Speech Enhancement based on Stacked Denoising Autoencoder with Wiener Filter

Chu Man Chung
Department of Electrical and Electronic Engineering, The University of Hong Kong

1. Abstract

In this research, a speech enhancement and noise removal model based on Stacked Denoising Autoencoder(SDAE), Signal to Noise Ratio Estimation(SNR) and Frequency Domain Wiener Filter is proposed. The performance evaluation shows that the model improves the speech quality by 0.7 PESQ scores. In conclusions, the neural network based method can provide a satisfactory noise removal and enhance the speech for speech recognition system.

Index Terms: Speech Enhancement, Autoencoder, Deep Neural Network

2. Introduction

For the mobile communication application and speech recognition system which operate in relatively complex environments, background noise is severe and hinder the speech recognition systems from recognizing the speech correctly. As a result, it is necessary to apply noise reduction technic to remove the background noise and improve the speech quality.

Model background noise reduction technic can back to 1970s when several powerful algorithms have been developed to remove noise, such as frequency domain Wiener Filter proposed by J. Lim and A. V. Oppenheim, MMSE Short Time Spectral Amplitude estimator proposed by Y. Ephraim and wavelet thresholding method developed by D. L. Donoho. These noise removal algorithms base on either the additive nature of background noise or the statistical difference between the noise and speech. However, with the fact that the relation between noise and speech is complex, it is more reasonable to apply an adoptive and non-linear model for finding the relationship between speech and noise in time and frequency domain. Neural network has these properties and as a result it has long been thought as better algorithms to enhance the speech quality.

Neural network has many different types of model which are suitable for different tasks and the one for noise reduction is called Denoising Autoencoder (DAE) which is a special version of Autoencoder and first introduced in 2008. Since adopting neural network requires huge computational power which could not be achieved in the past, DAE has showed its robustness in removing noise and enhancing speech quality in recent year with the advanced technology in graphic processing unit.

Goals of the research include:

- To design a model based on SDAE
- To maximize the background noise reduction
- To minimize the distortion between restored speech and clean speech
- To maximizer perceptual quality of estimated clean speech

Since most automatic speech recognition systems available online have some kind of noise reduction technics, it may be inaccurate to conduct experiment and get the Word Error Rate (WER) with the speech recognition system and the noise reduction model proposed in this research. Therefore, this research will focus more on speech enhancement part and Perceptual Evaluation of Speech Quality Improvement (PESQI) based on ITU-T Recommendation P.862 Standard, Log-Spectral Distortion (LSD) and Noise Reduction are proposed to evaluate the performance of model. Since this research focus on enhancement of speech and ability of noise removal, the model's ability on improving the automatic speech recognition(ASR) system's performance is undetermined.

The paper is organized as follow. First, the theory of methods, such as STFT, MFCC and SDAE, are introduced in part two. Second, two unsuccessful models proposed for the project are briefly introduced, evaluated and discussed in part three. Since two of them give relatively bad result, only the final model has detail comparison with different parameters. Third, the performance of the final models is discussed and evaluated with three evaluation criteria, PESQI, LSD and Noise Reduction. Finally, a conclusion of the research is drawn.

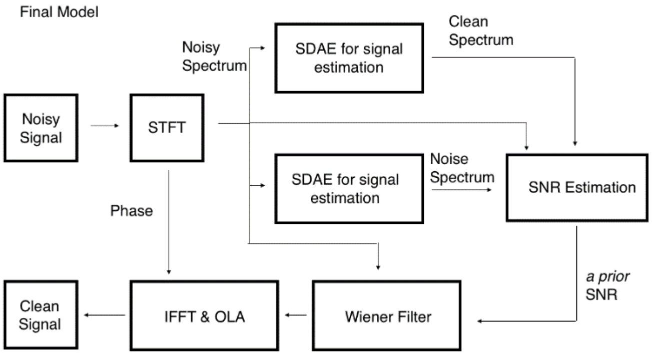


Figure 1: The proposed model

3. Methodology

In this part, some methods of the proposed model in Figure 1 will be introduced in order to provide solid ground to the models.

In addition, the removing additive noise is the main goal of the research and other types of noise are not discussed.

The nature of additive noise:

$$y[n] = x[n] + n[n] \ \ (1)$$

$$Y^2[\lambda,\delta] = X^2[\lambda,\delta] + N^2[\lambda,\delta] \ \ (2)$$

where n, x and y are the noise, clean and noisy speech.

3.1 Data Base and Noise

The data base, from Marathon Match - Contest: Spoken Languages 2, contain 57560 recorded speech files in 176 languages and they totally last for about 160 hours. Only 9600 files are selected for training and testing.

The noise file consists of car as well as road noise and lasts for 33 seconds. Since the noise is longer than the speech, a portion, about 9 to 10 seconds, of noise file is randomly picked and added to the speech. As a result, for all speech file, their background noise will be different.

3.1 STFT

STFT is one type of Fourier Transform and the difference between STFT and original Fourier transform is that STFT consider the non-stationary property of audio signal whose spectral content, or frequency content, changes over time. Since the Fourier coefficients are time-invariant function, applying original Fourier transform does not reveal the transitions in spectral content. To avoid this issue, the Fourier transform should be applied over a short time period that the audio speech is assumed stationary.

Usually, the audio signal is broken into 20ms-40ms frames with 50% or 75% overlap and the overlap is used to reduce artifacts at the boundary. Then, a window function, usually Hann and Hamming windows, is applied to each frames in order to provide a better frequency response since each frame is smoother and has less ripples. Then, Fourier transform is applied to each frame.

$$Hann\ Windows: w[\lambda] = 0.5 - 0.5\cos\left(\frac{2\pi\lambda}{N-1}\right) \ \ (3)$$

$$\begin{aligned} Hamming\ Windows: w[\lambda] \\ = 0.54 - 0.46\cos\left(\frac{2\pi\lambda}{N-1}\right) \ \ (4) \end{aligned}$$

where $1 \le \lambda \le N$ and N is the size of frames.

Here is the mathematics definition of discrete STFT.

$$X[\lambda,\delta] = \sum_{m=-\infty}^{\infty} x[m]w[\lambda - m]e^{-j\delta\lambda} \ \ (5)$$

3.3 Spectrum

Spectrum of STFT coefficients is a commonly input type in speech enhancement task.

$$Input = Y^2[\lambda,\delta] = y \ \ (6)$$

$$Desired\ Output = X^2[\lambda,\delta] = x \ \ (7)$$

However, the input and output is large numerically and the scale of difference between Input and Desired Output range from $10^1\ to\ 10^{10}$. At the beginning of training, the actual output $\hat{x}$ may be greatly different from the desired output when the speech is not presence in clean speech. For example, $x \approx 0$ because speech is not presence while $y \approx 10^{10}$ and $\hat{x} \approx 10^9$ because the noise is presence.

Therefore, the update of weight and bias will be large. Reference to back propagation training algorithm [11]:

$$\Delta W_{l-1} = \varepsilon\frac{\partial L}{\partial W_{l-1}} = \varepsilon\frac{\partial L}{\partial a_{l-1}}\frac{\partial \hat{x}}{\partial W_{l-1}} = 2\varepsilon(\hat{x} - x)h^T \ \ (8)$$

$$\Delta b_{l-1} = \varepsilon\frac{\partial L}{\partial b_{l-1}} = \varepsilon\frac{\partial L}{\partial a_{l-1}}\frac{\partial \hat{x}}{\partial b_{l-1}} = 2\varepsilon(\hat{x} - x) \ \ (9)$$

where l is the number of layers.

If $\hat{x} - x$ is large, the update of $\Delta W_{l-1}\ and\ \Delta b_{l-1}$ will also be large. However, $\hat{x} - x$ may be small for when the speech is presence. This problem will need to unstable update of $\Delta W_{l-1}\ and\ \Delta b_{l-1}$ and finally the training loss will go to infinity.

In order to solve the unstable training:

$$Input = log_{10}(Y^2[\lambda, \delta]) = y \quad (10)$$

$$Desired\ Output = log_{10}(X^2[\lambda, \delta]) = x \quad (11)$$

Taking logarithm on input can minimize the scale of difference between Input and Desired Output and solve this problem.

3.4 Feature Expansion

For the SDAE estimating the clean spectrum, feature expansion on input is adopted while the desired output remains the same.

$$Input = [y[\lambda - 1], y[\lambda], y[\lambda + 1], n[\lambda]] \quad (12)$$

$$Desired\ Output = x[\lambda] \quad (13)$$

The previous and following frame of noisy input are taken into consideration because they may have some information needed to estimate the current frame.

In addition, the input is expended by feeding the estimated noise spectrum based on the thought that it is beneficial to give information about the noise as the input to the neural network speech recognition [3]. This expansion is called noise aware and the noise spectrum is estimated by the SDAE estimating noise spectrum which refer to section 3.7.

3.5 SDAE

Before introducing SDAE, a brief description of Autoencoder (AE) and Denoising Autoencoder (DAE) is given

3.5.1 AE

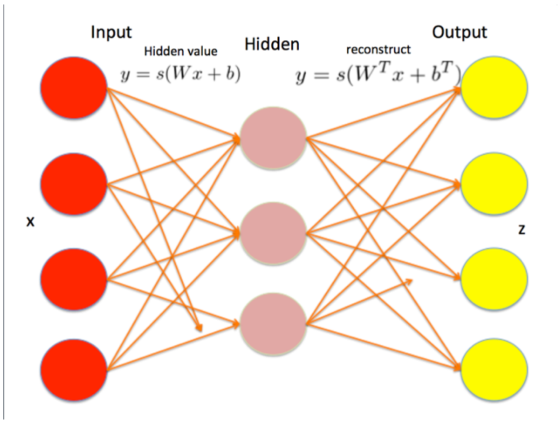


Figure 2: Auteocoder from Deep Learning Tutorials Denoising AutoEncoder, てれか

Autoencoder is one type of fully connected neural network while the hidden layer's size is usually smaller than the input and output layer's size. It includes one nonlinear encoding stage and one linear decoding stage. In encoding stage, the inputs are compressed into some coefficients with smaller size while the coefficients are decompressed back to original inputs in decoding stage thus it can be thought of compression and decompression. It is thought that if the input can be compressed into coefficients with smaller size, it may be a good representation of the input.

Assuming the output is x and input is y.

$$h(y_i) = \sigma(W_1 y_i + b) \quad (14)$$

$$\hat{x} = W_2 h(y_i) + c \quad (15)$$

where W_1 and W_2 are encoding and decoding matrix for the neural network connection weight. b and c are the vector of bias for hidden and output layers. The weights and biases are determined by optimizing the mean square error loss

$$L_{MSE} = \frac{1}{N} \sum ||x_i - \hat{x}_i||_2^2 \quad (16)$$

where x_i is the desired coefficients while $\hat{x}_i$ is the predicted coefficients.

3.5.2 DAE

Autoencder has a special version Denoising Autoencoder in which the input to DAE is a distorted output.

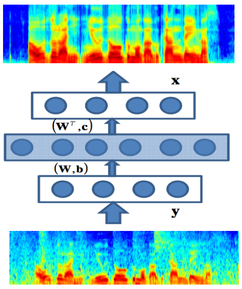


Figure 3: DAE from [17]

As shown in Figure 3, denoising Autoencoder is similar to Autoencoder except that the input and output are the same for Autoencoder while the input is noisy version of output for Denoising Autoencoder. It is thought that the noise is at higher dimension and if the input is compressed into coefficients with smaller size, the coefficients may mostly have the clean information and the noise information is filtered.

3.5.3 SDAE

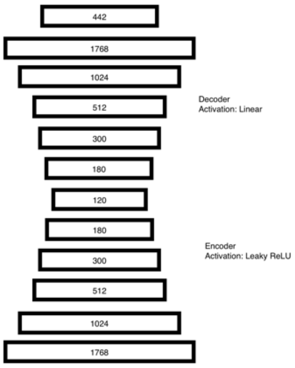


Figure 4: SDAE for clean spectrum estimation

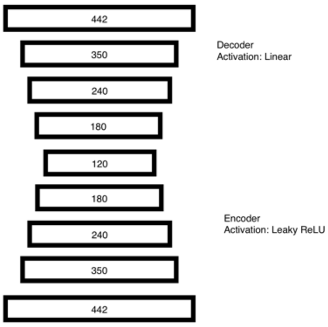


Figure 5: SDAE for noise spectrum estimation

By stacking several DAE, a SDAE is built as shown in figure 4. SDAE is commonly trained with pre-training. In pre-training stage, each layer of SDAE is separately trained as DAE. After pre-training stage, the DAEs are stacked together and form a multiple layer neural network. Then, it is trained as normal multiple layer neural network and this stage is called fine tuning stage.

Similar to AE, the activation in encoding part is non-linear function while the activation in decoding part is linear function.

The weights and biases of the neural network are determined by optimizing the following equation.

$$L = L_{MSE} + \frac{\lambda}{2n}\sum w^2 \quad (17)$$

where λ regulation coefficients and $\frac{\lambda}{2n}\sum w^2$ is L2 regulation which make the neural network learning small weight and avoid overfitting.

In training the SDAEs, greedy layer wise pre-training is adopted for accelerating the training.

3.6 Training Method of SDAE

In this section, the training methods will be introduced. They include objective, activation, regulation and pre-training.

3.6.1 Objective/Loss

In the research, two objective functions are proposed and they are Mean Square Error (MSE) and Weight Square Error (WSE).

$$MSE: L = \frac{1}{N}||x - \hat{x}||^2 \quad (18)$$

$$WSE: L = ||F \circ (x - \hat{x})||^2 \quad (19)$$

In MSE, all the input neurons have the same importance. However, different input neurons represent difference frequency for power spectrum as input. For example, a 44100Hz audio signal is transfer into STFT with 882(20ms) FFT size. Then, the frequency resolution is 50Hz. In other word, the 1st spectrum coefficient is 0Hz and 2nd is 50Hz. Since sensitivity of human ear to different frequency is different, it is reasonable to weight different frequency and this is the main idea of WSE.

In this research, Stochastic Back-Propagation training algorithm is applied to train the SDAE. The following show the effect of weighting function.

For simplicity, assume the DAE has one hidden layers, activation in encoding part is leaky relu and activation in decoding part is linear, the gradient of L with respective to

$$\hat{x} = a_2 = W_2 h + b_2$$

$$\frac{\partial L}{\partial a_2} = 2F \circ (\hat{x} - x) \quad (20)$$

The respective update of W_2 *and* b_2 is:

$$\Delta W_2 = \varepsilon \frac{\partial L}{\partial W_2} = \varepsilon \frac{\partial L}{\partial a_2}\frac{\partial \hat{x}}{\partial W_2} = 2\varepsilon F \circ (\hat{x} - x)h^T \quad (21)$$

$$\Delta b_2 = \varepsilon \frac{\partial L}{\partial b_2} = \varepsilon \frac{\partial L}{\partial a_2}\frac{\partial \hat{x}}{\partial b_2} = 2\varepsilon F \circ (\hat{x} - x) \quad (22)$$

where ε is the learning rate

Since the activation of hidden is Leaky Relu, h is expressed as:

$$h(j) = \sigma(a_1(j)) = \begin{cases} a_1(j)\ if\ a_1(j) > 0 \\ \alpha a_1(j)\ otherwise \end{cases} \quad (23)$$

$$\frac{\partial h(j)}{\partial a_1(j)} = \begin{cases} 1\ if\ a_1(j) > 0 \\ \alpha\ otherwise \end{cases} \quad (24)$$

Thus, the gradient of L with respective to

$$u_1 = W_1 y + b_1$$

$$\frac{\partial L}{\partial a_1(j)} = \frac{\partial L}{\partial h(j)}\frac{\partial h(j)}{\partial a_1(j)} = \begin{cases} \frac{\partial L}{\partial h(j)}\ if\ a_1(j) > 0 \\ \alpha \frac{\partial L}{\partial h(j)}\ otherwise \end{cases} \quad (25)$$

where N_{hid} is the hidden layer size.

Reference to the back-propagation,

$$\frac{\partial L}{\partial h} = W_2^T \frac{\partial L}{\partial a_2} = W_2^T[2F \circ (\hat{x} - x)] \quad (26)$$

Then, the respective update of W_1 *and* b_1 is:

$$\Delta W_1 = \varepsilon \frac{\partial L}{\partial W_1} = \varepsilon \frac{\partial L}{\partial a_1}\frac{\partial a_1}{\partial W_1} = \varepsilon \frac{\partial L}{\partial a_1} y^T \quad (27)$$

$$\Delta b_1 = \varepsilon \frac{\partial L}{\partial b_1} = \varepsilon \frac{\partial L}{\partial a_1}\frac{\partial a_1}{\partial b_1} = \varepsilon \frac{\partial L}{\partial a_1} \quad (28)$$

From the equation above, $\Delta W_1, \Delta b_1, \Delta W_2\ and\ \Delta b_2$ are controlled $\frac{\partial L}{\partial a_1}$ and $\frac{\partial L}{\partial h}$ which is proportional to the weighting function. Thus, they are controlled by the weighting function F which makes it possible to set an appropriate learning rate for each bin.

In this paper, the weighting function F is obtained by absolute threshold for hearing (ATH) which defines the minimum sound intensity in dB required to be detected by an average human ear with normal hearing in a quiet environment. The ATH is varied with different frequency and it is minimum for frequency around 3000Hz to 4000Hz. It is assumed that if ATH of a certain frequency is smaller which means the frequency is more detectable by human ear, the frequency is more important. The difference in importance of frequency can be used to construct the weighting function F.

Here is the detail operation.

Step 1: Since the frequency bins of spectrum represent different frequency, calculate ATH for different frequency bin.

$$Feq[\delta] = 50(\delta - 1)\ for\ 1 \le \delta \le 882\ (29)$$

$Feq[i]$ refer to the center frequency of each frequency bin.

$$ATH[\delta] = 3.64(\frac{Feq[\delta]}{1000})^{-0.8} - 6.5e^{-0.6(\frac{Feq[\delta]}{1000}-3.3)^2} + 10^{-3}(\frac{Feq[\delta]}{1000})^4\ for\ 1 \le \delta \le 882\ \ (30)$$

Step 2: Shift the ATH for making the minimum 1 and Calculate the frequency importance weights F

Since $Feq[\delta]$ only represent the center frequency of each frequency bin, the bin 1 do not only refer to 0Hz. In order to avoid assigning zero weighting to bin 1, the 4th weighting is assigned to bin 1.

The performances of MSE and WSE are compared in section 4.

3.6.2 Activation

In the final model, LeakyReLU is adopted as activation in encoding stage. Original ReLU has a serious problem called Dying ReLU problem. The ReLU neurons sometimes are in the state that they become inactive and output zero for all inputs. In this state, gradients flow backward through the neurons is zero and the neurons stick to this point and "die". For SDAE, the number of neurons in the middle is few and therefore Dying ReLU problem may lead to large construction error.

Leaky ReLU is an attempt to solve the problem.

$$\sigma(a_i) = \begin{cases} a_i\ for\ a_i > 0 \\ \alpha a_i\ otherwise \end{cases}\ (31)$$

where α is a parameter and usually very small.

The neurons can have response respect to all input, as a result it has gradient flow backward through the neurons and the neurons will not stick to some points.

3.6.3 Optimizer

In this model, three optimizers, Stoschastic Gradient Descent(SGD) optimizer, Adadelta optimizer, Adam optimizer, are proposed.

Adadelta optimizer and Adam optimizer are special version of SGD optimzer and the difference is that Adadelta adopts adaptive learning rate while Adam adopts adaptive momentum values. In order to examine their ability in training the neural network, their performances are compared in section 4.

3.7 Noise Estimation

In the research, *a prior* SNR estimation is adopted and it needs the estimated clean and noise spectrum. In figure 1, one of the SDAEs is constructed to estimate the noise spectrum and their respective structure refers to figure 4 and 5. Their differences are the structure and training data while all other settings are the same.

As a result, one SDAE estimates the clean spectrum and one SDAE estimates the noise spectrum from the noisy spectrum.

3.8 Inverse STFT

For the research, two common inverse STFT methods are proposed and they are Overlap-Add method (OLA) and Real Time Signal Estimation from Modified STFT Magnitude Spectra (RTSE) [19].

OLA requires the modified STFT coefficients which have phase information. However, only the power spectrum of STFT coefficients will be used in the proposed models proposed in this project. Thus, the phase information used for OLA is unmodified. In other word, it is noisy phase.

In comparison, RTSE do not require phase information. However, it takes long time to estimate the signal and if the overlapping is less than 75%, the speech will be highly distorted and inaccurate.

RTSE is not used in this model and final model because it is computational difficult and it usually takes long time to estimate the phase. Besides, a good estimation of phase requires over 90% overlapping which makes the training data size becomes large. For example, a 1.8MB wav file will expend to 18MB data of power spectrum with 90% overlapping. This should be avoided because the data has large similarity and it may affect the training of neural network. Therefore, Overlap-Add method is used.

The merit of OLA is that it is computational simple and large overlapping is not required while the demerit is that the use of noisy phase may affect the speech quality. However, Dr. Wang and Lim [5] have public a paper in which they investigated the importance of phase information and they found that the spectrum is more important in restoring the signal than the phase. In order to prove their finding, a similar experiment was done.

The noisy speech is made by the clean speech combined with 5dB noise and they are transformed to STFT spectrum with 882 FFT size and 441 overlapping. Then, OLA and IFFT are performed on the clean spectrum with noisy phase extracted from the noisy STFT coefficients before taking absolute value. The restored clean speech and original clean speech are down-sampled to 16000Hz and PESQ evaluation is conducted.

The resulting PESQ score is 3.241. Although the result is not perfect, it is a tradeoff between the performance and the efficiency of model.

3.9 a Prior SNR Estimation

In the research, a Posteriori SNR Controlled Recursive Averaging (PCRA) approach is proposed to estimate *a Prior* SNR. SDAE is very powerful in removing the background noise. However, the direct use of estimated clean power spectrum from SDAE will result in distorted speech when the period of speech is highly covered by background noise. Thus, PCRA takes this consideration and recover the distorted by using the speech information of previous frames and the noisy speech.

Here is the detailed operation:

Step 1: *a Posterior* SNR take the consideration of noisy spectrum and noise spectrum

$$\gamma(\lambda,\delta) = \max\left(\frac{Y^2(\lambda,\delta)}{\hat{\sigma}_d^2(\lambda,\delta)},1\right) \; (32)$$

where $\lambda\ and\ \delta$ are the frame index and frequency bin, $\hat{\sigma}_d^2(\lambda,\delta)$ is the restored noise spectrum and $Y^2(\lambda,\delta)$ is the noisy spectrum

Step 2: First Order Recursive Averaging is performed on the SNR for reducing the fast fluctuations along the time. And the compare the SNR with a predefined threshold T_γ and get $I(\lambda,\delta)$. It is assumed that if the speech component dominates on λth frame and ith bin, SNR will be high thus $I(\lambda,\delta) = 1$ indicate the presence of speech component. Then, first order recursive averaging is also performed on $I(\lambda,\delta)$ to reduce the fast fluctuations and it becomes probability function which indicate the probability of the speech component dominating on λth frame and ith bin,.

$$\bar{\gamma}(\lambda) = \alpha_\gamma\bar{\gamma}(\lambda-1) + (1-\alpha_\gamma)\gamma(\lambda) \; (33)$$

$$I(\lambda,\delta) = \begin{cases} 0\ if\ \bar{\gamma}(\lambda,\delta) < T_\gamma \\ 1\ otherwise \end{cases} \; (34)$$

$$p(\lambda,\delta) = \alpha_p p(\lambda-1,\delta) + (1-\alpha_p)I(\lambda,\delta) \; (35)$$

where α_γ and α_p are smoothing factor

Step 3: In order to take the previous frames into consideration, the probability function $p(\lambda,i)$ will become a smoothing factor for step 4.

$$\alpha_\xi(\lambda,\delta) = \alpha_{\xi min} + (1-p(\lambda,\delta))(\alpha_{\xi max} - \alpha_{\xi min}) \; (36)$$

where $\alpha_{\xi min}$ and $\alpha_{\xi max}$ are the minimum and maximum values of the smoothing factor.

Step 4: The *a priori* SNR takes the consider of previous frames, the noisy spectrum and noise spectrum. Therefore, it is calculated as follow.

$$\bar{\xi}(\lambda,\delta) = \alpha_\xi(\lambda,\delta)\bar{\xi}(\lambda-1,\delta) +$$
$$(1-\alpha_\xi(\lambda,\delta))[\beta\frac{\hat{X}^2(\lambda,\delta)}{\hat{\sigma}_d^2(\lambda,\delta)} + (1-\beta)(\gamma(\lambda,\delta)-1)](37)$$

in which β is the weighting factor. If β is smaller, more consideration of previous frame is taken.

The step 4 equation has three different parts

Part 1: $\bar{\xi}(\lambda-1,\delta)$ which is *a prior* SNR of previous frame. Tt is assumed that the large distortion of some frames of estimated clean spectrum from SDAE partly comes from the missing information of current frame. However, the previous frame may have less distortion and the missing information of current frame. Therefore, depending of the probability function $p(\lambda,\delta)$ and $\alpha_\xi(\lambda,\delta)$ which indicate if the current is highly distorted, the information of previous frame $\bar{\xi}(\lambda-1,i)$ will be used.

Part 2: $\frac{\hat{X}^2(\lambda,\delta)}{\hat{\sigma}_d^2(\lambda,\delta)}$ is the estimated SNR of current frame

Part 3: $\gamma(\lambda,\delta) - 1 = \frac{max(0,Y^2(\lambda,\delta)-\hat{\sigma}_d^2(\lambda,\delta))}{\hat{\sigma}_d^2(\lambda,\delta)}$ is the Maximum Likelihood(ML) estimate of current frame. The idea of ML is based on the additive nature of noise.

$$\frac{max(0,Y^2(\lambda,\delta)-\hat{\sigma}_d^2(\lambda,\delta))}{\hat{\sigma}_d^2(\lambda,\delta)} = \frac{X^2(\lambda,\delta)}{\hat{\sigma}_d^2(\lambda,\delta)}$$

if $\hat{\sigma}_d^2(\lambda,\delta)$ is accurate.

However, when it is not accurate, it will keep the noise.

Therefore, in the part $\beta\frac{\hat{X}^2(\lambda,\delta)}{\hat{\sigma}_d^2(\lambda,\delta)} + (1-\beta)(\gamma(\lambda,\delta) - 1)$, β indicate a tradeoff of distortion of speech and noise removal. The large β means larger noise while the information from ML estimate will also be smaller and it distorts the speech.

In order to examine the importance of PCRA and ML, three different settings are compared.

Settings 1: Reference to [5], a relatively high weighting of previous frame and ML is adopted.

$\alpha_\gamma = 0.8, T_\gamma = 1.5, \alpha_p = 0.95, \alpha_{\xi min} = 0.9, \alpha_{\xi max} = 0.98$ and $\beta = 0.75$

Setting 2: Since ML estimate will increase the noise level, it is not adopted. In addition, a relatively small weighting on previous frame is adopted

$\alpha_\gamma = 0.8, T_\gamma = 1.5, \alpha_p = 0.95, \alpha_{\xi min} = 0.3, \alpha_{\xi max} = 0.15$ and $\beta = 1$

Setting 3: Simple SNR estimation is adopted.

$$\bar{\xi}(\lambda, \delta) = \frac{\hat{X}^2(\lambda, \delta)}{\hat{\sigma}_d^2(\lambda, \delta)} \quad (38)$$

3.10 Wiener Filter

The *a priori* SNR is used to construct Wiener Filter in Frequency domain.

$$G(\lambda, b) = \frac{\bar{\xi}(\lambda, \delta)}{\bar{\xi}(\lambda, \delta) + 1} \quad (39)$$

4. Performance Evaluation and Discussion

In this section, the performances of different settings are compared. Since the comparison belong to tuning stage, only one random audio file is used. Before comparing the performances, the three performance evaluation are introduced. Finally, the performance of model is evaluated and the discussed.

4.1 Performance Evaluation

The three performance evaluations are Perceptual Evaluation of Speech Quality Improvement (PESQI), Log Spectral Distortion Improvement(LSDI) and Noise Reduction.

PESQ is a worldwide applied standard in evaluating the speech quality. The research adopts the ITU-T Recommendation P.862.1 standard proposed by International Telecommunication Union.

LSD focuses on the distortion of speech.

$$D_{LS} = \frac{1}{N_{frame}} \sum \sqrt{\frac{1}{N_{FFT}/2 + 1} \sum (10log_{10} \frac{X^2(\lambda, \delta)}{\hat{X}^2(\lambda, \delta)})^2} \quad (40)$$

where N_{frame} and $N_{FFT}/2 + 1$ are the number of frame and FFT frequency bin

Noise Reduction focuses on the noise removal

$$NR_{LS} = \frac{1}{N_{frame}} \frac{1}{N_{FFT}/2 + 1} \sum \sum (|\frac{X^2 - Y^2(\lambda, \delta)}{Y^2}|) \quad (41)$$

4.2 Objective

The performance of model adopting MSE or WSE is evaluated and it is the result.

	MSE	WSE
PESQI	0.939	0.742
LSDI(dB)	17.22	16.60
Noise Reduction	0.6896	0.6943

Although it was expected that WSE can perform better MSE, the result suggests that MSE get better performance. This result is unexpected and further investigation on it should be done.

4.3 Optimizer

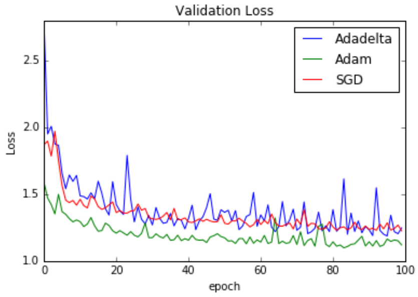


Figure 6: Validation Loss of different optimizer

Since the purpose of optimizer is finding the minimum construction error, it should be evaluated by the final error of validation data. The figure 6 suggests that Adam perform better than other two optimizers. The reason may be the momentum is important to train the network and the best momentum may be varied with different stage of training. As a result, an adaptive momentum approach generates the best performance.

4.4 a prior SNR Estimation

The three different settings refer to section 3.9.

SNR=5	PESQI	LSDI	Noise Reduction
Setting 1	0.168	3.87	0.50
Setting 2	0.849	16.32	0.68
Setting 3	0.775	16.41	0.808

The performance shows that the setting from [5] is not satisfactory and ML should not be considered. In addition, setting 2 and setting 3 have different advantages and their application should depend on the real situation. In this research, setting 2 is adopted.

4.5 Experiment Setup for final Evaluation

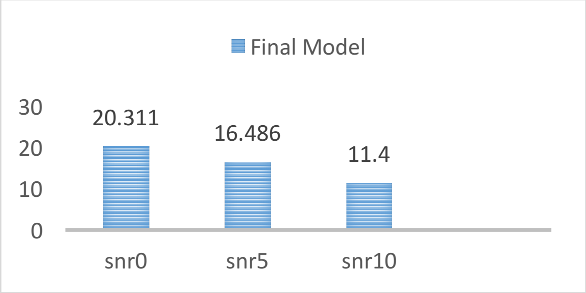
20 audio files are randomly selected from the data base excluding the audio file used for training and testing. The noise file is car and road noise last for 33 seconds and a portion of 9 to 10 seconds of the file is randomly selected. Then, the final model with setting 2 in section 3.5.3 is used to estimate the clean speech. All the speechs are down sampled to 16000Hz for PESQI.

The final evaluation score is the average of each evaluation. In addition, the SDAEs are trained with 5dB noise, therefore, the performance of model with different noise ratio is expected worse.

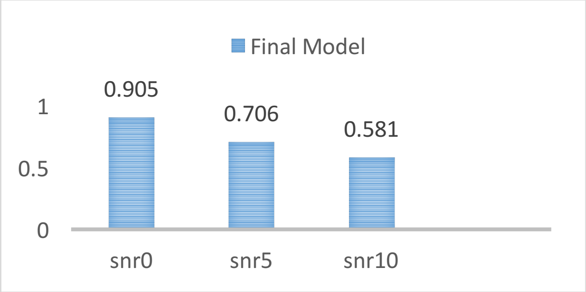
Moreover, the performance of traditional spectral subtraction algorithm with imcra noise estimation algorithms on PESQ is also evaluated in order to make a comparison with the final model.

4.6 Result

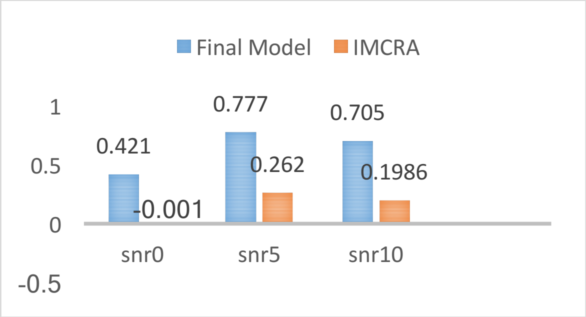
LSDI



Noise Reduction:



PESQI:



In addition to table, a speech is randomly selected from the 20 validation speech above and a graph of original speech, noisy speech and estimated speech are shown in the following.

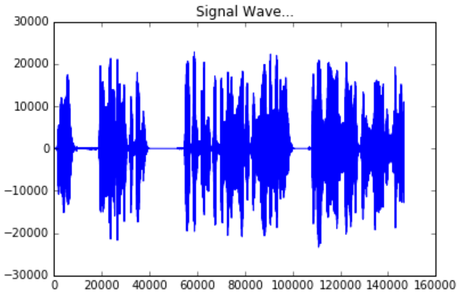


Figure 7: Original Speech

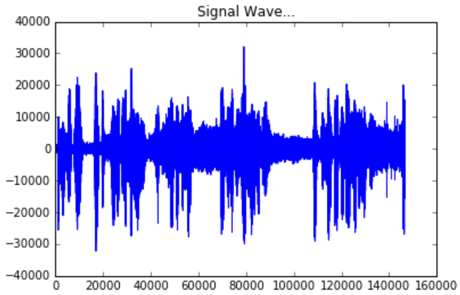


Figure 8: Noisy Speech

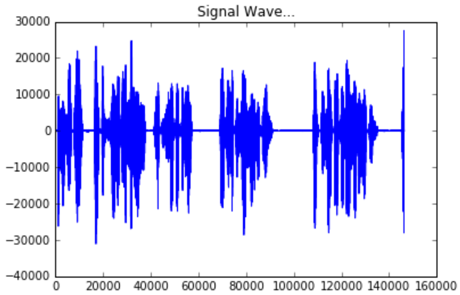


Figure 9: Estimated Speech

4.7 Discussion

In section 4.3, the result suggests that Adam which adopt adaptive momentum learning perform better than traditional SGD and Adadelta. In addition, the result of section 4.4 suggests that the ML of SNR may not be preferable in speech enhancement task although it has some useful information. Moreover, the result in section 4.2 suggest that MSE perform better than WSE although it was thought that WSE could have higher PESQI score than MSE.

The performance of model is satisfactory for removing the specific noise which is used in training the SDAEs. From figure 7,8 and 9, the estimated speech is closed to the original speech. In addition, comparing with traditional noise removal method, the proposed model performs better in PESQ improvement.

Therefore, SDAE with SNR estimation and Wiener Filter perform good in improving the speech quality and removing the background noise. Therefore, it may have potential in ASR system and other applications requiring noise removal technic.

5. Conclusion

Our brain is incredible strong in handling background noise and speech separation which enable us to talk to each other in a noisy environment. However, the machine could not do it well before the full application of neural network. For investigating the power of neural network in term of noise reduction, my research tries to build a stacked denoising autoencoder which help to reduce the background noise of noisy speech in order to improve the speech quality and recognition accuracy of ASR system. This is complementary to the exiting speech recognition system in mobile device which operate in a relatively complex environment where background noise, such as car noise or street noise, is common and greatly affect the performance of speech recognition. In addition to improving the speech recognition accuracy, potential contribution may include background noise removal for music or conversation.

6. Conclusion

[1] "Deep Learning Tutorials — DeepLearning 0.1 documentation", Deeplearning.net, 2017. [Online]. Available: http://deeplearning.net/tutorial/. [Accessed: 07- Apr- 2017].

[2] "Python tutorial", www.tutorialspoint.com, 2017. [Online]. Available: http://www.tutorialspoint.com/python/. [Accessed: 07- Apr- 2017].

[3] A. Kumar and D. Florencio, "Speech Enhancement In Multiple-Noise Conditions using Deep Neural Networks", 2016.

[4] B. Sh. Mahmood and N. N. Ibrahim, "Processing Mel Speech Power Spectrum for Speech Restoration", International Journal of Enhanced Research in Science, Technology & Engineering, vol. 5, no. 9, pp. 39-45, 2016.

[5] B. Xia and C. Bao, "Wiener filtering based speech enhancement with Weighted Denoising Auto-encoder and noise classification", Speech Communication, vol. 60, pp. 13-29, 2014.

[6] D. Griffin and Jae Lim, "Signal estimation from modified short-time Fourier transform", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 2, pp. 236-243, 1984.

[7] E. Boucheron and L. De Leon, "On the Inversion of Mel-Frequency Cepstral Coefficients for Speech Enhancement Applications", 2017.

[8] E. Plourde and B. Champagne, "Multidimensional STSA Estimators for Speech Enhancement With Correlated Spectral Components", IEEE Transactions on Signal Processing, vol. 59, no. 7, pp. 3013-3024, 2011. [9] E. Plourde and B. Champagne, "Multidimensional STSA Estimators for Speech Enhancement With Correlated Spectral Components", IEEE Transactions on Signal Processing, vol. 59, no. 7, pp. 3013-3024, 2011.

[10] F. Chollet, "Building Autoencoders in Keras", Blog.keras.io, 2017. [Online]. Available: https://blog.keras.io/building-autoencoders-in-keras.html. [Accessed: 07- Apr- 2017].

[11] M. Nielsen, "Neural Networks and Deep Learning", Neuralnetworksanddeeplearning.com, 2017. [Online]. Available: http://neuralnetworksanddeeplearning.com/. [Accessed: 07- Apr- 2017].

[12] M. Zhao, D. Wang, Z. Zhang and X. Zhang, "Music Removal by Convolutional Denoising Autoencoder in Speech Recognition", 2015.

[13] P. Loizou, Speech enhancement, 1st ed. Boca Raton, Fla.: CRC Press, 2013.

[14] T. Gerkmann and M. Krawczyk, "MMSE-Optimal Spectral Amplitude Estimation Given the STFT-Phase", IEEE Signal Processing Letters, vol. 20, no. 2, pp. 129-132, 2013.

[15] T. Ishii, H. Komiyama, T. Shinozaki, Y. Horiuchi and S. Kuroiwa, "Reverberant Speech Recognition Based on Denoising Autoencoder", INTERSPEECH, 2013.

[16] T. Sainburg, "Spectrograms, MFCCs, and Inversion in Python", Tim Sainburg, 2017. [Online]. Available: https://timsainb.github.io/spectrograms-mfccs-and-inversion-in-python.html. [Accessed: 07-Apr-2017].

[17] Thomas F. Quatieri, "Discrete-Time Speech Signal Processing: Principles and Practice"

[18] X. Lu, Y. Tsao, S. Matsuda and C. Hori, "Speech Enhancement Based on Deep Denoising Autoencoder".

[19] X. Zhu, G. Beauregard and L. Wyse, "Real-Time Signal Estimation From Modified Short-Time Fourier Transform Magnitude Spectra", IEEE Transactions on Audio, Speech and Language Processing, vol. 15, no. 5, pp. 1645-1653, 2007.

[20] Z. Wu, S. Takaki and J. Yamagishi, "DEEP DENOISING AUTO-ENCODER FOR STATISTICAL SPEECH SYNTHESIS", 2015.