CNN-Based Speaker Verification and Speech Recognition in Tibetan

Zhenye Gan1,2*, Yue Yu1,2, Rui Wang1,2, Xin Zhao1,2

1College of Physics and Electronic Engineering, Northwest Normal University, Lanzhou, China
2Engineering Research Center of Gansu Province for Intelligent Information Technology and Application
*ganzy@nwnu.edu.cn

Abstract. In recent years, there have been a little studies on speaker and speech recognition in Tibetan, which are mainly based on traditional methods of probability statistics. With the development of deep learning, neural networks have been widely used in speaker and automatic speech recognition, which have achieved remarkable results. In this paper, we utilize end-to-end model to study speaker verification and speech recognition in Tibetan. This article uses the ResCNN network for Tibetan speaker verification. In speech recognition, we adopt the DFCNN-CTC structure, where connectionist temporal classification (CTC) directly outputs the probability of sequence prediction without external post-processing. We have made some improvements to the two models. Experiments show that the improved model reduces EER by 3% and WER by 18% in speaker verification and speech recognition, respectively.

1. Introduction
Speaker recognition can be divided into two types according to different recognition methods: speaker identification and speaker verification [1]. Between them, speaker verification (SV) is to confirm whether a certain speech is spoken by a designated person. According to the recognition object, the SV system can be divided into two categories, text-dependent and text-independent. Text-dependent [2] SV systems require fixed or prompted text phrases to generate speech, while the text-independent SV systems operate on unconstrained speech. In this paper, we mainly study tasks about speaker verification with text-independent.

[3] [4] and [5] introduced end-to-end neural speaker verification systems. [3] used the vector output from the last layer of the hidden layer of the DNN model as speaker embedding, [4] used the last frame output of a LSTM model as an utterance-level speaker embedding, and [5] used a NIN, then used the utterance layer pooling layer to aggregate the frame-level representation.

Automatic Speech Recognition (ASR) [6] aims to automatically transcribe speech into text. In the past few years, deep learning has been successfully used in ASR to improve the accuracy of recognition. Recently, end-to-end models have become mainstream methods, and Convolutional Neural Networks (CNN) are more attractive than other deep learning networks.

Tibetan is a small language, and it is difficult to obtain a corpus, leading to a late start in the study of Tibetan speaker recognition. Liu Bin used probability and statistics algorithms to study speaker recognition and speech recognition in the Tibetan Amdo dialect.

The rest of the paper is organized as follows. In section 2, we first describe the research method.
And in section 3, we describe the deep architecture. The experimental results and analysis are presented in Section 4. Finally, the conclusion is given in section 5.

2. Research Method

2.1. Neural Network Structure

Figure 1 shows the deep architecture. The structure can realize speaker verification and speech recognition. The data processing of the two tasks is the same in the early period, the core architecture uses different frameworks and different loss functions.

In speaker verification, this paper uses the deep speaker embedding system [7], which is based on the ResCNN [8]-based framework for training. The core architecture of the system is the combination of convolution residual block [9] and CNN, so that the neural network reaches a deeper level to better represent the characteristics of speech. To begin with, the average sentence layer converts the frame-level input into the utterance-level speaker representation. Secondly, an affine layer and a length normalization layer map the temporally-pooled features to a speaker embedding. Finally, the cosine similarity is measured by triple loss.

In speech recognition, the core architecture of the system uses the DFCNN structure as the acoustic model. The CTC function can relax this one-to-one correspondence requirement without aligning data and labeling. Training requires only one input sequence and one output sequence.

2.2. Triplet Loss

As shown in Figure 2,[10] introduced the triplet loss function in detail, the triple loss is taken as three samples of input, an anchor (from a specific speaker's utterance), a positive example (from another utterance of the same speaker), and a negative example (from a different speaker). The purpose is to make the cosine similarity between the anchor and the positive example is greater than the cosine similarity between the anchor and the negative example. Formally,

\[
s_{ap} - a > s_{an}
\]  

(1)

2.3. CTC Loss

CTC [11] does not demand any prior alignment between input and output. To align the network outputs with the label sequences, an intermediate representation of CTC path is introduced in. We denote the set of CTC paths for \( z \) as \( \phi(z) \). The likelihood of \( z \) can thus be evaluated as a sum of the probabilities of its CTC paths as follows:

\[
P(z|X) = \sum_{p \in \phi(z)} P(p|X)
\]  

(2)

where \( X \) is the utterance consisting of speech frames and \( P \) is a CTC path. Therefore, CTC loss to be:
\[ L_{\text{c}}(T) = -\ln(P(z|X)) \]  \hspace{1cm} (3)

3. Deep Architecture

3.1. Speaker Verification

In this paper, the bottleneck residual block is used to replace the conventional residual block (ResBlock). Among them, the 1×1 convolution kernel can increase the nonlinearity of the network. Three bottleneck ResBlocks are stacked in our architecture. In the context of this work, the bottleneck residual block can effectively reduce the parameters and improve the operation efficiency.

| Module | Layer name | Structure | Dim |
|--------|------------|-----------|-----|
| Simple ResCNN | Conv64-s | 1×1.64, 3×3.64, 1×1.64 | 2048 |
| | Res64 | 5×5, 64 | |
| | Conv128-s | 1×1.128, 3×3.128, 1×1.128 | 2048 |
| | Res128 | 5×5, 128 | |
| | Conv256-s | 1×1.256, 3×3.256, 1×1.256 | 2048 |
| | Res256 | 5×5, 256 | |

Table 1 shows that the detailed information of Simple-ResCNN architecture. On the basis of the original model, the convolutional layer and the residual layer are reduced and other parameters remain unchanged. We use a single convolutional layer with a filter size of 5×5. The frequency dimension is kept constant in all convolution layers. ReLU and batch-norm layers are not shown.

This architecture has only 5 million parameters compared to the original model (24 million).

3.2. Tibetan Speech Recognition

As shown in Figure 3, this article uses a network structure based on DFCNN for Tibetan speech recognition. In the chart, CTC directly outputs the probability of sequence prediction to the model, and it does not require external post-processing. In order to improve the experimental effect and reduce WER, this article ameliorates the original model. Compared with the original network, the number of convolutional layers with 128 channels has doubled on the basis of the original network, and the other convolutional layers remain unchanged. The purpose is to increase the number of convolution layers to further increases the depth of the network and enhances the expressive power of the neural network.
4. Experimental Evaluation

4.1. Data Sets and Configuration
The corpus used in this experiment. Meanwhile, Tibetan speakers recorded by professional microphones. The sampled rate is 16 KHz, 16bit single-channel WAV format. As is shown in Table 2, the corpus consists of 47 speakers, each with a total of 500 sentences per person and a total of 23500 utterances. Average length of each speaker is 2.5 seconds and more than 16 hours of recorded.

The server configuration is as follows: Operating system is CentOS Linux release 7.7.1908 (Core), Intel Xeon E5-2620 CPU, Tesla K40C GPU with 11G memory.

|          | #speaker | #utterance | #utterance/speaker |
|----------|----------|------------|--------------------|
| train    | 30       | 15000      | 500                |
| dev      | 5        | 2500       | 500                |
| test     | 12       | 6000       | 500                |
| total    | 47       | 23500      |                    |

4.2. Training Details
First, we need to perform voice activity detection on the data, the purpose of which is to delete the silent segment. Secondly, the audio is converted to 64-dimensional Fbank coefficients, normalized to have zero mean and unit variance. Last of all, selecting the best from the randomly selected batches for trained. This feature uses a Hamming window with a width of 25ms and a step size of 10ms generated as a sliding window. The learn rate is set to 0.0001.

4.3. Training Loss for Speaker Verification
Figure 4 shows that the train loss during speaker verification. As can be seen from the left figure, the train loss of the original model is difficult to decrease in the early stage. When the number of iterations reaches 13000, the model loss drops to about 0.55. The figure on the right is the train loss after the model is improved. When the number of iterations reaches 7600 times, the training loss drops to about 0.55. Comparing with the two loss graphs, it can be seen that the time which takes for Simple-ResCNN to reduce the number of iterations of the original model is reduced by at least 40%.

4.4. Evaluation Result and Analysis
For the evaluation of speaker verification, the triple loss function is used to calculate the similarity among anchor and positive and negative, which are then evaluated by EER. Table 3 shows that the verification-results of Tibetan speakers based on the original model. It can be seen that as the number of iterations increases, loss value and EER are gradually reduced. The finally EER reached 13.84%. It can be seen from Table 4 that when the loss values of both models are around 0.55, the EER of the
Simple-ResCNN model is 13.42%. Compared with the ResCNN model, the EER decreases 3% roughly. Table 5 shows the results of Tibetan speech recognition. As can be seen from the table, increased the convolutional layer can reduce WER to 47.59%.

Table 3 Tibetan speaker verification based on original model

| iteration | loss  | EER[%] |
|-----------|-------|--------|
| 200       | 5.19  | 50.75  |
| 5200      | 1.97  | 20.42  |
| 13000     | 0.56  | 13.84  |

Table 4 Results of original and improved models in Tibetan speaker verification

| Model      | Iterate | Loss  | EER[%] |
|------------|---------|-------|--------|
| ResCNN     | 13000   | 0.55  | 13.84  |
| Simple-ResCNN | 7600  | 0.55  | 13.42  |

Table 5 Tibetan speech recognition results

| Model | Epoch | Time[s] | WER[%] |
|-------|-------|---------|--------|
| Basic | 3     | 5190    | 58.46  |
| Our   | 3     | 8564    | 47.59  |

5. Conclusion

This paper used an end-to-end model for speaker verification and speech recognition in Tibetan. Meanwhile, we have improved the network structure on the basis of the original model. In speaker verification, this article proposed a simplified model, which greatly reduced the parameters and the amount of calculation. The results show that it can greatly reduce the training time, reduce the loss value fastly, and EER also has a certain improvement. In speech recognition, this experiment increased the depth of the network, thereby reduced WER, but it need more time.

Acknowledgments

This work is supported by the research fund from the National Natural Science Foundation of China (Grant No.11664036). Additionally, part of this work is performed in High School Science and Technology Innovation Team Project of Gansu (Grant No.2017C-03).

References

[1] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khandpur, “Deep neural network embeddings for text-independent speaker verification,” Proc. Interspeech 2017, pp. 999–1003, 2017.

[2] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, “End-to-end text-dependent speaker verification,” in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016, pp. 5115–5119.

[3] E. Variani, X. Lei, E. McDermott, I. Moreno, and Javier Gonzalez-Dominguez, Deep neural networks for small footprint text-dependent speaker verification, in IEEE International Conference on Acoustics, Speech, and Signal Processing, 2014.

[4] Heigold, G., Moreno, I., Bengio, S., Shazeer, N. End-to-end text-dependent speaker verification. in IEEE International Conference on Acoustics, Speech, and Signal Processing, 2016.

[5] David S., Pegah G., Daniel P., Daniel G.R., Yishay C, Sanjeev K. Neural Network-Based Speaker Embeddings for End-To-End Speaker Verification. in IEEE Spoken Language Technology Workshop (SLT), 2016.

[6] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rn-transducer,” in Automatic Speech Recognition and Understanding Workshop (ASRU), 2017.

[7] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, “Deep speaker: an end-to-end neural speaker embedding system,” arXiv preprint arXiv:1705.02304, 2017.

[8] Yu, Dong, Xiong, Wayne, Droppo, Jasha, Stolcke, Andreas, Ye, Guoli, Li, Jinyu, and Zweig,
Geoffrey. Deep convolutional neural networks with layer-wise context expansion and attention. Interspeech 2016.

[9] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.

[10] Schroff, F., Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 815-823.

[11] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al., “Deep speech 2: End-to-end speech recognition in english and mandarin,” in International Conference on Machine Learning, 2016, pp. 173–182.