Adaptive Filter Banks using Autoencoders

Robert Viehweg
Technische Universität Ilmenau
robert.viehweg@tu-ilmenau.de

***Abstract*—In this seminar project, a deep learning–based approach is evaluated for designing adaptive filterbanks using an autoencoder architecture. The primary objective is to assess the reconstruction quality of the learned filterbank in comparison to a fixed Mel-scale filterbank. Separate autoencoder models were trained for each filterbank. Experimental results show that, for speech data, the adaptive filterbank achieves better reconstruction quality according to the mean absolute error (MAE) than the static Mel-scale filterbank.**

I. Introduction

Traditionally, filter banks are sets of band-pass filters that divide a signal into multiple components, each carrying information from different frequency ranges. In the past, these filters were typically static. For example, Mel-filterbanks [1], inspired by the human auditory system, have historically been successful for feature extraction. However, Mel-filterbanks also have limitations: they are based on auditory experiments whose original designs have had to be revised multiple times due to replication issues [2].

While human perception provides useful inductive biases for certain applications, such as music understanding, other tasks may require fine-grained resolution at higher frequencies—something not well supported by human perception–based filterbanks. These limitations of static filterbanks, such as Mel-filterbanks, have motivated the use of learnable neural layers as alternatives [3], [4].

In this work, we explore deep learning approaches—particularly the autoencoder architecture—to design adaptive filterbanks. The filterbank implementation is based on SincNet [5], where the learnable filters of the first CNN layer are replaced with parametrized sinc functions defined solely by their low and high cutoff frequencies. Here, the output of the first layer is fed into an autoencoder that attempts to reconstruct the raw input audio. We analyze both the quality and reconstruction accuracy of the learned filterbank.

II. Related Work

Initial attempts at learning filterbanks focused on adapting the filters of Mel-filterbanks in the spectrogram domain. Later studies [6], [7] proposed learning convolutional filters directly from raw waveforms. In these works, the learned filters were followed by different architectures: in one case, LSTM cells with a fully connected layer, and in another, a stack of four fully connected layers.

Further developments included SincNet [5], which applies convolution using a sinc function parametrized by low and high cutoff frequencies. The resulting filtered audio is then processed by a convolutional architecture.

As mentioned earlier, our approach is inspired by SincNet but differs in that the filtered audio produced by the learnable filters is processed by an autoencoder architecture. This design distinguishes our work from prior approaches and enables the exploration of learning filterbanks within an autoencoder framework.

III. Methodology

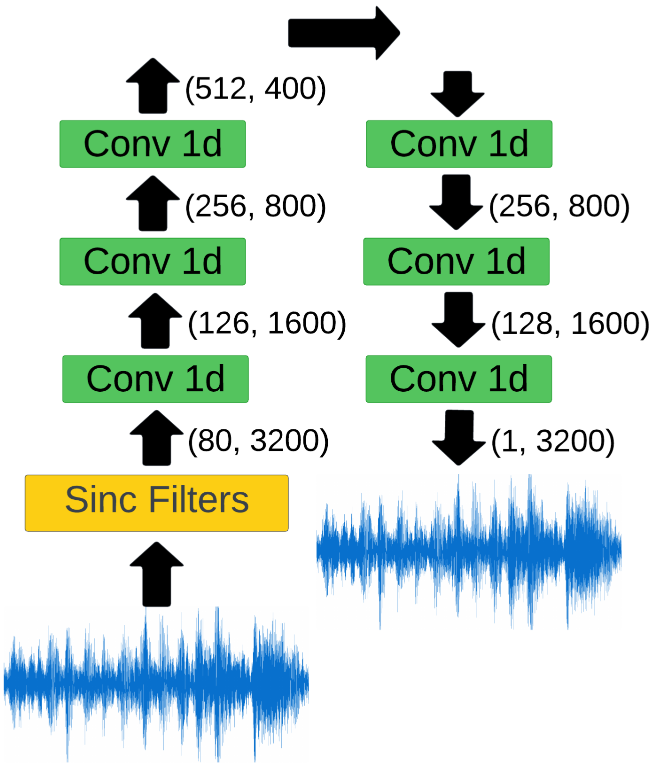

Fig. 1. Architecture of the project setup.

A. Model

As shown in Figure 1, the learnable SincNet filters are used to extract relevant features from the raw audio input. Following SincNet [5], the filters perform a convolution with the time-domain audio:

$$y[n] = x[n] * g[n, \theta],$$

where $x[n]$ denotes the raw time-domain audio input and $g[n, \theta]$ represents the filterbank of rectangular band-pass filters. In the time domain, these filters are expressed as:

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

where f_1 and f_2 are the learned low and high cutoff frequencies, respectively. The lower cutoff frequency f_1 is learned directly, while the upper cutoff frequency f_2 is learned indirectly by adapting the bandwidth:

$$\Delta f = f_2 - f_1.$$

To provide a meaningful initialization for the filter parameters, they are set according to the Mel scale, as in the original SincNet implementation[1]. The passband of each filter is constructed to cover a region around the Mel-scale center frequencies. This initialization offers a starting point that roughly mimics the human auditory system, from which the filters can adapt to the specific task of analyzing the speech data in the given dataset.

The learning of meaningful filters is guided by the autoencoder architecture shown in Figure 1. Each `Conv1d` block consists of a one-dimensional convolution followed by a ReLU activation. The final output layer uses a `tanh` activation. By learning to compress and reconstruct the original audio, the model is encouraged to develop filterbank representations that meaningfully capture the structure of the input signal.

B. Experimental Setup

The model was implemented in Python using the PyTorch deep learning library. As the dataset, the DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus [8] was used. This dataset is designed to support the acquisition of acoustic-phonetic knowledge. For training in this project, speech audio snippets were randomly sampled for each audio file, with a fixed length of `wlen = 3200` samples[2] and a sampling rate of `fs = 16000` Hz.

The architecture was divided into two separate PyTorch models: `Filterbank`[3], taken directly from the SincNet project, and `ConvAutoencoder`[4], representing the autoencoder architecture implemented in this work. The `Filterbank` was instantiated with `N_filt` = 80 filters, each of length `filt_dim` = 251.

Training used a single RMSprop optimizer [9] for both models, with an initial learning rate of 0.001. A learning rate scheduler decreased the learning rate by a factor of 0.95 every 25 epochs. The loss function was the Mean Absolute Error (MAE) between the autoencoder's decoder output and the original audio input. MAE provided more stable results than Mean Squared Error (MSE), as the latter caused the decoder output to converge towards small random values instead of accurately reproducing the input audio.

The final training ran for 600 epochs (10,800 steps in total) on an NVIDIA RTX A5000 GPU, resulting in a total training time of 1 hour and 12 seconds. The evolution of the loss function over the course of training is shown in Figure 2.

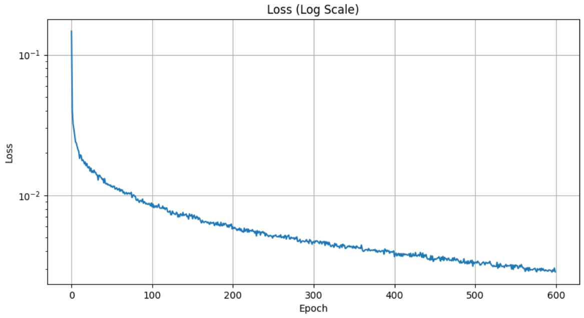

Fig. 2. Decreasing of loss on logarithmic scale during training over epochs.

C. Audio Reproduction Quality

The first indicator of the learned filterbank's performance is the quality of the audio reproduced by the autoencoder. To evaluate this, an inference script[5] was used. In this evaluation, a random audio segment of length `3200` samples, with a sampling rate of `fs = 16000` Hz, was fed into the autoencoder containing the learned filterbank and trained weights.

A comparison between the one-dimensional input audio waveform and the reconstructed output is shown in Figure 3. From looking at the time domain waveforms they look almost perfectly replicated and aligned, which indicates good replication capabilities of the filterbank and autoencoder.

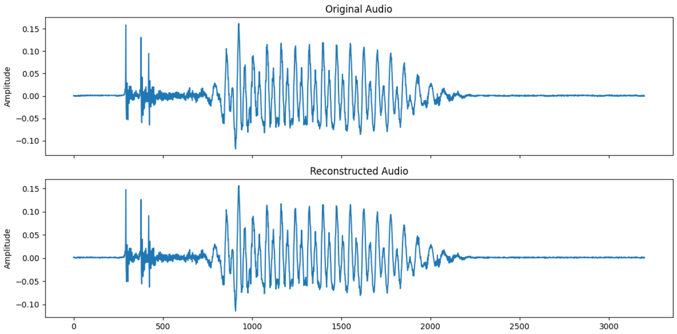

Fig. 3. Comparison of original audio and reconstructed audio by the model.

D. Learned Filterbank

Before training, the filterbank was initialized with the lower cutoff frequency and bandwidth set to cover the center frequencies of the Mel scale. This initialization is shown in Figure 4 for the time domain and in Figure 5 for the frequency domain. The filters are denser and more fine-grained

[1] `filterbank.py, lines 38--54`

[2] `speech_dataset.py, lines 6--27`

[3] `filterbank.py`

[4] `backbone.py`

[5] `inference.py`

at lower frequencies, while becoming wider and sparser at higher frequencies, mimicking the basic characteristics of the human auditory system.

The learned filterbank after training is shown in Figure 6 for the time domain and in Figure 7 for the frequency domain. Compared to the initialized filters, the trained filters are less ordered and less symmetric. Notably, many filters—particularly those with finer resolution—are concentrated in the range of approximately 1.5–4.5 kHz. This corresponds to the frequency range in which speech predominantly operates, encompassing critical formant frequencies and consonant energy. Compared to the holistic Mel-scale filterbank, the adaptive filterbank therefore focuses more precisely on speech-relevant frequencies present in the dataset, providing a finer and more task-specific representation of the data.

The exact reason for the emergence of this particular filter configuration is difficult to determine. However, it is evident that the learned representation is better adapted to the characteristics of the dataset than the fixed Mel-scale initialization.

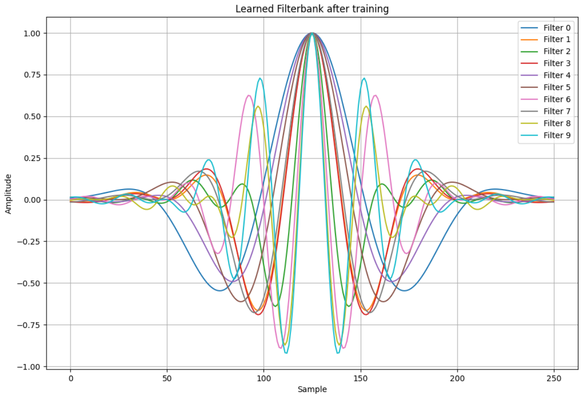

Fig. 4. Initialized time domain filterbank (first 10 filters) according to the Mel-scale.

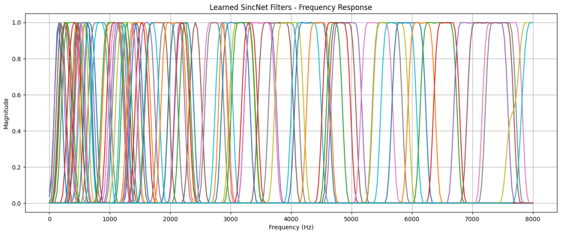

Fig. 5. Initialized frequency domain filterbank according to the Mel-scale.

E. Comparison of Adaptive Filterbank and Fixed Mel-scale Filterbank

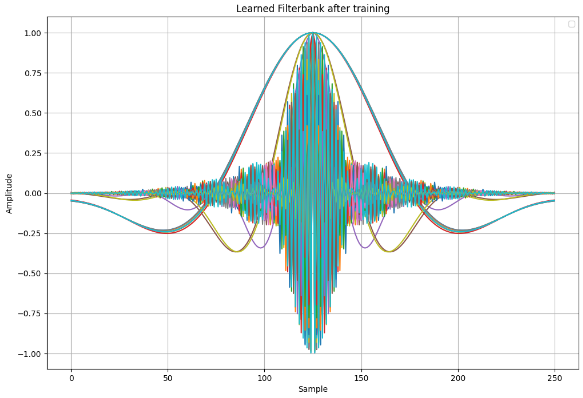

Fig. 6. Trained time domain Filterbank.

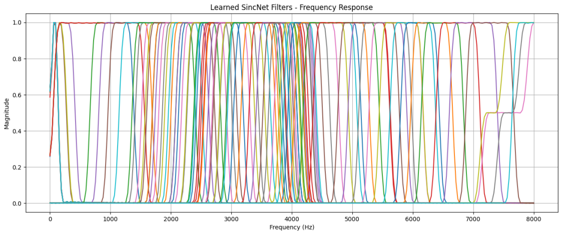

Fig. 7. Trained frequency domain Filterbank.

To evaluate whether the trained filterbank indeed results in better audio reconstruction, an identical training was conducted with the previously learnable filter parameters set to static. In this way, the new autoencoder model learns to make the best use of the provided Mel-scale filters, ensuring that the comparison between the adaptive and fixed filterbanks depends solely on the filterbank design.

For comparison, both models were fed the same randomly sampled audio snippet from the dataset. The outputs of both models were then compared to the original input audio using the mean absolute error (MAE). This process was repeated 1000 times to minimize the effect of outliers. Finally, the mean of all MAE values was computed separately for each model to determine which filterbank could reproduce the audio more accurately. This evaluation was implemented in the script [6].

The results were as follows: for the static Mel-scale filterbank, MAE $= 0.00259$, and for the adaptive filterbank, MAE $= 0.00215$. In terms of the mean absolute error, this indicates that the adaptive filterbank was able to capture more relevant audio features necessary for accurate reconstruction compared to the static Mel-scale filterbank.

IV. Discussion

A. Audio Comparison

The mean absolute error (MAE) is used as a metric for evaluating audio reproduction both during loss computation and when comparing the static and adaptive filterbank models. Since MAE operates sample-by-sample in the time domain, it measures the average magnitude of the difference between

[6] `evaluate.py`

two waveforms. While this is effective for assessing precise, pointwise reconstruction, it does not capture perceptual audio quality. Human hearing is highly nonlinear and frequency-dependent, meaning that two audio files can have a low MAE yet still sound different if there are small time shifts, phase shifts, or other perceptually relevant distortions.

B. Comparison of Mel-scale and Adaptive Filterbank

Although the learned filterbank outperforms the static Mel-scale filterbank for this specific speech dataset, this does not necessarily hold for broader audio tasks, such as environmental sound classification or music analysis. For such tasks, the Mel-scale filterbank may already provide a well-adapted representation, whereas the adaptive filterbank could be more beneficial for specialized or niche tasks like speech recognition.

V. Conclusion and Future Work

There remains room for further improvement. Future work could involve experimenting with alternative loss functions for both evaluation and training, such as the Spectral Loss or Log-Magnitude Spectral Loss, which capture human auditory perception more effectively than the purely time-domain approach of the MAE.

In addition, it would be worthwhile to explore other model architectures for audio reproduction that take the adaptive filterbank as input. Possible directions include adjusting the number of layers, experimenting with variational autoencoder architectures, or simply scaling up the training process. The latter could involve increasing the number of epochs, as the loss function was still decreasing even at 600 epochs.

Finally, it would be valuable to examine whether the adaptive filterbank also outperforms the static Mel-scale filterbank in more holistic audio tasks, involving a wider variety of sounds such as environmental noise or music. Providing the model with audio that covers the entire frequency spectrum more evenly might result in a learned filterbank that resembles the fixed Mel-scale filterbank. However, confirming this hypothesis remains a topic for future experimentation.

References

[1] N. Zeghidour, O. Teboul, F. de Chaumont Quitry, and M. Tagliasacchi, "LEAF: A Learnable Frontend for Audio Classification," in *Proc. International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: https://openreview.net/forum?id=jM76BCb6F9m

[2] D. D. Greenwood, "The Mel Scale's disqualifying bias and a consistency of pitch-difference equisections in 1956 with equal cochlear distances and equal frequency ratios," *Hear. Res.*, vol. 103, no. 1–2, pp. 199–224, 1997. [Online]. Available: https://doi.org/10.1016/s0378-5955(96)00175-x

[3] D. Palaz, M. Magimai-Doss, and R. Collobert, "Convolutional Neural Networks-based continuous speech recognition using raw speech signal," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4295–4299, doi: 10.1109/ICASSP.2015.7178781.

[4] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-training for Speech Recognition," *CoRR*, vol. abs/1904.05862, 2019. [Online]. Available: http://arxiv.org/abs/1904.05862

[5] M. Ravanelli and Y. Bengio, "Speaker Recognition from Raw Waveform with SincNet," *arXiv preprint* arXiv:1808.00158, 2019. [Online]. Available: https://arxiv.org/abs/1808.00158

[6] T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform CLDNNs," in *Proc. Interspeech*, 16th Annual Conf. of the International Speech Communication Association, 2015.

[7] Y. Hoshen, R. Weiss, and K. W. Wilson, "Speech acoustic modeling from raw multichannel waveforms," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

[8] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus*, Linguistic Data Consortium, Philadelphia, 1993. LDC93S1.

[9] G. Hinton, *Lecture 6e – RMSprop: Divide the gradient by a running average of its recent magnitude*, Coursera: Neural Networks for Machine Learning, 2012.
https://www.cs.toronto.edu/ tijmen/csc321/slides/lecture$_{s}lides_{l}ec6.pdf$