Future Generation Computer Systems 100 (2019) 751–758

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A lighten CNN-LSTM model for speaker verification on embedded devices

Zitian Zhao [a], Hancong Duan [a,*], Geyong Min [b], Yue Wu [a], Zilei Huang [a], Xian Zhuang [a], Hao Xi [a], Meirong Fu [a]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China
[b] Department of Computer Sciences, University of Exeter, Exeter, Devon, EX4 4QF, United Kingdom

HIGHLIGHTS

- Defects of Additive Angular Margin loss could lead to training failure are analyzed.
- An end-to-end lighten network scheme with bilateral LSTM cells is proposed.
- A three-phase training method is introduced for training and fine-tuning the model.
- The model reaches state of the art in a 200 persons verification test.
- The model can run in real time on a commercial off-the-shelf embedded platform.

ARTICLE INFO

Article history:
Received 11 December 2018
Received in revised form 26 March 2019
Accepted 21 May 2019
Available online 23 May 2019

ABSTRACT

Augmented by deep learning methods, the performance of speaker recognition pipeline has been drastically boosted. For the scenario of smart home, the algorithms of speaker recognition should be user friendly and has high speed, high precision and low resource demand. However, most of the existing algorithms are designed without considering these four performance requirements simultaneously. To fill this gap, this paper proposes a text-independent speaker verification model. Specifically, the lighten network scheme is constructed using one convolution layer, two bilateral Long Short-term Memory (LSTM) layers and one fully connected layer. Utterance segments are mapped to a hypersphere where cosine similarity is used to measure the degree of difference between speakers. Then we analyze the defects of Additive Angular Margin (AAM) loss and propose a 3-stage training method. Softmax pre-training is used for avoiding divergence. After pre-training, AAM loss is adopted to boost training process. In the end, we use triplet loss to further fine-tune the model. Short-term speech utterances are used in training and testing. The experimental results demonstrate that the proposed model reaches 1.17% Equal Error Rate (EER) on a 200 persons benchmark with real-time inference speed on a generic embedded device.

1. Introduction

Voice print, similar to finger print, is one of the unique characters for human identification. Voice Print Recognition (VPR), also termed as Speaker Recognition (SR), is the technology that automatically recognizes human voice prints, which is then used to identify the speakers. There are two types of tasks in SR: speaker identification and speaker verification. For speaker identification, each audio is assigned a known label associated with the corresponding speakers. While identification is a close-set problem, speaker verification is an open-set one and decides whether two audios belong to the same speaker. According to different application scenarios, speaker recognition tasks can be divided into two categories: text-dependent and text-independent. In the former scenario, users will be requested to pronounce the specified text transcript. For the text-independent style, speech of arbitrary verbal contents can be utilized. It does not need the extra cooperation of users and is in line with human habits.

Previously, Reynolds et al. [1] proposed GMM–UBM (Gaussian Mixture Model–Universal Background Model) in the speaker verification task. One major limitation of this work is that Gaussian components are independent from each other in GMM–UBM. To address this problem, Kenny et al. [2] proposed Joint Factor Analysis (JFA) and i-vector model in which speaker model is mapped into the low dimensional subspace. Recently, Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN) are used to extract phonetic/speaker features in training stage.

* Corresponding author.
E-mail address: duanhancong@uestc.edu.cn (H. Duan).

https://doi.org/10.1016/j.future.2019.05.057
0167-739X/

D-vector [3] uses DNN to extract features instead of generative Factor Analysis models, which achieves good accuracy especially for text-dependent tasks. However, its performance relies on the sample number of enrollment. Besides, d-vector only accumulates the output vectors, resulting in that context information is not taken into account.

Following d-vector, some end-to-end models [4–6] are put forward. In these works, metric learning is introduced into speaker verification systems and speech segments are mapped to hyperspace, where speaker similarity can be measured by a certain mathematic metric (cosine similarity in particular). Microsoft [4] and Google [5] perceptively proposed text-dependent models using large scale datasets. Baidu [6] used deep networks as speaker feature extractor. All the works were trained on private datasets and designed for device with high computation ability.

For face recognition, the end-to-end deep learning algorithms have achieved great success in face embedding, such as local metric learning methods (e.g., contrastive loss [7] and Triplet loss [8]) and global hypersphere metric learning based methods (e.g., SphereFace [9], CosineFace [10] and ArcFace [11]). Compared to face embedding, however, speaker embedding is confronted with the uncertainty of the utterance length. Fully connected layers suffer from the scarcity of computing ability with different input sizes. The convolution layer can handle data of indefinite length but the size of output feature is not same. Thence we should seriously design the network architecture.

As for realistic application scenarios, especially consumer electronics for "smart home", a good speaker recognition algorithm needs to have the following four characteristics such as: (1) low demand for enrollment speech segments; (2) low power consumption while real-time computation; (3) stronger discriminating ability than human; (4) text-independency for diverse application scenarios [12–15].

In response to the above requirements, we propose a lighten solution for speaker verification. We use a 3-stage training method to train the lighten CNN-LSTM network and achieve the best Equal Error Rate (EER) of 1.17%.

The core contributions of this paper can be summarized as:

(1) We review the AAM loss and analyze the defects that could draw forth training failure. Softmax pre-training is used to guarantee convergence of the training and avoid the defects.

(2) We propose an end-to-end neuron network with bilateral Long Short Term Memory (LSTM) cells for extracting speaker features. It is proved that the design of network can run on a commercial off-the-shelf embedded platform in real time.

(3) A three-phase training method is introduced for training and fine-tuning the embedding network for speaker verification. To the best of our knowledge, this is the first work to embed AAM loss function in training speaker recognition model.

(4) We evaluate the performance of the text-independent speaker verification system using short-term voices. The experimental results show that the proposed lighten model reaches state-of-the-art in a verification test with 200 persons. Especially, an evaluation for "smart home" is conducted.

The rest of this paper is organized as follows: Section 2 introduces speaker embedding methods. Section 3 introduces the network construction and the training method of the Bilateral LSTM model proposed in this paper. Section 4 shows the experiments settings, evaluation results and comparative analysis on the results. In Section 5 we discuss the future work. Section 6 concludes this work.

2. Related work

2.1. Speaker recognition system

Traditional algorithms for speaker recognition, such as i-vector, PLDA [16], heavy-tailed PLDA [17], and Gauss-PLDA [18], are mostly used in text-dependent scenario. Recently, DNNs, RNNs and convolution neural networks (CNNs) with an end-to-end loss $log\{p(accept/reject)\}$ have been investigated to discriminate between the same-speaker and different-speaker pairs for global keyword (e.g., "OK Google" and "Hey Cortana") speaker verification tasks [4,5], and shown to achieve the better performance compared with conventional techniques. Li et al. [6] adopted metric learning method to train the model and used cosine similarity to measure the degree of speaker difference, which achieves 1.83% Equal Error Rate(EER) on a text-independent mandarin benchmark containing 200 persons. As the development of neural networks, LSTM network has effectively avoided the problem of the vanishing gradient in the conventional RNN training process due to its own special structure design. Herve and Bredin [19] proposed a speaker embedding model that uses triplet loss with Euclidean distance to train LSTM net.

2.2. Metric learning in recognition task

The contrastive loss [7] and Triplet loss [8] are based on single task optimization. The contrastive loss function respectively utilizes the positive pairs and negative pairs. The gradients of the loss function pull together positive pairs and push apart negative pairs. The triplet loss computes loss according to the positive pair and the negative pair in triplets. But such methods suffer from slow convergence speed.

Liu et al. [20] proposed a large margin Softmax (L-Softmax) to enhance discriminative capability through introducing boundary condition into Softmax. SphereFace [9] maps identities to a hypersphere by weights normalization and adding multiplicative margin in angular space. The modified piece-wise function and Softmax loss are combined in SphereFace to guarantee the monotonicity of loss function. But the complex loss function and hyper parameter of Softmax weight make it difficult to train and implement. To overcome the defects of SphereFace, ArcFace [11] changes the multiplicative angular margin to constant angular margin, called Additive Angular Margin (AAM), similar to Triplet loss. The implementation and training of AAM are much easier than SphereFace. It is easy to implement and reaches state-of-the-art performance on LFW, CFP-FP [21] and AgeDB-30 [22]. Compared to single task losses, AAM loss takes advantage of Softmax to boost the speed of convergence.

3. Speaker embedding model

Table 1 illustrates the scheme of our speaker embedding system. Firstly, raw audio is converted to data frames by the method detailed in Section 3.4. Then, data frames are projected to utterance-level speaker embedding on a hypersphere by the networks described in Section 3.3. Considers that AAM loss can accelerate convergence but is not stable in the early phase of training, we use Softmax as loss function followed by AAM loss, as explained in Section 3.1 and Section 3.2. Then classification layer is removed and triplet loss is used to fine-tune the model further. From the start to the end, the optimization target is always maximizing the cosine similarities of embedding pairs from the same speaker, and minimizing those from different speakers.

3.1. AAM loss

AAM (Additive Angular Margin) loss is firstly proposed in training of end-to-end face embedding systems. By feature normalization and weights normalization in classification layer, it maps identity features to a hypersphere. An additive angular margin is added to improve feature discrimination. AAM loss

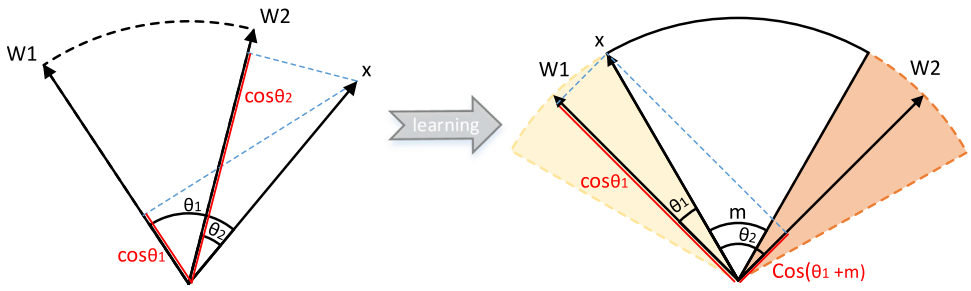


Fig. 1. The AAM loss in hypersphere. In the right part of this figure, the zones with different colors represent feature spaces of different classes. "m" is the angular margin between different classes.

comes from the most widely used Softmax loss which is shown in Eq. (1):

$$L = -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{C} e^{W_j^T x_i + b_j}} \tag{1}$$

where $x_i \in R^d$ denotes the feature of the ith sample, belonging to the y_ith class. To lower the quantity of weights in the last fully-connected layer of backbone, the output vector d of embedding is set as 128-dimensionality rather than 512-dimensionality [6] in this paper. $W_j \in R^d$ and $b_j \in R^C$ denote the jth column of weights and bias term in the classification layer, respectively. n is the batch size and C is the category number. After $b_j = 0$ is applied, $W_j^T x_i$ can be expressed as Eq. (2):

$$W_j^T x_i = \left\|W_j\right\| \|x_i\| \cos\theta_{ij} \tag{2}$$

where θ_{ij} is the angular between vector W_j and vector x_i

By fixing the value of $\|x_i\|$, $\left\|W_j\right\|$ and rescaling $\|x_i\|$ to s, prediction outputs only rely on θ_{ij}, as shown in Eq. (3):

$$W_j^T x_i = s\cos\theta_{ij} \tag{3}$$

The loss function can be expressed as

$$L = -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cos\theta_{iy_i}}}{\sum_{j=1}^{C} e^{s\cos\theta_{ij}}} \tag{4}$$

In this situation, the decision boundary of the output feature x_i is defined by

$$\theta_{iy_i} = \theta_{ij}(y_i \neq j) \tag{5}$$

By adding the angular margin on θ, AAM loss is defined as:

$$L = -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cos(\theta_{iy_i}+m)}}{e^{s\cos(\theta_{iy_i}+m)} + \sum_{j=1,j\neq y_i}^{C} e^{s\cos\theta_{ij}}} \tag{6}$$

In this situation, the decision boundary of the output feature x_i is defined as

$$\theta_{iy_i} + m = \theta_{ij}(y_i \neq j) \tag{7}$$

As shown in Fig. 1, the training under the supervision of AAM loss expands the distance between classes and compresses the size of feature space within one class. There will be an angular margin between the samples from different classes after training. By combining Softmax loss and metric learning, AAM loss has a more explicit optimization target and better geometric interpretation.

3.2. Importance of softmax pre-train

AAM (Additive Angular Margin) loss performs marvelously in the training of end-to-end face embedding system. Nevertheless, when we take advantage of it, instability of training results appears. If we adopt AAM loss to train networks directly after random initialization such as cropped Gaussian distribution initializer, there is a great possibility of divergence of training accuracy and falling into local optimum. In these cases, we observe that although the loss will drop to about one third of the initial value after a few iterations, the accuracy keeps around zero throughout.

We expose this process by showing train loss, classification accuracy and distribution of $\cos\theta_{ij}$ in training. θ_{ij} is the angular between $x_i (i = 1, 2, \ldots, N)$ and $W_j (i = 1, 2, \ldots, C)$. As depicted in Fig. 2, although the loss is reduced (Fig. 2(b)), the accuracy does not rise (Fig. 2(a)). Furthermore, it is accompanied by an abnormal angular distribution (Fig. 2(c)). In Fig. 2(c), we depict the distribution of $\cos\theta_{ij}$ following training steps. District of darker color stands for the larger proportion of values that are located in this interval. At the beginning, almost all θ_{ij} are similar and approximately equal to 90 degrees (corresponding to cosine value of 0). As the training progresses, majority of θ_{ij} are pushed to 180 degrees (corresponding to cosine value of -1).

For simplicity, we investigate a special case in which every θ_{ij} equals to each other so as to explore the reason of training failure. As can be seen from Fig. 3, loss value quickly drops to the minimum during the increase of θ_{ij} from $\pi/2$ to π. Based on this observation, we guess that the AAM loss is closely associated with the distribution of θ_{ij} in a batch.

In order to confirm our conjecture, we implement a simulation experiment. We assume that θ_{ij} in one batch obeys a normal distribution. The AAM loss values of distributions with different mean values and standard deviations are calculated. Batch size is 256, the same as that in training. Loss value is averaged among batches. We set angular margin $m = 0.4$ and radius $s = 16$. The simulation results are shown in Fig. 4.

In Fig. 4, different colors represent different standard deviations(std). We discover that the bigger standard deviation is always companied with bigger loss value when std < 0.5. Smaller standard deviation brings about smaller start point of decreasing interval. When std < 0.05, the start point is smaller than $\pi/2$. The mean value and standard deviation of the initial θ_{ij} distribution are 0 and 0.028 according to our statistics of the actual training. Hence, the training will push the mean value to somewhere nearing π and make standard deviation shrink to 0.

Compared to Softmax loss, optimization goal of AAM loss is to lessen θ_{ij} between x and W of the same class $(\theta_{ij}, i = j)$ while to enlarge that of different classes$(\theta_{ij}, i \neq j)$.

However, from a global perspective in a train batch, some categories lack positive samples when batch size is less than the number of categories. If the gradient of enlarging dominates, all W vectors could be pushed to the positions with the maximum angular (180 degrees) from x. Hence, if AAM loss is employed to train a embedding network from scratch, failure of training may occur.

The optimization goal of Softmax is to maximize the logits of correct classes and minimize the others. But Softmax does not

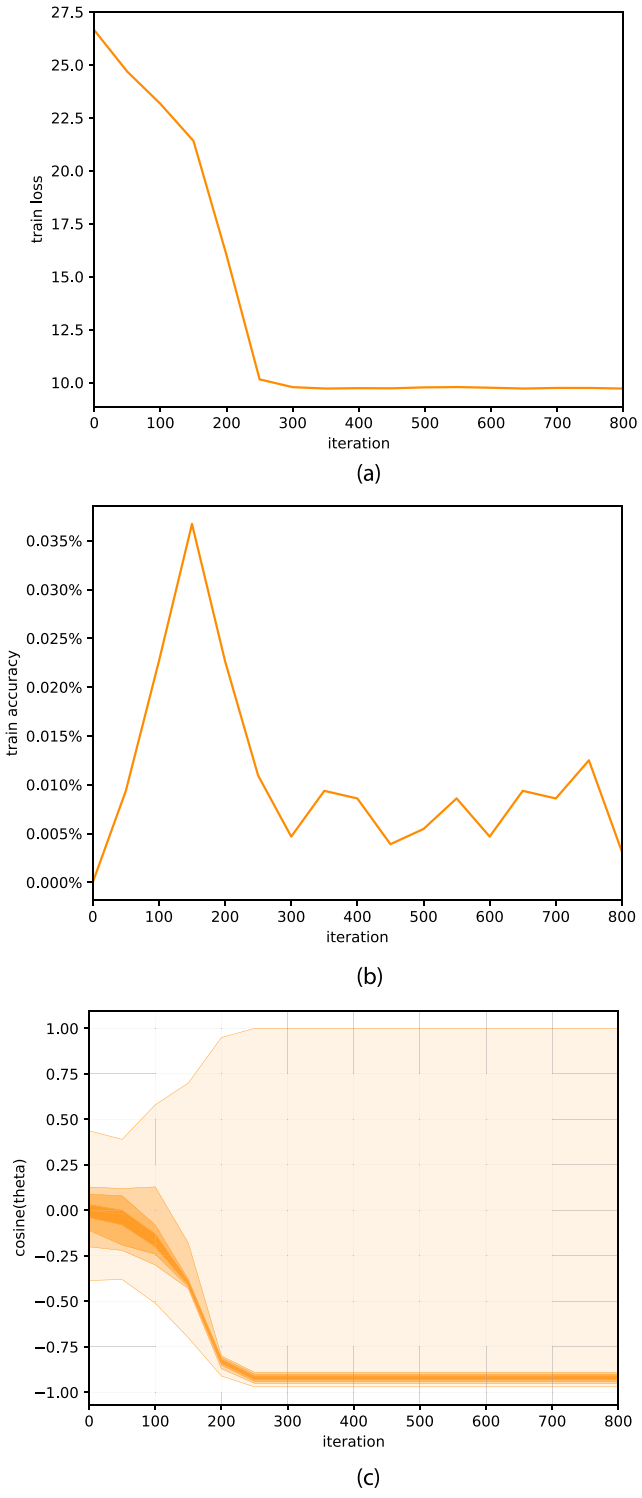


Fig. 2. Indicators in the case of divergence (a) train loss in early iterations, (b) classification accuracy in early iterations, (c) distribution of $cos\,\theta_{ij}$.

have such ability to manipulate the angular of embedding feature and weight vector in classification layer. After pre-training by several epochs, the weights in the classification layer will not change drastically for strong constrain of classification loss even if AAM loss is adopted. Therefore, Softmax pre-training plays an important role to guarantee convergence of training a network.

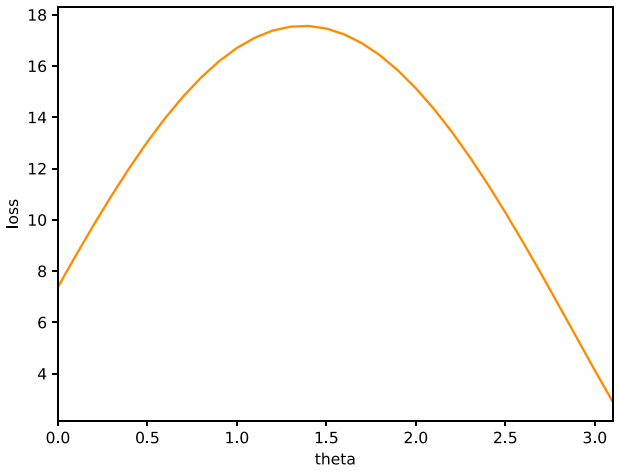


Fig. 3. Loss in a special case in which every θ_{ij} equals (m = 0.4, s = 32).

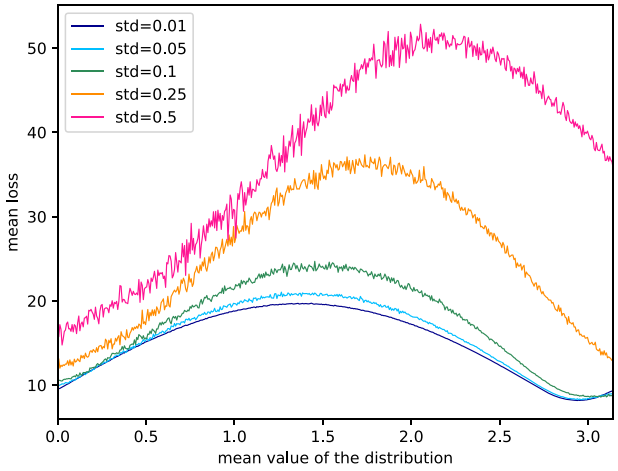


Fig. 4. AAM loss at different θ_{ij} distribution.

3.3. Bilateral LSTM network

We design our backbones network for feature extracting in utterance level. Since there is inertia in the motion of the vocal organ, it can be assumed that speech signal is stationary in short temporal (10∼30 ms). Under this assumption, after framing and windowing, speech signal can be transformed to frequency feature frame by frame through STFT (Short Time Fourier Transform) and MEL filters (described in detail in Section 3.4).

Some studies [3,23] only exploited the features from single frame or several frames. However, deviations between frames can offer huge amount of information that cannot be utilized by algorithms based on only a few frames. LSTM shows superiority in sequence processing [24]. It can effectively resolve the problem of gradient explosion and gradient dispersion when encountering long term input compared to vanilla RNN [25]. Therefore, we choose LSTM to construct the backbones. Convolution layers are widely used in image processing on width dimension and height dimension, which belong to one physical dimension. Yet Fbank coefficient data is composed by two different physical dimensions: time domain generated by successive windowing and frequency domain generated by STFT. The property that convolution layer is designed to aim at local feature makes it awkward in handling long term sequence data. So we put a convolution layer before LSTM cells to take advantage of its effectiveness in local feature extracting.

Table 1
Structure of bilateral LSTM networks.

Layer name	Structure	Time dimension @train stage	Frequency dimension	Parameters number
Input	–	256	64*3	–
Conv-16	5*5*2	128	32*16	1.2K
LSTM-64	[64]*2	[128]*2	[64]*2	290K
LSTM-64	[64]*2	[128]*2	[64]*2	66K
Time-average	–	[1]*2	[64]*2	0
Affine	[64*128]*2	[1]*2	[128]*2	16K
Group-average	–	1	128	0
Total		–	–	379K

The details of the proposed bilateral LSTM network architecture are shown in Table 1. We apply a convolution layer with $5*5$ filter size and $2*2$ stride in the front of the whole network to decrease both the time dimension size and frequency dimension size. Two Long Short-Term Memory (LSTM) blocks take the output of the head convolution layer. They have two layers of 64 hidden cells respectively. The first one processes the sequence in chronological order, but contrary for the second one. To cope with different length of input audio, we adopt an average layer for each LSTM block on time dimension following [6]. For now, we get two feature maps averaged in time dimension. For each feature map, we use 1 affine layer with batch normalization to project the averaged output of LSTM to fixed length speaker feature vectors. Finally, speaker feature vectors are averaged among groups.

3.4. Three phases training methodology

We train the model in three stages: Softmax pre-training for 2 epochs, 8 epochs training with AAM loss for global optimization and 3 epochs fine-tuning with triplet loss for local optimization. Softmax pre-training is conducted to avert divergence as mentioned in Section 3.2. Unlike [11] that Euclidean distance is used in triplet fine-tuning, cosine similarity is used in both AAM loss training and triplet loss fine-tuning. By using the same distance metric, we maintain the consistency of training goal: narrowing the distance within class and increasing distance between classes. In triplet loss fine-tuning, we use special hard mining strategy similar to that in [8]. For every training iteration, we randomly select N samples for C random identities. N and C are set to 8 and 32, respectively, in this paper. Then, we search for hard triplet pairs in which deviation between positive pair and negative pair is higher than threshold (0.1 in this paper) among every possible triplet pairs.

3.5. Implementation details

Input features

64-dimensional log Fbank coefficients, delta and double delta are computed from raw audio in a sliding window fashion using a window of width 25 ms and step 10 ms (without window function). Mean and variance normalization is performed on every frequency bin of the spectrum. During the training stage, raw audio files belong to the same speaker are randomly jointed and cropped to pieces of 2.575 s (each utterance will be extracted to exactly 256 frames of spectrum) before Fbank coefficients are extracted.

Training details

Bilateral LSTM Network is trained using TensorFlow on one NVIDIA 1080ti GPU. In all three training stages as mentioned in Section 3.4, we employ Adam optimizer with 0.001 learning rate for the first 7 epochs (Softmax pre-training, first 5 epochs of AAM loss training) and 0.0001 for the last 6 epochs (last 3 epochs of AAM loss training and triplet loss fine-tuning). Learning rate and loss function ploy is described in Table 3 more clearly. Batch size is set as 256 for the whole training process. More importantly, in the whole process of training, maximum memory usage is below 1.5G. It means that the proposed model can be trained on most mobile devices, instead of expensive GPU for servers.

Table 3
Learning rate and loss function ploy.

Epoch	1∼2	3∼7	8∼10	11∼13
Loss function	Softmaxloss	AAMloss	AAMloss	Triplet loss
Learning rate	0.001	0.001	0.0001	0.0001

4. Experiments

4.1. Dataset and evaluation

Datasets

AISHELL-1 Corpus [26] and **AISHELL-2 Corpus** [27] are mandarin speech datasets collected by Beijing Shell Shell Technology Co., Ltd.

AISHELL-1 includes 400 speakers from different accent areas in China. The recording is put in quiet indoor environment, using 3 different devices at the same time: high fidelity microphone (44.1 kHz, 16-bit,); Android-system mobile phone (16 kHz, 16-bit), iOS-system mobile phone (16 kHz, 16-bit). Audios in high fidelity were re-sampled to 16 kHz. The total duration is 178 h.

AISHELL-2 contains more than 1 million utterances of 1991 persons which are composed by 1146 females and 845 males from all around China. The raw signal of the whole corpus is recorded by devices of iOS at the sample rate of 16 KHz in silent rooms with the total duration of 1000 h.

Evaluation

Previous study [28] unearthed that the increasing duration of utterance can promote the performance of models in speaker recognition. Yet too long time for speaker enrollment and verification is extremely user unfriendly. In this benchmark, we pick 2.56 s samples as enrollment samples and subject samples.

A verification benchmark of the whole AISHELL-1 dataset is generated, namely **veri-test-1**. It includes 200k pairs of test samples composed by half positive samples and half negative ones. In order to ensure the balance of samples between each speaker and each sample, we carefully design positive and negative samples producing strategies. First, we collect unanimous numbers (250 pairs) of unrepeated positive samples for each speaker. For every speaker, we randomly choose positive verification pairs with the same probability of occurrence for every combination. Then,

Table 2
Speaker verification results.

System	Network	Loss function	EER on veri-test-1(%)	EER on veri-test-2(%)
T-DS	cropped Deep Speaker	Softmax + triplet	4.36	1.30
A-DS	cropped Deep Speaker	Softmax + AAM	4.69	2.11
T-BLSTM	Bilateral LSTM	Softmax + triplet	4.85	1.18
A-BLSTM	Bilateral LSTM	Softmax + AAM	4.51	1.39
3-stage BLSTM	Bilateral LSTM	Softmax + AAM + triplet	**4.27**	**1.17**

we build the negative test pairs. For every speaker, 250 anchor samples are picked out during samples not appearing in positive pairs. Another 250 negative samples not appearing in positive pairs are selected stochastically from the samples that belong to the other speakers.

We split AISEHLL-2 Corpus into the train set and test set with 1791 persons and 200 persons respectively. In the similar way to build veri-test-1, a speaker verification test benchmark called **veri-test-2** is constructed based on test set of AISHELL-2. 100k verification pairs of samples are generated.

Specially, we build a benchmark for "smart home" scenario, namely eight people top-1 test. The number of enrollment users is set as 8 in "smart home", which can cover most situations. So eight people top 1 accuracy is an appropriate evaluation metric.

To establish this benchmark, we randomly generate unrepeated 1000 groups of eight people combination from the 200 people in train set of AISHELL-2. For each group, we select one sample as enrollment sample for each person. One speaker is treated as ground truth and another unrepeated speech sample is selected as the tested sample. In this test, we respectively compute cosine distance between the tested sample and enrollment samples according to the features extracted by the trained CNN-LSTM models. The speaker label of the shortest distance is treated as prediction.

Metric for verification

Equal Error Rate (EER) is adopted in this paper to evaluate the capability of models in the experiments of speaker verification.

In verification task, accuracy just unilaterally reveals discriminative capability among detected positive samples. However, there are two types of error existing in verification tasks: false negative (FN) yielded when voice segments from the same person are predicted as pronounced by different persons, and false positive (FP) is the case that the samples not from the same one are classified as pronounced by the same person. Amplifying threshold will lead to declining false positive rate (FPR) but uprising false negative rate (FNR). The situation turns contrast while lowering the threshold.

The system with too high FNR or FPR makes user experience of the products terrible in industry. EER is a valid evaluation metric since it efficaciously takes into account the target missing and false alarm at the same time.

4.2. Baseline speaker verification model

The baseline speaker verification model is built on [6]. We choose the residual CNN network because it performs better than GRU network in the experiments of mandarin text-independent speaker verification. In order to prevent overfit in training on AISHELL-2 train set and to make it fair in control experiments, we suppress the Deep Speaker residual CNN via reducing the channels by 8 times. The suppressed Deep Speaker model has 364 K parameters close to that of the proposed network.

To fairly compare the effects of different network structure and train methodologies, we feed the same features as that of the proposed algorithm into suppressed Deep Speaker. The same learning rate and loss function strategies are used in control experiments which are detailed described in Section 4.3.

4.3. Verification test

In Table 2, we exhibit comparison of speaker verification between the modified Deep Speaker CNN network and proposed B-LSTM network. Further analysis is as follows.

In this task, the proposed model performs better than the baseline only pre-trained by Softmax and fine-tuned by triplet loss (T-DS). We reach state-of-the-art on veri-test-2 using the model with low requirement for computing resources. In all experiments carried out in this section, hyper parameter m of AAM loss is set as 0.4 according to ablation experiments in Section 4.6 and threshold of triplet loss is set as 0.1 by [6].

Table 4
Learning rate and loss function ploy in control experiments.

System	Epoch	1∼2	3∼7	8∼13
A-BLSTM & A-DS	Loss function	Softmax	AAM	AAM
	Learning rate	0.001	0.001	0.0001
T-BLSTM & T-DS	Loss function	Softmax	Triplet	Triplet
	Learning rate	0.001	0.001	0.0001

Table 5
Results for inference time.

network	inference time (s)
B-LSTM	**0.605**
Deep Speaker CNN	3.517

We also conduct control experiments of network structures and training methodologies. The learning rate and loss function ploy are represented in Table 4. By comparing the models with the same network structure but different training methods (T-DS and A-DS, T-BSLTM and A-BSLTM), it is shown that triplet loss performs even better than AAM loss by the special hard mining strategies. By comparing the models with the same train method but different network structure (T-DS and T-BSLTM, A-DS and A-BSLTM), it is demonstrated that our proposed bilateral LSTM network has more powerful ability to extract the speaker feature from sequence voice data.

In the experiments on the larger dataset AISHELL-1 (400 persons), 3-stage BLSTM still obtains 4.27% EER, which proves the strong robustness of the model.

4.4. Inference time evaluation on embedding device

To demonstrate that the proposed network is a real-time architecture, we evaluate the TensorFlow inference time of networks on Raspberry Pi. Raspberry Pi is an embedded device. It is equipped with a quad-core processor of Armv7 architecture working at 1.9 GHz and 4 GB memory. The performance of the hardware just reaches the average level of AMRv7 platforms which is widely used by smart household appliances for its high performance, versatility and cheap price. So we choose Raspberry Pi as the evaluation platform.

We compare the inference time of our proposed network structure with that of the original Deep Speaker CNN network. Raw wave data of two pieces of utterance are feed into the networks. Inference time is composed by log Fbank feature computing, feature extraction by neuron network and computing cosine similarity between two samples.

As shown in Table 5, there is significant difference in inference time between two architectures. The proposed bilateral LSTM network is almost 6 times faster than the Deep Speaker CNN. Furthermore, what of great significance is that time consuming of processing the original speech signal (0.605 s) is much shorter than the length of the elapsed (2.575 s). It exhibits that the speaker verification system could run continuously on embedded devices.

4.5. Eight people top1 test

In Table 6, we present the top-1 accuracy of eight people test. We reach 99.62% on the test set of AISHELL-2 and 97.64% on more challengeable AISHELL-1. The precision is good enough for practical application in the household smart appliances. This

Table 6
Eight people top1 test.

System	on AISHELL-1	on AISHELL-2 test set
3-stage BLSTM	**97.64%**	**99.62%**

Table 7
EER of veri-test-2 with different values of m.

m	EER
0.2	1.64%
0.3	1.54%
0.4	**1.39%**
0.5	1.44%
0.6	1.50%
0.7	1.51%

experiment demonstrates that our CNN-LSTM model is suitable for "smart home" application from the perspective of recognition accuracy performance.

4.6. Ablation experiments of hyper parameter m in additive angular margin loss

In this part, we exhibit the deviation of network performance affected by the hyper parameters configuration and seek out the best configuration.

In the study undertaken by Arcface [11], it is demonstrated that the setting of $m = 0.5$ outperforms others in face tasks. However, the same configuration might not work in other tasks. We design ablation experiments to search for the best suitable value of m for speaker recognition task.

In the experiments of this section, we use bilateral LSTM network and AAM loss to train models on AISHELL-2 train set. Networks are trained 8 epochs at different m after 2 epochs pretrain by Softmax (m = 0). The initial learning rate is set as 0.001 and decayed to 0.0001 at the last three epochs. Batch size is 256 for every network.

The result is presented in Table 7. We can observe a relationship between the test metric and m:the larger the m, the better the performance result; but when m exceeds a threshold, the situation reverses. What is different is that when $m = 0.4$ we get optimal networks.

5. Discussion

There would usually be more than one million identities (such as MS-Celeb-1M [29], WebFace [30] and WilderFace [31]) in face datasets. However, open source datasets of speech suffer from shortage of identities, especially mandarin corpus. It is known that deep learning is a data-driven algorithm. Therefore, the bigger dataset for training could boost performance.

A VAD (Voice Activity Detection) system can identity useful talking segments from background sound. It can further promote the performance of non-cooperating speaker recognition and decreasing unnecessary cost for feature extraction. That is one of our future work to integrate VAD to our algorithm.

6. Conclusion

In this paper, we propose an end-to-end lighten frame for mandarin text-independent speaker verification. It is equipped a group of double-layer bilateral LSTM cells to make full use of the deviation in the voice signal. We prove that the proposed model is more suitable for speaker feature extraction from raw audio sequences. By carefully designing the network, the proposed model can run in real time on embedded devices. Besides, we introduce AAM loss into the training of speaker embedding model and propose a training methodology by fusing Softmax loss, AAM loss and triplet loss which share unified optimization objective.

The experiments results show that the EER value obtained by the 3-stage trained bilateral LSTM net is lower than that of the baseline, the suppressed Deep Speaker CNN model, which indicates that our method has a better verification performance. What calls for special intention is that our model has the lower time cost on computing on Raspberry Pi than audio duration, which signifies it is a real-time model for micro embedded devices. We explore various hyper parameter m of AAM loss and certify the best setting of m is 0.4 in speaker recognition tasks. Meanwhile, we compare different combinations of training methods and network structures. The experiments results illustrate that proposed network plus proposed training method are the best. The results of eight people top1 test demonstrate that our scheme achieves high precision and is suitable for application in "smart home".

Acknowledgments

This work is funded by project "Identification system based on audio and video big data and its application on the people's livelihood" whose fund source is Sichuan Provincial Economic and Information Commission, China. The contract number of the fund is 2017JXW001.

Conflict of interest

None.

Declaration of competing interest

The authors declared that they had no conflicts of interest with respect to their authorship or the publication of this article.

References

[1] W.M. Campbell, D.E. Sturim, D.A. Reynolds, A. Solomonoff, SVM Based speaker verification using a GMM supervector kernel and NAP variability compensation, in: Int. Conf. Acoust. Speech, Signal Process., 2006, pp. 97–100, http://dx.doi.org/10.1109/ICASSP.2006.1659966.
[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, IEEE Trans. Audio. Speech Lang. Process. 19 (2011) 788–798, http://dx.doi.org/10.1109/TASL.2010.2064307.
[3] E. Variani, X. Lei, E. Mcdermott, I.L. Moreno, J. Gonzalezdominguez, Deep neural networks for small footprint text-dependent speaker verification, in: Int. Conf. Acoust. Speech, Signal Process., 2014, pp. 4052–4056, http://dx.doi.org/10.1109/ICASSP.2014.6854363.
[4] S. Zhang, Z. Chen, Y. Zhao, J. Li, Y. Gong, End-to-end attention based text-dependent speaker verification, Spok. Lang. Technol. Work (2016) 171–178, http://dx.doi.org/10.1109/SLT.2016.7846261.
[5] G. Heigold, I.L. Moreno, S. Bengio, N. Shazeer, End-to-end text-dependent speaker verification, in: Int. Conf. Acoust. Speech Signal Process., 2016, pp. 5115–5119, http://dx.doi.org/10.1109/ICASSP.2016.7472652.
[6] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, Z. Zhu, Deep Speaker: an End-to-End Neural Speaker Embedding System. ArXiv Comput. Lang. (2017).
[7] R. Hadsell, S. Chopra, Y. Lecun, Dimensionality reduction by learning an invariant mapping, in: Comput. Vis. Pattern Recognit., 2006, pp. 1735–1742, http://dx.doi.org/10.1109/CVPR.2006.100.
[8] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, Comput. Vis. Pattern Recognit. (2015) 815–823, http://dx.doi.org/10.1109/CVPR.2015.7298682.
[9] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, Sphereface: Deep hypersphere embedding for face recognition, Comput. Vis. Pattern Recognit. (2017) 6738–6746, http://dx.doi.org/10.1109/CVPR.2017.713.
[10] F. Wang, J. Cheng, W. Liu, H. Liu, Additive margin softmax for face verification, IEEE Signal Process. Lett. 25 (2018) 926–930, http://dx.doi.org/10.1109/LSP.2018.2822810.
[11] J. Deng, J. Guo, S. Zafeiriou, ArcFace: Additive Angular Margin Loss for Deep Face Recognition. ArXiv Comput. Vis. Pattern Recognit. (2018).
[12] X. Liu, R. Zhu, B. Jalaian, Y. Sun, Dynamic spectrum access algorithm based on game theory in cognitive radio networks, Mob. Netw. Appl. 20 (2015) 817–827, http://dx.doi.org/10.1007/s11036-015-0623-2.

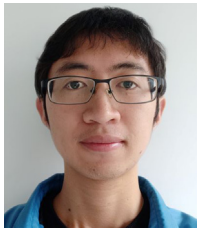
[13] R. Zhu, X. Zhang, X. Liu, W. Shu, T. Mao, B. Jalaian, ERDT: Energy-efficient reliable decision transmission for intelligent cooperative spectrum sensing in industrial iot, IEEE Access 3 (2015) 2366–2378, http://dx.doi.org/10.1109/ACCESS.2015.2501644.
[14] L. Chen, C.D. Nugent, H. Wang, A knowledge-driven approach to activity recognition in smart homes, IEEE Trans. Knowl. Data Eng. 24 (2012) 961–974, http://dx.doi.org/10.1109/TKDE.2011.51.
[15] M.A.A. Pedrasa, T. Spooner, I. Macgill, Coordinated scheduling of residential distributed energy resources to optimize smart home energy services, IEEE Trans. Smart Grid 1 (2010) 134–143, http://dx.doi.org/10.1109/TSG.2010.2053053.
[16] S. Cumani, O. Plchot, P. Laface, Probabilistic linear discriminant analysis of i-vector posterior distributions, in: Int. Conf. Acoust. Speech, Signal Process., 2013, pp. 7644–7648, http://dx.doi.org/10.1109/ICASSP.2013.6639150.
[17] P. Matejka, O. Glembek, F. Castaldo, M.J. Alam, O. Plchot, P. Kenny, L. Burget, J. Cernocky, Full-covariance UBM and heavy-tailed PLDA in i-vector speaker verification, in: Int. Conf. Acoust. Speech, Signal Process, 2011, pp. 4828–4831, http://dx.doi.org/10.1109/ICASSP.2011.5947436.
[18] S.H. Ghalehjegh, R.C. Rose, Deep bottleneck features for i-vector based text-independent speaker verification, in: Ieee Autom. Speech Recognit. Underst. Work, 2015, pp. 555–560, http://dx.doi.org/10.1109/ASRU.2015.7404844.
[19] H. Bredin, Tristounet: Triplet loss for speaker turn embedding, in: Int. Conf. Acoust. Speech, Signal Process, 2017, pp. 5430–5434, http://dx.doi.org/10.1109/ICASSP.2017.7953194.
[20] W. Liu, Y. Wen, Z. Yu, M. Yang, Large-margin softmax loss for convolutional neural networks, Int. Conf. Mach. Learn. (2016) 507–516.
[21] S. Sengupta, J. Chen, C.D. Castillo, V.M. Patel, R. Chellappa, D.W. Jacobs, Frontal to profile face verification in the wild, in: Work. Appl. Comput. Vis., 2016, pp. 1–9, http://dx.doi.org/10.1109/WACV.2016.7477558.
[22] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, S. Zafeiriou, AgeDB: The first manually collected, in-the-wild age database, in: Comput. Vis. Pattern Recognit., 2017, pp. 1997–2005, http://dx.doi.org/10.1109/CVPRW.2017.250.
[23] D. Snyder, D. Garciaromero, D. Povey, S. Khudanpur, Deep neural network embeddings for text-independent speaker verification, in: Conf. Int. Speech Commun. Assoc., 2017, pp. 999–1003, http://dx.doi.org/10.21437/Interspeech.2017-620.
[24] B. Yuan, J. Panneerselvam, L. Liu, N. Antonopoulos, Y. Lu, An inductive content-augmented network embedding model for edge artificial intelligence, IEEE Trans. Ind. Inform. (2019) 1, http://dx.doi.org/10.1109/TII.2019.2902877.
[25] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, NY Univ., 2008, http://dx.doi.org/10.1007/978-3-642-24797-2.
[26] H. Bu, J. Du, X. Na, B. Wu, H. Zheng, AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline, ArXiv Comput. Lang. (2017). http://dx.doi.org/10.1109/icsda.2017.8384449.
[27] J. Du, X. Na, X. Liu, H. Bu, AISHELL-2: Transforming Mandarin ASR Research Into Industrial Scale. ArXiv Comput. Lang. (2018).
[28] C. Zhang, K. Koishida, J.H.L. Hansen, Text-independent speaker verification based on triplet convolutional neural network embeddings, IEEE Trans. Audio. Speech. Lang. Process. 26 (2018) 1633–1644, http://dx.doi.org/10.1109/TASLP.2018.2831456.
[29] Y. Guo, L. Zhang, Y. Hu, X. He, J. Gao, MS-Celeb-1m: A dataset and benchmark for large-scale face recognition, in: Eur. Conf. Comput. Vis., 2016, pp. 87–102, http://dx.doi.org/10.1007/978-3-319-46487-9_6.
[30] D. Yi, Z. Lei, S. Liao, S.Z. Li, Learning Face Representation from Scratch, ArXiv Comput. Vis. Pattern Recognit. (2014).
[31] S. Yang, P. Luo, C.C. Loy, X. Tang, WIDER FACE: A face detection benchmark, Comput. Vis. Pattern Recognit. (2016) 5525–5533, http://dx.doi.org/10.1109/CVPR.2016.596.

Zitian Zhao received the B.S. degree in the measurement and control technology and instrument (2015) from UESTC. Currently he is studying toward his Ph.D. degree in the School of Computer Science and Engineering of UESTC and engaged in research of deep learning, voice process and computer vision.

Hancong Duan received the B.S. degree in computer science from Southwest Jiaotong University in 1995, the M.E. degree in computer architecture in 2005, and the Ph.D. degree in computer system architecture from UESTC in 2007. Currently he is a professor of computer science at UESTC. His current research interests include Deep Learning, Large-Scale P2P Content Delivery Network, Distributed Storage and Cloud Computing.

Geyong Min received the Ph.D. degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. He is a Professor of High Performance Computing and Networking in the Department of Mathematics and Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, UK. His research interests include Future Internet, Computer Networks, Wireless Communications, Multimedia Systems, Information Security, High Performance Computing, Ubiquitous Computing, Modeling and Performance Engineering.

Yue Wu is a Professor in UESTC. He had served as Dean of the School of Computer Science and Engineering at UESTC from 2001 to 2006. He was the Dean of the School of Information and Software Engineering in UESTC from 2002 to 2006. He worked as the Dean of School of Software in Chengdu College of UESTC from 2002 to 2004. He is a committee member of several journals including the Journal of UESTC, the Journal of Computer Applications, and Software World. He has presided over a number of international conferences such as CIDE 2005 and ISNN 2006. He has published more than 70 research papers and authored 3 books. His current research focuses on Deep Learning, Grid Computing, Database Systems and Data Mining.



Zilei Huang received his B.S. degree from School of Software in Central South University, China, in 2016. Currently, He is a gradu ate student in the School of Computer Science and Engineering of UESTC. His current research direction is computer vision and artificial intelligence.

Xian Zhuang received her B.S. degree from Chongqing University of Posts and Telecommunications in 2018. From 2018, she studies a master's degree in the School of Computer Science and Engineering of UESTC. Her research interests are Deep Learning and Voice Print Recognition.

Hao Xi received the B.S. degree from school of Communication and Information in Chongqing University of Posts and Telecommunications in 2018. From 2018, he studies for a M.E. degree in School of Computer Science and Engineering in UESTC. His research interest is Deep Learning.

Meirong Fu received the B.S. degree from School of Information Engineering in Chang'an University, China, in 2018. Currently, she has is studying for her M.E. degree of computer science and engineering in UESTC. Her research interests are Deep Learning and Computer Vision.