Chinese Dialect Recognition Using X-vector with Multi-task Learning

Jingzhe Yang\textsuperscript{1,*} and Tongrui Yu\textsuperscript{2}

\textsuperscript{1}Shanghai Pinghe Bilingual School, Shanghai, China
\textsuperscript{2}RDFZ Xishan School, Beijing, China

*Corresponding author email: yangjingzhe@shphschool.com

Abstract. Speech recognition is often hard for language in countries with diverse accents, like Mandarin. However, the recognition of different dialects may help to improve the later speech recognition. In this paper, we conduct some experiments on a ten-dialect recognition task. We improve the system with different methods, like x-vector and multi-task learning with phone recognition. Finally, with the combination of the methods, we improve the accuracy of baseline model from 76.48\% to 88.25\%, where the relative improvement is 15\%.

1. Introduction

Recognition of Language is to recognize the speaker’s language or dialect from acoustic inputs. The most of the methods depends on the language-specific differences in audio for discrimination [1]. In [2], the authors conduct that there have four kind of methods for language identification, which are classification based on such as GMM (Gaussian Mixture Model) or PRLM (single-language phone recognition followed by language-dependent language modeling) and others. However, this methods are older with the advance of deep learning [3].

In recent years, the deep neural networks (DNN) based methods have been widely adopted in speech related tasks, like speech recognition[4,5], emotion recognition[6], speech enhancement[7,8], speech separation[9], speaker recognition[10] and so on. For language recognition task, there also have some works.

The problem of single channel speech enhancement has always been a difficult task. The traditional method uses a signal estimation method and introduces new noise. The emergence of deep neural networks has improved the research level in this field, for example, DNN can be used to estimate the smoothed ideal ratio mask. The Reference [8] used multi-conditional data to build a regression model to map the relationship between raw data and clear data. Information such as speaker identity and SNR are used as information.

The separation of sound signals could increase the performance based on speech recognition under noise and other factors, and can also extract target music from audio signals. But the separation of mono sound signals is very difficult because most of the signals are not additive mixing but multiplicative mixing. More common methods of sound signal separation include non-negative matrix factorization. Deep neural networks perform better in the separation of sound signals from non-linear models, because the non-linear hidden representation is learned and the output layer can reconstruct signals based on the above information. The literature [9] proposed a joint optimization deep learning model with soft masking.

The deep neural networks used on the automatic speech recognition triggered DNNs for other speech applications. It has been shown that putting DNN into language recognition and speaker recognition
tasks works well. Its main method includes using DNN as a classifier to directly distinguish on speaker recognition. Or DNN extracts frame-level features and so on for training i-vector. The main contribution of [10] is to study the use of DNN for ASR. Previous work has studied its application. We describe the initial LR experiment, which stimulated interest in two indirect approaches. DNN was used to direct do frame-level classification or extract bottleneck feature for i-vector based classifier. Joint factor analysis and i-vector have been used in acoustics and phoneme language recognition. The usual step in these methods is to first model the spectral features. The purpose of statistical modeling of phoneme n-grams is to use speech methods to identify languages. The method of extracting prosodic information from speech and applying it to phoneme language recognition is also a more mainstream method. For example, features of rhythm, accent, and intonation can be proposed according to prosody. Pitch contours are also important for prosody modeling, so speaker recognition can calculate pitch contours and energy contours [11, 12]. Automatic language recognition is widely worked on multilingual translation systems. Although previous studies have introduced phonology, prosody, etc. as information supplements, the new automatic language recognition still requires acoustic model modeling, especially i-vector and classifiers as the main methods [13]. The fully connected network was adopted when direct classification and the results showed that the DNN based system outperformed i-vector based system. In [14], x-vector was used to perform spoken language recognition, which is also used in speaker recognition [15, 16]. The results showed that x-vector outperformed our systems. In this paper, we conduct experiments on a Chinese ten-dialect recognition task. The first one is the x-vector [14], which is often used in speaker recognition task. The other one is the multi-task learning, where the architecture do both the dialect classification and phone/senone recognition tasks. The paper was organized as follows: the part II will introduce some related works; followed by the proposed methods in part III; there are some experimental results in Section 4; and the last section is the conclusions.

2. Prior Work

2.1. X-vector

X-vector, which maps different length speech to the fix-embedding was firstly proposed in [16] for speaker recognition, which is shown in Fig.1 and Fig.2. To obtain fixed-dimensional feature, the mean and standard deviation of the middle hidden layer’s outputs are computed among time axis and concatenated as the input to the final softmax layer for classification. The results showed that x-vector outperformed two i-vector systems. In literature [14], x-vector was used to the task of spoken language recognition. The results showed that the x-vector model trained on large-scale dataset outperform the i-vector baseline significantly. Besides, x-vector system is trainable in a DNN architecture, while i-vector system can just use the DNN bottleneck feature, which is a multi-stage optimization.

| Layer          | Layer context | Total context | Input x output |
|----------------|---------------|---------------|----------------|
| frame1         | \( t-2, t+2 \) | 5             | 120x512        |
| frame2         | \( t-2, t+2 \) | 9             | 1536x512       |
| frame3         | \( t-3, t+3 \) | 15            | 1536x512       |
| frame4         | \( t \)        | 15            | 512x512        |
| frame5         | \( t \)        | 15            | 512x1500       |
| stats pooling  | \( 0, T \)     | T             | 15007x3000     |
| segment6       | \( \{ 0 \} \)  | T             | 3000x512       |
| segment7       | \( \{ 0 \} \)  | T             | 512x512        |
| softmax        | \( \{ 0 \} \)  | T             | 512xN          |

**Figure 1.** The DNN embedding architecture with x-vectors.
Figure 2. Results on various systems.

For deep neural network systems, more data can reduce the phenomenon of overfitting. x-vector uses multiple classes of cross entropy to construct multiple classifiers, and calculates the entropy according to the label. This network can still maintain correct identification when there is a mismatch in channel or time; x-vector and i-vector certain similarities. For example, LDA and PLDA are used. In addition, x-vector can merge new deep neural network structures such as LSTM to further improve performance. The advantages of x-vector include: very fast training speed; no language-specific training set is needed, because an embedding layer is set to extract features and directly scored by PLDA; the recognition rate is higher than previous systems.

2.2. Multi-task DEEP Learning

The multiple task learning, where several tasks may share lower layers in a single DNN architecture, but has different output layers for different task. Thus, multi-task learning will have more than one loss, where different losses focus on different task. Then during the training, the loss of different tasks may help to improve the shared lower layers to extractor better deep features for input audio.

Multi-task learning generally means the simultaneous and parallel of multi-related tasks, the simultaneous back-propagation of gradients, and multiple tasks to help each other learn through the underlying shared representation.

Compared with single-task learning, the benefit of it is that during single-task learning, the model space between each task is independent of each other, and the backpropagation of gradients tends to fall into local minima. The small values are in different positions, and the interaction can help the hidden layer escape from the local minimum. When multi-task learning, the model space between multiple tasks is shared.

3. X-vector and Multiple Task Learning with Phone Recognition

The primary idea of our method is indicated in Fig.3. The input to the neural network is Fbank feature. 3-layer long short-term memory (LSTM) [19] layers are the long-time dependency across time axis with dropout [20] applied to prevent over-fitting.

After LSTM layers, we can adopt two strategies to map variable-length output to fix-dimension vector, or embedding vector. The first one is simply averaging the outputs of LSTM layer across time axis. The second one is to use x-vector, i.e., to compute the parameters of LSTM’s output axis and then concatenated statistics feature is feed to next layer.

The embedding vector is then feed to two fully connected (FC) layers, with ReLU and Softmax activation functions. When training with multi-task learning, another one or two separated FC layers will be added to the architecture for output phone or senone probability vector for speech recognition related task. The final loss function is the combination of dialect recognition (DR) task and automatic speech recognition (ASR) task.

$$L = L_{DR} + \alpha L_{ASR}$$

Where $\alpha$ is the weight of ASR task. To ensure ASR task make the early stage training easily and the final stage training mainly focus on DR task, we also propose a progressive training strategy, where $\alpha$ is set to relative larger in the beginning of training, and will be decayed by 0.8 after every epoch.
4. Experimental Results

4.1. Datasets
The dataset used for experiments is provided by iFlytek’s AI challenge [21]. The dataset contains 10 Chinese dialects, which are Minnan, Nanchang, Kejia, Changsha, Shanghai, Hebei, Hefei, Sichuan, Ningxia and Shanxi. The training set contains 60k sentences, with 6k sentences and 30 speakers each language. The test set contains 5k sentences, which are consist of about 2500 long sentences (>3s) and about 2500 short sentences (<3s). The later short sentences are more harder for recognition. The feature inputs to LSTM are 40-dim Fbank extracted by Kaldi toolkit [22]. The training of the above described architecture uses PyTorch toolkit [23]. Each LSTM layer has 512 cell states. The training will continue with 20 epochs with initial rate of 0.001 until there are 4 epochs with any improvement on the dev-set. Once the performance on it is decreased, the learning rate will be halved and the model of that epoch is deprecated. The Adam [24] optimization is used for neural network training.

4.2. Experiments Based on X-vector
The baseline system use the strategy of averaging the LSTM’s output for embedding vector extracting. We then improve the baseline with x-vector based NN structure. Shown in Tab. 1, it is the accuracy (Acc.) included long, short and total test sentences are listed. As indicated in the table, the x-vector system outperforms the simple average strategy by about 5% relative increasing in accuracy. Also, the accuracy on shorter sentences is a little worse than the longer sentences.

4.3. Experiments based on Multi-Task Learning
To obtain the ASR frame-level phone/senone label, we use Kaldi toolkit to build a TDNN-HMM [25] based speech recognition system with phone transcription provided in the dataset. The system has 178 phones and 2888 senone states. The alignments of training set is generated by the TDNN-HMM based system.
First, we compared the performance of phone and senone based ASR labels and the results are in Tab. 2. The phone or senone based multiple task learning will improve the performance and phone recognition task outperforms senone recognition task slightly. We conduct that the other task will help the neural networks learn better representations of input features and can also prevent the neural networks from over-fitting. We also notice that with the combination of both phone and senone recognition, the system has a relative 8.83% improvement than the pure x-vector system.

4.4. Progressive Training
In this subsection, we will show the benefits of progressive training in Table 3. As illustrated in the table, the progressive training can still improve the system slightly by absolute 0.94%, which is line with our assumptions in Section 3.

| Table 1. The recognition results of average based and x-vector based systems. |
|-----------------------------|------------------|------------------|------------------|
| Embedding Strategy | Acc. (short) | Acc. (long) | Acc. (Total) |
| average | 75.28% | 77.68% | 76.48% |
| x-vector | 79.56% | 80.88% | 80.22% |

| Table 2. The recognition accuracy of multi-task learning system. |
|-----------------------------|------------------|------------------|------------------|
| Methods | Acc. (short) | Acc. (long) | Acc. (Total) |
| x-vector | 79.56% | 80.88% | 80.22% |
| x-vector + multi-task(phone) | 85.46% | 88.96% | 87.21% |
| x-vector + multi-task(senone) | 85.62% | 88.04% | 86.83% |
| x-vector + multi-task(phone+senone) | 85.79% | 88.84% | 87.31% |

| Table 3. The performance of progressive training. |
|-----------------------------|------------------|------------------|------------------|
| Methods | Acc. (short) | Acc. (long) | Acc. (Total) |
| x-vector + multi-task(phone+senone) | 85.79% | 88.84% | 87.31% |
| x-vector + multi-task(phone+senone) + progressive training | 86.39% | 90.12% | 88.25% |

5. Conclusion
In this paper, one combine x-vector and multi-task learning for Chinese dialect recognition is proposed. The x-vector system is significantly improved with the help of multi-task learning. Also, the proposed progressive training also benefits the system from the training process. Next, we are going to continue improve the system with some data augmentation techniques.

References
[1] Shen, Wade, Nancy Chen, and Douglas A. Reynolds. "Dialect recognition using adapted phonetic models." Ninth Annual Conference of the International Speech Communication Association. 2008.
[2] Zissman, Marc A. "Comparison of four approaches to automatic language identification of telephone speech." IEEE Transactions on speech and audio processing 4.1 (1996): 31.
[3] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.
[4] Seltzer, Michael L., Dong Yu, and Yongqiang Wang. "An investigation of deep neural networks for noise robust speech recognition." 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013.
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.

Han, Kun, Dong Yu, and Ivan Tashev. "Speech emotion recognition using deep neural network and extreme learning machine." Fifteenth annual conference of the international speech communication association. 2014.

Xu, Yong, et al. "A regression approach to speech enhancement based on deep neural networks." IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 23.1 (2015): 7-19.

Xu, Yong, et al. "An experimental study on speech enhancement based on deep neural networks." IEEE Signal processing letters 21.1 (2013): 65-68.

Huang, Po-Sen, et al. "Deep learning for monaural speech separation." 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014.

Richardson, Fred, Douglas Reynolds, and Najim Dehak. "Deep neural network approaches to speaker and language recognition." IEEE signal processing letters 22.10 (2015): 1671-1675.

Richardson, Fred, Douglas A. Reynolds, and Najim Dehak. "A Unified Deep Neural Network for Speaker and Language Recognition." Sixteenth Annual Conference of the International Speech Communication Association. 2015.

Martinez, David, et al. "iVector-based prosodic system for language identification." 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2012.

Lopez-Moreno, Ignacio, et al. "Automatic language identification using deep neural networks." 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2014.

Snyder, David, et al. "Spoken Language Recognition using X-vectors." Odyssey. 2018.

"X-vector DNN refinement with full-length recordings for speaker recognition", Daniel Garcia-Romero, David Snyder, Gregory Sell, Alan McCree, Daniel Povey, Sanjeev Khudanpur, Interspeech 2019.

David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, "X-vectors: Robust DNN Embeddings for Speaker Recognition", Sanjeev Khudanpur, ICASSP 2018

Seltzer, Michael L., and Jasha Droppo. "Multi-task learning in deep neural networks for improved phoneme recognition." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.

Chen, Nanxin, Yanmin Qian, and Kai Yu. "Multi-task learning for text-dependent speaker verification." Sixteenth annual conference of the international speech communication association. 2015.

Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.

Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.

http://challenge.xfyun.cn/2018/aicompetition/tech. (2019, December 2).

Povey, Daniel, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann et al. "The Kaldi speech recognition toolkit." In IEEE 2011 workshop on automatic speech recognition and understanding, no. CONF. IEEE Signal Processing Society, 2011.

Peddinti, Vijayaditya, Daniel Povey, and Sanjeev Khudanpur. "A time delay neural network architecture for efficient modeling of long temporal contexts." Sixteenth Annual Conference of the International Speech Communication Association. 2015.