Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 9, No. 1, March 2021, pp. 84~90
ISSN: 2089-3272, DOI: 10.11591/ijeei.v9i1.2494

Noise reduction system by using CNN deep learning model

Haengwoo Lee
Namseoul University/91, Daehak-ro, Cheonan-si, Chungcheongnam-do, South Korea

Article Info

Article history:

Received May 24, 2020
Revised Feb 11, 2021
Accepted Mar 19, 2021

Keyword:

Noise reduction
Deep learning
Convolutional neural network
Neural filter

ABSTRACT

In this paper, we propose a new algorithm to reduce the acoustic noise of hearing aids. This algorithm improves the noise reduction performance by the deep learning algorithm using the neural network adaptive prediction filter instead of the existing adaptive filter. The speech is estimated from a single input speech signal containing noise using a 80-neuron, 16-filter convolutional neural network(CNN) filter and an error backpropagation algorithm. This is by using the quasi-periodic property of the voiced sound section in the speech signal, and it is possible to predict the speech more effectively by applying the repeated pitch. In order to verify the performance of the noise reduction system proposed in this research, a simulation program using Tensorflow and Keras libraries was coded and a simulation was done. As a result of the experiment, the proposed deep learning algorithm improves the mean square error(MSE) of 28.5% compared to using the existing adaptive filter and 17.2% compared to using the FNN(full-connected neural network) filter.

Corresponding Author:

Haengwoo Lee,
Namseoul University/91, Daehak-ro, Cheonan-si,
Chungcheongnam-do, South Korea
Email: haengwoolee@hanmail.net

1. INTRODUCTION

When using a hearing aid, noise is a factor that makes users uncomfortable and makes speech recognition difficult. A speech improvement technology that reduces the noise included in the speech signal is required, and many studies have been conducted so far. There are two types of techniques for noise reduction. First, there are spectral subtraction methods[1,2] and Wiener filter methods[3,4] based on short-term spectrum estimation. These methods are suitable when the estimated spectrum of noise is subtracted from the input speech signal, or a clean speech spectrum is estimated, and the statistical characteristics of the noise and the obtained speech signal are known. Second, there are Comb filters[5] and adaptive filter methods[6,7] that use quasi-periodic characteristics of speech signals. The Comb filter method is used when the noise has a specific frequency band, and the adaptive filter method automatically adjusts the coefficients of the filter, so it is not necessary to know the statistical characteristics of the noise in advance.

The adaptive noise reduction system is divided into a single input and a multiple input system according to the number of acoustic sensors. The single input system[8] inputs a speech signal through a single microphone. Since the voiced section of the speech signal has a quasi-periodic characteristic, a signal having a high correlation with the input speech signal can be obtained by delaying the microphone input signal containing noise by one or two pitches. This signal is used as a reference signal for the adaptive filter. . To obtain the pitch delay value, the input signal is divided into approximately 30 ms intervals in which the statistical characteristics of the speech are not changed, and the autocorrelation function is calculated for each section to obtain the pitch delay value.

Deep learning[9] is a complex machine learning model that uses a large number of hidden layers based on neural networks. The recent deep learning model is making great progress in many fields because it has developed a technology that can learn multi-layered neural networks composed of many layers. An error backpropagation algorithm[10] that trains a multi-layer neural network can learn deep neural networks composed of many layers by pre-learning the synapses of the lower layer before learning the upper layer[11].

The most used deep learning model is Convolutional Neural Network. In this study, we propose a method to reduce noise using deep learning algorithm of neural network filter instead of adaptive filter of adaptive noise reduction system. Since the CNN model shows excellent performance in feature extraction, it has been recently used in video and audio signal processing[12,13].

The content of the paper looks at the adaptive noise reduction system in Section II, Section III describes the learning algorithm of the multilayer neural network, and Section IV proposes a new deep learning model structure. In addition, in Section V, the simulation and the results of this system are described, and finally, in Section VI, a conclusion is drawn.

2. ADAPTIVE NOISE REDUCTION SYSTEM

Figure 1 is a single-input adaptive noise reduction system that estimates the current speech from signals delayed by more than one sample by the adaptive prediction method using the quasi-periodic characteristics of the speech signal. One or two pitch-delayed input signals have a high correlation with speech signal components, but little correlation with noise components. Therefore, the speech signal is independent of noise and converges to be the least square error of the target value.

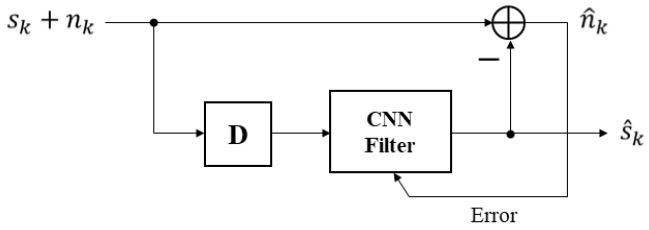

Figure 1. Adaptive noise reduction system

The output of the CNN filter estimates the voiced sound of the speech signal included in the input signal, and when this signal is subtracted from the input signal, it becomes an estimated noise signal. The noise estimation signal is used as an error signal for adjusting the coefficient of the CNN filter, and the average power of this error signal is shown in Equation (1).

$$E\{\hat{n}_k^2\} = E\{(s_k - \hat{s}_k)^2 + 2(s_k - \hat{s}_k)n_k + n_k^2\} \quad (1)$$

Here, $E\{\cdot\}$ represents the average value, and assuming that the speech and noise signal are not correlated with each other,

$$E\{\hat{n}_k^2\} = E\{(s_k - \hat{s}_k)^2 + n_k^2\} \quad (2)$$

Since the noise energy in any frame is a fixed value,

$$min(E\{\hat{n}_k^2\}) = min(E\{(s_k - \hat{s}_k)^2\}) + E\{n_k^2\} \quad (3)$$

That is, minimizing $E\{\hat{n}_k^2\}$ is to minimize the estimation error of the speech signal $E\{(s_k - \hat{s}_k)^2\}$. At this time, the estimated value of the speech signal $\hat{s}_k$, which is the output of the filter, best estimates the speech signal. Therefore, minimization of $E\{(s_k - \hat{s}_k)^2\}$ means minimizing $E\{(n_k - \hat{n}_k)^2\}$ and error signal $\hat{n}_k$ estimates noise.

Since the microphone input signal, which is a mixture of speech and noise signal, has quasi-periodic characteristics in the voiced section, one or two pitch delayed signals have a high correlation with the speech signal. At this time, the output of the filter becomes a speech estimation signal having a minimum square error and a speech signal in the input signal by minimizing the energy of the error signal.

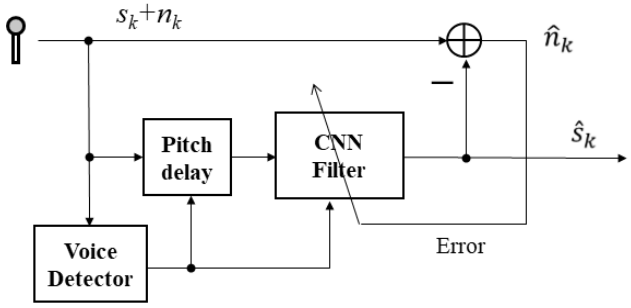

Figure 2. Adaptive noise reduction system with pitch detector

To obtain the pitch delay information, the pitch detector obtains autocorrelation function for every analysis section. The pitch delay time for which this value is the maximum is selected between 2-20 ms. In

addition, the coefficient of the filter is updated only in the voiced sound section, and the adjustment of the filter coefficient is stopped in the voiced sound section (Figure 2).

3. LEARNING ALGORITHM OF MULTI-LAYER NEURAL NETWORK

A multi-layer perceptron has the structure of a multi-layer forward neural network with one or more hidden layers. Figure 3 shows a multi-layer perceptron consisting of an input layer with l input neurons, a hidden layer with m hidden neurons, and an output layer with n output neurons. The values of the input neurons of the multi-layer perceptron are represented by the l-dimension vector $x = [x_1, x_2, \cdots, x_i, \cdots, x_l]$, the values of the hidden neurons are represented by the m-dimension vector $a^1 = [a_1^1, a_2^1, \cdots, a_j^1, \cdots, a_m^1]$, and the values of the output neurons are represented by the n-dimensional vector $a^2 = [a_1^2, a_2^2, \cdots, a_k^2, \cdots, a_n^2]$. The weight and bias between the input layer and the hidden layer is represented by w_{ij}^1, b_j^1, the weight and bias between the hidden layer and the output layer is expressed by w_{jk}^2, b_k^2. Also, the weighted sum inputted to the j-th hidden neuron is called u_j^h, the weighted sum inputted to the k-th output neuron u_k^o, and the activation function of the hidden neuron is expressed as $\emptyset_h$, the activation function of the output neuron $\emptyset_o$.

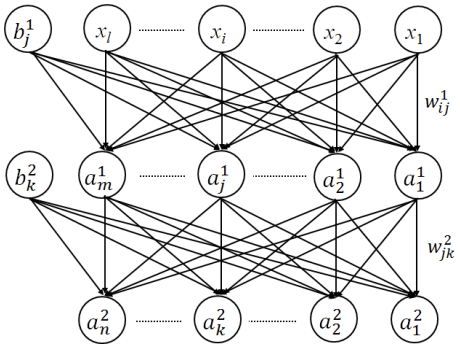

Figure 3. Multi-layer neural network

The output values of the hidden neurons and output neurons can then be expressed by the equation (4) and (5).

$$a_j^1 = \emptyset(u_j^h) = \emptyset_h\left(\sum_{i=1}^{l} w_{ij}^1\, x_i + b_j^1\right) \tag{4}$$

$$a_k^2 = \emptyset(u_k^o) = \emptyset_o\left(\sum_{j=1}^{m} w_{jk}^2\, a_j^1 + b_k^2\right) \tag{5}$$

If we denote all weights and biases as a parameter θ, we can express the value of the k-th output neuron as a function $f_k(x, \theta)$ given the input x.

$$f_k(x, \theta) = a_k^2 = \emptyset_o\left(\sum_{j=1}^{m} w_{jk}^2\, \emptyset_h\left(\sum_{i=1}^{l} w_{ij}^1\, x_i^1 + b_j^1\right) + b_k^2\right) \tag{6}$$

The error backpropagation learning algorithm was developed by Geoffrey Hinton in the mid-1980s. The supervised learning of the multi-layer perceptron should be based on the target output value and the cost function using the difference of the values outputted by the multi-layer perceptron. When the learning data and the target output value are given as a pair of input and output orders $(x_i, t_i)(i = 1, \cdots, N)$, the error for the whole learning data X can be defined as a mean square error as shown in the following equation.

$$E(X, \theta) = \frac{1}{2N} \sum_{i=1}^{N} \|t_i - f(x_i, \theta)\|^2 \tag{7}$$

In the above equation, the error function $E(X, \theta)$ is set to one value given the data set X and the parameter θ. The data set X is the value given from the outside, and the target to be optimized is θ. Therefore it can be written $E(\theta)$. The back-propagation learning algorithm uses the gradient descent method to find the parameters to minimize the error function $E(\theta)$. The gradient descent method is an algorithm that finds a parameter that minimizes the value of a cost function iteratively.

$$\theta(t+1) = \theta(t) + \Delta\theta(t) = \theta(t) - \eta \frac{\partial E(\theta)}{\partial \theta} \tag{8}$$

Where η is the learning rate that controls the speed of learning. In the multi-layer perceptron, the back-propagation learning uses the error function $E(X,\theta)$ for a data by applying a stochastic gradient descent method of updating a data for each weight.

$$E(x,\theta)=\frac{1}{2}(t_k-a_k^2)^2=\frac{1}{2}\left(t_k-\emptyset_o\left(\sum_{j=1}^{m}w_{jk}^2\,a_j^1+b_k^2\right)\right)^2 \quad (9)$$

The parameter θ, which should be corrected through learning in the above equation, is the weight w_{jk}^2 and bias b_k^2 between the hidden layer and the output layer, and the weight w_{ij}^1 and bias b_j^1 between the input layer and the hidden layer. Therefore, if the error function is partially differentiated by the output-side parameter, it is as follows.

$$\frac{\partial E}{\partial w_{jk}^2}=\frac{\partial E}{\partial u_k^o}\frac{\partial u_k^o}{\partial w_{jk}^2}=-\phi_o'(u_k^o)(t_k-a_k^2)a_j^1=\delta_k a_j^1 \quad (10)$$

$$\frac{\partial E}{\partial b_k^2}=\frac{\partial E}{\partial u_k^o}\frac{\partial u_k^o}{\partial b_k^2}=-\phi_o'(u_k^o)(t_k-a_k^2)=\delta_k \quad (11)$$

Here, $\phi_o'(u_k^o)$ is the derivative value of the activation function of the output neuron, and is generally a unit step function $\phi_o'(u_k^o)=u(t)$ since the ReLU(Rectified Linear Unit) function ($\max\{0,u_k^o\}$) is widely used. And then δ_k is the effect of the output neuron on the error. Next, the error function is partially differentiated by the input-side parameters as follows.

$$\frac{\partial E}{\partial w_{ij}^1}=\frac{\partial E}{\partial u_j^h}\frac{\partial u_j^h}{\partial w_{ij}^1}=\phi_h'(u_j^h)\sum_{j=1}^{m}w_{jk}^2\,\delta_k x_i=\delta_j x_i \quad (12)$$

$$\frac{\partial E}{\partial b_j^1}=\frac{\partial E}{\partial u_j^h}\frac{\partial u_j^h}{\partial b_j^1}=\phi_h'(u_j^h)\sum_{j=1}^{m}w_{jk}^2\,\delta_k=\delta_j \quad (13)$$

Taken together, it can be seen that the parameters between the input layer and the hidden layer are affected by the sum of multiplication the weights between the hidden and output layers by the effect δ_k of each output neuron on the error. Since the error of the output neuron propagates backward to the hidden neuron and influences the parameter control of the hidden neuron, the gradient descent learning method of the multi-layer perceptron is named as the error backpropagation learning algorithm, and finally each parameter is updated by the equations (14) - (17).

$$w_{jk}^2(t+1)=w_{jk}^2(t)+\eta\phi_o'(u_k^o)(t_k-a_k^2)a_j^1 \quad (14)$$

$$b_k^2(t+1)=b_k^2(t)+\eta\phi_o'(u_k^o)(t_k-a_k^2) \quad (15)$$

$$w_{ij}^1(t+1)=w_{ij}^1(t)-\eta\phi_h'(u_j^h)\sum_{j=1}^{m}w_{jk}^2\,\delta_k x_i \quad (16)$$

$$b_j^1(t+1)=b_j^1(t)-\eta\phi_h'(u_j^h)\sum_{j=1}^{m}w_{jk}^2\,\delta_k \quad (17)$$

4. STRUCTURE OF DEEP LEARNING MODEL

The deep learning model in Figure 4 used in this paper has a three-stage structure using a CNN layer.

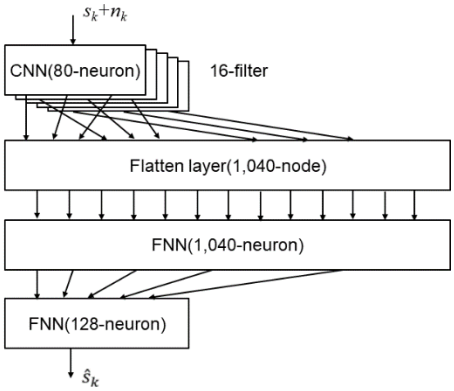

Figure 4. Structure of deep learning model

The CNN layer in the first stage consists of 80 neurons and 16 feature filters, and the size of the kernel is 16 samples, with the kernel at every sample interval. The input signal is composed of 80 × 16 data for each sample, and ReLU is applied as an activation function at the output. The output of the CNN layer is flattened in one dimension through the next Flatten layer and is expanded to (80-16 + 1) × 16 = 1,040 nodes. These signals are input to the Fully-connected Neural Network layer with 1,040 neurons, and the ReLU function is applied again at the output. Subsequently, it is output as one signal through the FNN layer having 128 neurons as the last layer. To reduce the amount of calculation, the batch size was set to 30, and the total number of parameters to be calculated in this model is 133,248.

Adam and the error backpropagation algorithm are used as the weight update algorithm. Since this system is classified as supervised learning, training data and learning target values are prepared as single input data.

5. SIMULATIONS AND ANALYSES

In order to verify the performance of the proposed speech noise reduction system, a simulation program was created using the Tensorflow and Keras libraries. The input signal was sampled at 8 kHz with a mixture of speech and white noise, and 900,000 samples (112.5 sec) were prepared. Since this system is for supervised learning, the input data is internally composed of an input array of 80 × (900,000-79) samples and a target value of (900,000-79) samples.

Figure 5 shows the waveforms of the audio signal, mixed signal and output signal.

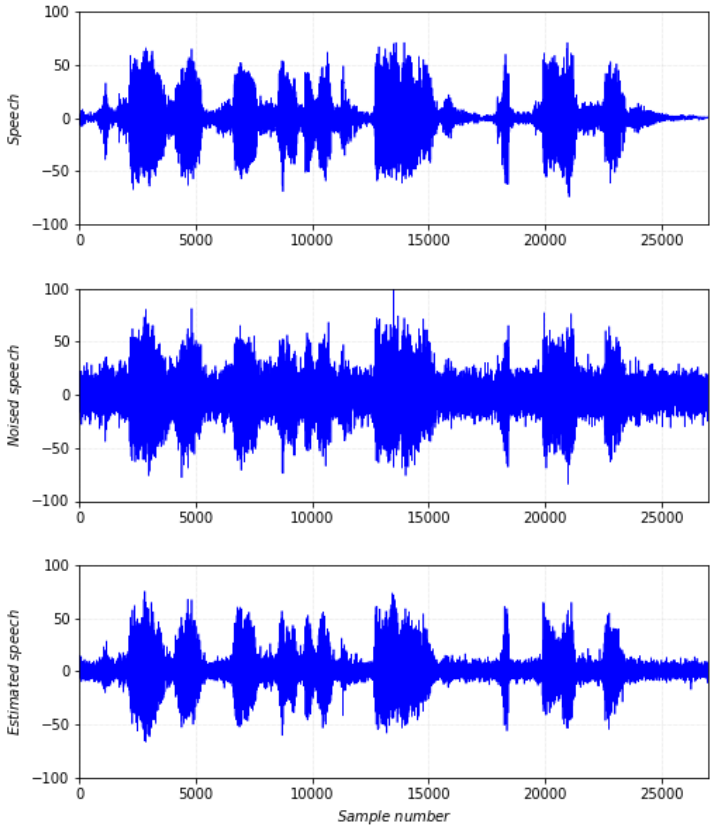

Figure 5. Waveform of input and output signal

Mean Square Error was used to evaluate the performance between systems. MSE refers to the error of the speech predicted value for the input signal, which is the target value.

$$MSE(k) = \frac{1}{N}\sum_{n=0}^{N-1}(s_k + n_k - \hat{s}_k)^2 \quad (18)$$

Figure 6 shows the MSE performance for the three filters.

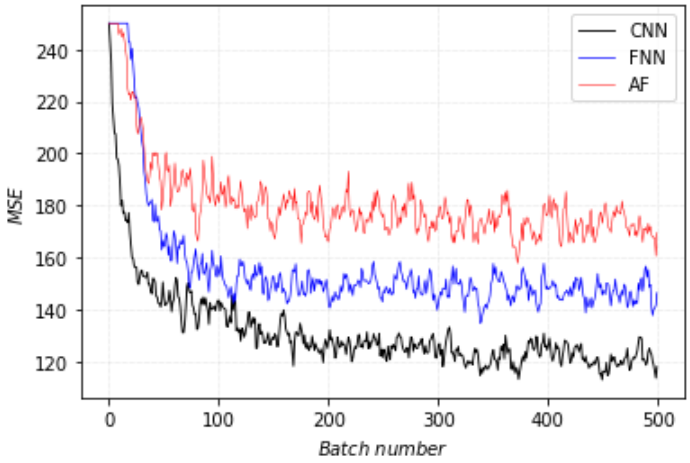

Figure 6. Comparison of MSE for the filters

In the case of using the existing adaptive filter and FNN and CNN filter, the MSE performance showed that the deep learning model using the CNN filter was the best. Simulation results showed that when using CNN filter, MSE improved by 28.5% compared to adaptive filter and 17.2% compared to FNN filter. The reason is that the adaptive filter reduces only linear noise between samples, while the FNN filter can reduce even nonlinear noise. That is, since the adaptive filter has an FIR (Finite Impulse Response) structure, it reduces the linear component of noise, and the FNN filter can reduce nonlinear components because each neuron is interconnected with all input signals and it is composed of two or more layers. Furthermore, the CNN filter achieved better performance because it also finds and utilizes several features between samples.

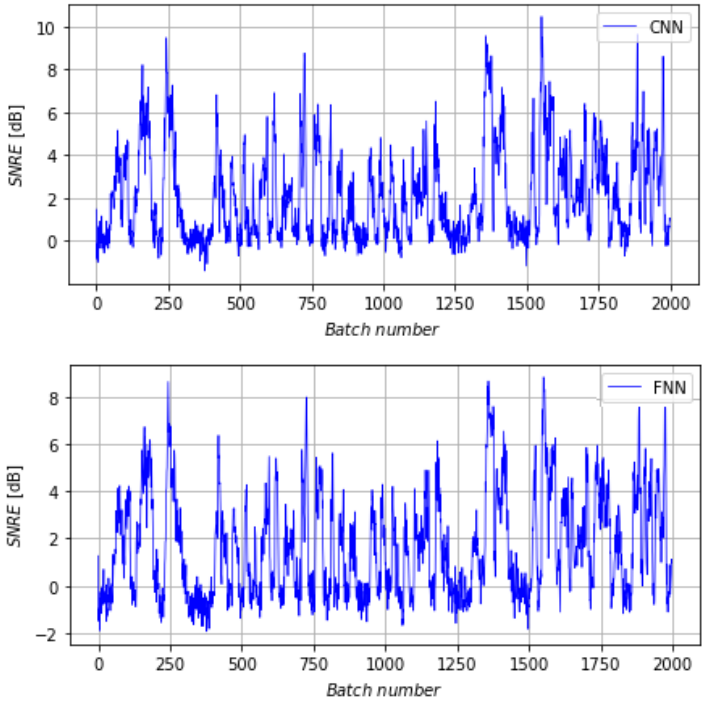

Figure 7. SNRE of the filters

Figure 7 shows the signal to noise ratio enhancement (SNRE) of a noise attenuator using a CNN filter and an FNN filter. From this figure, it can be seen that the SNRE of the noise attenuator is about 2 dB higher when using a CNN filter than when using an FNN filter.

6. CONCLUSIONS

In order to improve the speech recognition performance of hearing aids, it is required to develop an excellent noise attenuator. In this paper, we proposed a new noise reduction system using deep learning technology. Using the CNN filter, the noise reduction performance is improved by deep learning using a neural network instead of the existing adaptive filter.

The noise reduction system achieved a significant performance improvement using a 80-neuron, 16-filter CNN filter and deep learning error backpropagation learning algorithm. As a result of the study, this system has an effect of reducing MSE by 28.5% compared to adaptive filter and 17.2% compared to FNN filter.

ACKNOWLEDGEMENTS

Funding of this paper was provided by Namseoul University.

REFERENCES

[1] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-29, pp.113-120, Apr. 1979.
[2] A. Schaub and P. Schaub, "Spectral sharpening for speech enhancement/noise reduction," IProc. of Int. Conf. on Acoust., Speech, Signal Processing, vol.2, pp.993-996, May 1991.
[3] J. S. Lim and A. V. Oppenheim, "All-pole modeling of degraded speech," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-26, pp.197-210, Jun. 1978.
[4] J. Hansen and M. Clements, "Constrained iterative speech enhancement with to speech recognition," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-39, no.4, pp.21-27, Apr. 1989.
[5] J. S. Lim, A. V. Oppenheim and L. D. Braida, "Evaluation of an adaptive comb filtering method for enhancing speech degraded by white noise addition," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-26, no.4, pp.354-358, Apr. 1991.
[6] S. F. Boll and D. C. Pulsipher, "Suppression of acoustic noise in speech using two microphone adaptive noise cancellation," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-28, no.6, pp.752-753, Dec. 1989.
[7] W. A. Harrison, J. S. Lim and E. Singer, "A new application of adaptive noise cancellation," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-34, pp.21-27, Feb. 1986.
[8] M. R. Sambur, "Adaptive noise canceling for speech signals," IEEE Trans. Acoust., Speech, Signal Processing, vol.ASSP-26, pp.419-423, Oct. 1978.
[9] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural Networks, vol.61, pp.85–117, 2015.
[10] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," Cognitive modeling, vol.5, pp.3, 1988.
[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner "Gradient-Based Learning Applied to Document Recognition," Proceedings of the IEEE, vol.86, no.11, pp.2278-2324, November 1998.
[12] E. W. Healy, S. E. Yoho, Y. X. Wang, and D. L. Wang, "An algorithm to improve speech recognition in noise for hearing-impaired listeners," J. Acoust. Soc. Amer., vol.134, no.4, pp.3029-3038, 2013.
[13] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," IEEE Transactions on Audio, Speech and Language Processing, vol.23, no.1, pp.7–19, 2015.