CleanMachine

(proof of concept presentation)

Julio Ramirez

Project Description

- Idea behind the project is to build a machine learning model that upon inference can calculate a frequency filter that when “applied” to some input signal will attenuate all frequencies estimated to contain noise.
  - Ideally the model would be small (in task or computation) so that it could be used for real time speech enhancement in the future.
- Interface
  - Likely no GUI
  - CLI application for batch processing of audio
    - Options for selecting different denoising algorithms

Concepts

Source separation.

$$s[n] = x[n] + d[n]$$

Noisy input signal {s_n} is composed of a clean signal {x_n} and a distorted signal {d_n}

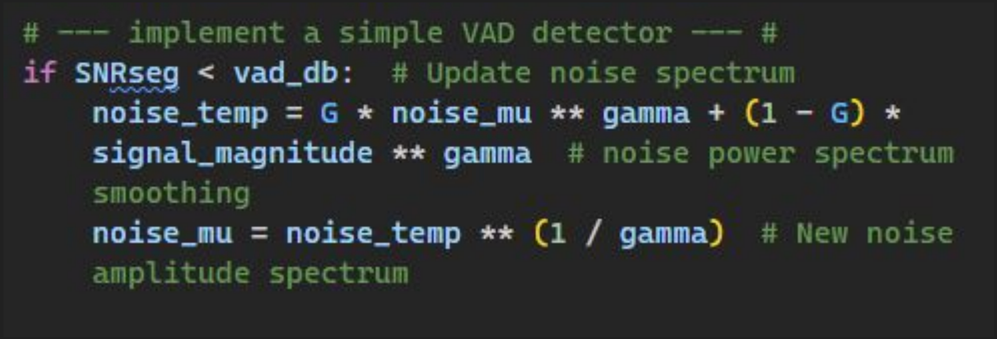
This holds in the frequency domain of the Fourier Transform and by extension the TF-domain of the STFT

$$S(\omega) = X(\omega) + D(\omega) \implies S[n, \omega] = X[n, \omega] + D[n, \omega]$$

In practice, **exact** information of about the distortion noise is not known.

Most Algorithms use simple voice activity detection(VAD) modules

- When no speech is detected the noise estimation can be updated.



```
# --- implement a simple VAD detector --- #
if SNRseg < vad_db:  # Update noise spectrum
    noise_temp = G * noise_mu ** gamma + (1 - G) *
    signal_magnitude ** gamma  # noise power spectrum
    smoothing
    noise_mu = noise_temp ** (1 / gamma)  # New noise
    amplitude spectrum
```

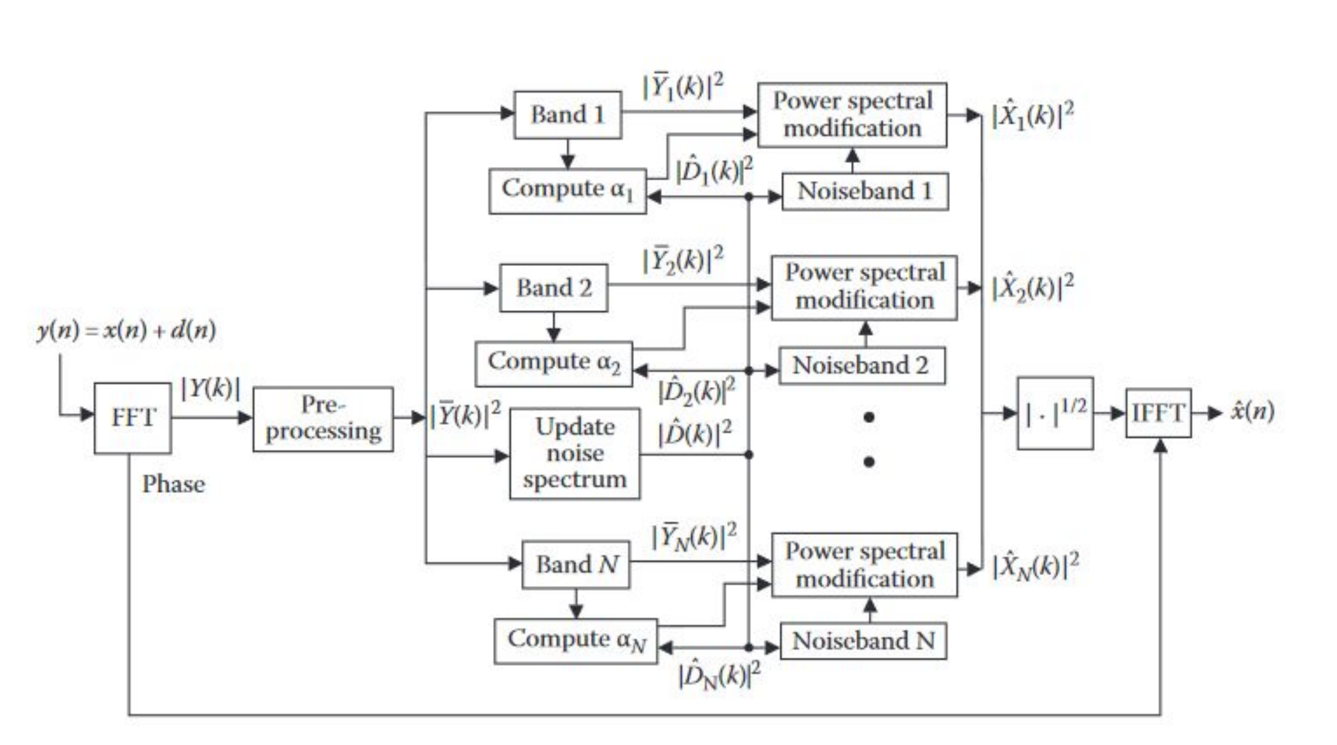
- The fundamental idea is subtracting the noise estimate from the noisy signal and returning a clean signal. (Spectral Subtraction).

- Subtraction problem can also be interpreted as a multiplication problem.
- Instead of subtracting 2 spectra
  a. Compute a gain function using the noise estimate.
  b. Multiply this gain function with the original noisy signal to create the clean signal estimate.
- Posed as an error estimation problem. The optimal filter {H} is sometimes called a “Wiener Filter”

$$|X(\omega)| = H(\omega) \cdot |S(\omega)|$$

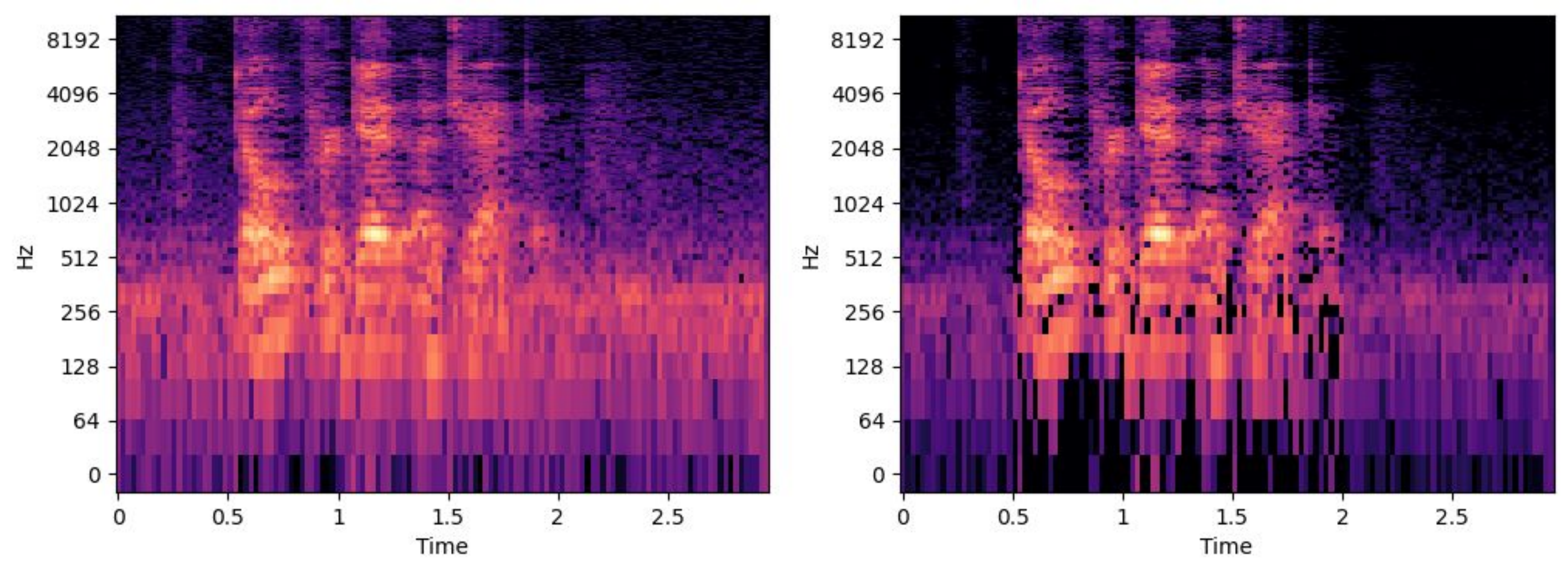
$$H(\omega) = \left(\frac{|S(\omega)|^2 - \alpha |\hat{D}(\omega)|^2}{|S(\omega)|^2} \right)^{1/2}$$

A Spectral subtraction block diagram



Loizou, Philipos C. *Speech Enhancement: Theory and Practice*. 2nd ed. CRC Press, 2013.

https://doi.org/10.1201/b14529.


Hz
8192
4096
2048
1024
512
256
128
64
0
0
0.5
1
1.5
2
2.5
Time
Hz
8192
4096
2048
1024
512
256
128
64
0
0
0.5
1
1.5
2
2.5
Time

What about machine learning?

Complex Ratio Masking (Williamson, Donald S., et al. “Complex Ratio Masking for Joint Enhancement of Magnitude and Phase.” 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 5220–24. IEEE Xplore, https://doi.org/10.1109/ICASSP.2016.7472673)

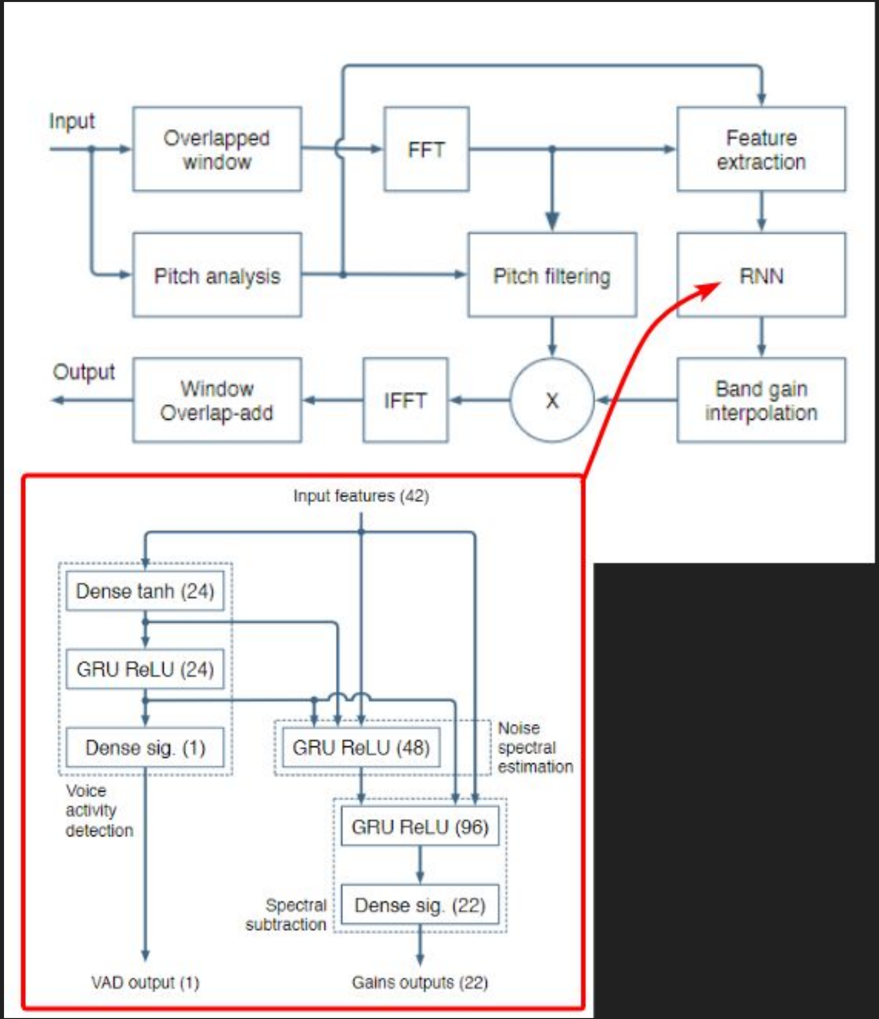
- Complex Ratio mask is closely related to Wiener filters in the complex domain.
    - Ideal masks are hard to generate but can jointly enhance magnitude and phase.
- Proposes using a simple Fully Connected Deep Neural Network to jointly estimate the complex and real coefficient of the complex ideal ratio mask.
- This mask is then multiplied to the dirty signal and compared with the clean target signals during training.

A Fully Convolutional Neural Network for Speech Enhancement (Park, Se Rim, and Jinwon Lee. September 22, 2016. https://doi.org/10.48550/arXiv.1609.07132.)

- Uses Convolutional Network in an Encoder-Decoder like architecture to estimate a “mapping” from the spectrum of a noisy speech signal to the spectrum of a denoised speech signal.
- Does not make use of fully connected layers so it is more computationally suitable for real-time applications
- Does not attempt to estimate phase only magnitude of the noisy spectrum.

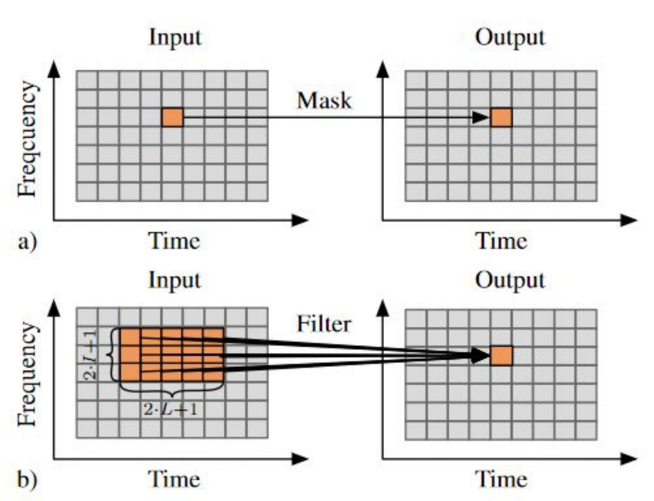
The Hybrid Approach (Valin, Jean-Marc. “A Hybrid DSP/Deep Learning Approach to Real-Time Full-Band Speech Enhancement.” arXiv, May 31, 2018. https://doi.org/10.48550/arXiv.1709.08243.)

- Uses a Recurrent Neural Net (RNN) to estimate “critical gain bands” for ranges of frequency
  - Gain coefficients are easier to calculate than entire spectrograms.
- Applies DSP pitch filtering after gains are applied to reduce musical noise.
- Assumed the Network is smart enough to learn what the different model components are responsible for



Deep Filtering(Mack, Wolfgang, and Emanuël A. P. Habets. “Deep Filtering: Signal Extraction and Reconstruction Using Complex Time-Frequency Filters.” IEEE Signal Processing Letters 27 (2020): 61–65. https://doi.org/10.1109/LSP.2019.2955818.)

- Calculates a ratio mask for each time and frequency bin in an stft using an area around it.
- Filter mask is applied and directly evaluated against the given clean speech bin during training serving as the loss function
- Adds notch filters and zeroed time frames to the noisy signal during training to teach the system to reconstruct
- Uses BiDirectional LSTM DNN



And many more…

Hu, Yanxin, Yun Liu, Shubo Lv, Mengtao Xing, Shimin Zhang, Yihui Fu, Jian Wu, Bihong Zhang, and Lei Xie. “**DCCRN: Deep Complex Convolution Recurrent Network for Phase-Aware Speech Enhancement,**” August 1, 2020. https://doi.org/10.48550/arXiv.2008.00264.

Braun, Sebastian, Hannes Gamper, Chandan K. A. Reddy, and Ivan Tashev. “**Towards Efficient Models for Real-Time Deep Noise Suppression.**” arXiv, May 19, 2021. https://doi.org/10.48550/arXiv.2101.09249.

Jia, Xupeng, and Dongmei Li. “**TFCN: Temporal-Frequential Convolutional Network for Single-Channel Speech Enhancement.**” arXiv, January 3, 2022. https://doi.org/10.48550/arXiv.2201.00480.

Schröter, Hendrik, Alberto N. Escalante-B., Tobias Rosenkranz, and Andreas Maier. “**DeepFilterNet: A Low Complexity Speech Enhancement Framework for Full-Band Audio Based on Deep Filtering.**” arXiv, February 1, 2022. https://doi.org/10.48550/arXiv.2110.05588.

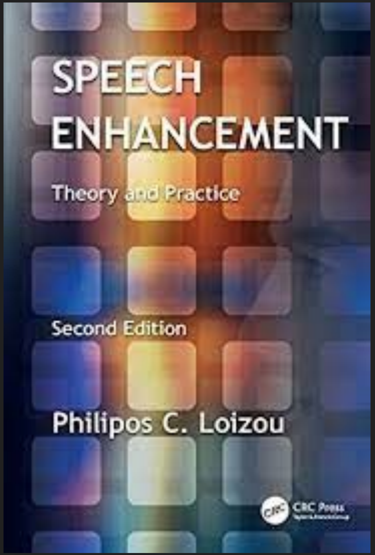
Richter, Julius, Simon Welker, Jean-Marie Lemercier, Bunlong Lay, and Timo Gerkmann. “**Speech Enhancement and Dereverberation with Diffusion-Based Generative Models.**” arXiv, August 11, 2022. https://doi.org/10.48550/arXiv.2208.05830.

Welker, Simon, Julius Richter, and Timo Gerkmann. “**Speech Enhancement with Score-Based Generative Models in the Complex STFT Domain.**” arXiv, July 7, 2022. https://doi.org/10.48550/arXiv.2203.17004.

Rethage, Dario, Jordi Pons, and Xavier Serra. “**A Wavenet for Speech Denoising.**” arXiv, January 31, 2018. https://doi.org/10.48550/arXiv.1706.07162.

So where am i?

- I had 0 knowledge about DSP
    - Read about 60% of this book
    - Had like 5% knowledge of DSP after.
- Spent a week trying to implement algorithms presented but found it difficult to translate the math to code.
    - This was aided with libraries
        - **Librosa**
        - **Numpy**
    - No previous experience with complex values
- Finally managed to get a working filter using spectral subtraction
    - Better understanding achieved
    - Made re-reading the presented algorithms easier.



- Have collected data sets of clean speech (no noise), and background noise.
  - Learning how to implement the filters gave me a better idea of how to mix the noise and clean signals to create a training set.
  - UrbanSound8K (https://urbansounddataset.weebly.com/download-urbansound8k.html)
    - Different categories noise samples
      - Restaurant, babble, city sounds, etc.
  - Noisy speech database for training speech enhancement algorithms and TTS models(https://datashare.ed.ac.uk/handle/10283/2791)
    - Using the clean dataset only
    - Recorded at higher fidelity. (48khz)
    - 56 different speakers for model robustness
- Have a leftover Keras CNN model used for a previous speech emotion detection project.

where to?

- Possible to repurpose mentioned model for this task
  - Would require modifying layer structure if i wanted to do block-wise inference.
- Possible to just create a model from scratch.
  - Fully Convolutional Nets look promising based on the papers i’ve read.
- Possible to just drop the Machine learning idea all together
  - Instead work on implementing more optimizations to classical dsp algorithms
  - Create a module with a couple different algorithms to choose from
  - Would allow for more time to create a well rounded “program” with “features”
    - Have experience building webapps wouldn't be too hard to build a simple gui frontend

¯_(ツ)_/¯

Thank you