SPEECH ENHANCEMENT: AN INVESTIGATION WITH RAW WAVEFORM

Yujia Yan
University Of Rochester
Electrical And Computer Engineering

Ye He
University Of Rochester
Electrical And Computer Engineering

ABSTRACT

Speech enhancement has a vast amount of demand in many areas. Previous works were usually formulated using time-frequency representations. Time-frequency representation has two limitations: firstly, it is a trade-off to choose between time and frequency resolutions; secondly, phase information is usually discarded and is very difficult to work with. This project serves as an investigation for building a system that operates on raw audio waveforms directly. We proposed a lattice-ladder structured neural networks with gated dilated convolutional layers as its basic building block. We performed training on the dataset we built, with a lot of operations for data augmentation. We evaluated this system with unseen speeches, unseen noises with unseen room impulse response. Our results indicate that this approach is able to produce better speech for low input quality. Due to limited time and resources and high computational burden, many properties of this kind of systems are still remained for further investigation.

1. INTRODUCTION

Real world speeches are noisy. Increasing the overall quality, at least intelligibility has a vast demand nowadays, in areas such as communications, hearing aids, speech recognition and content production, etc. The goal of our project is to explore both traditional statistical spectrum domain methods and methods formulated with neural networks for speech enhancement.

Speech Enhancement is traditionally formulated as a source separation problem, ie., separating clean speech part from its mixture with noises. Due to the the approximate (w-)disjoint orthogonality of speech signal [12], which corresponds to the assumption that speech signal can be separated by masking the spectrogram, methods using time-frequency representations are prevalent. Masks can be estimated with either statistical estimation [3] [7] [1] or a neural network [6] [14].

With the increasing popularity of convolutional neural networks which is designed and restricted to learn time-invariant operators, and with the idea of building something from scratch (Tabura Rasa), some attempts [8] [10] have been made trying to directly work on time domain with raw audio waveforms and without any notion of the well-established set of basis, namely, the Fourier Transform. Directly working on the time domain may have the potential to overcome the limits (time-frequency uncertainty, phase reconstruction, etc.) of using a time-frequency representation. However, training a system of this type is time consuming which requires a huge amount of resources.

In this project, we made an investigation in this direction. We proposed a lattice-ladder structured neural network inspired by the IIR lattice filter implementation. We performed training for this system on the dataset we built from varies sources, with diversified speech quality.

This paper is structured as follows: In section 2, we give a description on the systems we propose and implement in this work. In section 3, We talk about the datasets and data augmentation process we used we present evaluations on algorithms we implemented.

2. ALGORITHM DESCRIPTION

2.1 Wiener Filtering

We implemented a spectral domain Wiener filter to work as our baseline method. Wiener filter gives an estimate of power spectrum which has the minimum mean square error (MMSE) to the target signal. MMSE is more suitable for speech signal, compared with directly subtracting estimated amplitude spectrum of noise(can be over-subtracted), since large errors will be reduced more and small errors will be reduced less. Human ears may not be sensitive to the small errors, which in turn, less artifacts will be introduced.

For filtering out independent and additive noise, the frequency response of the filter is given by:

$$H(\Omega) = \frac{P_{xx}(\Omega)}{P_{yy}(\Omega)} \tag{1}$$

where $P_{xx}(\Omega)$ is the power spectral density of the signal x. Hence, the spectrum of the estimated signal is

$$S(\Omega) = H(\Omega)Y(\Omega) \tag{2}$$

where $S(\Omega)$ is the spectrum of the estimated signal and $Y(\Omega)$ is the noisy signal.

We process with the above formula frame by frame. The estimated signal at time k and frequency bin m, $S_m(k)$ is

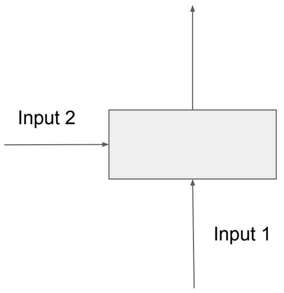

Figure 1: Gated Convolutional Layer

given by

$$S_m(k) = H_m(k)Y_m(k) \tag{3}$$

where

$$H_m(k) = \frac{P_{xx,m}(k)}{P_{yy,m}(k)} \tag{4}$$

However, the power spectrum of the clean signal, $P_{xx,m}$ is unknown. Therefore, we have to estimate it from signal. Equation 4 can be reformulated with SNR term [3]

$$H_m(k) = \frac{P_{xxm}(k)}{P_{xxm}(k) + P_{nnm}(k)} = \frac{\eta_m}{1+\eta_m} \tag{5}$$

where $\eta_m = \frac{P_{xx,m}(k)}{P_{nn,m}(k)}$, the Signal-to-Noise Ratio. Then the assumption is that one estimate of SNR actually executed at previous time is close to the target signal of the current frame. Then we have a smoothing equation for η_m

$$\eta_m = \alpha_\eta \frac{|S_m(k-1)|^2}{P_{nn,m}(k)} + (1-\alpha_\eta)\max(0, \gamma_m(k)-1) \tag{6}$$

where $\gamma_m(k) = \frac{P_{yy,m}(k)}{P_{nn,m}(k)}$ is a posteriori SNR and α_η is a smoothing parameter. The noise power spectrum P_{nn} is estimated directly by taking medians of all frames in the spectrogram.

2.2 Convolutional Lattice Neural Network

The proposed neural network structure is inspired by traditional lattice filters, which implements an IIR filter in a way that signal goes though a series of simple all pass sections, after which, the output of the filter is a linear combination of the outputs from these all-pass sections.

2.2.1 Gated Dilating Convlutional Layer

We incorporate similar idea as used by Wavenet [8], but the difference is that how we apply gating. The basic layer in our architecture uses dilated convolution without pooling. The dilated convolution is defined as

$$(x\ [*]_k\ y)[n] = \sum_m x[m]y[n-km] \tag{7}$$

where $[*]_k$ represents dilated convolution with dilating step k, which can be intuitively explained as convolving with skip step k. There are no downsampling operations after the convolution. Therefore the length of the input and output signal can be the same if zero paddings are used. This enables us to design a layer that has a highway/residual connection. Denote the two inputs and the output (shown in figure 1) of our layer as x_1, x_2, y respectively,

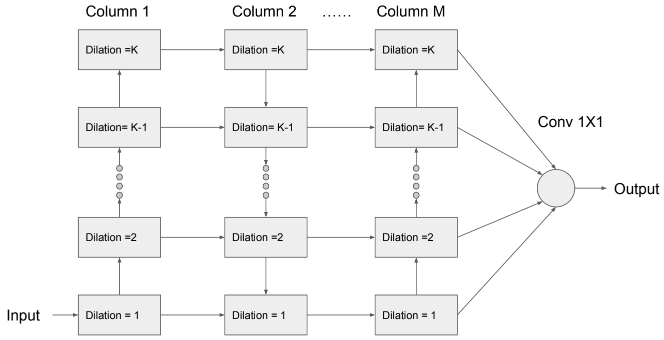

Figure 2: The lattice Architecture

$$\begin{aligned} g &= \sigma(w_1^{gate}[*]_k x_1 + w_2^{gate}[*]_k x_2 + b^{gate}) \\ \tilde{y} &= c\tanh(w_1^{out}[*]_k x_1 + w_2^{out}[*]_k x_2 + b^{out}) \\ y &= g \circ \tilde{y} + (1-g)\circ(x_1+x_2) \end{aligned} \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid function,c is the scale parameter, and $\circ$ is the element-wise product. g can be interpreted as the gate to determine which portions of the input and the transformed input should pass the layer.

2.2.2 The lattice-ladder Architecture

Our neural network architecture is shown in figure 2. In this architecture, we have M columns of dilating convolution chains with alternating directions. The dilation step k for each dilating convolutional layers is calculated as follows

$$k = base^{dilation-1} \tag{9}$$

the filter width in each convolutional layer is chosen according to $base$ such that it at least covers the entire span of $base$, which is essential to build the whole receptive field.

In addition, we have skip connections between consecutive columns to allow the signal bypass the lattice and make gradient back-propagating easier.

The outputs of each gated dilating convolution layers in the last column are concatenated and then passed through one length-1 convolution layer for obtaining the final output for this network.

Each column can be viewed as a neural network counterpart of a classical filterbank. Columnwisely, they form a multi-layered filterbank structure.

3. IMPLEMENTATION AND EXPERIMENT

3.1 Dataset

Our dataset have three pieces: *clean speech(clean)*, *additive noise(noise)*, and *room impulse responses(ir)*. Room impulse responses here are not served as convolutional

	Sources
CLEAN	100hours of Librispeech [9], THCHS-30 [2]
noise	MUSAN [13]
IR	MUSAN [13], Simulated Room IR [5]

Table 1: A simple table

noise we want to deduct(known as dereverberation), but as a way to make variations to noises.

We built our dataset from various sources. Table 1 gives details on where they comes from. We then reserve some samples from the whole dataset exclusively for generating validation and test data. Note that our data set includes both English and Chinese Speeches for training. However Chinese Speeches are not used for evaluation and they are simply a grub-and-place data for regularizing what is learned in the neural network.

3.2 Data Sampling

All samples are generated following procedure outlined in algorithm 1. Samples generated by this data augmentation algorithm are actual samples we use. During training, samples are generated on the fly in background threads. A queue with maximum size of 1000 is used for storing generated samples during training. A set of 500 samples is generated for validation and test set respectively with their exclusive raw samples.

We choose parameters in our data generation process in order to diversify the speech quality in our dataset, and to have a wide range in metrics we use.

Algorithm 1 Generating data from all pieces of data

```
procedure SAMPLEACLIP
    randomly select a clip of clean speech $x$
    perform pitch shifting and time stretching on $x$, with $ratio \sim U[0.9, 1.1]$
    sample $k \sim U[0, 18]$
    Initialize $n$ to be zero vector
    for i = 0: k do
        select a random noise clip
        perform pitch shifting and time stretching on $x$, with $ratio \sim U[0.9, 1.1]$
        sample a random room impulse response and apply it to the noise clip
        apply random spectral envelope to the noise clip
        add this clip to $n$
    end for
    Sample a SNR value
    mix $x$ and $n$ according to SNR
    sample a loudness value
    adjust the mixture to the loudness just sampled, adjust the clean speech clip accordingly
    return the clean speech clip and the final mixture
end procedure
```

3.3 Neural Network Training

We trained our neural network with 6 columns, dilate base 2 and dilate levels $k = 16$. Each convolutional layer outputs 8 channels. We applied Dropout and gradient noise for regularization. For training, block size of 3 seconds audio is directly fed into the neural network. Due to limited time and resources(ie., GPU memory, training time, etc.), we use Adam Optimizer with batch size 1. We use mean square error (MSE) as the objective function

$$\min_{\theta} \frac{1}{2} ||y_{GT} - f_{\theta}(x)||_2^2 / N \tag{10}$$

where y_{GT} is the clean speech, f is our system, θ is the parameters we want to optimize, and N is the length of points in the waveform. We also experimented on weighting the objective function with A-weighting Curve [4] and a combination with Kullback-Leibler divergence on spectrogram. However, it does not improve the result.

3.4 Evaluation Metrics

In this work, we use PESQ and SSNR as our metrics to evaluate the results. PESQ [11] is a standard evaluation methods. We use the wide-band version in its reference implementation which outputs a MOS-LQO (Mean Opinion Score - Listening Quality Objective) ranging from 1 to 5. The Segmental Signal-to-Noise Ratio (SSNR) used in this work is calculated by firstly framing the signal, secondly calculating the SNR frame by frame, and then averaging certain frames that are within the range of $[-10, 35]db$.

3.5 Results and Discussion

Unlike most works on speech enhancement, we do not evaluate the system with the mean of metrics on selected data set: we are interested in how the quality of the output will change according to different levels of the input quality. Our results are shown in figure 3 for PESQ, and figure 4 for SSNR. From the result we can see that our neural network approach performs better when the quality of the input is low. Performances of both methods drops with increasing input quality. This phenomenon is caused by the imperfection of reconstruction. From our observation, the degeneration of quality is due to the loss of high frequency components in the denoised version produced by our neural network. It may have three causes: firstly, the model size we use(limited by time and resources we have) may not have enough capacity resulting in under-fitting of our model; secondly, the model is not well trained(also limited by time and resources we have); thirdly, the MSE objective penalizes too much for the low frequencies and for a dataset with many samples of extremely low quality, it may be more conservative to focus more on the low frequency components.

4. CONCLUSION

In this project, we proposed a gated dilated convolutional lattice-ladder neural network for speech enhancement, which works directly on raw audio waveforms. We

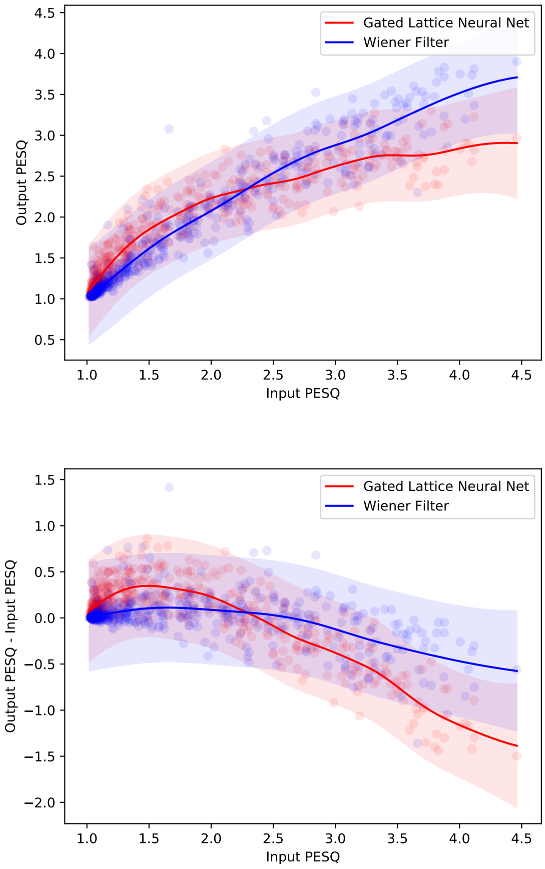

Figure 3: PESQ results: raw PESQ score versus input PESQ(above), PESQ improvement versus input PESQ(below)

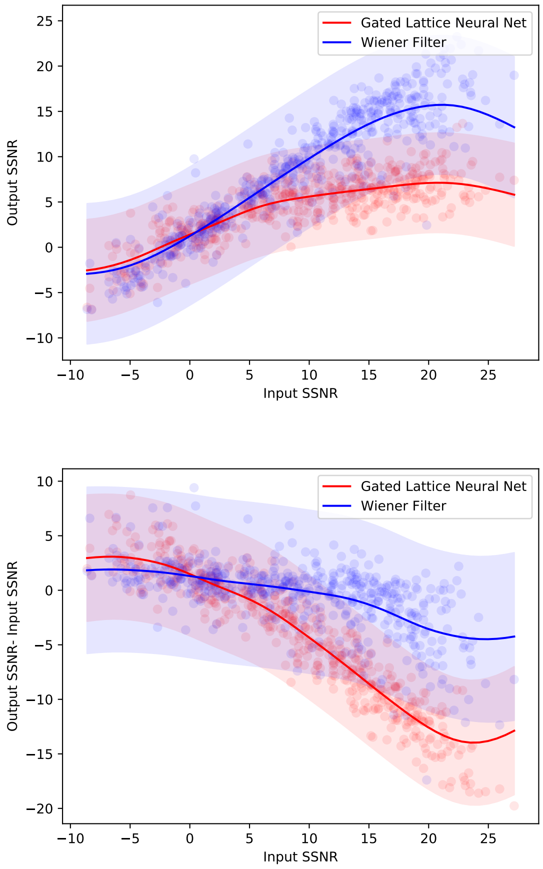

Figure 4: SSNR results: raw SSNR score versus input SSNR(above), SSNR improvement versus input SSNR(below)

trained and evaluated this system with the dataset we built that has a wide range of quality. The result produced on unseen speeches, unseen noises with unseen room impulse responses suggests that our proposed model is able to outperform our baseline Wiener filter for inputs with low quality. Operating directly on raw audio waveforms is still remained for further investigation.

5. REFERENCES

[1] Israel Cohen. From volatility modeling of financial time-series to stochastic modeling and enhancement of speech signals. *Speech enhancement*, pages 97–113, 2005.

[2] Zhiyong Zhang Dong Wang, Xuewei Zhang. Thchs-30 : A free chinese speech corpus, 2015.

[3] Yariv Ephraim and David Malah. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(6):1109–1121, 1984.

[4] IEC IEC. 61672: 2003: Electroacoustics–sound level meters. Technical report, Technical Report, IEC, 2003.

[5] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5220–5224. IEEE, 2017.

[6] Xugang Lu, Yu Tsao, Shigeki Matsuda, and Chiori Hori. Speech enhancement based on deep denoising autoencoder. In *Interspeech*, pages 436–440, 2013.

[7] Rainer Martin. Statistical methods for the enhancement of noisy speech. *Speech Enhancement*, pages 43–65, 2005.

[8] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[9] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5206–5210. IEEE, 2015.

[10] Santiago Pascual, Antonio Bonafonte, and Joan Serrà. Segan: Speech enhancement generative adversarial network. *arXiv preprint arXiv:1703.09452*, 2017.

[11] ITU-T Recommendation. Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. *Rec. ITU-T P. 862*, 2001.

[12] Scott Rickard and Ozgiir Yilmaz. On the approximate w-disjoint orthogonality of speech. In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 1, pages I–529. IEEE, 2002.

[13] David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

[14] Yan Zhao, Zhong-Qiu Wang, and DeLiang Wang. A two-stage algorithm for noisy and reverberant speech enhancement. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 5580–5584. IEEE, 2017.