AN AUTOENCODER BASELINE IN CHANNEL NORMALIZATION SYSTEM

Ge Zhu
University of Rochester
Electrical and Computer Engineering

ABSTRACT

Channel normalization system aims to remove the channel effects, presented in speech signals, e.g. quantization noise in coder or speech distortion, such system is able to increase the robustness of automatic speech verification systems. However, to the best of our knowledge, previous works were usually engaged with additive noise or convolutional noise. This project first creates a channel corrupted speech dataset and applies an autoencoder neural network architecture to serve as a baseline for deep-learning based channel normalization.

1. INTRODUCTION

Automatic speech verification (ASV) is a significant problem in speech enhancement and is essential part to automatic speech identification (ASI). Usually ASV systems use speech signal to generalize parameterized features to represent speakers individually based on various models representing different properties. ASV in varying conditions is a challenging problem since clean speech seldom present in reality, some problems may result from additive noise and convolutional, while some result from differences between channels.

In commercial ASV systems, speech signals are usually recorded through communication devices of various qualities like telephones, cell phones, laptops etc., and different types of devices have different bandwidths and codec standards. On purpose of transmission across different channels, audio codecs algorithms are usually utilized to convert analog signals into digital signals and then perform compression. Different channels usually have different codec standards. To cover a wider users, a lower bitrate codec are offen applied, but at the same time, it will introduce distortions and packet loss. The above mentioned conditions may cause various channel distortions and further affect the performance of ASV.

Neural network based ASV system have been used to minimize channel effects for years, this method is able to achieve great performance as long as training data and test data have similar properties or channel distortions have already been set as prior knowledge. However, in real world, ASV usually encounter diverse acoustic situations, and channel mismatch often happen between training data and test data.

The most natural way to solve this mismatch problem is to develop a supervised method by dividing channel effects removal task into recognition step and compensation step separately, specifically, train a universal channel classifier first and compensate for them correspondingly.

In this project, we first applied an acoustic-simulator to simulate channel corruption on a clean speech signal and thereby created a channel corrupted speech dataset. Inspired by homework 5 in the course, we develop an autoencoder neural network to compensate for channel distortion and set this architecture as a baseline for future research.

This paper is organized as follows, in section 2, we introduce the process of generating channel corruption dataset. Section 3 briefly describes neural network architecture. Details on experimental results are given in section 4. Section 5 gives the conclusion about this project.

2. GENERATING CHANNEL CORRUPTION SPEECH DATASET

A channel distortion simulator can be treated as an ideal and affordable way of generating a corrupted channel speech dataset. In acoustic simulator [3], 12 different speech codecs are included. These comprise mainly four types of codec standards: landline, cellular, satellite and Voice over Internet Protocol (VoIP). For each of these conditions, codec parameters such as bit rate, dtx or packet loss as well as noise recordings, and device impulse responses can be specified based on requirements. Figure 1 shows a codec-corruption only block diagram of the acoustic simulator.

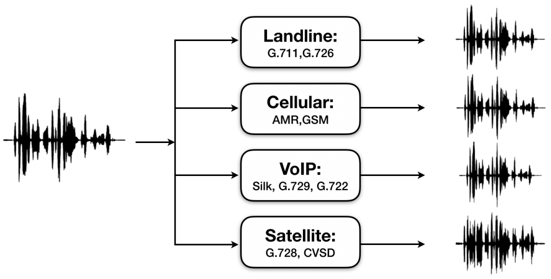

Figure 1: Diagram for Acoustic Simulator.

We first apply this acoustic simulator to CSTR VCTK corpus to generate 14 different codec corruptions corresponding to one clean speech.

3. METHOD: AUTOENCODER

Inspired by homework 5 [5] in computer audition class, we will use autoencoder neural network to serve as baseline system for removing channel corruption in this project.

Similar to codec algorithms, autoencoder neural networks first learn to compress input data into code and then uncompress this code back, reconstruct the original data and simutaneously remove corruption or noise. The aim of this autoencoder is to learn robust representations from the data.

In mathematical notation [4], the encoder and the decodercan be defined as transitions ϕ and ψ, such that:

$$\begin{aligned}&\phi : \mathcal{X} \mapsto \mathcal{F}\\&\psi : \mathcal{F} \mapsto \mathcal{X}\\&\phi, \psi = \underset{\phi,\psi}{\operatorname{argmax}} \, ||X - (\phi \circ \psi)X||^2\end{aligned}$$

In source separation task in homework 5, we are trying to estimate a mask function for mixture for further filtering, this whole framework can also be adapted in additive noise speech enhancement problem.

However, in our case, codec involves data compression and decompression, the resulting channel corruption is therefore nonlinear. For each time-frequency (T-F) bin in corrupted speech, we are unable to infer to some degree it's noise or simply determine whether it is or not. Due to this change, we have to change the loss function to fit the new problem.

The simplest autoencoder form consists of two parts, encoder and decoder which are basically multi-layer perceptrons (MLP).

Since pure fully connected layers are unclear to deal with time series data, autoencoder with recurrent neural networks (RNN), especially long-short term memory (LSTM), is a solution to this problem, this architecture has a memory block to store sequential information or temporal information.

4. IMPLEMENTATION AND EXPERIMENT

4.1 Dataset

For the corruption dataset, we applied continuously variable slope delta (cvsd) modulation with 128kb/s. And due to limited time, our orignal clean dataset only contains 6626 pieces of few-second-long speech sampling at 48000Hz from 8 different people, of all the speaking corpus, 2995 pieces are for training, 1225 for validation and 1631 for test.

4.2 Preprocessing

In commercial use, speech signals are usually sampled at 8kHz or 16kHz, which can speed up processing and also training in our case. In STFT step, our window length is 0.064s and hop size is 0.032s. To better estimate uncorrupted speech, we also perform log to the original STFT spectrogram magnitude.

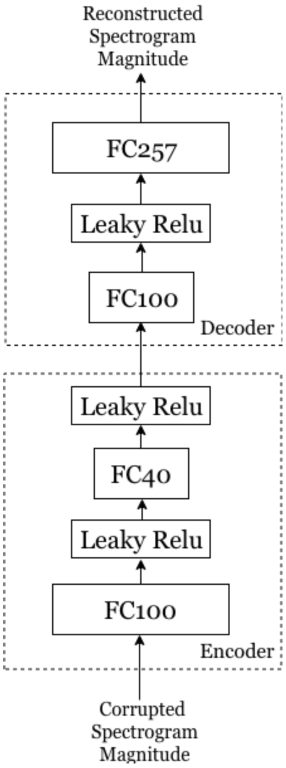
Reconstructed Spectrogram Magnitude
FC257
Leaky Relu
FC100
Decoder
Leaky Relu
FC40
Leaky Relu
FC100
Encoder
Corrupted Spectrogram Magnitude

Figure 2: Fully Connected Autoencoder.

4.3 Neural Network Architecture

Figure 2 and Figure 3 shows the simplest autoencoder and RNN model architecture separately.

In feedforward autoencoder architecture, the encoder receives chunks of magnitude of a short-time Fourier transform (STFT) of the corrupted speech, it is consisted of two fully connected layer with dimension of 100 and 40 separately while the decoder consists of two fully connected layers with dimension 100 and 257. In both encoder and decoder, we choose rectified linear unit (ReLU) as activation layer. But for the autoencoder output, since we also want a spectrogram magnitude output, we choose a linear output layer instead of sigmoid activation layer to avoid bounding values between 0 and 1.

In RNN autoencoder architecture, the additional LSTM layer takes the output of encoder as input with dimension 20; another two-layered fusing neural network concatenates the output of LSTM layer and encoder with dimension of 120 and 40.

4.4 Training

In both of the two neural network architecture training, we feed 64 time frames (around 1s) by 257 frequency bins with batchsize of 256 at a time into the network. During training, we applied dropout with a rate of 0.025 for regularization.

To train the networks, we consider mean square error

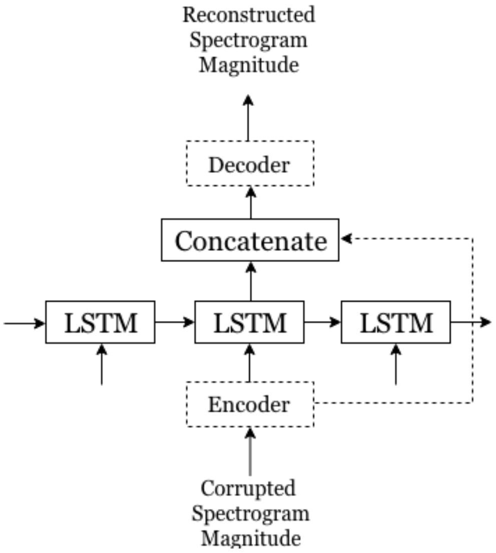
Reconstructed Spectrogram Magnitude
Decoder
Concatenate
LSTM
LSTM
LSTM
Encoder
Corrupted Spectrogram Magnitude

Figure 3: Fully Connected Autoencoder with LSTM layer.

(MSE) as loss function with a learning rate of 1e-3, L1 loss function can also be applied here. The MSE objective function minimize the reconstruction error of the T-F bins of the speech source.
Due to limited time and computation resources, we only train the two autoencoder networks on a CPU both for 100 epoches, and during each epoch we perform 500 iterations.

4.5 Evaluation Metrics

We use perceptual evaluation of speech quality (PESQ) [1] and short time objective intelligibility (STOI) [2] to perceptually evaluate our baseline channel corruption removal system. Both are common metrics in speech enhancement and have positive correlation with speech intelligibility.

5. RESULTS AND DISCUSSION

Figure 4 and Figure 5 show the training loss and validation loss.

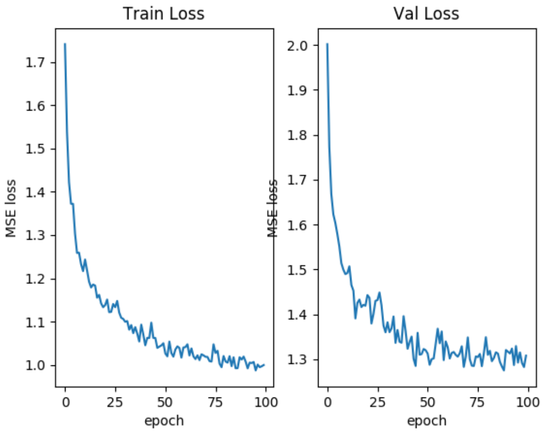

Figure 4: MSE Loss in Autoencoder

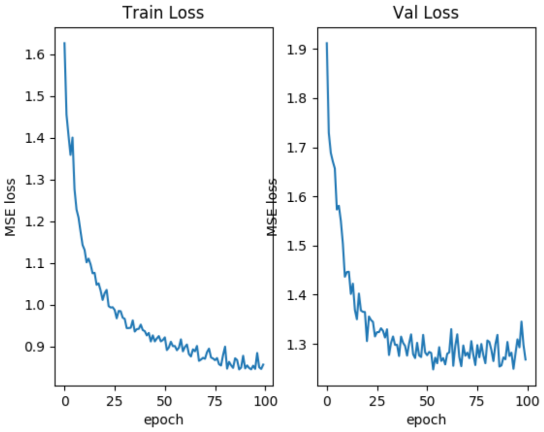

Figure 5: MSE Loss in Autoencoder with LSTM layer

Within this limited training epoches, we can find that in these two architecture, both training loss and validation loss are decreasing, which means that further training are needed. It also can be seen that it converges slowly in later epoches. Techniques are needed to speed up the this converging. After training, the recovered magnitude of spectrogram from the corrupted speech in the first epoch and 100th epoch are separately shown in Figure 6 and Figure 7.

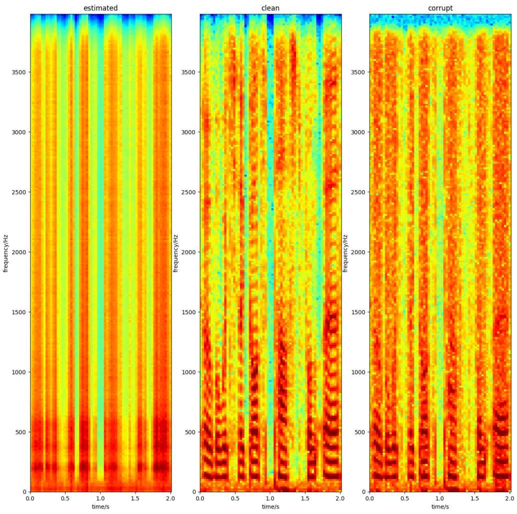

Figure 6: Estimated Spectrogram in First Epoch

Differences between first epoch and last epoch are apparent by comparing Figure 6 and Figure 7. We can see that in the first epoch, this encoder-decoder network tries to learn an approximate shape from the clean speech and after many epoches, more details are filled in the spectrogram. It is expected that after relative long epoches, the estimated spectrogram would be close enough to the clean speech, but further optimization should be applied to speed up this training.

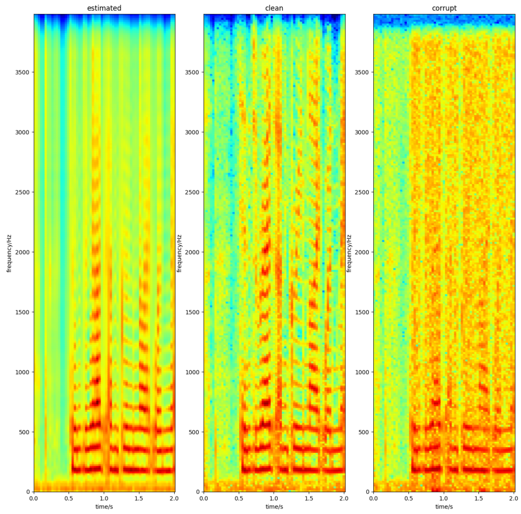

Figure 7: Estimated Spectrogram in Last Epoch

Figure 8 and Figure 9 show the PESQ improvement and STOI on test set after 100 epoch training. From the PESQ curve and STOI curve, we can see the improvement from corrputed speech to recovered speech to some degree. The average PESQ improvement in simple autoencoder is 0.46 and 0.49 for autoencoder with LSTM layer.

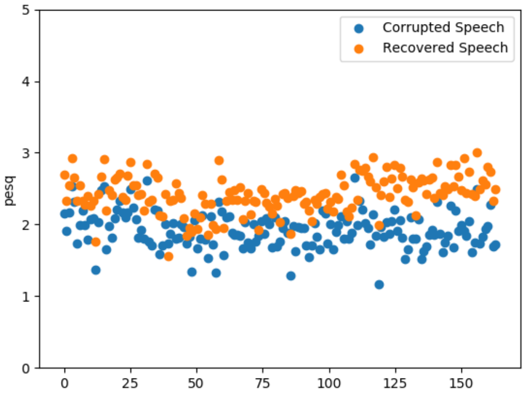

Figure 8: PESQ

6. CONCLUSION

In this project, we develop an autoencoder baseline to compensate for a certain channel corruption. We trained and evaluated this system on our created dataset. Results show that this model needs further training and parameters tuning.

Since this project only develops a baseline for channel compensation, much more work should be done for future work, a better neural network architecture for more generalized dataset containing various types of channel compensation, such as convolutional encoder-decoder (CED) network. Also, device impulse responses are another big issue involved in the problem.

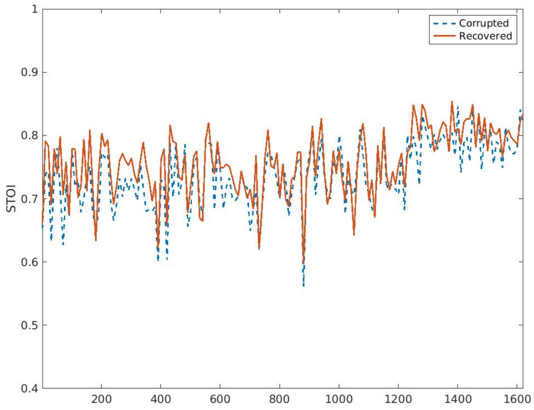

Figure 9: STOI

7. REFERENCES

[1] Antony W. Rix et.al. *Perceptual Evaluation of speech quality (PESQ) - A new Method for Speech Quality Assessment of Telephone Networks and Codecs*. ITU-T Recommendation.862, ITU, 2001.

[2] C.H. Taal et.al. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. In *IEEE Transactions on Audio, Speech, and Language Processing*, pages 2125 – 2136, 2011.

[3] M. Ferras et.al. A large-scale open-source acoustic simulator for speaker recognition. *IEEE Siganl Processing Letters*, 23(4):527–531, 2016.

[4] Wikipedia. *Autoencoder*. Free Encyclopedia, https://en.wikipedia.org/wiki/Autoencoder, 2018.

[5] Y. Yan and Z. Duan. *HW5: Singing voice separation with neural networks*. U of R Press, Wilmot, 2018.