Learning FIR Filter Coefficients from Data for Speech-Music Separation

Boyang Wang and Jafar Saniie

Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago IL, U.S.A.

***Abstract*—An Finite Impulse Response (FIR) filter is a widely used digital filter technology whose impulse response has a finite duration. An FIR filter is usually favored for many reasons such as easy to design, easy to implement on a variety of system architectures. An FIR filter can be easily designed with a linear phase response and its output is more predictable since it doesn't have feedback components. There are both engineer and mathematical methods for designing an FIR filter so that machine learning doesn't play an important role in the FIR filter design. In this paper, we present an alternative to traditional filter design methods to direct learn the FIR filter coefficients from input data with machine learning algorithm. With the proposed algorithm, we can easily design an FIR filter from the input data mixed with designed all spectrum noise signal. To show the capability of this algorithm, an example application of suppressing background music from speech or vice versa is demonstrated in this paper. Despite that the music and speech have a lot of overlap in their spectrum, the filter designed by our algorithm can successfully suppress music or speech in a mixture of music and speech signals.**

Keywords—FIR Filter, Convolutional Layer, Filter Design, Selective Filtering, Machine Learning, TensorFlow

I. INTRODUCTION

An Finite Impulse Response (FIR) filter has a finite duration and is widely used in many signal filtering applications such as communication, image processing, and many other signal processing methods that require signal conditioning due to stability [1] [2]. Equation 1 shows the formula of filtering a signal x[n] with an FIR filter of N taps.

$$y[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n-k] \qquad (1)$$

The term h[k] is the impulse response of the FIR filter, it is also referred to as FIR coefficients. Each tap in an FIR filter is a multiply-accumulate (MAC) unit which contains a register, a multiplier, and an adder as shown in Figure 1. This formula can also be interpreted as a convolution between the input signal and the FIR filter kernel impulse response. Figure 1(a) is a classical FIR filter design schematic with 20 taps. A disadvantage of this type of architecture is that the critical path is $T_{mult}+20T_{adder}$, this will dramatically reduce the maximum system clock and jeopardize the speed of the FIR filter realization. Figure 1(b) shows a transposed implementation of the FIR filter, it is also called a broadcast FIR filter since the input signal will be directly broadcasted to all multipliers [3]. It is a more preferred architecture since the critical path of this design is always the $T_{mult}+T_{adder}$ regardless of the number of taps. However, the input signal is broadcasted to N multipliers, the fanout must be considered when we design a filter with a large number of taps.

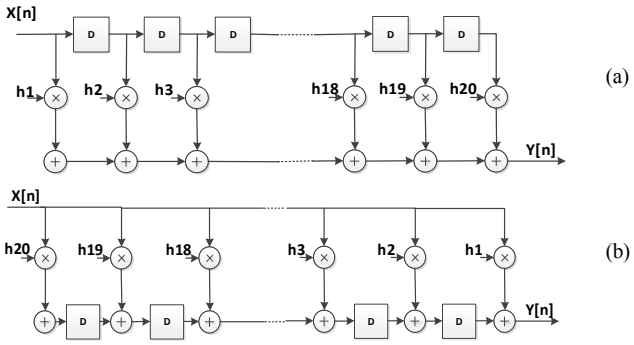


Figure 1. FIR Filter Architecture

There are many special types of FIR filters such as raised cosine filter and differentiator filter [4] [5]. These filters have their own specific design concepts and methods. There are multiple methods for designing typical FIR filters with specified frequency response including window design method and frequency sampling method [6] [7]. These conventional FIR design methods are optimized mathematically or offer an efficient engineering solution. Machine learning usually doesn't play a role in the FIR design when there is already a direct solution. In this paper, we present a machine learning model that can learn directly from the input signals and come up with an optimized FIR filter solution. For example, we have a speech signal mixed with a music signal and need to be separated. This doesn't seem to be a problem that can be solved by a conventional FIR filter since speech and music have notable spectral overlap. With the proposed machine learning algorithm, we can learn a special FIR filter that can decompose speech signals from music signals adaptively.

Section II of this paper discusses the FIR filter design method with machine learning algorithms and validation procedures. Section III demonstrates an example of separating music and speech.

II. MACHINE LEARNING FIR FILTER DESIGN METHODS

This section introduces FIR machine learning models and their validation. Figure 2 illustrates *Signal A* has significant spectral overlap with unwanted *Signal B*. The goal of FIR design is to separate *Signal A* from *Signal B*.

978-1-7281-5317-9/20/$31.00 ©2020 IEEE

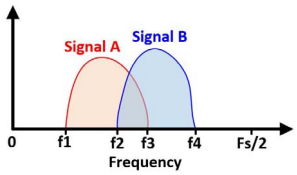


Figure 2. Mixed signals with an overlapped spectrum

A. Design of Machine Learning Model

Figure 3 is the block diagram of the proposed model for learning FIR filter coefficients. The training input is a mixture of three signals as shown in equation 2. *Signal A,* $s_A[n]$, needs to be estimated. *Signal B,* $s_B[n]$, must be filtered. Noise, $v[n]$, is used to eliminate the irrelevant frequency components in the process of filtering *Signal B*. Equation 3 represents an FIR convolution layer, with coefficient w[k]. The model is designed that this output to estimate *Signal A*. The expected training output $\bar{y}$ is defined as Signal A. The loss function of the model is defined as Mean Square Error (dMSE) presented in equation 4. The training of the coefficient is done by the backpropagation algorithm [8]. When training this model, the gradients of the loss function with respect to each individual weight is computed and are used to update these weights to minimize MSE. Once the training is done, we can extract the weights of the convolutional layer and use it as the FIR filter impulse response.

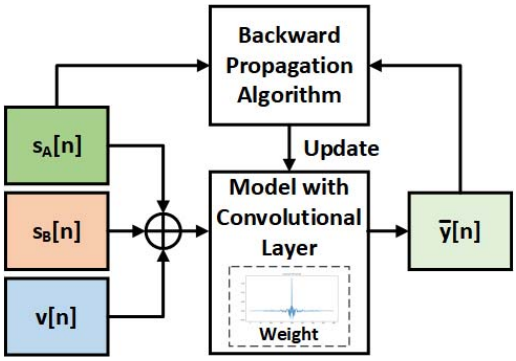


Figure 3. Model for Learning FIR Filter Coefficients

$$x[n] = s_A[n] + s_B[n] + v[n] \tag{2}$$

$$\bar{s_A}[n] = \sum_{k=0}^{N-1} w[k] \cdot x[n-k] \tag{3}$$

$$MSE(\bar{s_A}[n], s_A[n]) = \frac{1}{N} \sum_{i=1}^{N-1} (s_A[i] - \bar{s_A}[i])^2 \tag{4}$$

TensorFlow is used in this paper to build and train the model. With Keras API in the TensorFlow library, it is flexible to structure and control the designed model.

B. FIR Filter Design

To show that the model can automatically compute the optimized filter kernel with the desired frequency response, a randomly generated all spectrum noise is used for training the model. In the validation experiment, a uniformly distributed noise is used as the training data. As shown in Figure 4, the generated noise is transformed into the frequency domain using FFT and multiplied by the desired frequency response mask [9]. After this step, the noise signal will then be converted back to the time domain to be used as the expected training output.

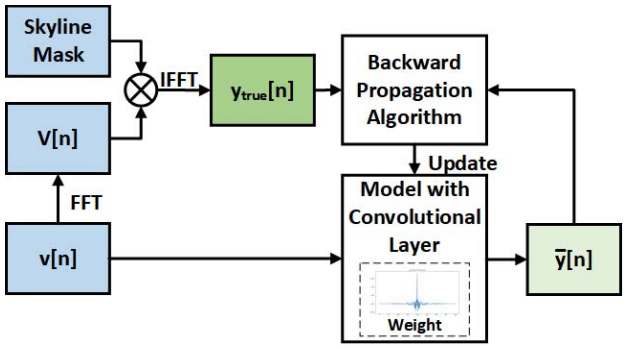
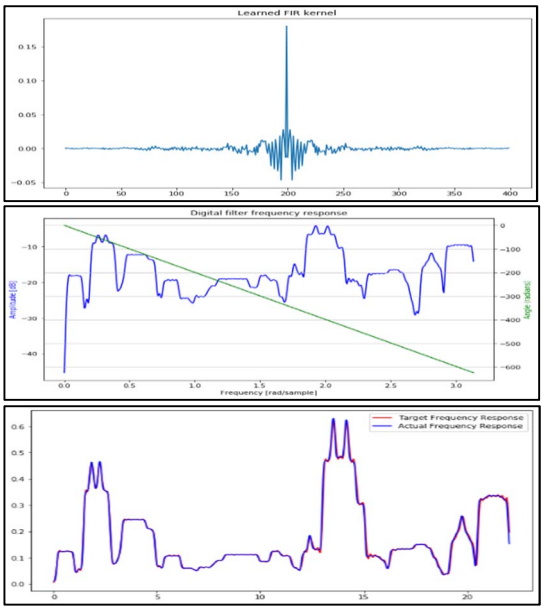


Figure 4. Model validation learning block diagram

The model is trained with 10,000 randomly generated noise signals, each of these noise signals is uniformly distributed with the length of 8,192. For demonstration, the frequency response mask is designed to look like Chicago Skyline. Figure 5(a) is the extracted FIR filter coefficient from the trained model after training the convolutional layer with 400 coefficients for 100 epochs. Figure 5(b) is the digital filter frequency and phase response. Figure 5(c) is the frequency response of the designed filter kernel plotted against the frequency response mask that is used to modulate the noise in the frequency domain. This method provides a very close solution compared to the IDCT digital FIR filter design method but will never perform better since IDCT is the direct solution to the problem. The result in this experiment shows that our model is fully capable of learning the FIR filter kernel from time-domain input data.



(a) (b) (c)

Figure 5. FIR Filter kernel learned from the desired mask spectrum

Another important validation experiment is to show that the model can characterize different frequencies by directly learning from the input data. In this experiment, a training data mixed by

8 sinusoidal signals that have different center frequencies, amplitudes, and phases are created. Figure 6(a) is the generated training data and its frequency response. We choose three out of these eight frequency components to create the corresponding training output as is shown in Figure 6(b). In order to suppress the irrelevant frequency band an all-spectrum uniformly distributed noise to the training data. Figure 6(c) is the acquired FIR filter impulse response with 300 taps. Figure 6(d) is the digital filter frequency response. To be noticed is that the learned filter frequency response shows that the designed filter has additional attenuation when the unwanted frequency components have higher energy. The result shows that our model is fully capable of separating different frequency components from the time domain.

This FIR filter coefficients design method has very little control over its phase response when the frequency component is been attenuated to almost zero. This is because the frequency components that are zeroed have very little contributed to the backpropagation algorithm computation. The proposed algorithm only computes the statistically optimized solution, which is not always the best solution to the problem when the specific frequency response is known by the designer.

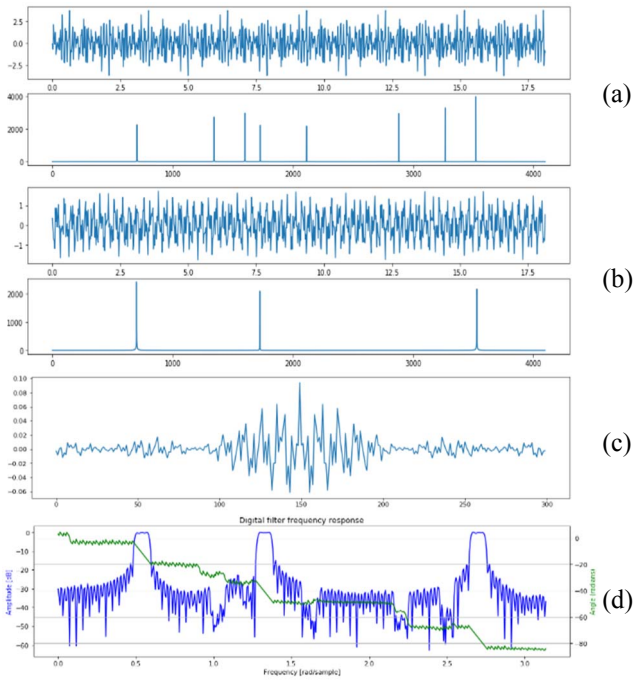


Figure 6. Experiment to show that the model can be used to separate different frequency components

III. Example Application on Music and Speech

Source separation is a classical problem in speech processing and other signal processing algorithms [10] [11]. Clean separation is difficult to achieve when the sources to be separated have overlapped frequency components. However, with the proposed machine learning filter coefficient design method, we can achieve acceptable source suppression among two overlapped voices on the spectrum with a single FIR filter. Figure 7 shows the spectrum of flute music, speech and a mixture of both. The sampling rate for all the voices presented in this paper is 44,100 Hz. As can be observed in the figure, the music and speech signals have a lot of overlap on the frequency domain, which raises the difficulty of being separated.

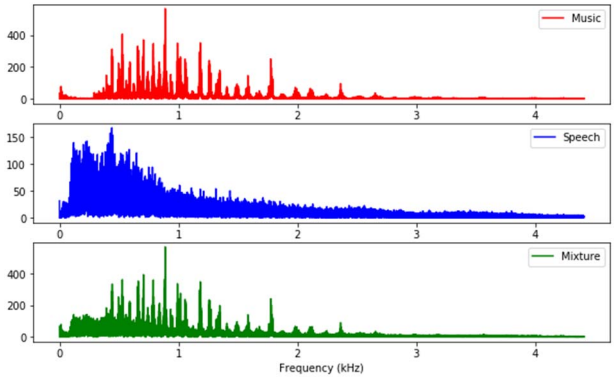


Figure 7. The spectrum of music, speech and their mixture

During training, we use the mixture of the music signal, speech signal and all spectrum noise as the training input. A selection of either a music signal or speech signal is used as the expected training output. In order to select the optimized FIR tap number, we train the model with different tap numbers for 50 epoch and plotted the final training loss against its tap number as shown in Figure 8.

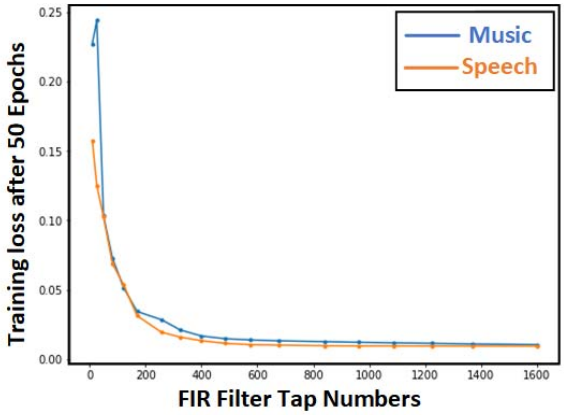


Figure 8. Training Loss after 50 epochs plot against different FIR Tap Number

After training the model with 2000 taps for 100 epochs, Figure 9 shows the MSE training loss against epochs.

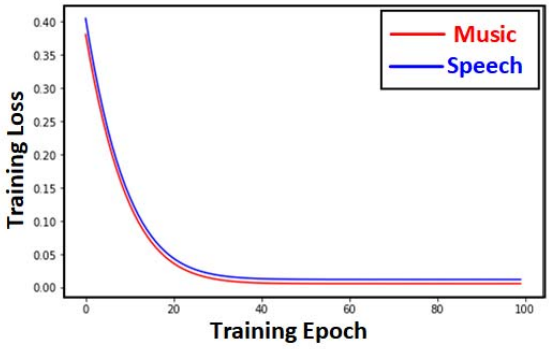


Figure 9. Training loss against epochs

Figure 10 shows the designed FIR filter impulse responses and their frequency responses. As can be seen from the figure, music signal extractor and speech signal extractors have overlap on the frequency axis. The designed filter has extra attenuation on a certain spectrum when the other signal to be suppressed has high energy on that frequency band.

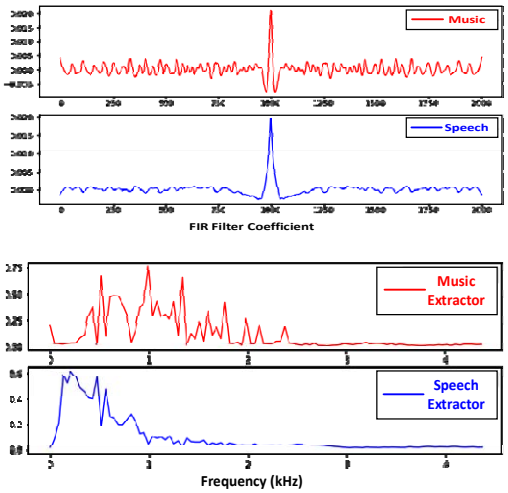


Figure 10. (a) FIR filter impulse responses and (b) frequency responses

Figure 11 is the processed mixture speech and music signal filtered by the designed FIR coefficients using machine learning. As can be observed in the figure, the filters successfully suppress the unwanted signal. Experiment result shows that when voice signal is suppressed, the mean square error between filtered signal and the music signal is 0.0035. On the other hand, when music signal is chosen to be filtered out, the mean squre error between the filtered output and voice signal is 0.006. When the filtered signals are played after restored back into wav files, the unwanted signal will sound just like a background whisper but cannot be completely removed since this method is only an FIR approach after all.

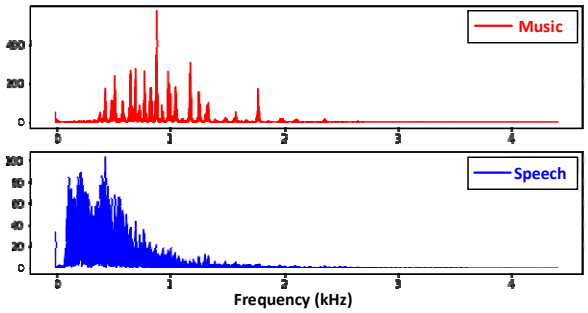


Figure 11. Extracted music and speech signal spectrums

IV. Conclusion

In this paper, we provide a simple but effective machine learning model for learning FIR filter coefficients directly from the input data. The proposed algorithm can produce FIR filter impulse response that can separate highly spectrum overlapped signals. Uniformly distributed noise is added to the training data to eliminate the irrelevant frequency components. This method will find the statistically optimized FIR filter impulse response with the given number of taps according to the training data set. The learned FIR filter coefficients will have limited control over the highly attenuated frequency components. The designed FIR filter provides a linear phase response for most of its passbands. An example application of speech or music signal extractor from a mixture of two is demonstrated. The result shows that our model can achieve a very complicate design by simply training with the input data.

References

[1] M. B. Trimale and Chilveri, "A review: FIR filter implementation," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 2017.

[2] M. Ferrario, A. Spalvieri and R. Valtolina, "Design of transmit FIR filters for FDM data transmission systems," *IEEE Transactions on Communications,* vol. 52, no. 2, pp. 180-182, 2004.

[3] Xilinx, "PG149 LogiCORE IP FIR Compiler v7.1, Product Guide," 2 April 2014. [Online]. Available: https://www.xilinx.com/support/documentation/ip_documentation/fir_compiler/v7_1/pg149-fir-compiler.pdf.

[4] N. S. Alagha and P. Kabal, "Generalized raised-cosine filters," *IEEE Transactions on Communications,* vol. 47, no. 7, pp. 989-997, 1999.

[5] C.-C. Tseng, "Digital differentiator design using fractional delay filter and limit computation," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 52, no. 10, pp. 2248-2259, 2005.

[6] A. E. Cetin, O. N. Gerek and Y. Yardimci, "Equiripple FIR filter design by the FFT algorithm," *IEEE Signal Processing Magazine,* vol. 12, no. 2, pp. 60-64, 1997.

[7] M. G. Shayesteh and M. Mottaghi-Kashtiban, "FIR filter design using a new window function," in *2009 16th International Conference on Digital Signal Processing*, 2009.

[8] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT Press, 2016.

[9] E. O. Brigham and R. E. Morrow, "The fast Fourier transform," *IEEE Spectrum,* vol. 4, no. 12, pp. 63-70, 1967.

[10] E. M. Grais and H. Erdogan, "Single channel speech-music separation using matching pursuit and spectral masks," in *2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU)*, Antalya, 2011.

[11] P. Mowlaee, A. Sayadian, M. Sheikhan and M. Fallah, "Single-channel music/speech separation using non-linear masks," in *2008 International Symposium on Telecommunications*, Tehran, 2008.