THUEE system description for NIST 2019 SRE CTS Challenge

Abstract

This paper describes the systems submitted by the department of electronic engineering, institute of microelectronics of Tsinghua university and TsingMicro Co. Ltd. (THUEE) to the NIST 2019 speaker recognition evaluation CTS challenge. Six subsystems, including etdnn/ams, ftdnn/as, eftdnn/ams, resnet, multitask and c-vector are developed in this evaluation.

Introduction

This paper describes the systems developed by the department of electronic engineering, institute of microelectronics of Tsinghua university and TsingMicro Co. Ltd. (THUEE) for the NIST 2019 speaker recognition evaluation (SRE) CTS challenge BIBREF0. Six subsystems, including etdnn/ams, ftdnn/as, eftdnn/ams, resnet, multitask and c-vector are developed in this evaluation. All the subsystems consists of a deep neural network followed by dimension deduction, score normalization and calibration. For each system, we begin with a summary of the data usage, followed by a description of the system setup along with their hyperparameters. Finally, we report experimental results obtained by each subsystem and fusion system on the SRE18 development and SRE18 evaluation datasets.

Data Usage

For the sake of clarity, the datasets notations are defined as in table 1 and the training data for the six subsystems are list in table 2, 3, and 4.

Systems ::: Etdnn/ams

Etdnn/ams system is an extended version of tdnn with the additive margin softmax loss BIBREF1. Etdnn is used in speaker verification in BIBREF2. Compared with the traditional tdnn in BIBREF3, it has wider context and interleaving dense layers between each two tdnn layers. The architecture of our etdnn network is shown in table TABREF6. It is the same as the etdnn architecture in BIBREF2, except that the context of layer 5 of our system is t-3:t+3 instead of t-3, t, t+3. The x-vector is extracted from layer 12 prior to the ReLU non-linearity. For the loss, we use additive margin softmax with $m=0.15$ instead of traditional softmax loss or angular softmax loss. Additive margin softmax is proposed in BIBREF4 and then used in speaker verification in our paper BIBREF1. It is easier to train and generally performs better than angular softmax.

Systems ::: ftdnn/as

Factorized TDNN (ftdnn) architecture is listed in table TABREF8. It is the same to BIBREF2 except that we use 1024 nodes instead of 512 nodes in layer 12 and 13. The x-vector is extracted from layer 12 prior to the ReLU non-linearity. So our x-vector is 1024 dimensional. More details about the architecture can be found in BIBREF2.

Systems ::: eftdnn/ams

Extended ftdnn (eftdnn) is a combination of etdnn and ftdnn. Its architecture is listed in table TABREF10. The x-vector is extracted from layer 22 prior to the ReLU non-linearity.

Systems ::: resnet

ResNet architecture is also based on tdnn x-vector BIBREF3. The five frame level tdnn layers in BIBREF3 are replaced by ResNet34 (512 nodes) + DNN(512 nodes) + DNN(1000 nodes). Further details about ResNet34 can be found in BIBREF5. In our realization, acoustic features are regarded as a single channel picture and feed into the ResNet34. If the dimensions in the residual network don't match, zeros are added. The statistic pooling and segment level network stay the same. For the loss function, we use angular softmax with $m=4$. The x-vector is extracted from first DNN layer in segment level prior to the ReLU non-linearity. It has 512 dimensions.

Systems ::: multitask

Multitask architecture is proposed in BIBREF6. It is a hybrid multi-task learning based on x-vector network and ASR network. It aims to introduce phonetic information by another neural acoustic model in ASR to help speaker recognition task. The architecture is shown in Fig. FIGREF13.

The frame-level part of the x-vector network is a 10-layer TDNN. The input of each layer is the sliced output of the previous layer. The slicing parameter is: {t - 2; t - 1; t; t + 1; t + 2}, { t }, { t - 2; t; t + 2 }, {t}, { t - 3; t; t + 3 }, {t }, {t - 4; t; t + 4 }, { t }, { t } , { t }. It has 512 nodes in layer 1 to 9, and the 10-th layer has 1500 nodes. The segment-level part of x-vector network is a 2-layer fully-connected network with 512 nodes per layer. The output is predicted by softmax and the size is the same as the number of speakers.

The ASR network has no statistics pooling component. The frame-level part of the x-vector network is a 7-layer TDNN. The input of each layer is the sliced output of the previous layer. The slicing parameter is: {t - 2; t - 1; t; t + 1; t + 2}, {t - 2; t; t + 2}, {t - 3; t; t + 3}, {t}, {t}, {t}, {t}. It has 512 nodes in layer 1 to 7.

Only the first TDNN layer of the x-vector network is shared with the ASR network. The phonetic classification is done at the frame level, while the speaker labels are classified at the segment level.

To train the multitask network, we need training data with speaker and ASR transcribed. But only

Phonetic dataset fits this condition and the data amount is too small to train a neural network. So, we need to train a GMM-HMM speech recognition system to do phonetic alignment for other datasets. The GMM-HMM is trained using Phonetic dataset with features of 20-dimensional MFCCs with delta and delta-delta, totally 60-dimensional. The total number of senones is 3800. After training, forced alignment is applied to the SRE, Switchboard, and Voxceleb datasets using a fMLLR-SAT system.

Systems ::: c-vector

C-vector architecture is also one of our proposed systems in paper BIBREF7. As shown in Fig. FIGREF15, it is an extension of multitask architecture. It combines multitask architecture with an extra ASR Acoustic Model. The output of ASR Acoustic Model is concatenated with x-vector's frame-level output as the input of statistics pooling. Refer to BIBREF7 for more details.

The multitask part of c-vector has the same architecture as in the above section SECREF12 ASR Acoustic Model of c-vector is a 5-layer TDNN network. The slicing parameter is { t - 2; t - 1; t; t + 1; t + 2 }, { t - 1; t; t + 1 }, { t - 1; t; t + 1 }, { t - 3; t; t + 3}, { t - 6; t - 3; t}. The 5-th layer is the BN layer containing 128 nodes and other layers have 650 nodes.

A GMM-HMM is also trained as like in section SECREF12 to do phonetic alignment for training datasets.

feature and back-end

23-dimensional MFCC (20-3700Hz) is extracted as feature for etdnn/ams, ftdnn/as, eftdnn/ams, multitask and c-vector subsystems. 23-dimensional Fbank is used as feature for ResNet 16kHz subsystems. A simple energy-based VAD is used based on the C0 component of the MFCC feature BIBREF8.

For each neural network, its training data are augmented using the public accessible MUSAN and RIRS_NOISES as the noise source. Two-fold data augmentation is applied for etdnn/ams, ftdnn/as, resnet, multitask and cvector subsystems. For eftdnn/ams subsystem, five-fold data augmentation is applied.

After the embeddings are extracted, they are then transformed to 150 dimension using LDA. Then, embeddings are projected into unit sphere. At last, adapted PLDA with no dimension reduction is applied.

The execution time is test on Intel Xeon E5-2680 v4. Extracting x-vector cost about 0.087RT. Single trial cost around 0.09RT. The memory cost about 1G for a x-vector extraction and a single trial. In the inference, only CPU is used.

The speed test was performed on Intel Xeon E5-2680 v4 for etdnn_ams, multitask, c-vector and ResNet system. Test on Intel Xeon Platinum 8168 for ftdnn and eftdnn system. Extracting embedding cost about 0.103RT for etdnn_ams, 0.089RT for multitask, 0.092RT for c-vector, 0.132RT for eftdnn, 0.0639RT for ftdnn, and 0.112RT for ResNet. Single trial cost around 1.2ms for etdnn_ams, 0.9ms for multitask, 0.9ms for c-vector, 0.059s for eftdnn, 0.0288s for ftdnn, 1.0ms for ResNet. The memory cost about 1G for an embedding extraction and a single trial. In the inference, we just use CPU.

Fusion

Our primary system is the linear fusion of all the above six subsystems by BOSARIS Toolkit on SRE19 dev and eval BIBREF9. Before the fusion, each score is calibrated by PAV method (pav_calibrate_scores) on our development database. It is evaluated by the primary metric provided by NIST SRE 2019.