Data Driven Filter Learning

A report on the Bachelor's Thesis Project
in partial fulfillment of the
requirements for the Degree of

Bachelor of Technology

by

Aditya Suryawanshi
Roll No. 210150004

Mehta Family School of Data Science and Artificial Intelligence

Indian Institute of Technology Guwahati

Guwahati-781039

Jul-Nov 2024

Aditya Suryawnshi
Data Driven Filter Learning

SUPERVISOR:
Dr. Neeraj Kumar Sharma
SPIN Lab
Assistant Professor
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology Guwahati

Report Certificate

This is to certify that this report titled, *Data Driven Filter Learning*, submitted by *Aditya Suryawanshi* to the *Mehta Family School of Data Science and Artificial Intelligence, IIT Guwahati* is a bonafide record of work carried with the supervision of *Dr. Neeraj Kumar Sharma.* The contents of this report/thesis, in full or in parts, have not been submitted to any other institute or university for the award of any degree.

Signature

Date:
Place: Guwahati

Acknowledgement

I am grateful to Dr. Neeraj Kumar Sharma for his invaluable guidance, patience, and encouragement throughout this research.

I would like to thank Nischay Nilabh for his help with the training of the model.

Finally I am grateful to Mehta Family School of Data Science and AI for providing me with the opportunity to work on this project.

Contents

Abstract

Speaker recognition has evolved significantly over the years, transitioning from early models like Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs)In the pre-2000s, they were elliptic models like GMMs, while in the 2000s they were probabilistic and statistical models. I-vectors became common in the 2010s but were later replaced by deep neural networks (DNNs) as the state-of-the-art approach. Traditional methods relied on hand-crafted features, but recent advancements leverage convolutional neural networks (CNNs) to learn low-level speech representations directly from raw waveforms. This method enhances the precision by identifying relevant characteristics as pitch and formants.

The core challenge still remains optimizing feature extraction to reduce uncertainties in raw audio waveforms. SincNet, a specialized CNN architecture using parameterized sinc functions, directly tackles this problem by implementing band-pass filters to pick more intuitive and interpretable features, unlike traditional CNNs that may focus on irrelevant or non-explainable features. SincNet uses only high and low cutoff frequencies as parameters, to improve their efficiency and interpretability.

Softwares Used

Software toolkits: The following software tools aided in the computer implementation of the research presented in this thesis.

(1) Python for coding of the proposed algorithms.
(2) PyTorch is used to make the models
(3) Kaggle platform is used to run the model.

List of Figures

Introduction

In a convolutional neural network (CNN) architecture, filters are the basic building blocks that learn and extract the relevant features of the input data. Simply put, filters are nothing more than feature detectors that look at the input data whether it is an an image or a waveform and pick up all patterns, textures or structures on them.

Specifically, in speech and audio processing, CNN filters opt for raw waveforms and are capable of recognizing fundamental features such as pitch, energy distribution, temporal and spectral characteristics. This is especially relevant in tasks such as speaker recognition, where minute changes in speech can lead to performance degradation, and small changes in the recording equipment can mean losing the experiment. The effectiveness of these filters depends on how well they are designed and chosen, allowing the network to pay attention to the features of primary interest and disregard noise or other irrelevant details, directly impacting the accuracy and efficiency of the model.

SincNet applies filter transformation technique to a generic speaker recognition CNN model and analyzes their effects. Speaker recognition is a type of biometric technology that recognizes or authenticates a person based on their unique voice data. It differentiates between people based on features of their voice, including pitch, tone and vocal tract shape. Speaker recognition can broadly be separated into two tasks: speaker identification, which seeks to determine who is speaking from a pool of known voices, and speaker verification, which authenticates a claimed identity of the speaker.

Recent speaker recognition systems have advanced significantly over previous probabilistic models like GMMs-UBMs [6], i-vectors[2] and DNNs[1].Although there are previous DNN based methods for directly classifying speakers, these methods mostly treat hand-crafted features such as FBANK and MFCCs as inputs. Although such features work well, they prevent the model from exploiting the raw waveforms fully. For instance, standard features such as MFCC smooth the speech spectrum, which can make it challenging to extract important narrow-band speaker characteristics such as pitch and formants. Recent efforts have focused on training features end-to-end directly on the raw signal, which allows for both better accuracy and generalization potential.

Due to weight sharing, local filters, and pooling which help to extract robust and invariant representations, CNN are the best architecture choice to process raw speech samples. Also CNN architecture has the first convolution layer which can be thought of the most important layer because this layer to extracts features from the high dimensional inputs gives them to the further layers and so filters learned from here are very important.

Especially with scarce training data, CNN filters often take noisy and unpredictable multi-band shapes. While such filters may work well for the network, they are not human-readable and likely not the best representation of the speech signal.

While just having tons of feature maps gain improved results, SincNet[5] is a CNN architecture which attempts at optimum usage of the learnt filters in the first convolution layer by applying certain constraints according to the shape of function of the filters.This also makes it far more efficient in terms of convolution, as filters now only have to learn the low and high cut-off frequencies. This way, the network learns filters that have meaningful, general effects instead of extremely specific to the training image or irrelevant details. This therefore leads to a significant improvement in the speaker recognition performance over standard CNNs.

Understanding the SincNet Architecture

In CNN, the first layer operates on the input data (raw audio waveform in this case) by applying time-domain convolutions. Alternatively it uses a bank of Finite Impulse Response (FIR)[4] filters to extract features from the waveform. Here is the function for output of of the CNN layer :

$$y[n] = x[n] * h[n] = \sum_{l=0}^{L-1} x[l] \cdot h[n-l] \tag{0.1}$$

here n is time instance x[n] is input waveform and h[n] is filter The filter h is of size L and this is based on the learning from the data. However, in SincNet, the filter uses a fixed g that instead of L learnable filters uses few learnable parameters θ, therefore the input is convoluted with g[n,θ].

In the frequency domain, the filter is a bandpass filter which can be seen through the subtraction of two low-pass filter in the equation:

$$G[f, f_1, f_2] = \text{rect}\left(\frac{2f_2}{f}\right) - \text{rect}\left(\frac{2f_1}{f}\right) \tag{0.2}$$

where f1 and f2 are the low and high cut-off frequency to be learnt and rect is the rectangular function in freq domain. Using inverse Fourier transform [4] to go back to the time domain we get:

$$g[n, f_1, f_2] = 2f_2 \,\text{sinc}(2\pi f_2 n) - 2f_1 \,\text{sinc}(2\pi f_1 n) \tag{0.3}$$

here sinc(x)=sin(x)/x.

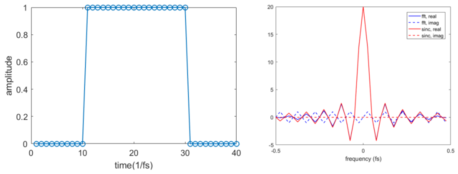


Fig. 0.1 Rect and Sinc in Time and Frequency domain respectively

Cut-off frequencies need to initialized properly on the mel-scale filter-bank as more of the speaker features for speaker identification are located on the lower half of the spectrum.

For the cutoffs to remain positive and f_2 must be larger than f_1 We apply the following transformation:

$$f_1^{\text{abs}} = |f_1| \tag{0.4}$$

$$f_2^{\text{abs}} = f_1 + |f_2 - f_1| \tag{0.5}$$

Ideal bandpass filters are theoretical filters where the passband (the range of frequencies allowed to pass) is perfectly flat, and the stopband (the range of frequencies that are attenuated) has infinite attenuation. However, creating an ideal filter is practically impossible because it requires an infinite number of elements (or coefficients), denoted as L. In reality,filters are truncated many times , which leads to filtering of something like the ideal filter (i.e. approaching the ideal filter as L→∞).The fact that filters are truncated leads to phenomena such as ripples in the passband (the desired flat frequency range) and limited attenuation in the stopband (insufficient removal of undesired frequencies). Most of the time, however, this is handled via some sort of windowing. Windowing is the convolution of the truncated filter g[n,f_1,f_2] with a window function w[n] to smooth the sudden discontinuities created at the edges of the filter. This minimizes the distortions that result from truncating the perfect filter. The window function smooth out the transition from the filter passband to stopband, it yield lower ripples for better frequency selectivity.

$$g[n, f_1, f_2] = g[n, f_1, f_2] \cdot w[n] \tag{0.6}$$

the choice of band pass filter by the SincNet authors is the popular Hamming window[3]:

$$w[n] = 0.54 - 0.46 \cdot \cos\left(\frac{2\pi n}{L}\right) \tag{0.7}$$

The filters g are symmetric, so they can be more computationally efficient, since we can consider only one side to compute the updated parameters, and copy it on the other side. All the operations involved in SincNet are fully differentiable. This is significant because it enables the network to be trained by gradient-based optimization methods such as Stochastic Gradient Descent (SGD). After the first sinc-based convolution, the pipeline is structured as any regular CNN. This means all ops like pooling, normalization, activations or dropout. From here on, more convolutional layers, fully connected layers or even recurrent layers may be stacked for better performance at feature extraction. Final classification of the speaker is done using a softmax classifier,which give likelihood of every class.

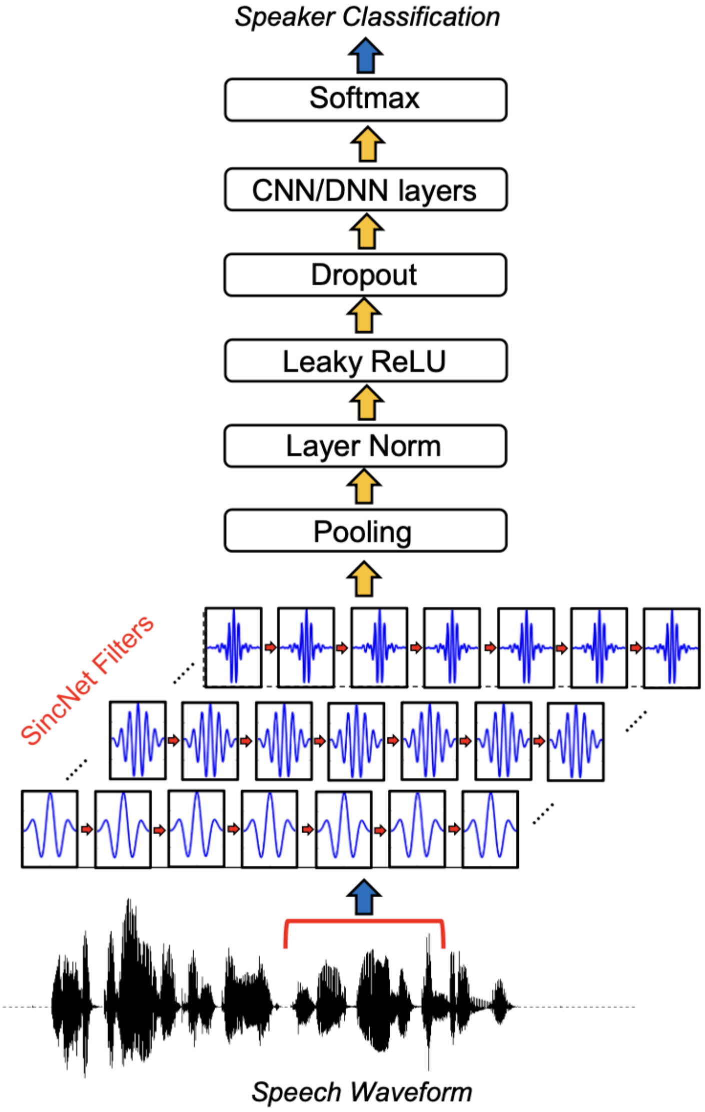


Fig. 0.2 The CNN architecture used

0.1 Advantages of SincNet

- By having custom filter that only focuses on the parts that have the biggest role in differentiating the speaker , the model converges faster. Even though this method relies on incorporating prior knowledge about the shape of filters commonly used in signal processing the parameters are learned by the model making it adapt to the data.

- The parameters to learn are significantly lower, considering F filters of L length the CNN has F*L parameters to learn but SincNet has to learn only two i.e high and low cut-off frequencies per filter making the number of filters 2*F.
- Model becomes more explainable as the new outputs of the convolution layer are more intuitive and interpretable.

0.2 Experiment

0.2.1 *DataSet*

The experimentation is performed on TIMIT(TIMIT Acoustic-Phonetic Continuous Speech Corpus) Dataset.The dataset has 630 speakers of eight American English dialects and Each speaker reads 10 phonetically-rich sentences(A 16-bit, 16kHz speech waveform file for each phrase). To ensure the model is fed with only speech data, the non-speech part at the beginning and the end of the waveforms were removed. The calibration that is the utterance that has the same text for all individuals is removed for the speaker recognition to be text independent. The train test split is in a ratio 5:3 .

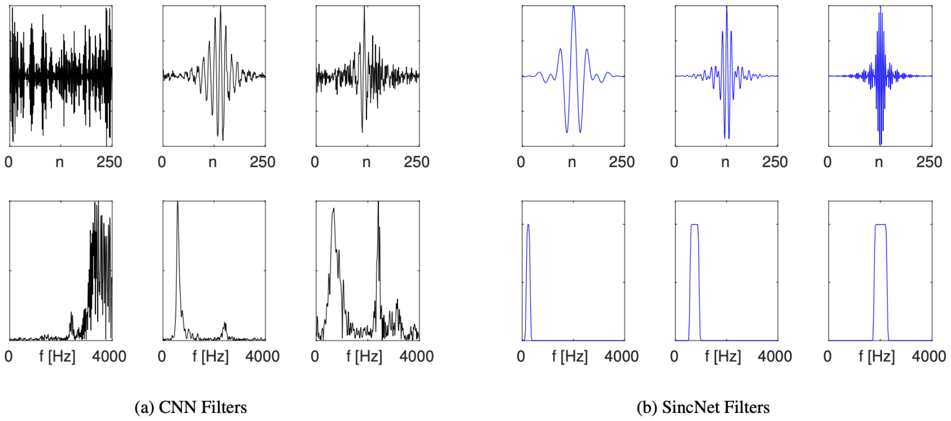
0.2.2 *SincNet*

The input waveforms are split into utterences of 200ms. The hyperparameters for the first layer of convolution are F=80 and L=251. After this two more convolution layers are applied of F=60 and L=6.The outputs after batch normalization were fed to a Fully connected layer(FCC) having 2048 neurons.The activation function chosen is leaky ReLU. As discussed earlier the initial cut-off frequencies were set to the mel-scale cutoff frequencies. At the end softmax is used to get the classification output. Training used RMSprop optimizer, with lr = 0.001, $\alpha = 0.95$, $\epsilon = 107$.

A speaker verification system is built upon the speaker identification neural network and supports two modes. The first, known as the d-vector framework, extracts a fixed-length embedding (d-vector) from the network's final hidden layer and compares it to the claimed speaker's d-vector using cosine distance, accepting the claim if the distance is below a set threshold. The second DNN class framework, depends on the softmax posterior score, corresponding directly to the corelation of test sample to guessed speaker, with indicating greater confidence. These approaches provide flexibility, one emphasizes on the first using direct similarity comparisons on document embeddings, and the second using direct classification probabilities. A baseline CNN, with the same architecture as SincNet but standard convolutions instead of sinc-based filters, was tested on raw waveforms.

Results of the Experimentation

The one point to realize is the change induced by the form of filter employed. Thus, the filters after training the CNN are shown in the figure below, as well as the SincNet implementation. However, As you can see the CNN is not really able to learn correct filters on well defined raw waveforms. The CNN filters exhibit noise and bands. But the SincNet filters are way more clearly defined.



(a) CNN Filters

(b) SincNet Filters

Fig. 0.3 Representation of filters learned by CNN and SincNet

We can plot the cumulative frequency response of these filters as shown in fig 0.4, where we can note SincNet filters have three clear peaks but the CNN ones don't. The first peak pitch, the second first formants (500 Hz), and the third (900–1400 Hz) important second formants.

The fig 0.5 plots the error rate of CNN and SincNet with the training epochs . As observed the Frame Rate error(FER) for the SincNet drops much faster than the CNN, shows better performance is achieved by SincNet much faster.

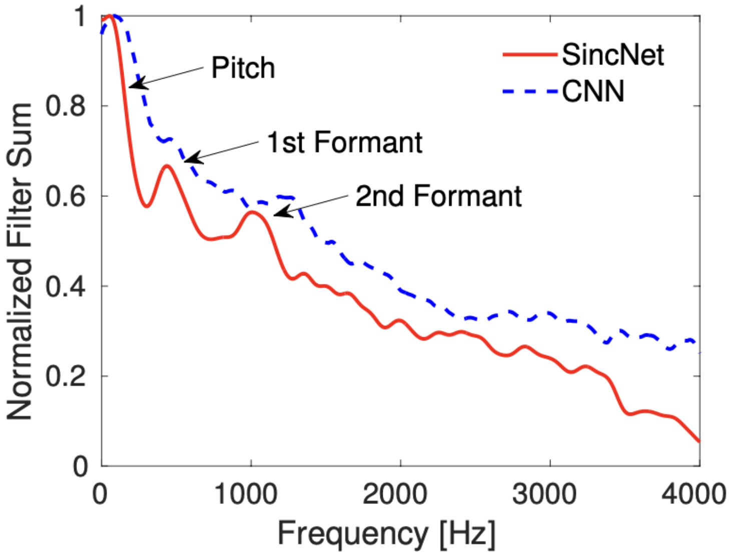


Fig. 0.4 Cumulative Frequency response of filters

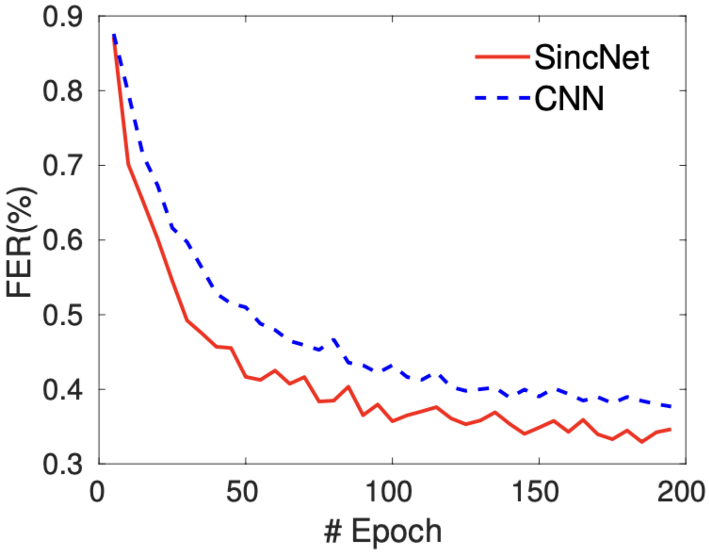


Fig. 0.5 FER(%) over epochs

The Classification Error Rate(%) for speaker recognition is 1.72 using CNN and 1.24 using the SincNet architecture in TIMIT dataset .

Future Work

After realizing that the choice of filters matters, particularly in context of noisy data and lot of features, next thing on agenda is custom filter in 2D CNNs. Raw waveforms have been used directly as input to model building, however, they provide a rather poor feature representation of the signal, as seen with light-based analysis, where a spectrogram derived from the raw waveform reveals both time and frequency domain features much more clearly, thus making them better suited for extracting finer speaker characteristics. To enable such custom filters, approaches such as parametrising them with learnable functions (as in SincNet) or using domain-specific priors such as Gabor filters and Mel-scale inspired transformations will be explored. These filters are intended to highlight speaker-relevant characteristics and reduce noise. Performance will be compared to SincNet on speaker verification and identification on the TIMIT dataset.

Bibliography

[1] G. Dahl, D. Yu L. Deng, and A. Acero. Context- dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2012.

[2] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet. Context-dependent pre-trained deep neural networks for large vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 2012.

[3] S. K. Mitra. Digital Signal Processing. *McGraw-Hill*, 2005.

[4] L. R. Rabiner and R. W. Schafer. Theory and Applica- tions of Digital Speech Processing. *Prentice Hall*, 2011.

[5] Mirco Ravanelli and Mirco and Bengio. SPEAKER RECOGNITION FROM RAW WAVEFORM WITH SINCNET. 12 2018. doi: 10.1109/slt.2018.8639585.

[6] D. A. Reynolds, T. F. Quatieri, and R.B. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 2000.