# TIME DELAY DEEP NEURAL NETWORK-BASED UNIVERSAL BACKGROUND MODELS FOR SPEAKER RECOGNITION

*David Snyder, Daniel Garcia-Romero, Daniel Povey*

Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA
david.ryan.snyder@gmail.com, dgromero@jhu.edu, dpovey@gmail.com

## ABSTRACT

Recently, deep neural networks (DNN) have been incorporated into i-vector-based speaker recognition systems, where they have achieved state-of-the-art performance. In these systems, a DNN replaces the Gaussian mixture model (GMM) as the universal background model (UBM). In this study the DNN is a recently developed time delay deep neural network (TDNN) that has achieved promising results in LVCSR tasks. We believe that the TDNN-based system achieves the best reported results on SRE10 and it obtains a 50% relative improvement over our GMM baseline in terms of equal error rate (EER). For some applications the computational cost of a DNN is prohibitive. Therefore we also investigate a lightweight alternative in which a supervised GMM is derived from the TDNN posteriors. This method maintains the speed of the traditional unsupervised-GMM but achieves a 20% relative improvement in EER.

***Index Terms***— speaker recognition, deep neural networks, time delay neural networks

## 1. INTRODUCTION

Modern speaker recogntion system are based on i-vectors [1]. In this paradigm, a universal background model (UBM) is used to collect sufficient statistics for i-vector extraction and a probabilistic linear discriminant analysis (PLDA) backend computes a similarity score between i-vectors [2, 3, 4, 5, 6, 7].

Recent speaker recognition systems have improved performance by replacing the GMM-based UBM with a DNN [8, 9]. Usually a DNN trained as the acoustic model in an automatic speech recognition (ASR) system is repurposed for speaker recognition, where it assumes the role of the UBM. In an ASR system the output layer of a DNN provides soft alignments for phonetic content, often tied triphone states (or senones). As is standard in GMM-based speaker recognition systems, these DNN posteriors are aggregated along with speaker recognition features to create sufficient statistics for i-vector extraction [1]. The advantage of the DNN over the GMM may be due to its ability to directly model phonetic content, rather than an arbitrary acoustic space [8, 9]. In [9] it was found that improvements to DNNs in terms of ASR word error rate (WER) may translate into improvements in speaker recognition performance. Recently, recurrent neural networks (RNN) and TDNNs [10] have outperformed traditional DNNs for a variety of LVSCR tasks [11, 12, 13]. In particular, the multisplice TDNN [13] had an 11% WER on Switchboard, better than RNN systems on the same task. Our DNN is based off of [13]. TODO: Are TDNNs ever used in DNN-based SID? Does anyone care?

The DNN-based speaker recognition methods achieve excellent results, but the performance comes at the cost of increased computational complexity. During i-vector extraction, the role of the UBM is to produce frame-level posteriors. For a DNN, the computation is nontrivial. In a resource limited application that nonetheless requires realtime performance, the DNN-based system may not be practical. Ideally, a supervised GMM could be be created with the speed of the traditional GMM-based UBM but with heightened phonetic awareness. In [14] a GMM-based ASR acoustic model replaced the usual GMM-UBM to create a phonetically aware GMM, but the improvements were only consistent during model combination [14].

Usually DNN-based speaker recognition systems employ a supervised-GMM derived from the DNN posteriors and speaker recognition features (often not the same as the DNN features) [15, 8, 9]. However, this GMM is not typically used as a UBM; it has a minor role during i-vector extractor training. Promoting this supervised GMM to the role of the UBM was explored in [8], but it did not improve on their baseline. It was speculated that this is due to the GMM's limited ability to model phonetic information. However, that supervised GMM was diagonal, which possibly reduced its modelling compacity. In this paper we reexamine the value of this supervised-GMM as a lightweight alternative to the

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1232825. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

DNN-based speaker recognition system, and find that it consistently outperforms the baseline.

## 2. EXPERIMENTAL SETUP

### 2.1. Datasets

We evaluate our systems on the condition 5 extended task of SRE10 [16]. The test consists of conversational telephone speech in enrollment and test utterances. In total there are 416,119 trials, over 98% of which are nontarget comparisons.

The UBM and i-vector extractor training data consists male and female utterances from SWB and NIST SREs prior to 2010. The SWB data contains 1,962 speakers and 20,905 utterances of SWB Cellular and SWB 2 Phases II and III. The SRE dataset consists of 3,805 speakers and 36,614 utterances. The PLDA backend is trained only on the SRE data. We train the TDNN on roughly 1800 hours of Fisher English.

### 2.2. DNN Recipe

The system is a 6-layer deep neural network based on the multisplice time delay DNN described in [13]. In the multisplice system, a narrow temporal context is provided to the first layer and increasingly large temporal contexts are available to the subsequent hidden layers. The result is that higher levels of the network are able to learn greater temporal relationships. TODO: Why was this DNN picked?

The features are 40 MFCCs (without cepstral truncation) with 25ms frames. Cepstral mean subtraction is performed over a window of 6 seconds. Five frames are spliced together together at the input layer and an increasingly wide context is provided to subsequent hidden layers. The hidden layers use the $p$-norm (where $p = 2$) activation function [17]. To facilitate faster computation, the size of the network is reduced from what is described in [13]. The hidden layers have an input dimension of 350 and an output dimension 3500. The softmax output layer computes posteriors for 5297 triphone states. No fMLLR or i-vectors are used for speaker adaptation.

### 2.3. GMM-UBM Baseline

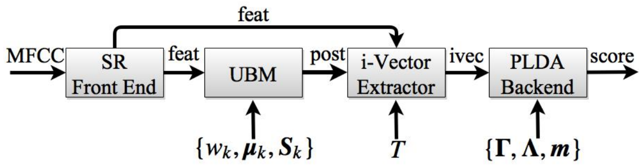
feat
MFCC
SR Front End
feat
UBM
post
i-Vector Extractor
ivec
PLDA Backend
score
$\{w_k, \boldsymbol{\mu}_k, \boldsymbol{S}_k\}$
$\boldsymbol{T}$
$\{\boldsymbol{\Gamma}, \boldsymbol{\Lambda}, \boldsymbol{m}\}$

**Fig. 1**: GMM-based speaker recognition schema.

The UBM in our baseline system illustrated in Figure 1 is a full-covariance GMM with several thousand mixture components. We compare systems with 2048, 4096, and 5297 components. The front-end consists of 20 MFCCs which are mean normalized over a 3 second window, plus $\Delta$ and $\Delta\Delta$ to create a 60 dimension frame-level feature vector. The nonspeech frames are then eliminated using energy-based Voice Activity Detection (VAD). The GMM is trained on SWB and the SRE datasets. The full-covariance GMM is initially a diagonal covariance GMM which is trained for 4 iterations of EM followed by an additional 4 iterations using a full-covariance matrix. A 600 dimensional i-vector extractor is also trained on SWB + SRE for 5 iterations of EM. In Figure 1 the PLDA backend also includes the i-vector mean $\boldsymbol{m}$ subtraction and length normalization. The between-class and within-class covariance matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ of the PLDA backened along with the vector $\boldsymbol{m}$ are trained on the SRE dataset described in Section 2.1.

### 2.4. Supervised GMM-UBM

To create the supervised-GMM, DNN posteriors from the system in Section 2.2 and speaker recognition features described in Section 2.3 are computed on the training data (SWB + SRE) using the equations in 1. The DNN parameters are collectively labeled $\boldsymbol{\Theta}$ and $Pr(k \mid \boldsymbol{y}_i, \boldsymbol{\Theta})$ is the probability of senone $k$ given the DNN features $\boldsymbol{y}_i$. The corresponding speaker recognition features are denoted $\boldsymbol{x}_i$. In contrast to [8], our supervised-GMM is full-covariance.

$$\begin{aligned}
z_k^{(i)} &= Pr(k \mid \boldsymbol{y}_i, \boldsymbol{\Theta}), \\
w_k &= \sum_{i=1}^{N} z_k^{(i)}, \\
\boldsymbol{\mu}_k &= \frac{1}{w_k} \sum_{i=1}^{N} z_k^{(i)} \boldsymbol{x}_i, \\
\boldsymbol{S}_k &= \frac{1}{w_k} \sum_{i=1}^{N} z_k^{(i)} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^{\top}.
\end{aligned} \tag{1}$$

Since the DNN output layer has 5297 senones, the supervised GMM also has 5297 components. The supervised and unsupervised GMMs differ only in the UBM training proceedure. Training of the $\boldsymbol{T}$ and PLDA parameters $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}$ are unchanged from Section 2.3.

### 2.5. TDNN-UBM

The TDNN system uses the supervised-GMM described in the preceding section to initialize the i-vector extractor $\boldsymbol{T}$ matrix. However, updating the $\boldsymbol{T}$ matrix and extracting i-vectors uses DNN posteriors and speaker recognition features. As in Sections 2.3 and 2.4, the i-vectors are 600 dimensional and the PLDA backend is the same.

In order to maintain the correct temporal context for the DNN, VAD is not used at the frontend. Instead, the voice

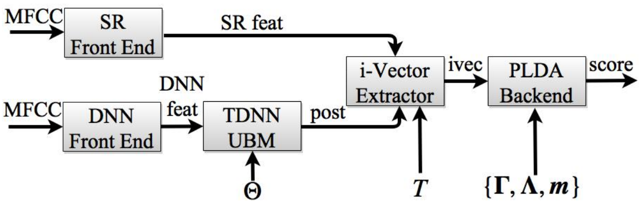

**Fig. 2**: TDNN-based speaker recognition schema.

activity detection from the speaker recognition features are reused to filter the posteriors of the network.

### 2.6. System Design

Experiments used ASR and speaker recognition modules in the Kaldi speech recognition toolkit [18]. The multisplice DNN recipe and our speaker recognition systems are available in the Kaldi code repository.

## 3. RESULTS

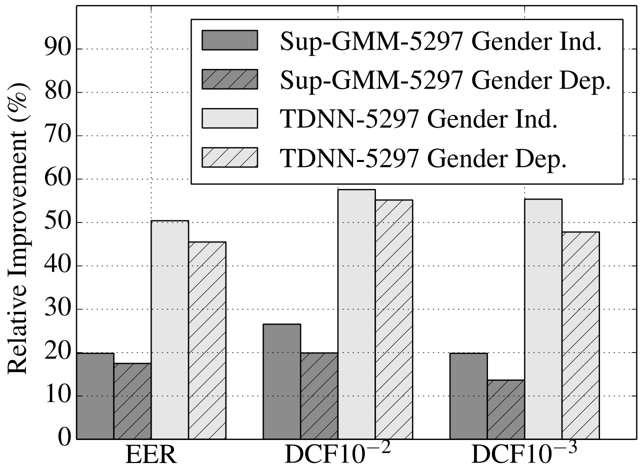

**Fig. 3**: Relative improvement over the GMM-5297 baseline.

We compare gender independent and gender dependent versions of the baseline GMM, sup-GMM and TDNN systems. The gender independent systems each have a single pipeline which evaluates all of the SRE10 extended condition 5. The gender dependent systems share most of the same components with the gender independent systems. The SRE data is partitioned into male and female sets and two PLDA backends are trained. Accordingly, we evaluate the gender dependent models on just the male or female portions. To avoid overly large tables we only report the performance for pooled gender dependent and independent scores.

| System | EER(%) | DCF$10^{-3}$ | DCF$10^{-2}$ |
|---|---|---|---|
| Sup-GMM-5297 | 1.94 | 0.388 | 0.213 |
| TDNN-5297 | 1.20 | 0.216 | 0.123 |
| GMM-2048 | 2.49 | 0.496 | 0.288 |
| GMM-4096 | 2.56 | 0.468 | 0.287 |
| GMM-5297 | 2.42 | 0.484 | 0.290 |

**Table 1**: Performance comparison of gender independent models on SRE10 C5.

| System | EER(%) | DCF$10^{-3}$ | DCF$10^{-2}$ |
|---|---|---|---|
| Sup-GMM-5297 | 1.65 | 0.354 | 0.193 |
| DNN-5297 | 1.09 | 0.214 | 0.108 |
| GMM-2048 | 2.16 | 0.417 | 0.239 |
| GMM-4096 | 1.96 | 0.414 | 0.227 |
| GMM-5297 | 2.00 | 0.410 | 0.241 |

**Table 2**: Performance comparison of gender dependent models on SRE10 C5.

Because there isn't much of a performance difference between the unsupervised GMMs with 2048, 4096, and 5297 components we choose GMM-5297 as our primary baseline, since it has, by a small margin, the highest gender independent results of the baseline models.

Figure 3 illustrates the relative improvement of the TDNN and sup-GMM over the GMM-5297 baseline. Across the three operating points on the gender indendent and dependent systems we see a relative improvement of 13.65%-26.55% by the sup-GMM and 47.80%-57.59% by the TDNN. Although the performance of the sup-GMM is less than the TDNN, it nevertheless outperforms the baseline by a significant margin. In similar methods such as [8] and [14] the supervised-GMM did not bring any significant improvements by itself. Perhaps the underlying reason lies in the overall quality of the TDNN which the sup-GMM is based off of. Additionally, the use of a full-covariance may allow the sup-GMM to retain modeling capacity.

Tables 1 and 2 detail system performance with gender independent and dependent models on the three operating points. Figure 4 provides an illustration of the EER of the three primary systems.

The primary advantage of a GMM-based method over a DNN-based method is in its computational efficiency during i-vector extraction. Table 3 and Figure 5 show the average amount of time to process a 5 minute utterance, broken down by different parts of the pipeline. Ten 5 minute utterances were selected at random from the SRE10 test and each system processed these utterances from feature extraction to i-vector extraction 30 times. The experiment was performed on an Intel x86-64 machine with 48 2000Mhz CPUs. The duration was recorded by the Linux tool `time` and we only report the usr+sys portion.

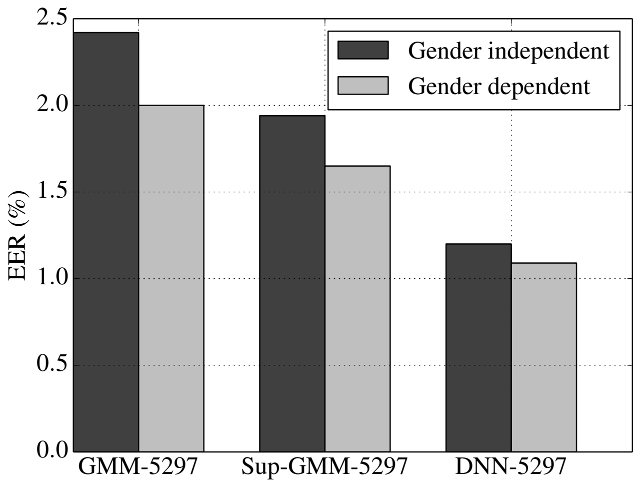

**Fig. 4**: Comparison of EERs.

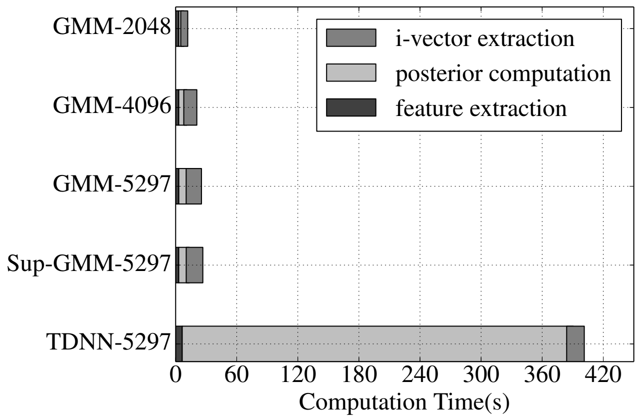

**Fig. 5**: Comparison of average total CPU time for system pipelines

The GMM-2048 system is about twice as fast as the larger GMMs with 4096 or 5297 components during posterior and i-vector extraction. Since the TDNN system needs to compute features for both the DNN and for speaker recognition this stage of the pipeline is about twice as slow. The vast majority of the computation is spent in the posterior calculation and is about 10x heavier than the GMM-5297 system. We see that the computational cost of the sup-GMM-5297 system is comparable to the GMM-5297 baseline.

In practice we would perform the DNN posterior matrix calculations in CUDA. However, by comparing the total CPU time between the systems, we can better expose the overall computational load of the DNN, and facilitate a comparison of compute-cost vs. performance of the three systems.

**Table 3**: CPU timing comparison for primary stages of the system pipelines.

| System | Feat.(s) | Post.(s) | i-Vec.(s) | Tot.(s) |
|---|---|---|---|---|
| Sup-GMM-5297 | 3.05 | 10.39 | 16.32 | 29.76 |
| TDNN-5297 | 6.46 | 384.05 | 17.30 | 407.81 |
| GMM-5297 | 3.09 | 10.33 | 15.04 | 28.46 |
| GMM-4096 | 3.04 | 8.00 | 12.82 | 23.86 |
| GMM-2048 | 3.04 | 5.04 | 6.85 | 14.93 |

## 4. CONCLUSION

We explored the use of TDNNs for speaker recognition on the SRE10 task. We found that this DNN yields a large relative improvement over the unsupervised GMM baseline on EER and DCF operating points. With the TDNN-UBM we also achieve a 1.20% EER, which we believe is the best reported on the task. We also found that the supervised-GMM, normally of minor use in the DNN-based system, can be used on its own as a fast alternative to the DNN with better performance than the baseline. TODO: Consider reflecting on the use of unique attributes of the TDNN for this task, and possibly comparing the relative performance gains here vs related work.

## 5. REFERENCES

[1] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[2] S.J.D. Prince and J.H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, Oct 2007, pp. 1–8.

[3] Niko Brümmer and Edward De Villiers, "The speaker partitioning problem.," in *Odyssey*, 2010, p. 34.

[4] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Odyssey*, 2010, p. 14.

[5] Jesús A Villalba and Niko Brümmer, "Towards fully bayesian speaker recognition: Integrating out the between-speaker covariance.," in *INTERSPEECH*, 2011, pp. 505–508.

[6] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems.," in *Interspeech*, 2011, pp. 249–252.

[7] Daniel Garcia-Romero, Xinhui Zhou, and Carol Y Espy-Wilson, “Multicondition training of gaussian plda models in i-vector space for noise and reverberation robust speaker recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4257–4260.

[8] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Moray McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1695–1699.

[9] Daniel Garcia-Romero, Xiaohui Zhang, Alan McCree, and Daniel Povey, “Improving speaker recognition performance in the domain adaptation challenge using deep neural networks,” in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 378–383.

[10] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K.J. Lang, “Phoneme recognition using time-delay neural networks,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 3, pp. 328–339, Mar 1989.

[11] Hasim Sak, Andrew W. Senior, and Françoise Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” *CoRR*, vol. abs/1402.1128, 2014.

[12] George Saon, Hagen Soltau, Ahmad Emami, and Michael Picheny, “Unfolded recurrent neural networks for speech recognition,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[13] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” 2015.

[14] M. Omar and J. Pelecanos, “Training universal background models for speaker recognition,” in Odyssey: The Speaker and Language Recognition Workshop, 2010.

[15] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, P Ouellet, and J Alam, “Deep neural networks for extracting baum-welch statistics for speaker recognition,” in *Proc. Odyssey*, 2014.

[16] “The nist year 2010 speaker recognition evaluation plan,” 2010.

[17] Xiaohui Zhang, J. Trmal, D. Povey, and S. Khudanpur, “Improving deep neural network acoustic models using generalized maxout networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 215–219.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, et al., “The kaldi speech recognition toolkit,” 2011.