修士論文概要書

Summary of Master's Thesis

Date of submission: 1/29/2020

専攻名（専門分野） Department	Computer Science and Communications Engineering	氏名 Name	Hangyu Song	指　導 教　員 Advisor	渡辺 裕　　㊞
研究指導名 Research guidance	Audiovisual Information Processing	学籍番号 Student ID number	CD 5118FG07- 8		
研究題目 Title	DNN Based Speaker Recognition System				

1. Introduction

Machine learning is a popular research field in computer science since the age of 1960s. Among the methods of machine learning, DNN (deep neural network) is a method that is widely used in image procession [1] and NLP (natural language procession) [2]. With the fact that, DNN has been already applied in other problems in speech procession field [3], we try to introduce DNN to solve the problem of speaker recognition. Especially, the problem of multi-speaker recognition is a difficult point in this field under the situation that multi-speaker speaking at the same time. In this research, we try to build a speaker recognition system with 1D convolution, 2D convolution or LSTM (long short-term memory) for different input pattern for typical scenes to be used, e.g. office with multiple staffs or family.

With 3 stages' experiments, the methods that analyze the voice with CNN (convolutional neural network) in the pattern of spectrograms that based on STFT (short time Fourier transformation) give the best result that with accuracy of 78.58% for voice with length of 8 seconds.

2. Past Research

Researchers had worked on the problem of speaker recognition since 1960s. Before the rise of DNN, the methods used for speaker recognition can be divided into 2 stage: traditional methods (mainly pattern match) and early machine learning, including HMM (hidden Markov model), SVM (supported vector machine) and ANN (artificial neural network, i.e. early neural network).

For traditional methods (represented by pattern match), the accuracy of speaker recognition had already reached a relatively high level before 1980s. However, one limitation of the method pattern match is that, the system is text-dependent, i.e. only with certain content of the voice, the system can recognize the speaker of the voice samples, which is obviously not possible to fulfill the requirement of us.

With the development of computer science, machine learning gradually be used in the field of speech procession (speaker recognition is a branch of speech procession). Machine learning methods can extract features from voice, which made it possible for system to recognize speakers regardless of the content of voice.

The problem is that, for methods of the early machine learning, there is still room for improvement. Except for this, the system with early machine learning methods only deal with speaker recognition problem when only one speaker is speaking, while the scenes to be used in our expectation can be more complex.

Source	Method	Text	Error
Markel and Davis [4]	Long Statistics	Independent	2%@39s
Li and Wrench [10]	Pattern Match	Dependent	21%@3s
Hermann [15]	Pattern Match	Dependent	2%@3s
Reynolds and Rose	HMM	Dependent	0.8%@10s
Tisby	HMM	Independent	2.8%@1.5s
Campbell	SVM	Independent	6.1%@15s
Yegnanarayana and Kishore	ANN	Independent	6.9%@15s

Table 1 Some of Achievement of Methods before DNN

3. Proposed Approach

In our research, we applied DNN on voice waveform to make the speaker recognition. Also, different pre-procession methods were applied to waveform for networks with different structures.

3.1. Pre-procession

We applied different methods for different networks. for networks that use CNN with 1D convolution, the waveform needs no per-procession. The network directly accept waveform as the input of the system.

While for CNN with 2D convolution and RNN (recurrent neural network). The waveform needs to be transformed into 2D matrix, which was done by MFCC (Mel-frequency cepstral coefficients) and STFT.

We know the definition of Fourier transformation:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i t\omega}dt$$

However, making Fourier transformation for waveform will loss the information about time. So, we divided waveform into fragments with certain length and overlap and make Fourier transformation for each fragment. By combining the Fourier transformation results following the time order, a 2D matrix can be made, which is directly the result of STFT. In our research, we only care about the real part of STFT, which means the intensity of waveform in frequency domain and time domain.

For MFCC, it made some follow-up calculation based on the real spectrogram from STFT.

3.2. DNN

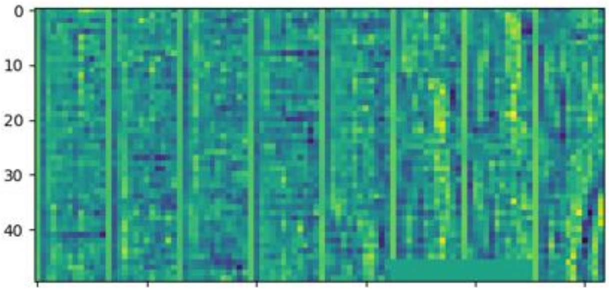

Figure 1 A Sample of MFCC Spectrogram

The DNN methods that used in our research were mainly 1D CNN, 2D CNN and RNN. DNN was widely used in image procession field [1]. The spectrograms are 2D matrix, which is in the similar pattern with image, this suggested that DNN could be used for speech recognition. In fact, DNN had been already used for speech recognition [3].

Similarly, 1D CNN was used for speech recognition [4]. This suggested us that 1D directly applied on waveform might be a good choice.

4. Experiments

The experiments were done in 3 stages.

For the first stage, we used CNN, RNN, CRNN and RCNN on MFCC samples of 2 speakers with only one speaker speaking. For all the network, the accuracy was obviously higher than 50% (random choice). This means that CNN and RNN are available methods for feature extraction for speaker recognition.

In the second and third stage, the samples are the random mix of waveform of 2 speakers' voice. For every 0.5 seconds, a label is made for the sample for if a speaker is speaking in that period.

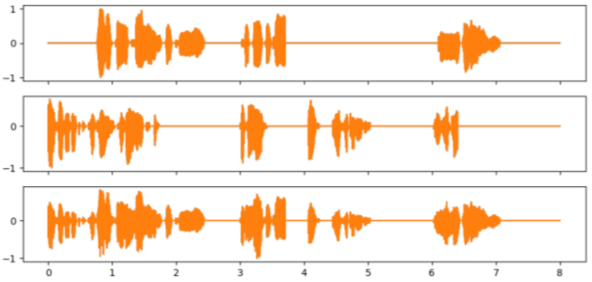

Figure 2 A Sample of Waveform

For example, in Figure 2, the 3 waveform is relatively the voice of Speaker A, Speaker B and mix.

The accuracy of Stage II is about 60.6% with recall value at 78.3% while that of Stage III is 78.58% with recall value ate 80.88%.

5. Conclusion

In our research, we implement systems of speaker recognition that based on DNN with different structure. The results show that, DNN can extract the features of voice waveform about speaker recognition.

There is still room for improvement for networks and will be optimized in our future research.

References

[1] Krizhevsky, Alex, et al. "ImageNet Classification with Deep Convolutional Neural Networks." Communications of the ACM, vol. 60, no. 6, ACM, May 2017, pp. 84–90, doi:10.1145/3065386.

[2] Fan, Yuchen, et al. "TTS synthesis with bidirectional LSTM based recurrent neural networks." *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

[3] Yajie Miao, et al. "EESEN: End-to-End Speech Recognition Using Deep RNN Models and WFST-Based Decoding." *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 167–74, doi:10.1109/ASRU.2015.7404790.

[4] Oord, Aaron van den, et al. "Parallel WaveNet: Fast high-fidelity speech synthesis." *arXiv preprint arXiv:1711.10433* (2017).

DNN Based Speaker Recognition System

A Thesis submitted to the Department of Computer Science and Communications Engineering,

the Graduate School of Fundamental Science and Engineering of Waseda University

in Partial Fulfillment of the Requirements for the Degree of Master of Engineering

Submission Date: January 29th, 2020

Hangyu SONG

(5118FG07-8)

Advisor: Prof. Hiroshi Watanabe

Research guidance: Research on Audiovisual Information Processing

To my beloved mother,

Runqing XU,

who supported my study, my research and **my whole life.**

Acknowledgements

First and foremost, I would love to express my sincere thanks to my supervisor Prof. Hiroshi Watanabe of Department of Computer Science and Communications Engineering, Graduate School of Fundamental Science and Engineering, Waseda University for offering me a precious position in his lab. Without the continuous support from him, it will be impossible for me to finish my study and research in the past two years.

I would also express my thanks to all the members in audio team of Watanabe Lab. The seminar discussion with you was joyful and could always give me inspirations about my research and trends of deep learning field.

Moreover, special thanks are extended to Shufeng Jia and Hongrui Lyu, my best friends since my university time, who helped me get out of confusion and frustration over and over again.

Abstract

Speaker recognition has been an important task in computer science since 1960s. Researches had already made great achievements with methods including pattern matching, hidden Markov model, supported vector machine etc. However, in the conventional approaches, the results of text-independent recognition and the situation that multi-speaker speaking together are not satisfying.

The development of deep learning technologies in recent years gave us the inspirations about application of DNN (deep neural network) in the field of speaker recognition.

With 3 of experiments, we make attempts to build a text-independent speaker recognition system for a multi-speaker case.

In the first stage, we test the performance of CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), CRNN, RCNN on samples of 2 speakers, for which only 1 speaker speaking at the same time. The accuracy of all the network is obviously higher than 50% (random choice), which proves that DNN is available for feature extraction of speaker recognition.

In the following 2 stages, we build 2 different networks with 1D CNN (Stage II) and 2D CNN (Stage III) to make speaker recognition on samples with random mix of voice of 2 speakers. The accuracy of 2 networks are 60.6% (Stage II) and 78.58% (Stage III) respectively.

Key words: DNN, CNN, RNN, Speaker recognition, text-independent, multi-speaker

Contents

1. Introduction

1.1. Motivation

Nowadays, smart equipment with AI technology plays an important role in everyday life for everybody. In the past, people need to input the commands to the equipment by keyboard, mouse, buttons, etc. With the development of computer science and multimedia technologies, it became possible for a piece of equipment with relative sensors (e.g. camera, microphone, pressure sensor, etc.) to recognize and analyze movements, speech and other information given from users as the inputs to the equipment.

Thus, in certain scenes to be used for equipment, the equipment needs to be aware of the user issuing the commands, especially for the scenes in which the user management and user authentication may decide whether the commands will be done or the way to execute the commands. As an instance, the face recognition technology allows the equipment to recognize the person giving gesture or motion command to it and decide how to handle those commands.

At the same time, recent developments in natural language processing enable many platforms to recognize voice commands. Whereas speech recognition technology converts voice information into words, semantic recognition technology allows equipment to understand commands from words.

It is obvious that, for voice commands, in some of the scenes, the sender giving out the voice command also need to be recognized. This requires the equipment to offer the feature of speaker recognition.

Although research in the integrated circuit field has illustrated the fact that it became harder and harder for modern industry to increase the density of transistors on a chip [1], which suggested that the limitation of Moore's law [2] might be reached in foreseeable future, in the past decades, the computing power of integrated circuits has evolved in an incredible amount of time. This became large amount new methods based on hardware in computer science field that introduced in past several years. In the early 2010s, due to the considerable computation power of modern GPUs (graphics processing unit), neural networks, a method that can be traced back to 1960s, have achieved a series of break throughs in traditional subjects such as image classification and text processing. [5, 7, 8, 9, 23]. This highlights the possibility of applying neural networks in speaker recognition. Therefore, in this study, a series of studies on the application of neural networks in speaker recognition is conducted.

1.2. Problem Statement

In past decades, researchers have developed a series of means to recognize speakers of voice. However, these methods suffer from various problems. For instances, pattern matching, which was introduced to this field in 1970s is strongly content (i.e. texts) depended. [3] Otherwise, it requires a voice with a considerable length. [4] This is obviously not a proper choice for this task. Other methods may lack accuracy or require long computation time. About the weakness of past research, the description in details will be found in the second part of this paper, which will not be discussed in this part. In general, past methods cannot be used directly in this task.

As for the neural network approach, the first challenge for this task is to convert voice into network-recognizable information. Voice is recorded as waveform signal, actually 1D array, by microphone that records vibration as an electric signal. Neural networks accept 2D matrix data such as images as input information patterns. [5] This suggests that transforming voice into spectrogram with STFT (short-time Fourier transform) might be a good idea. However, the network can also accept 1D array as input [6]. Experiments are required to decide the appropriate input pattern for the networks.

Another problem with neural networks is that the structure of network varies from case to case. For differently structured networks, the results may be totally different. Much research has been done on the structure of neural network. For Image classification task only, in past few years, researchers have tried to create many networks with different structures. [5, 7, 8, 9]

The question of what is a proper structure of the network that should be applied to this task is also an important question.

1.3. Research Objectives

To solve the problem mentioned in the previous part, we need to make a series of experiments to decide the proper pattern of the inputting voice, the structure of the network. The specific objectives of this study are summarized as follows:

- Collection of the data used in training of and evaluation of networks
- Making labels for samples in dataset
- Attempts about the pre-processing method of voice
- Basic verification of neural network application in speaker recognition
- Construction of networks with possible structure that can be used in speaker recognition
- Training of networks
- Evaluation of networks
- Optimization of structures of networks

1.4. System Overview

For the system used in this research, the input of the system is obviously the voice, i.e. the wave file. After the pre-processing of the wave, (e.g. STFT), the signal will be used as the input of the neural network. After going through the hidden layers of the networks, it will give out a time-sequence results about the speakers at each time point. We will compare the time-sequence results with the labels of the samples to calculate the accuracy and recall value of the network to decide is this network is an available one in this task.

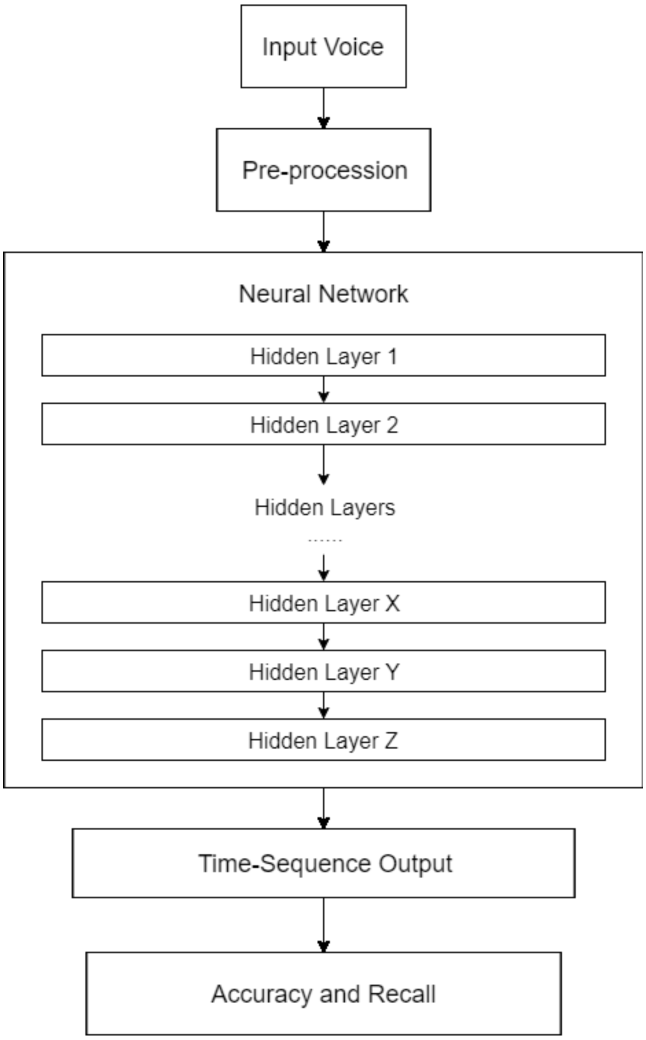

Figure 1.1 System Overview

1.5. Outline

The outline of this thesis is organized as follows:

Chapter 1: A brief introduction of this thesis. In this chapter, the motivation and problems of this research will be introduced. After that, the main aims and the overview of the system to be used is shown.

Chapter 2: Previous works of this thesis. In this chapter, the previous trials that researchers had done for speaker recognition will be introduced. There is mainly 2 parts about it, one is about the traditional methods in voice processing, mainly pattern match. The other one is about the popular machine learning methods used in 1990s in speaker recognition, especially HMM (hidden Markov model). After that, the theory of neural network including CNN (convolutional neural network) and RNN (recurrent neural network) will be introduced. The application of the neural network in other speech processing task be the last part of this chapter.

Chapter 3: The first stage of this research. In this stage, a simple verification research was done to test whether neural network was an available method in the speaker recognition. In this stage, networks with different structure were tested with samples from 2 speakers. In each sample, only voice of one speaker will be recorded. The output of the network is only a single label of the sample.

Chapter 4: The second stage of this research. In this stage, the theory of WaveNet inspired us. We tried to construct a network with similar structure of WaveNet. This network directly processes the signal of wave. The input of this stage is voice samples that may record voice of both speakers. The output of the network should be a time-sequence results to show at each time point, whether the speaker is speaking.

Chapter 5: The third stage of this research. In this stage, we change the structure of network to make it fit the 2D matrix as input. Voice samples are first transformed into STFT spectrogram, and then be used as the input of the network. The input voice samples and output results are in the same pattern of previous stage.

Chapter 6: In this chapter, the conclusion of this research is shown.

2. Previous Work

It is not our team that raised the subject of speaker recognition. It was a subject that had been researched for decades. To understand this subject and the way in which to solve it, paper review was done. The work of paper review mainly contains two parts. One is about the achievement of speaker recognition reached by other researchers. The other part is about the method that will be applied to this field by us, neural network.

According to the past research about our subject, speaker recognition, the room for improvement in this field is discovered. If this problem had been already perfectly solved by other researchers, there would be no necessity for us to do further research in this field. The research about speaker recognition based on computer science started in 1960s. Traditional method that represented by pattern match and traditional statistics model were applied. By reading past papers, the fact that content-independent (i.e. text-independent) speaker recognition is still not satisfied appeared. The method used in the past resulting in lack of accuracy [10] or requiring voice with considerable length [4].

Later, in the time of 1990s, machine learning was introduced in the field of speech processing. The accuracy of speaker recognition became higher compared with that of traditional methods under the condition of text-dependent recognition. [14] However, the improvement in content-independent speaker recognition is limited. Especially that, little research had been done about the multi-speaker recognition. This suggested the purpose of our research should be getting better results for speaker recognition of multi-speaker voice under the condition of content-independence.

Although neural network was proved performing obviously better than methods before, e.g. image processing [5], speech recognition [11] and even disease diagnosis [12]. In some of the fields, it is still not the best choice due to the computing power needed, limitation of acceptable patterns of signal, requirement of large numbers of training samples, etc. To understand the principle of neural network, study need to be done by reading related papers. It is also necessary for us to learn from the application of neural networks in other fields of speech, such as speech recognition and speech generation, which is important reference about the structure and inputting form of the networks that used in our research.

2.1. Previous Work about Speaker Recognition

Thanks to the paper "Speaker Recognition: A Tutorial" by Joseph P. Campbell, Jr in 1997, which is a comprehensive and detailed literature review about the research in speaker recognition by 1997, the achievement made by 1997 was illustrated clearly to us. After that, another paper found by us on *Speech Communication*, "An Overview of Text-Independent Speaker Recognition: From Features to Supervectors" summarized the development of speaker recognition until 2010. Roughly, the development of speaker recognition before the times of deep neural networks can be divided into two parts: traditional methods (mainly pattern match) before 1990s and early machine learning methods (e.g. support vector machine or hidden Markov model) after 1990s.

2.1.1. Pattern Match in Early Years

In early years (i.e. before 1990s) the method most used in speaker recognition is pattern match. When mentioning pattern match, the actual work needs to do is to compare to what extent the two distributions are similar. By calculating the differences between the two

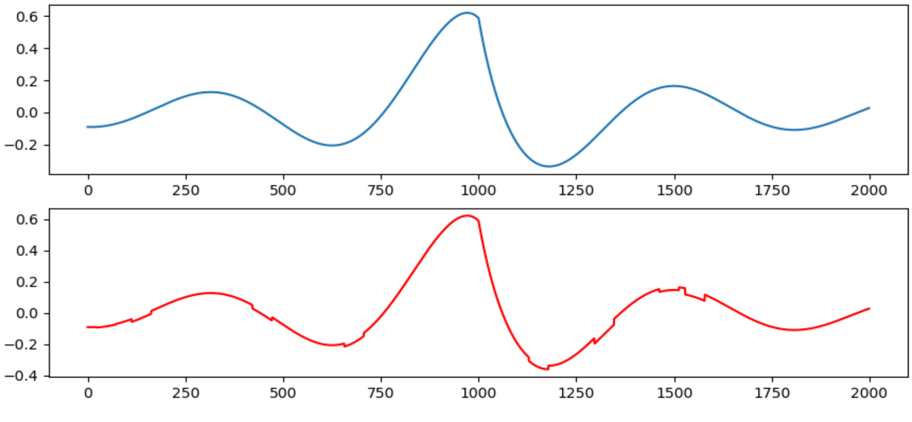

Figure 2.1 Sample of Wave Pattern a (blue) and b (red)

distribution, we can know how much they are different. Given a threshold, it can be decided that if the two patterns are same.

For example, we assume that there are two different curve, a and b. (shown in Figure 2)

It is obvious that there is some little noise on b compared with a. So, b is different to a.

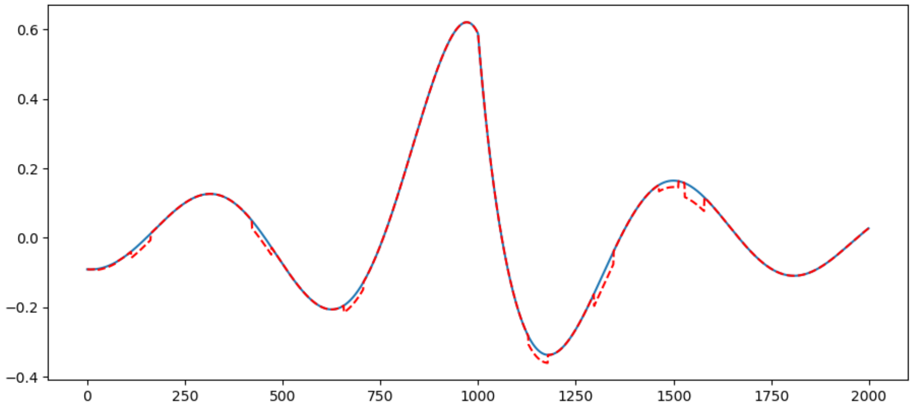

Figure 2.2 a and b are put together

However, if we put a and b together, it is easy to see, the differences between a and b are very limited (shown with Figure 2.2). We can set a rule to calculate the differences between them, for example, MSE (mean squared error) between a and b. If the MSE between a and b is less than the threshold set before, we can claim that a and b are in the same pattern.

When talking about the pattern match in speaker recognition, at most of the times, it needs to be text-dependent. It is obviously that, if the content of the voice is changed, the wave pattern will change by the content.

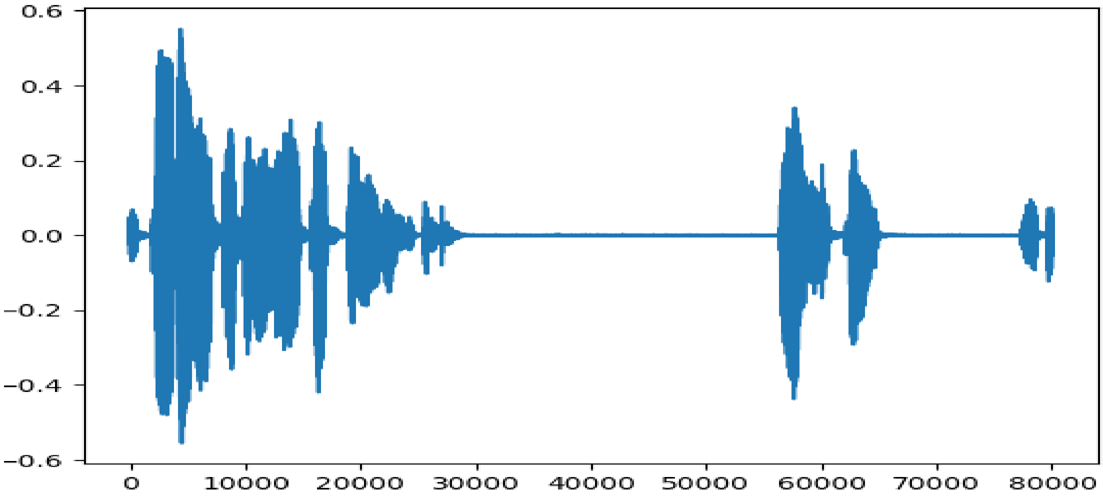

Figure 2.3 A sample of sound wave pattern of voice (5 seconds at sampling rate 16kHz)

So, for the pattern match methods in early years, this technology was also known as "voiceprint" recognition. The work it able to do is to accept a series of sound wave to decide if this wave is similar to the wave pattern that recorded by the system to know if this is the voice by the certain speaker.

As what we do today, in early years, researchers did not only deal with the wave pattern of voice, but also transformed voice into STFT spectrogram to extract the information in frequency domain (will be mention in following chapter). Because the sound wave is a series of value related to the time, i.e. it carries time-sequence information. This means the way to say the same word or the speed of speech will also affect wave pattern of voice. For example, a same sentence that being said fast and slowly, will make 2 different wave patterns. There will be a displacement between 2 wave patterns. By doing easy comparison, 2 wave patterns will be judged as "different" even if the same word is said by one person at different speech speed. One part of the important work had been done in early years is to overcome this effect [15]. For the differences of intensity of wave pattern, the normalization was also done on wave patterns.

Although the research done in early years seems totally different from the work we need to do, we still learnt valuable experience from the pre-processing of sound wave. Some of the methods was also used in our research.

2.1.2. Machine Learning Methods before DNN

With the development of computer science, machine learning methods were introduced to speaker recognition from 1990s. In this period, the typical methods applied in this field contains HMM [16], GMM (Gaussian mixture model) [13, 17], SVM (supported vector machine) [18] and ANN (artificial neural network, i.e. neural network before 2010s) [19].

With the methods of machine learning, researchers finally achieved some acceptable results in text-independent speaker recognition. By training from samples, the system of machine learning could obtain or update the parameters in model. With pre-trained model, the system will give out a result about inputting signal. For machine learning methods, they do not compare patterns mechanically, but also be with the ability to extract features from inputting information. By continually updating the parameters in model, system can also build a model with large quantity of parameters and complex mathematics relation that exceed the Intuitive understanding of human. This makes it possible for machine learning methods to make speaker recognition under the condition of text-independent.

2.1.2.1.Hidden Markov Model

HMM was created based on the principle of Markov process. Markov process is a concept in probability theory and statistics that describes a stochastic process with Markov property, which was defined by Russian Mathematician Andrey Andreyevich Markov. For a stochastic process, if the future probability distribution of it is only decided by the present state of system and be independent of the past state, which means "memoryless" for this process, it is with Markov property. Specially, if the states are dispersed, it is also called "Markov chain". For Markov chain, the probability of events is only decided by the last events. For data in computer science (or in digital form), it is obviously dispersed.

If a series events that can be observed is recorded as X with

$$X = x_1, x_2, x_3, x_4 \dots x_n \qquad (2.1.1).$$

And for each event in X, it is related to a series of hidden states Y where

$$\mathrm{Y} = y_1, y_2, y_3, y_4 \dots y_n \qquad (2.1.2)$$

And Y is a Markov chain, we have

$$P(X) = \sum_{X} P(X|Y)P(Y) \qquad (2.1.3).$$

Which means that for each hidden state, it will lead to an observable state at certain probability. This called hidden Markov model (as shown in Figure 5, hidden states are a Markov chain, and observable states depends on the hidden states). It is easy to know, hidden Markov model is a proper choice to describe or simulate a series of events of time-sequence, which means the order of value or events are decisive. With this Characteristic, before the times of RNN, HMM was widely used in NLP (natural language processing), for example

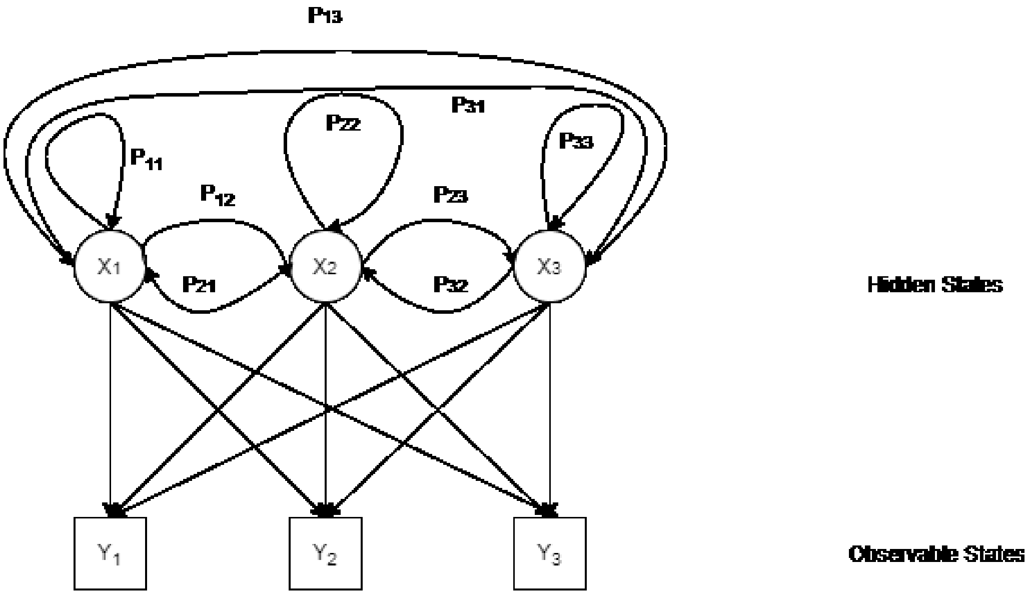

Figure 2.4 A Schematic Diagram of HMM

machine translation [20]. As for voice information, it is also information of time-sequence, i.e. strongly related to the time order, which is in the pattern of Markov chain. In speaker recognition, the only information we can receive is the voice from speakers, while we do not know whom the speaker is. Speakers information is hidden states. For HMM in speaker recognition, the mission is to train a model that always give out the highest possibility between observed states (voice information) and hidden states (speaker information). When solving the problem of speaker recognition, another machine learning method that often used together with HMM is GMM (Gaussian mixture model). GMM is a model assume that assume a series value is the sum of the pattern of Gaussian distribution, i.e. for a value y that related to value x, we have

$$y = f(x) \qquad (2.1.4).$$

And for the mapping $f(x)$ it comes with:

$$f(x) = \sum_{i=0}^{n} k_i G(\mu_i, \sigma_i) \qquad (2.1.5),$$

where $G(\mu, \sigma)$ is for Gaussian distribution with average value μ and standard error σ. Figure 6 is a schematic diagram for GMM, curve a (red), b (green) and c (blue) are 3 curves with different patterns of Gaussian distribution, and curve k (black) is the sum of a, b and c. By training the GMM with samples, it will give out a set of parameters for Gaussian distributions that most close to the given curve. GMM is used to analyze the waveform of voice in speaker recognition. We assume that the waveform of a fragment of voice is approximately made of sum of Gaussian distributions. GMM transform the waveform of voice into parts with the pattern of Gaussian distributions. The parameters as the results of GMM is treated as the "observable states" of the HMM, rather than waveform of voice itself.

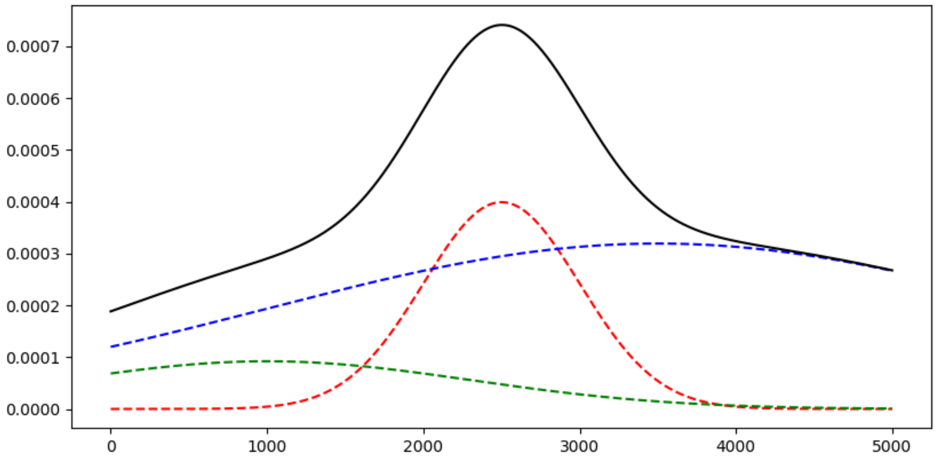

Figure 2.5 A Schematic Diagram for GMM

The problem of HMM is evident: for HMM, the hidden states are only related to the previous state. This means, firstly, the links between data is obviously short, while the voice lasts for several seconds. This makes the model cannot make a fully use of the whole voice information. Another problem for HMM is that, it works well in some situations of uni-direction. However, for bi-direction problem, the performance of HMM is not as good as expectation, which means the model can hardly make fully use of the whole voice information. Thus, it is reasonable that with better model, a better result can be obtained.

2.1.2.2. Supported Vector Machine

SVM (support vector machine) is a machine learning model designed for 2 classes classification problem. The aim of SVM is to find a hyper plane that divide the points from 2

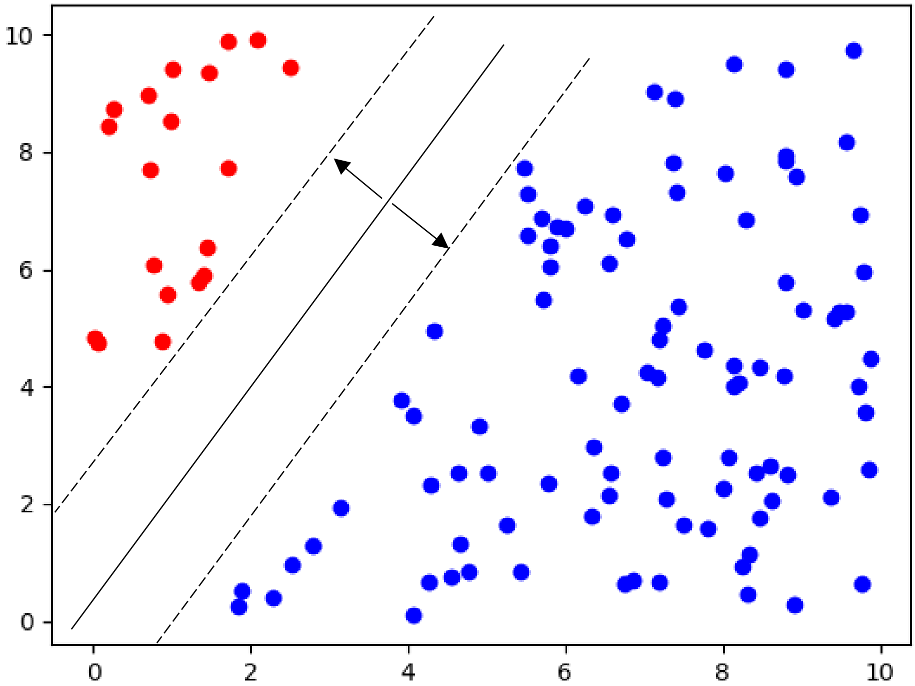

Figure 2.6 SVM: Hyper Plane for 2-Classes Classification Problems

different classes. As shown in Figure 7, an ideal hyper plane will divide the space (2D space in example) into 2 parts and at the location that with the max distance to 2 area. The example in Figure 7 is very simple, which seems that SVM does the similar work as linear programming. However, practically the problem is more difficult, which can be complex enough that obviously beyond the limitation for linear programming. For example, the problem in Figure 8 is obviously very complex that it is hard to be solved by linear programming. Even if a complex combination of linear relationship is applied, it is still difficult to be solved in a 2D space. Of course, with a complex mapping relationship that transform these points into space with higher dimension may be a solution to this problem. For example, we can use a certain relationship of x and y that maps the points into a surface in 3D space like:

$$f(x,y) - b = 0 \qquad (2.1.6)$$

By looking for a plane in the 3D space that can divide the points into 2 different space, the problem can be solved.

However, the complexity of system will burst with the increase of dimensions. Especially that, many problems themselves are already in high dimension. Without methods for reducing dimensions, an image classification problems with images at size 100×100 is a problem of 30000 dimensions. This inspires us that, we can solve the problem in a higher dimension space.

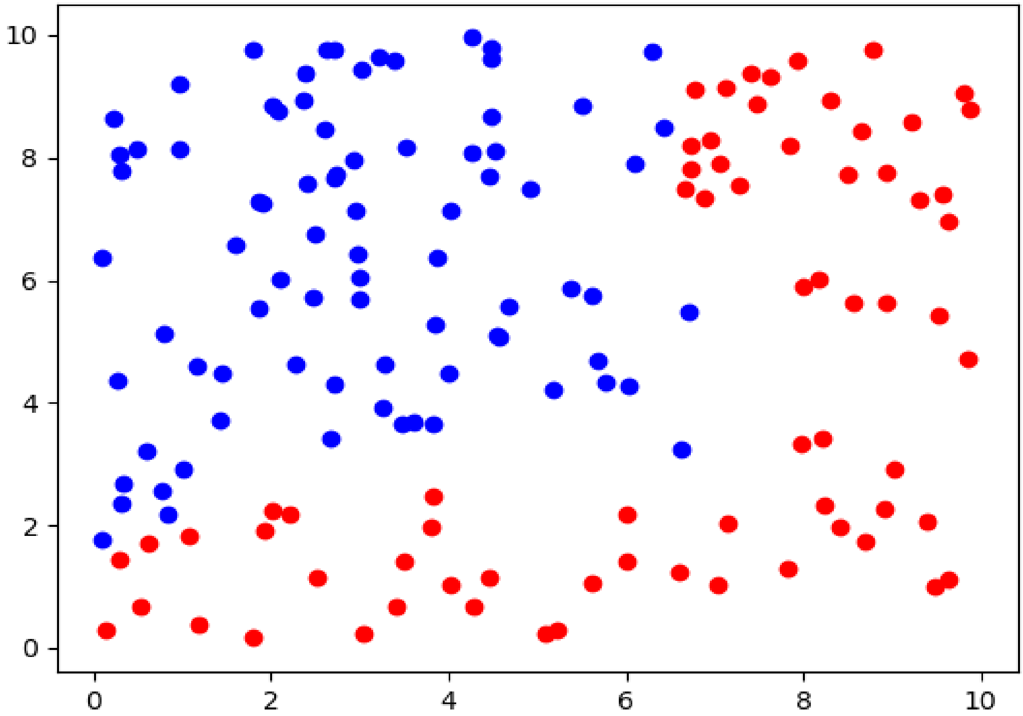

Figure 2.7 Classification Problem that beyond the Limitation of Linear Programming

However, after that, is it possible to make it back to the original space? This is the reason why SVM is created. Of course, the problems that we need to solve is more complex than 2-classes classification problems. With certain strategies, we can make them into problems of combinations of 2-classes classification problems.

SVM was widely used in image processing [21]. In the field of speaker recognition, pre-processing is done for the waveform of voice to extract features of voice. By making the MFCC of waveform as feature extracting methods, in fact for speaker recognition, the problem was treated as image classification [18]. GMM was also used together with SVM in speaker recognition as an option of feature extraction [18].

The problem of SVM is that, the system complexity of SVM is limitation. On one hand, this means the computing efficiency of it is considerably high. One the other hand, for some complex problem, there is still some room between the performance of SVM and expectation. Without any doubt, SVM was one of the best machine learning methods before the development of deep neural networks.

2.1.2.3. Early Neural Networks

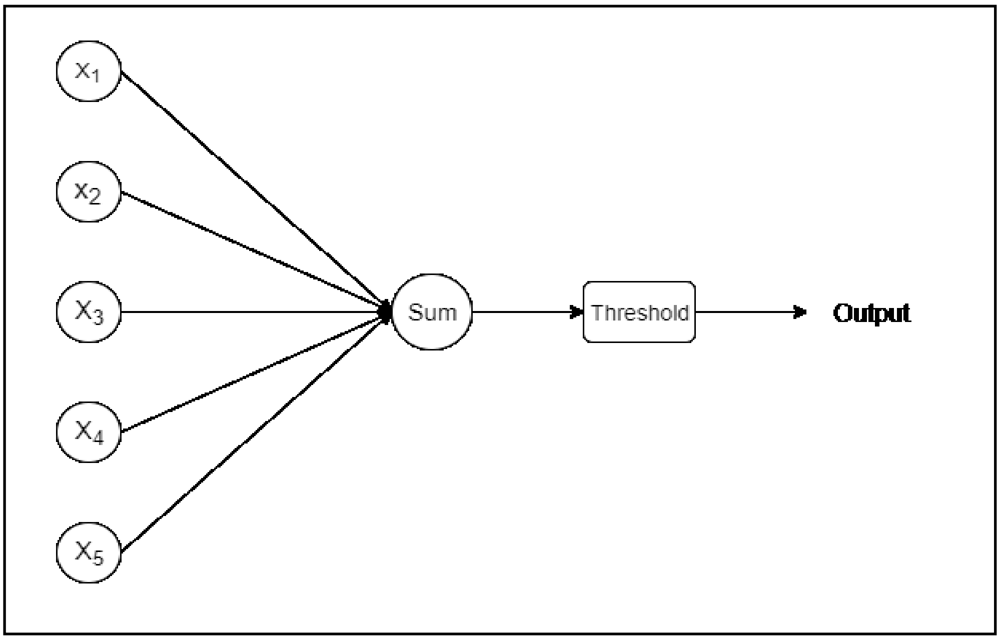

Figure 2.8 Circuit for Simulation of Working Principle of Neuron

Neural network is a concept that was introduced in 1940s as a result of the research of Neurophysiology [22]. The idea of neural network comes from the structure and working principle of brain. At first, a small circuit was designed to simulate the way in which neurons work (shown as Figure 9). It is always a dream that constructing a machine that with the ability to “think” as human. So, it is not a surprise that apply the idea of how neurons work back to computer science. However, in the early years, constructing a system as the organization of brain is extremely difficult. The quantity of neurons in brain is extremely large. Building such a system with so many computing nodes seems to be an impossible mission. Not to mention the computing power needed by this system and the computing power that even larger for the training of this system.

After late 1980s, machine learning became popular. The technologies of neural networks also developed. The form and principle of neural networks is already similar to that of the neural networks we used for deep learning at that time, which will be introduced in the next part of this paper.

Compared with the networks known as DNN (deep neural network), the number of layers and the number of neurons of the neural networks used in that time are lower. Especially that, one of the typical networks used in that time is a kind of network called AANN (autoassociative neural network) [19]. The characteristic of this network is that the number of inputting cells and that of outputting is the same.

Although the ANN that used that time is similar to DNN that we use today. Because of the small number of layers and neurons, when simulating complex relationship, the network cannot give out a result that as accurate as DNN used today.

2.1.2.4. A Brief Summary of Works before DNN

According to the previous parts, it is easy to be found that the methods used for speaker recognition developed greatly. The task of speaker recognition also developed from text-dependent to text-independent.

Source	Method	Text	Error
Markel and Davis [4]	Long Statistics	Independent	2%@39s
Li and Wrench [10]	Pattern Match	Dependent	21%@3s
Hermann [15]	Pattern Match	Dependent	2%@3s
Reynolds and Rose [13]	HMM	Dependent	0.8%@10s
Tisby [16]	HMM	Independent	2.8%@1.5s
Campbell [18]	SVM	Independent	6.1%@15s
Yegnanarayana and Kishore [19]	ANN	Independent	6.9%@15s

Table 2.1 Methods for Speaker Recognition Mentioned in this Chapter

2.2. Deep Neural Network and the Application in Speech Processing

With time going beyond 2010s, the method of neural networks got back to the sight of computer science. Because of the development of GPU, it became possible for researchers to make large scale parallel computing, which is exactly the computing power for neural networks since that the neurons are independent in the same layer.

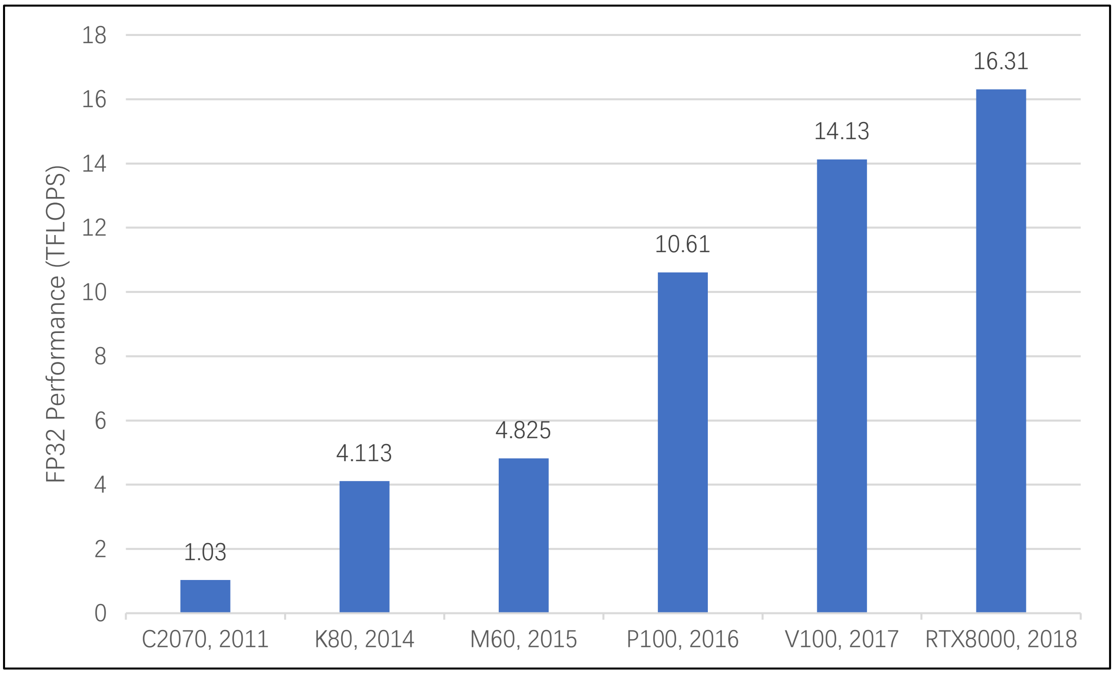

Chart 2.1 FP32 Performance of Flagship Products of Nvidia Tesla Series

The technologies of DNN was first introduced in the field of text processing and CV (computer vision). Later, it was also used to solve the problems in speech processing such as speech recognition (i.e. speech to text) [23].

2.2.1. DNN (Deep Neural Network)

The first difference between DNN and early neural networks is that the number of layers and number of cells in each layer are obviously higher, which means that it is possible for DNN to simulate relationships that very complex. The variety of styles of DNN is another feature of it. The 2 common styles of DNN that most be used are CNN (convolutional neural network) and RNN (recurrent neural network). CNN is designed for 2D matrix with extra ability to process the relationship of one pixel and pixels around it, while RNN is designed for time-sequence data.

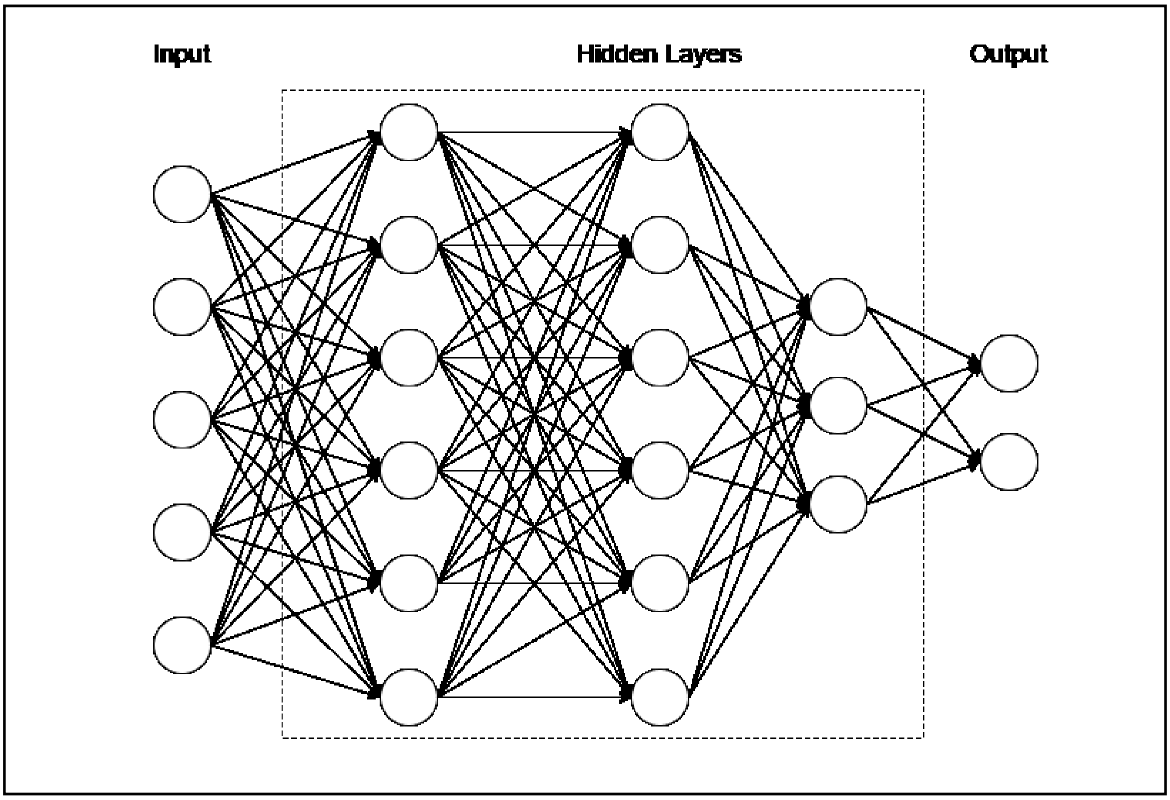

Figure 2.9 Neural Network

Basically, one neural network is made up with input, hidden layers and output. Set fully connected layer as an example, for the later layer $y\ (y_1, y_2, y_3, \dots\ y_n)$ where $y_1, y_2, y_3, \dots\ y_n$ are cells of that layer, we have:

$$y_k = \sum_{i=1}^{m} a_{ki} x_i \qquad (2.2.1),$$

where y_k is one of the cells in the layer and $x_1, x_2, x_3, \dots\ x_m$ are cells in last layer. In some situation, linear relationship can be too simple for the relationship to be simulated, an activation function $G(x)$ will be introduced, so we have:

$$y_k = G\left(\sum_{i=1}^{m} \omega_{ki} x_i\right) \qquad (2.2.2).$$

Sigmoid function, tanh function, ReLU (rectified linear unit) are most commonly used activation function in DNN. By computing layer by layer, the results from inputting data can be calculated, which is known as the "forward propagation". It is easy to see that cells in a single layer are independent, which means value for each cell can be calculated independently. This is the reason that GPU is a good choice for DNN.

The way to train the network is to update the parameters with SGD (stochastic gradient descent) method (although there are varies of optimizing methods, the other methods are only derivative pattern of SGD).

Firstly, loss value that represent errors between output and ground truth need to be calculate by certain function. That is to say:

$$L = l(y, g) \qquad (2.2.3),$$

where g stands for ground truth. For example, cross entropy is often used to calculate differences between probability distributions.

Then, calculate partial derivative of parameters to loss, i.e. gradient. With a certain value "learning rate" r, the parameter can be updated as:

$$\omega' = \omega - \frac{\partial L}{\partial \omega} r \qquad (2.2.4).$$

The problem is the way to calculate the gradient of parameters. To calculate the gradient, back propagation is introduced. Let us take a simple example to show how back propagation works. (shown as Figure 11)

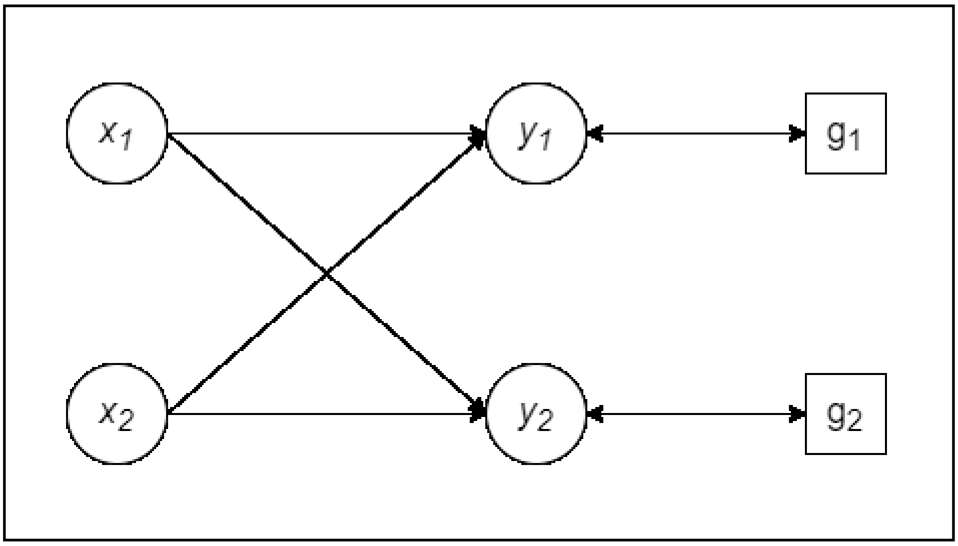

Figure 2.10 A Simple Example of Neural Network

For this figure, we have:

$$\begin{cases} y_1 = \omega_{11}x_1 + \omega_{21}x_2 \\ y_2 = \omega_{12}x_1 + \omega_{22}x_2 \end{cases} \qquad (2.2.5).$$

To calculate the gradient of loss and parameter (set ω_{11} as example), we know that, loss L s related to y_1 and y_2. With chain rule, we have:

$$\frac{\partial L}{\partial \omega_{11}} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial \omega_{11}} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial \omega_{11}} \qquad (2.2.6).$$

Because y_2 is not related to ω_{11}, so actually, (2.2.6) is

$$\frac{\partial L}{\partial \omega_{11}} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial \omega_{11}} \qquad (2.2.7).$$

Since the relationship between L and y is known. So, for every batch, $\frac{\partial L}{\partial y_1}$ is a certain value. And $\frac{\partial y_1}{\partial \omega_{11}} = x_1$ which is also decided for each batch. So, the value of gradient is known. For network with multi-layers, the chain rule can be applied to the network layer by layer from the last layer. So, this is the back propagation of neural network.

The only work that the network can do during training is to reduce the loss value of every batch. With the update of parameters batch by batch, the model may reach a situation in which for all of the inputting data from training dataset, the losses between output and ground truth are always the smallest value.

2.2.1.1. CNN

CNN is a common style of DNN. For CNN, inputting value should be a 2D matrix with 1 or several channels. For example, a grayscale image is a 2D matrix with only 1 channel, while colored images are 2D matrix with 3 channels. Compared with fully connection layer mentioned in previous parts, parameters of convolutional layers are not simply numbers, but convolutional kernels. The number of convolutional kernels in one convolutional layer is $input\ channel\ \times output\ channel$. It is obvious that after all the convolutional calculation, the number of output maps are also $input\ channel\ \times output\ channel$. For each output channel, make the sum of feature maps, so the number of final output maps is $output\ channel$.

The definition of convolution is

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau \qquad (2.2.8).$$

Specially, for image or other 2D matrix, the convolution is discrete 2D convolution. In mathematics, it is defined as

$$M * X = \sum_{j=0}^{n}\sum_{i=0}^{m} M'_{ij} \odot X^{T} \qquad (2.2.9).$$

X is convolutional kernel, M is a 2D matrix. M'_{ij} is a matrix with the same size as X. $\odot$ stands for Hadamard product, which is defined as

$$M \odot N = \sum_{i=1}^{m}\sum_{j=1}^{n} M_{ij}N_{ij} \qquad (2.2.10).$$

In fact, the number of parameters in a CNN is considerable large. We do not really care about the location of numbers in a single convolutional kernel. So, to reduce the complexity of computing, at most of the time, the convolution is calculated as

$$M * X = \sum_{j=0}^{n}\sum_{i=0}^{m} M'_{ij} \odot X \qquad (2.2.11).$$

Convolution is a very useful mathematical tool in image processing. The result of convolution is strongly related to the relationship of a number in matrix and the numbers that around it. So, it is widely used in the feature extraction in image processing.

Before the time of CNN, convolution had already been widely used in image edition. For example, a well-known convolutional kernel $\begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix}$ is used for sharpening of image.

It is important for a kernel to set with sum 1, which makes the intensity of the image remains unchanged. If the sum of a kernel is 0, the kernel can be used to extract boundary of image. A widely used kernel is $\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1. \\ -1 & -1 & -1 \end{matrix}$

Except for feature extraction of image, convolution can also play other roles in image processing. Gaussian blur is a common way to remove noise point on image. In fact, Gaussian blur is to make convolution on the image with a convolutional kernel with Gaussian distribution.

In CNN, a huge number of convolution kernels are applied to the input data layer by layer. For example, in the famous image classification network VGG-16, it carries more than 1 million convolutional kernels. For the kernels in later layers in one network, the feature to extract is very abstract that we can hardly understand what work it does.

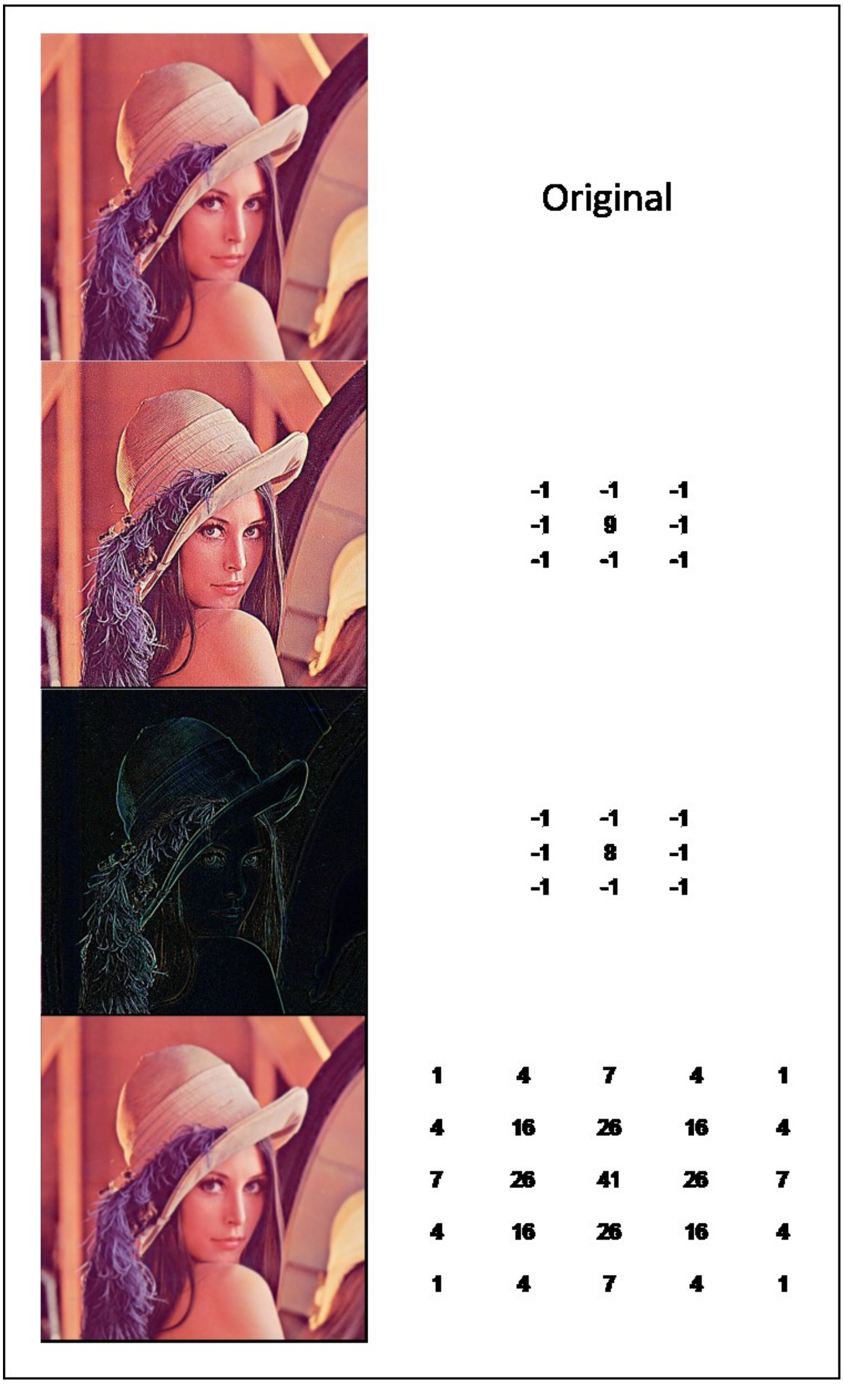

Figure 2.11 Convolution with Different Kernel

For typical CNN, pooling layer is also commonly used to change the scale of data during computing. By doing pooling, the scale of data gets smaller. With convolution kernel with same size, the “view” of the kernel on original image gets larger. There are 2 kinds of pooling layer that often used in CNN, max pooling and average pooling.

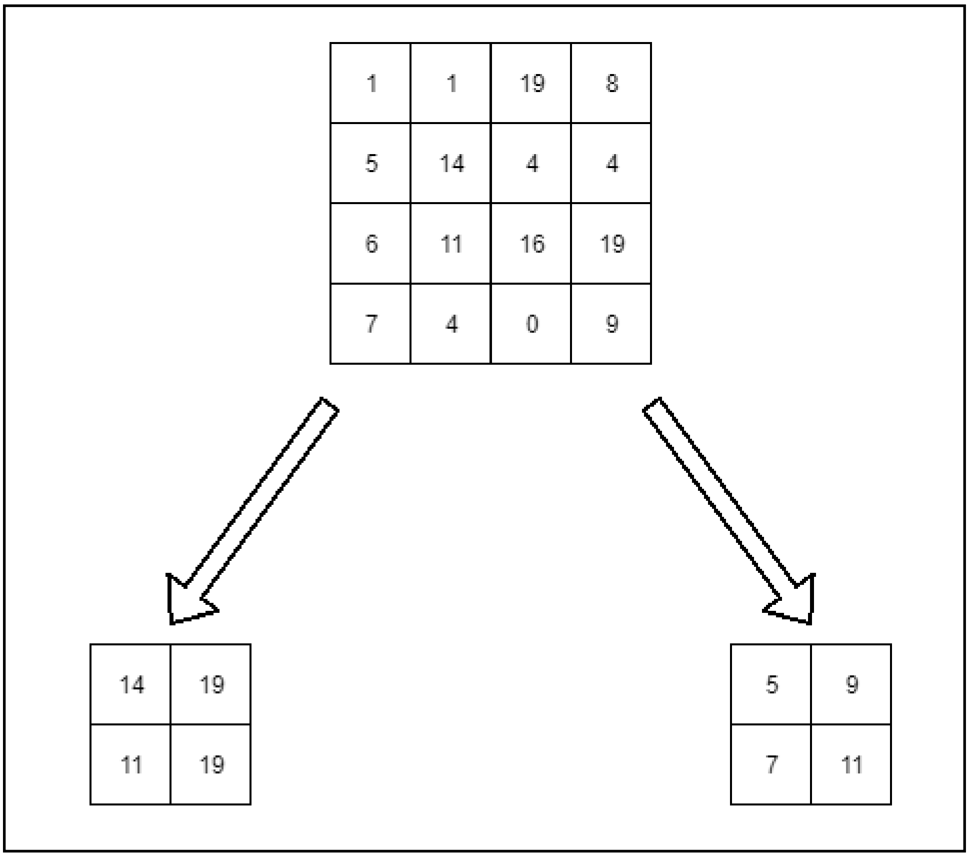

Figure 2.12 2 Kinds of Pooling, Max pooling (left) and Average Pooling (right)

Of course, it is not always done to reduce the scale of data in CNN. In some certain task, it is also necessary to increase the scale of data, for example, super resolution networks or generators in GAN (generative adversarial network).

“Deconvolution” is one of the methods used to increase the scale of data. To do deconvolution, zeros need to be inserted between the pixels of original data. Then, with certain convolutional kernels, do convolution on the data. This is widely used for super resolution problem in 2010s [24]. There are also other ways to expand the scale of data for CNN such as pixel-shuffle [25].

Although the parameters in CNN is not that easy as a fully connection layer, it is still possible to make back propagation in CNN like fully connection layer. The training of CNN follows the same principle.

2.2.1.2. RNN

RNN is another form of network that often used in DNN. The concept of RNN is that, layers in network is made up with RNN cells which make part of the value “go pass” it and part of value “remain” when a series of values go trough them. The cell is made up with certain “gate” and “route”, which makes it possible for cell to control data. During this period, the cell itself also updates.

This makes RNN sensitive to the order of input data. For the example in Figure 14, if the input is not in the order of (X_1, X_2, X_3, X_4), the output of the network may be totally different. The previous input will affect the output. This is similar to a Markov chain. However, the relationship of RNN is longer than that of Markov chain. The last output is still related to the input in the very beginning. One of the typical cells used in RNN is LSTM (long short-term memory), which can be traced back to 1990s [26]. Training of RNN is to adjust the parameters of gates in cells. RNN is widely used in the fields that sensitive to the order of information, such as NLP. For a sentence, if the order of words changes, the meaning of whole sentence might change. For example, although the words that make up sentence “Chinese team beats American team” is the same as sentence “American team beats Chinese team”, the meaning

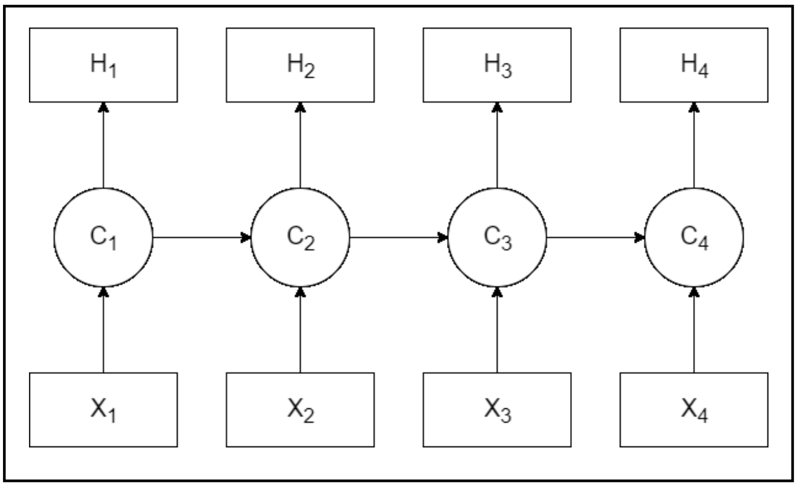

Figure 2.13 RNN

of these 2 sentences are totally different. Similarly, in the field of speech processing, the order

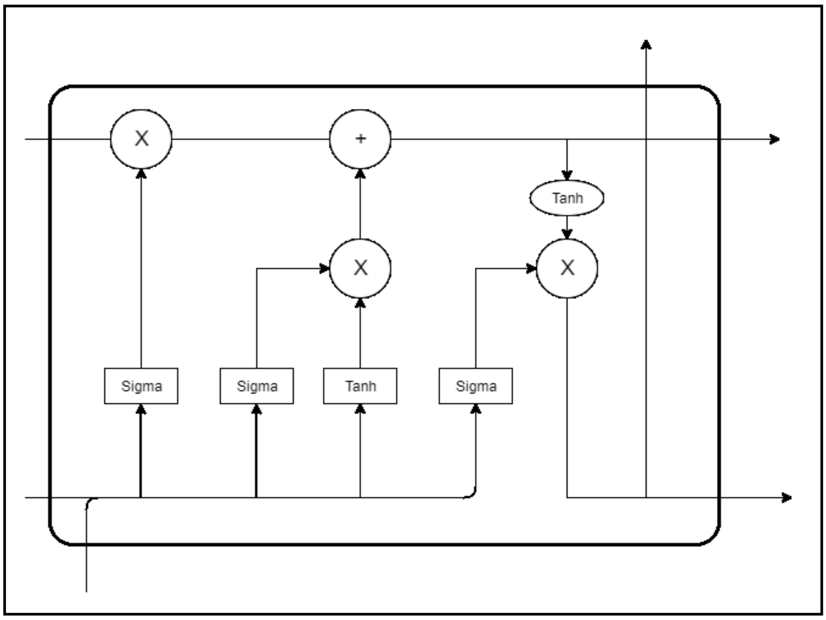

Figure 2.14 A LSTM Cell [27]

of speech is an important element that need to be carefully considered.

2.2.1.3. RCNN and CRNN

It is also common that to use CNN and RNN together. RCNN and CRNN are 2 forms of combination of CNN and RNN. The 2D matrix can be processed by CNN first, and use every column or row as the series input of RNN. This is called CRNN. CRNN is often used in the problems in which the order of information on image is important. For example, in OCR (optical character recognition) problems, the order of letters or kanji decides what exactly the word is. It is a common solution for OCR problem to use object detection system to decide the location of the word on an image. Then use a pre-trained CRNN to recognize the words on image [35].

For RCNN, the data first go through the RNN. A 2D matrix that made up with series output from RNN is used as the input of CNN. In some of the semantic understanding problems, word vectors of sentence first go trough an RNN, and with the output of every word vector connected, a 2D matrix is used as the input of CNN.

2.2.2. Neural Network Applied in Speech Processing

In the past research, DNN was applied to speech recognition (i.e. speech to text) [23]. DNN was used to recognize the phoneme of speech. By combining phoneme of speech, speech (in audio form) can be "translated" to words.

To make the network adapt to speakers with different speech speed, CTC (connectionist temporal classification) is often applied to speech recognition networks.

Speech generation is also an important work for speech processing. In this field, DNN had also been widely applied. There are 2 methods that used in audio generation with DNN. One is to directly generate waveform for audio (speech) [6]. With series of 1D convolution, network directly generate 1D array, which is in fact the waveform of a period of speech. For the method of WaveNet, researchers firstly trained the network to make it generate short period audio, which is similar to phoneme of speech. After that, researchers trained the network to generate speech of certain text. This method reached a better result compared with previous research with RNN [6, 28].

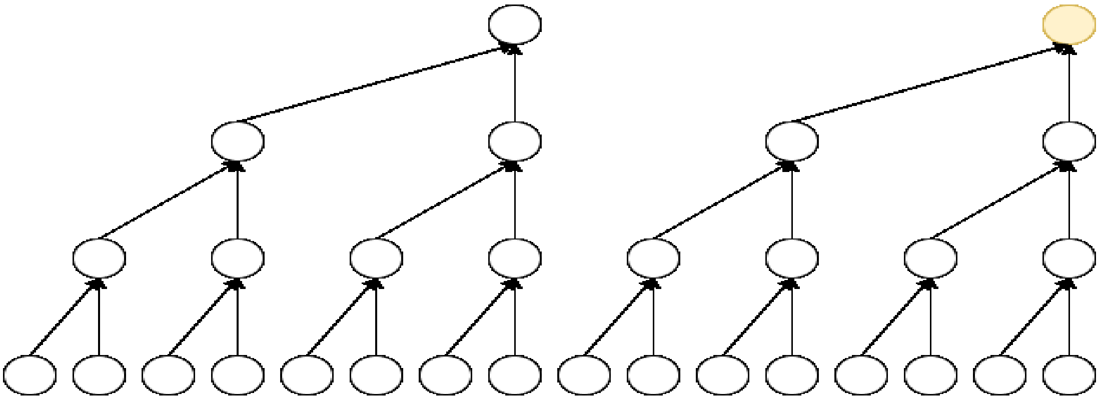

Figure 2.15 1D Convolution Applied in WaveNet Model [6]

The other way to generate speech (or other audio such as music) is to first use DNN to generate spectrograms (at most of the time, STFT intensity spectrogram) for audio with the methods that used for image generation (e.g. GAN). Then, transform the spectrogram to waveform. One problem is that, the STFT has 2 parts, the real part and the imaginary part. For real part, it stands for the intensity information of waveform in frequency domain (generally speaking). And the imaginary is about the information of phase. With only intensity spectrogram, the system is in the lack of some necessary information. Although in the past, researchers had already developed a method called Griffin-Lim algorithm to transform spectrogram back to waveform [30], the results of it is not satisfied. To make the inversion of spectrogram, researchers even used DNN technologies. For example, create a CNN to generate waveform from a spectrogram [31]. This made the problem even more complex, but it is still an idea for solving this problem.

Since that in recent years, the application of DNN in speech processing becomes common. This suggests us that, speaker recognition may be another field in which DNN can be applied.

3. Stage I: Simple Verification of 2 Speakers Recognition

To start the research of speaker recognition with DNN, first an experiment of verification was done. In this stage, CNN and RNN (and the combination of them, i.e. CRNN and RCNN) was used to make a simple classification for audio from 2 speakers to check that if CNN is an available method for speaker recognition.

3.1. Proposed Approach

There were 2 main part of the research in this stage, pre-precession and network building. For pre-precession of the audio, in this stage, we made MFCC (Mel-frequency cepstral coefficients) of the audio samples. For the networks used in this stage, 4 kinds of networks were built for verification, CNN, RNN, RCNN and CRNN.

3.1.1. MFCC

MFCC is a method to extract information of audio to a 2D matrix. It was popular in speaker recognition field in the past [32]. The MFCC of a period of audio can be done by following steps [32]:

- Firstly, divide the audio into small period with certain window scale and overlap.
- Make Fourier transform for each period of the audio
- Map the intensity (i.e. the real part of the result of Fourier transform) to mel-scale
- Take the logarithm of intensity
- Make DCT (discrete cosine transformation) of intensity, which treat the intensity of a window as a “signal”

By making MFCC of a period of audio, the information of audio in frequency domain is extracted. DCT is in fact a method for lossy compression, which furtherly reduce the scale of data. In this stage, aiming at a simple verification of DNN in speaker recognition, it is not necessary to build a very deep, very complex network for it. Controlling of the scale of input data is important for this stage, so, MFCC is a good choice.

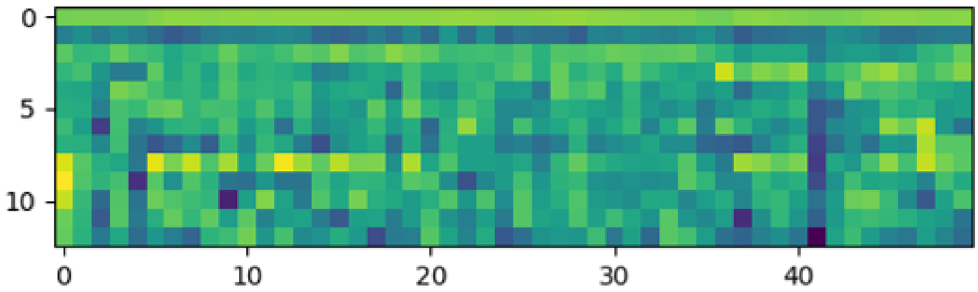

Figure 3.1 A Sample of MFCC

There are many factors that may affect the performance of MFCC [33], so, for networks in this stage, to control the variates in experiments, all of the networks will accept the MFCC with same factors.

3.1.2. CNN, RNN, RCNN and CRNN

In this stage, 4 kinds of networks are applied. Since MFCC is actually a 2D matrix, it is obvious that it can be treated as "image". So, CNN can be used for feature extraction of MFCC. At the same time, the columns of MFCC means features of audio in different period of it, which is a series of information of time-sequence. For every column, it can be used as an input of an RNN. Put columns of the MFCC into RNN one by one, and take the final output as an "eigenvector", the finally results can be made from a series of fully connection layers following the RNN.

Layer Index	Input Size	Layer Description	Output Size

1	$50 \times 100 \times 1$	Conv 3× 3×32 S:1	$50 \times 100 \times 32$
2	$50 \times 100 \times 32$	Conv 3× 3×32 S:1	$50 \times 100 \times 32$
3	$50 \times 100 \times 32$	Maxpooling 2× 2 S:2	$25 \times 50 \times 32$
4	$25 \times 50 \times 32$	Conv 3× 3×64 S:1	$25 \times 50 \times 64$
5	$25 \times 50 \times 64$	Conv 3× 3×64 S:1	$25 \times 50 \times 64$
6	$25 \times 50 \times 64$	Maxpooling 2× 2 S:2	$12 \times 25 \times 64$
7	$12 \times 25 \times 64$	Conv 3× 3×128 S:1	$12 \times 25 \times 128$
8	$12 \times 25 \times 128$	Conv 3× 3×128 S:1	$12 \times 25 \times 128$
9	$12 \times 25 \times 128$	Maxpooling 2× 2 S:2	$6 \times 12 \times 128$
10	$6 \times 12 \times 128$	Conv 3× 3×256 S:1	$6 \times 12 \times 256$
11	$6 \times 12 \times 256$	Conv 3× 3×256 S:1	$6 \times 12 \times 256$
12	$6 \times 12 \times 256$	Conv 3× 3×512 S:1	$6 \times 12 \times 512$
13	$6 \times 12 \times 512$	Conv 3× 3×512 S:1	$6 \times 12 \times 512$
14	$6 \times 12 \times 512$	Maxpooling 2× 2 S:2	$3 \times 6 \times 512$
15	9216×1	FC Unit:1024	1024×1
16	1024×1	FC Unit:2	2×1

Table 3.1 CNN Structure in Stage I

Layer Index	Input Size	Layer Description	Output Size
1	$50 \times 100 \times 1$	BiLSTM 128× 5	256×100
2	256×1	FC Unit:32	32×1
3	32×1	FC Unit:2	2×1

Table 3.2 RNN Structure in Stage I

In this stage, we use a CRNN that with similar structure as the CRNN used for music classification [34]. First the MFCC of audio was put into CNN. After the processing of CNN, a feature map was made. Use a convolutional layer with convolutional kernel with a certain size to reduce the dimension of tensor. Then, the feature sequence was put into RNN. Use the final output of RNN as a “eigenvector”, which was followed by 2 fully connection layer to get final result.

Layer Index	Input Size	Layer Description	Output Size
1	$50 \times 100 \times 1$	Conv 3× 3×32 S:1	$50 \times 100 \times 32$
2	$50 \times 100 \times 32$	Conv 3× 3×32 S:1	$50 \times 100 \times 32$
3	$50 \times 100 \times 32$	Maxpooling 2× 2 S:2	$25 \times 50 \times 32$
4	$25 \times 50 \times 32$	Conv 3× 3×64 S:1	$25 \times 50 \times 64$
5	$25 \times 50 \times 64$	Conv 3× 3×64 S:1	$25 \times 50 \times 64$
6	$25 \times 50 \times 64$	Maxpooling 2× 2 S:2	$12 \times 25 \times 64$
7	$12 \times 25 \times 64$	Conv 3× 3×128 S:1	$12 \times 25 \times 128$
8	$12 \times 25 \times 128$	Conv 3× 3×128 S:1	$12 \times 25 \times 128$
9	$12 \times 25 \times 128$	Maxpooling 2× 2 S:2	$6 \times 12 \times 128$
10	$6 \times 12 \times 128$	Conv 3× 3×256 S:1	$6 \times 12 \times 256$
11	$6 \times 12 \times 256$	Conv 3× 3×256 S:1	$6 \times 12 \times 256$
12	$6 \times 12 \times 256$	Conv 3× 3×512 S:1	$6 \times 12 \times 512$
13	$6 \times 12 \times 512$	Conv 3× 3×512 S:1	$6 \times 12 \times 512$
14	$6 \times 12 \times 512$	Maxpooling 2× 2 S:2	$3 \times 6 \times 512$
15	$6 \times 12 \times 512$	Conv 3× 1×256 S:1	$1 \times 6 \times 256$
16	6×256	BiLSTM 128× 5	256×6
17	256×1	FC Unit:32	32×1
18	32×1	FC Unit:2	2×1

Table 3.3 CRNN Structure in Stage I

For RCNN, series output from RNN was treated as a 2D matrix, which was used as the input of a CNN.

Although the structures of different networks cannot be in all the same complexity, we make them with similar structure as far as possible. For CRNN, we make it with CNN part similar to the CNN we used in this stage and RNN part similar to RNN we used in this stage. Similarly, in RCNN, the first 4 layers RNN part is directly built with same structure as that in the RNN used in this stage and CNN part the same as CNN used in this stage.

For different networks, although they have similar structure, we did not share parameters between different networks. All the networks are trained from beginning.

Layer Index	Input Size	Layer Description	Output Size
1	$50 \times 100 \times 1$	BiLSTM 128$\times 4$	256×100
2	256×100	BiLSTM 50$\times 4$	100×100
1	$100 \times 100 \times 1$	Conv 3$\times 3 \times 32$ S:1	$100 \times 100 \times 32$
2	$100 \times 100 \times 32$	Conv 3$\times 3 \times 32$ S:1	$100 \times 100 \times 32$
3	$100 \times 100 \times 32$	Maxpooling 2$\times 2$ S:2	$50 \times 50 \times 32$
4	$50 \times 50 \times 32$	Conv 3$\times 3 \times 64$ S:1	$50 \times 50 \times 64$
5	$50 \times 50 \times 64$	Conv 3$\times 3 \times 64$ S:1	$50 \times 50 \times 64$
6	$50 \times 50 \times 64$	Maxpooling 2$\times 2$ S:2	$25 \times 25 \times 64$
7	$25 \times 25 \times 64$	Conv 3$\times 3 \times 128$ S:1	$25 \times 25 \times 128$
8	$25 \times 25 \times 128$	Conv 3$\times 3 \times 128$ S:1	$25 \times 25 \times 128$
9	$25 \times 25 \times 128$	Maxpooling 2$\times 2$ S:2	$12 \times 12 \times 128$
10	$12 \times 12 \times 128$	Conv 3$\times 3 \times 256$ S:1	$12 \times 12 \times 256$
11	$12 \times 12 \times 256$	Conv 3$\times 3 \times 256$ S:1	$12 \times 12 \times 256$
12	$12 \times 12 \times 256$	Conv 3$\times 3 \times 512$ S:1	$12 \times 12 \times 512$
13	$12 \times 12 \times 512$	Conv 3$\times 3 \times 512$ S:1	$12 \times 12 \times 512$
14	$12 \times 12 \times 512$	Maxpooling 2$\times 2$ S:2	$6 \times 6 \times 512$
17	18432×1	FC Unit:1024	1024×1
17	1024×1	FC Unit:64	64×1
18	64×1	FC Unit:2	2×1

Table 3.4 RCNN Structure in Stage I

3.2. Experiments

3.2.1. Dataset

The dataset contains 3600 voice samples from 2 speakers (1800 for each speakers) with sampling rate at 16kHz. The length of a single sample is 2.25s. A sample was divided into 8 fragments with length 0.5s, which means cut the voice with window length 0.5s, with overlap of 0.25s.

Although the frequency range of human hearing is about 20Hz – 20kHz (this is the reason why commonly audio file with very high quality is recorded with sampling rate at 44kHz) [36], the frequency range for voice is much smaller than hearing range. Commonly, the high limitation of frequency of sopranos is about 1046Hz (which is known as soprano C). So, the frequency range for MFCC was set as 2200Hz.

For each fragment of the sample, MFCC is made. Together with other fragments in one sample, a 2D matrix was made.

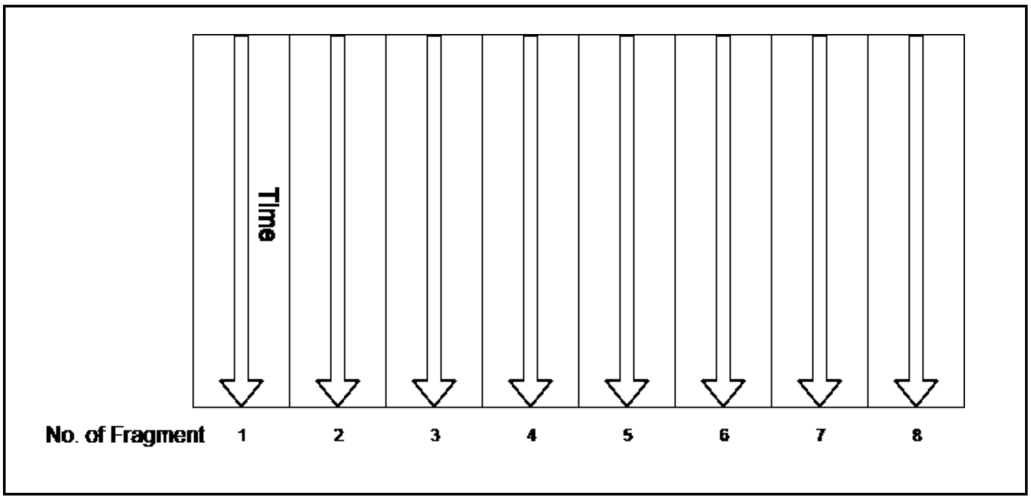

Figure 3.2 The Orgnization of Sample

For every sample, a one-hot label is attached. For samples of speaker A, the labels are recorded as $[0,1]$, while for speaker B, labels are $[1,0]$. The output of networks can be treated as a probability distribution of the speaker prediction, which means cross-entropy is a good choice for loss function of networks. For 2 discrete probability distributions about x, $p(x)$ and $q(x)$, cross-entropy is defined as

$$H(p,q) = -\sum p(x)\log q(x) \qquad (3.2.1).$$

The samples are divided into 2 parts, 2800 for training and 800 for test.

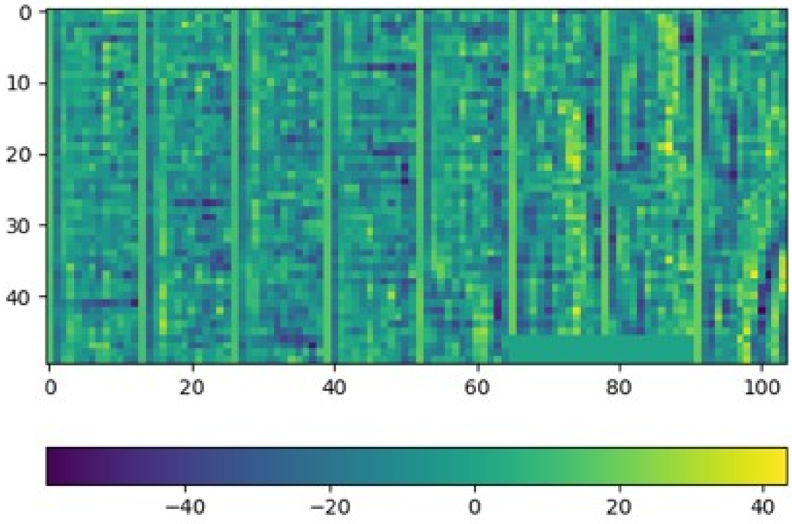

Figure 3.3 A Sample Used in Stage I

3.2.2. Results

After the loss and accuracy of each network reaching a stable value during the training, the parameters of networks are frozen. Test samples are put into each network to evaluate the

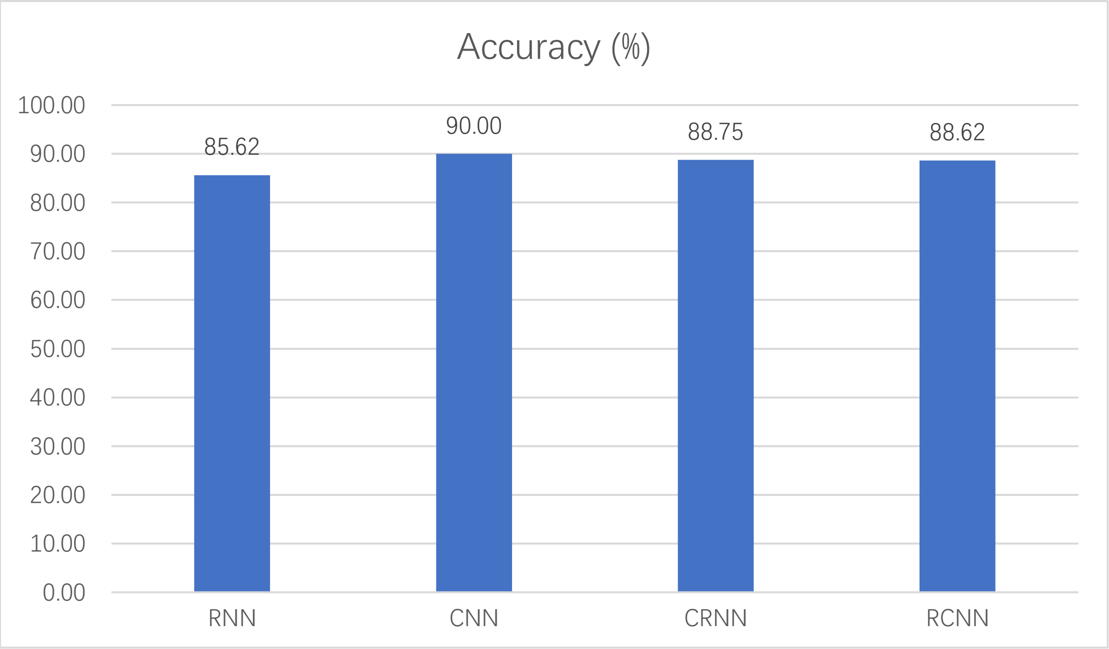

Chart 3.1 Accuracy of Networks in Stage I

performance of each networks.

According to the evaluating accuracy, although there should still been room for improvement for networks in this stage, it is obviously that, for any of the network in this stage, the accuracy is much higher than random choice (50%, for 2 speakers). This suggests that, neural network is an available method in speaker recognition field.

4. Stage II: Time Sequence Recognition

After the varication in Stage I, we will go further in the speaker recognition with neural network. For the samples in Stage I, there is only one speaker speaking. In this stage, we try to process the voice with at most 2 speakers speaking at the same time, which is closer to the scenes to be used for this system in our expectation, for example in an office in which multiple staffs working together.

We used MFCC in the last stage. However, MFCC loses part of the information of waveform of voice. In this stage (and the following stage) the policy is that, we will try to use all of the information from voice in experiments. So, except for the updating of the task in this research, pre-processing methods, or input pattern of the research also changes.

4.1. Proposed Approach

In this stage, we use waveform of the voice itself as the input of this system rather than MFCC in last stage. Of course, it is impossible for us to make deal with the analog signal in this system. However, compared with original analog signal, digital signal with sampling rate enough high can be approximately treated as signal that without loss. To make the processing of waveform itself, the 2D CNN used in Stage I is inappropriate. So, we will apply new methods in this stage.

4.1.1. 1D Convolution

We have talked about the definition of convolution in Chapter 2, including the definition of continues convolution and discrete 2D convolution. In this stage, the input of system is waveform, i.e. 1D array, so, it is normal to apply 1D convolution on signal.

With the definition of continues convolution, we can easily have the definition of 1D discrete convolution.

For 1D array $f(i)$ and convolutional kernel $g(i)$ with length n, the result $y(i)$ is that

$$y(i) = \sum_{m=0}^{n-1} f(i+m)g(m) \qquad (4.1.1).$$

A famous work of neural network with 1D array is WaveNet [6]. It is one of the best audio generation models by now. There are also many derivative versions of WaveNet such as parallel WaveNet [38]. It is mentioned that, WaveNet model with special training can be also used for speech recognition tasks [6], which inspired us to make a network like WaveNet to do the work of speaker recognition.

4.1.2. Network Structure

Layer Index	Input Size	Layer Description	Output Size
1	128000×1	Conv $16 \times 1 \times 32$ S:1	$128000 \times 1 \times 32$
2	$128000 \times 1 \times 32$	Conv $16 \times 1 \times 32$ S:1	$128000 \times 1 \times 32$
3	$128000 \times 1 \times 32$	Conv $16 \times 1 \times 64$ S:16	$8000 \times 1 \times 64$
4	$8000 \times 1 \times 64$	Conv $16 \times 1 \times 64$ S:1	$8000 \times 1 \times 64$
5	$8000 \times 1 \times 64$	Conv $16 \times 1 \times 64$ S:1	$8000 \times 1 \times 64$
6	$8000 \times 1 \times 64$	Conv $16 \times 1 \times 64$ S:1	$8000 \times 1 \times 64$
7	$8000 \times 1 \times 64$	Conv $16 \times 1 \times 64$ S:1	$8000 \times 1 \times 64$
8	$8000 \times 1 \times 64$	Conv $25 \times 1 \times 128$ S:25	$320 \times 1 \times 128$
9	$320 \times 1 \times 128$	Conv $16 \times 1 \times 128$ S:1	$320 \times 1 \times 128$
10	$320 \times 1 \times 128$	Conv $16 \times 1 \times 128$ S:1	$320 \times 1 \times 128$
11	$320 \times 1 \times 128$	Conv $16 \times 1 \times 128$ S:1	$320 \times 1 \times 128$
12	$320 \times 1 \times 128$	Conv $16 \times 1 \times 128$ S:1	$320 \times 1 \times 128$
13	$320 \times 1 \times 128$	Conv $10 \times 1 \times 256$ S:10	$32 \times 1 \times 256$
14	$32 \times 1 \times 256$	Conv $16 \times 1 \times 256$ S:1	$32 \times 1 \times 256$
15	$32 \times 1 \times 256$	Conv $16 \times 1 \times 256$ S:1	$32 \times 1 \times 256$

16	$32 \times 1 \times 256$	Conv $16 \times 1 \times 16$ S:2	$16 \times 1 \times 16$
17	$16 \times 1 \times 16$	Conv $16 \times 1 \times 2$ S:1	$16 \times 1 \times 2$

Table 4.1 Network Structure in Stage II

A full convolutional network is designed for the work in this stage, which is with the similar principle of WaveNet. The input of the network is a period of 8s audio with sampling rate 16kHz, i.e. the input is a 1D array with length of 128000. The output of the network is a tensor with size 16×1 in 2 channels, which is treated as a time-sequence label for the voice.

The details about dataset will be described later in the part about experiment.

4.1.2.1. Gated Activation Unit

In WaveNet, the network used a component named gated activation unit, which is defined as:

$$z = tanh(W_{f,k} * X) \odot \sigma(W_{f,k} * X) \qquad (4.1.2).$$

$\odot$ stands for Hadamard product, which was mentioned in Chapter 2 as $(2.2.10)$ and σ stands for sigmoid function. This unit replaces the activation function in part of the network. This theory was first introduced in PixelCNN [39]. It was used in WaveNet, and this time in our research, we also use it in our network.

The structure of gate activation unit is more complex than that of a single activation function, which means it is with the ability to simulate the relationship that more complex than a normal activation function. Of course, this also calls for more computing power than normal activation

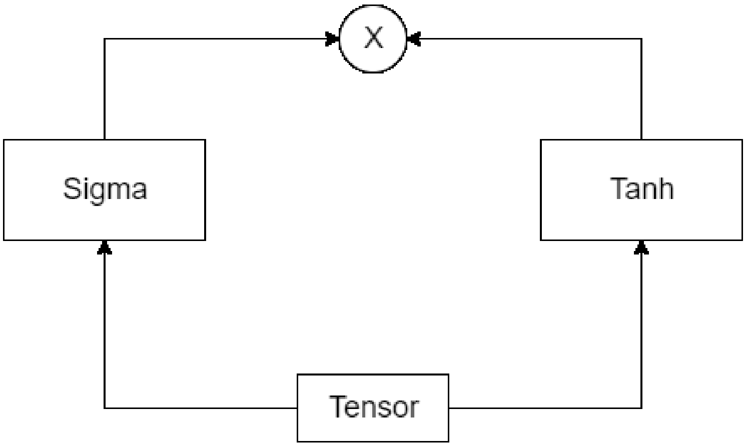

Figure 4.1 Gate Activation Unit

function.

4.2. Experiment

4.2.1. Dataset

We made 5000 training samples by random mix of the waveform of 2 speakers. For every 0.5s, we make a label for each speaker to record if in this period, the relative speaker is speaking. It the speaker is speaking, 1 will be recorded in this slot, otherwise, 0 will be recorded in this slot.

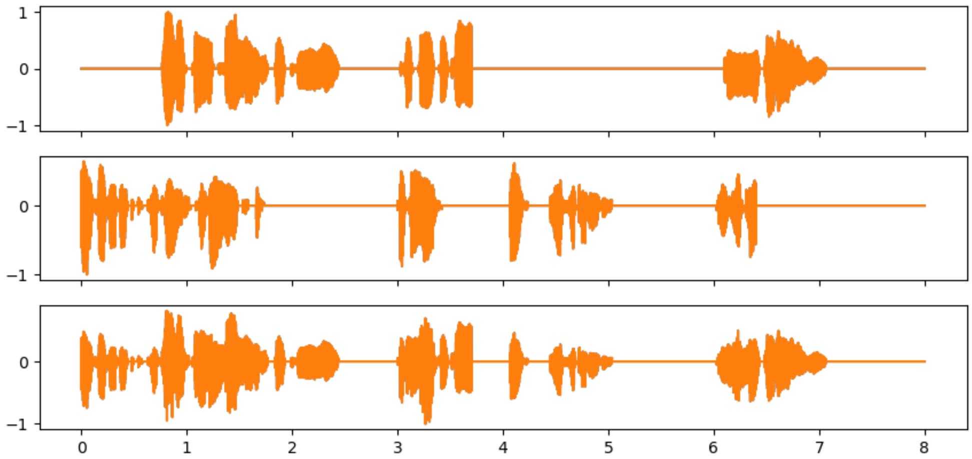

Figure 4.2 A Sample in Training Samples

Take the sample in Figure 24 as an example, the upper waveform is the voice of Speaker A and waveform in the middle is the voice of Speaker B. The waveform in the bottom is the mix of voice of Speaker A and Speaker B. The time-sequence label for this sample is $[[0,1,1,1,1,0,1,1,0,0,0,0,1,1,1,0\,],[1,1,1,1,0,0,1,0,1,1,1,0,1,0,0,0\,]]$.

4.2.2. Result

After 500 epochs' training, the loss of network remains stable. We evaluate the model with 1000 evaluating samples that made in similar way with training samples. For evaluating samples, the waveform that used to make them was not be used for training samples. The accuracy of network is at about 60.6% with recall at 78.3%.

Obviously, this result is not satisfied. The fact that recall value is obviously higher than that of accuracy suggests that incorrect positive result is the main part of error. By checking the evaluating results for every sample, it is found that, the network sometimes makes wrong result that when only one speaker is speaking. For this situation, the network may give out a high probability that 2 speakers speaking at the same time.

5. Stage III: Time Sequence Recognition on Spectrogram

The result of Stage II suggests that, it may be not enough for the network to only analyze waveform itself. The results show that, the distinguishing of 2 speakers for that network is still a difficult mission.

So, in this stage, we come back to the idea that extract the information in frequency domain.

5.1. Proposed Approach

5.1.1. Spectrogram

In this stage, we first make a spectrogram for the voice fragment. The method to make spectrogram of waveform in this stage, is STFT (short-time Fourier transformation). We know the definition of Fourier transformation: for a function $f(x)$, the Fourier transform of it:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-2\pi it\omega}dt \qquad (5.1.1)$$

However, this makes it lost the information of time domain, while we need the time information to make the time-sequence result. So, we use STFT to make the pre-process of voice waveform in this stage. We cut the waveform into fragments with certain window length and overlap and make Fourier transformation for each fragment. Together combine the Fourier transformation of each fragment by the order of time, a 2D matrix can be made. This is the result of STFT.

It is obviously that, the result of Fourier Transformation is a complex value. Specially, for waveform, which is a function of intensity about time, the real part of Fourier transformation of it is the proportion of the waveform in frequency domain. The imaginary part carries the phase information, which is not necessary for our research. So, we use the real part of STFT as the spectrogram in this stage.

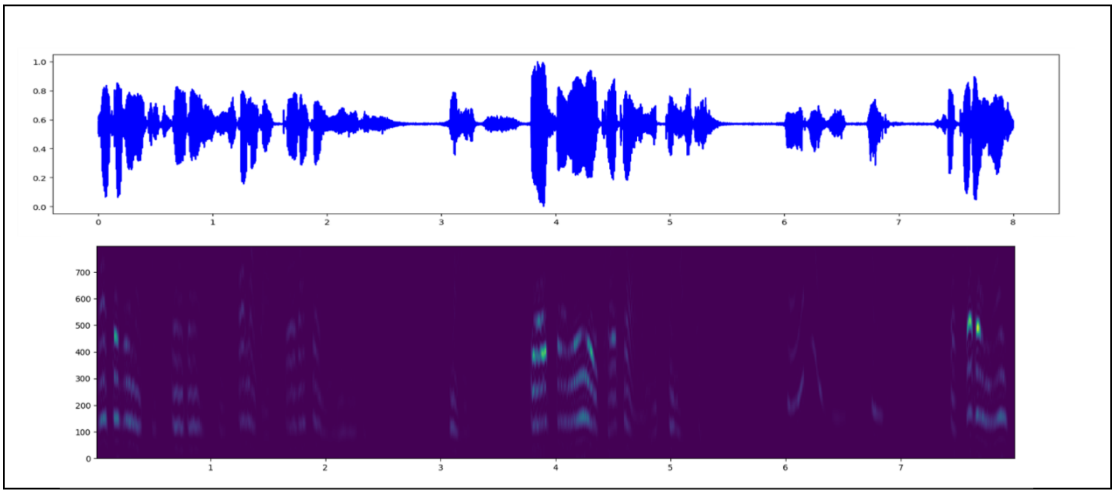

Figure 5.1 A Sample of Voice Waveform

It has been mentioned before that, the high limitation of frequency of voice is at about 1000Hz, so, in this stage, we still only need the spectrogram of 0 to about 1200Hz. Considering the input size of the network (which will be introduced later), we set the parameter of spectrogram as following:

Input File Format	WAV File
Sampling Rate	16kHz
Window Length	20ms (320 Points)
Overlap	5ms (80 Points)
DFT Resolution	2666
Spectrogram Size	$2666\ \times 533$
Input Size	$416\ (0 -\ 1246Hz) \times 512(\ about\ 7.68s)$

Table 5.1 Parameter of Spectrogram

5.1.2. Network

In this stage, we make a neural network that be made up with 2 parts. The first part is darknet-

53 that used in YoloV3 [40] and the second part is a network that designed by us.

5.1.2.1. Darknet-53

This network is designed for image feature extraction in YoloV3 [40]. The darknet-53 is a fully convolution network, with the structure in Figure 26.

Block Number	Input Size	Description	Output Size
1	$416 \times 512 \times 1$	Conv 3× 3×32 S:1	$416 \times 512 \times 32$
1	$416 \times 512 \times 32$	Conv 3× 3×64 S:2	$208 \times 256 \times 64$
1	$208 \times 256 \times 64$	Conv 3× 3×32 S:1	$208 \times 256 \times 32$
	$208 \times 256 \times 32$	Conv 3× 3×64 S:1	$208 \times 256 \times 64$
	$208 \times 256 \times 64$	Residual (+)	$208 \times 256 \times 64$
1	$208 \times 256 \times 64$	Conv 3× 3×128 S:2	$104 \times 128 \times 128$
2	$104 \times 128 \times 128$	Conv 3× 3×64 S:1	$104 \times 128 \times 64$
	$208 \times 256 \times 64$	Conv 3× 3×128 S:1	$104 \times 128 \times 128$
	$104 \times 128 \times 128$	Residual (+)	$104 \times 128 \times 128$
1	$104 \times 128 \times 128$	Conv 3× 3×256 S:2	$52 \times 64 \times 256$
8	$52 \times 64 \times 256$	Conv 3× 3×128 S:1	$52 \times 64 \times 128$
	$52 \times 64 \times 128$	Conv 3× 3×256 S:1	$52 \times 64 \times 256$
	$52 \times 64 \times 256$	Residual (+)	$52 \times 64 \times 256$
1	$52 \times 64 \times 256$	Conv 3× 3×512 S:2	$26 \times 32 \times 512$
8	$26 \times 32 \times 512$	Conv 3× 3×256 S:1	$26 \times 32 \times 256$
	$26 \times 32 \times 256$	Conv 3× 3×512 S:1	$26 \times 32 \times 512$
	$26 \times 32 \times 512$	Residual (+)	$26 \times 32 \times 512$
1	$26 \times 32 \times 512$	Conv 3× 3×1024 S:2	$13 \times 16 \times 1024$
4	$13 \times 16 \times 1024$	Conv 3× 3×512 S:1	$13 \times 16 \times 512$
	$13 \times 16 \times 512$	Conv 3× 3×1024 S:1	$13 \times 16 \times 1024$
	$13 \times 16 \times 1024$	Residual (+)	$13 \times 16 \times 1024$

Table 5.2 Darknet-53 Used in Stage III

Darknet-53 in YoloV3 is first pre-trained with samples from ImageNet as a classification network. It is proved that, compared with other networks used in the past, the accuracy of it can reach a very high point, while the computing time for it is obviously short [40]. So, we use this network to extract features from spectrogram in our research. However, compared with the network for YoloV3, we did not use the pre-trained network. We train it at the same time with the rest parts.

5.1.2.2. Network Designed for Speaker Recognition

After the darknet-53, we used another network to make the output of the whole network. This network is designed to furtherly extract features from the tensor and reduce the dimension.

Layer Index	Input Size	Layer Description	Output Size
1	$13 \times 16 \times 1024$	Conv 3$\times 3 \times 256$ S:1	$13 \times 16 \times 256$
2	$13 \times 16 \times 256$	Conv 3$\times 3 \times 256$ S: (2,1)	$6 \times 16 \times 256$
3	$6 \times 16 \times 256$	Conv 3$\times 3 \times 64$ S: 1	$6 \times 16 \times 64$
4	$6 \times 16 \times 64$	Conv $6 \times 1 \times 2$ S: 1	$1 \times 16 \times 2$

Table 5.3 Network Structure of Reducing Part

5.2. Experiment

In this stage, we used the same dataset and training system as that used in last stage. The only differences between the 2 stages is network and before inputting samples into network, we first make spectrogram of the waveform.

5.2.1. Results

After the training of 1200 epochs, the training loss of the network stays stable. The evaluating accuracy of network is at about 78.58% with recall value at 80.88%. Compared with the results in last stage, there is an obvious increase of accuracy.

6. Conclusion

In this work, we made an attempt to build a speaker recognition system based on DNN in 3 stages. For the first stage, the verification experiment proved that it was feasible to extract features that related to speakers, which is the basis of the following research. Considering that, speaker recognition is a part of the speech processing problem, the methods that proved available in other speech processing problems were introduced to multi-speaker recognition problem.

The methods that using 1D convolution to extract the features from waveform was first used in this problem, of which the idea came from the experience that WaveNet also worked for speech recognition. However, without the information in frequency domain (which is difficult to be extracted for a convolutional network), the network did not work as expected in distinguishing of 2 speakers. This suggested the necessity of pre-processing that able to extract information in frequency domain.

So, in the third stage, spectrograms illustrating the intensity in frequency domain that based on STFT was made for samples. With a network that had been proved to be effective in image processing in YoloV3, features in spectrograms was extracted in the similar way of images. The accuracy of the system increased compared with that of stage II, and finally reached the value of 78.58% with recall value 80.88%.

The results of stage III are not satisfied for practical application for typical scenes to be used. A system that designed for 2 speakers is obviously in the lack of practicality. In the future research, the requirement of practical application should be considered.

7. Appendix

7.1. List of Academic Achievements

[1] Hangyu, Song and Hiroshi, Watanabe. “Comparison of CNN based Illustration Drawing-Style Classification Systems,” *2018 ITE Winter Annual Convention*, 12D-3, Dec. 2018

[2] Hangyu, Song and Hiroshi, Watanabe. “Deep Learning Based Speaker Recognition System”, *ITE Annual Conference*, 13B-1, Aug. 2019

[3] Hangyu, Song and Hirosh, Watanabe. “Content-Independent Speaker Recognition System based on Neural Networks,” *IEICE General Conference*, D-12-9, Mar. 2019

7.2. Bibliography

[1] Iwai, H. “CMOS Technology after Reaching the Scale Limit.” *Extended Abstracts - 2008 8th International Workshop on Junction Technology (IWJT’ 08)*, IEEE, 2008, pp. 1–2, doi:10.1109/IWJT.2008.4540004.

[2] Moore, G.E. “Cramming More Components onto Integrated Circuits.” *Proceedings of the IEEE*, vol. 86, no. 1, IEEE, Jan. 1998, pp. 82–85, doi:10.1109/JPROC.1998.658762.

[3] Atal, B, and Atal, B. “Effectiveness of Linear Prediction of the Speech Wave for Automatic Speaker Identification and Verification.” *The Journal of the Acoustical Society of America*, vol. 55, no. 6, June 1974, pp. 1304–12, doi:10.1121/1.1914702.

[4] Markel, J, and Davis, S. “Text-Independent Speaker Recognition from a Large Linguistically Unconstrained Time-Spaced Data Base.” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 1, IEEE, Feb. 1979, pp. 74–82, doi:10.1109/TASSP.1979.1163201.

[5] Krizhevsky, Alex, et al. “ImageNet Classification with Deep Convolutional Neural Networks.” Communications of the ACM, vol. 60, no. 6, ACM, May 2017, pp. 84–90, doi:10.1145/3065386.

[6] Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).

[7] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[8] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[9] Szegedy, Christian, et al. "Inception-v4, inception-resnet and the impact of residual connections on learning." *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[10] Li, K. P., and Wrench, E. H. “Text-independent Speaker Recognition with Short Utterances.” *The Journal of the Acoustical Society of America*, vol. 72, no. S1, Acoustical Society of America, Nov. 1982, pp. S29–S30, doi:10.1121/1.2019810.

[11] Yajie Miao, et al. “EESEN: End-to-End Speech Recognition Using Deep RNN Models and WFST-Based Decoding.” *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, IEEE, 2015, pp. 167–74, doi:10.1109/ASRU.2015.7404790.

[12] Li, Rongjian, et al. "Deep learning-based imaging data completion for improved brain disease diagnosis." *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, Cham, 2014.

[13] Reynolds, D.A, and Rose, R.C. "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models." *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, IEEE, Jan. 1995, pp. 72–83, doi:10.1109/89.365379.

[14] Kinnunen, Tomi, and Li, Haizhou. "An Overview of Text-Independent Speaker Recognition: From Features to Supervectors." *Speech Communication*, vol. 52, no. 1, Elsevier B.V, 2010, pp. 12–40, doi: 10.1016/j.specom.2009.08.009.

[15] Ney, Hermann. "Automatic Speaker Recognition Using Time Alignment of Spectrograms." *Speech Communication*, vol. 1, no. 2, Elsevier B.V, 1982, pp. 135–49, doi:10.1016/0167-6393(82)90033-4.

[16] Tisby, N.Z. "On the Application of Mixture AR Hidden Markov Models to Text Independent Speaker Recognition." *IEEE Transactions on Signal Processing*, vol. 39, no. 3, IEEE, Mar. 1991, pp. 563–70, doi:10.1109/78.80876.

[17] Leung, Cheung-Chi, and Moon, Y. S. *GMM-Based Speaker Recognition for Mobile Embedded Systems*. ProQuest Dissertations Publishing, 1 Jan. 2004, http://search.proquest.com/docview/305041281/.

[18] Campbell, W.M, et al. "Support Vector Machines for Speaker and Language Recognition." *Computer Speech & Language*, vol. 20, no. 2-3, Elsevier Ltd, 2006, pp. 210–29, doi: 10.1016/j.csl.2005.06.003.

[19] Yegnanarayana, B, and Kishore, S.P. "AANN: An Alternative to GMM for Pattern Recognition." *Neural Networks*, vol. 15, no. 3, Elsevier Ltd, 2002, pp. 459–69, doi:10.1016/S0893-6080(02)00019-9.

[20] Deng, Yonggang, and William Byrne. "HMM word and phrase alignment for statistical machine translation." *IEEE Transactions on Audio, Speech, and Language Processing* 16.3 (2008): 494-507.

[21] Lin, Yuanqing, et al. "Large-scale image classification: fast feature extraction and svm training." *CVPR 2011*. IEEE, 2011.

[22] McCulloch, Warren, and Pitts, Walter. "A Logical Calculus of the Ideas Immanent in Nervous

Activity." *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, Kluwer Academic Publishers, Dec. 1943, pp. 115–33, doi:10.1007/BF02478259.

[23] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013.

[24] Dong, Chao, et al. "Learning a deep convolutional network for image super-resolution." *European conference on computer vision*. Springer, Cham, 2014.

[25] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[26] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[27] Olah, Christopher. The repeating module in an LSTM contains four interacting layers. "Understanding LSTM Networks", 27 Aug. 2015, https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[28] Fan, Yuchen, et al. "TTS synthesis with bidirectional LSTM based recurrent neural networks." *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

[29] Donahue, Chris, Julian McAuley, and Miller Puckette. "Synthesizing audio with generative adversarial networks." *arXiv preprint arXiv:1802.04208* (2018).

[30] Griffin, D, and Jae Lim. "Signal Estimation from Modified Short-Time Fourier Transform." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, IEEE, 1984, pp. 236–43, doi:10.1109/TASSP.1984.1164317.

[31] Arik, Sercan, and Diamos, Gregory. "Fast Spectrogram Inversion Using Multi-Head Convolutional Neural Networks." *arXiv.org*, Cornell University Library, arXiv.org, Nov. 2018, http://search.proquest.com/docview/2092791450/.

[32] Sahidullah, Md, and Saha, Goutam. "Design, Analysis and Experimental Evaluation of Block Based Transformation in MFCC Computation for Speaker Recognition." *Speech Communication*, vol. 54, no. 4, Elsevier B.V, May 2012, pp. 543–65, doi: 10.1016/j.specom.2011.11.004.

[33] Zheng, F, et al. “Comparison of Different Implementations of MFCC.” *Journal Of Computer Science And Technology*, vol. 16, no. 6, SCIENCE PRESS, Nov. 2001, pp. 582–89.

[34] Choi, Keunwoo, et al. “Convolutional Recurrent Neural Networks for Music Classification.” *arXiv.org*, Cornell University Library, arXiv.org, Dec. 2016, http://search.proquest.com/docview/2080825655/.

[35] Baoguang Shi, et al. “An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, IEEE, Nov. 2017, pp. 2298–304, doi:10.1109/TPAMI.2016.2646371.

[36] Cutnell, John D.;Johnson, Kenneth W.;Stadler, Shane;Young, David, Sr. *Physics, 10th Edition*. Wiley, 2015.

[37] Stark, James. *Bel canto: a history of vocal pedagogy*. University of Toronto Press, 1999.

[38] Oord, Aaron van den, et al. "Parallel WaveNet: Fast high-fidelity speech synthesis." *arXiv preprint arXiv:1711.10433* (2017).

[39] van Den Oord, Aaron, et al. “Conditional Image Generation with PixelCNN Decoders.” *arXiv.org*, Cornell University Library, arXiv.org, June 2016, http://search.proquest.com/docview/2079224893/.

[40] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

7.3. List of Figures

7.4. List of Tables

7.5. List of Chart