INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Deep Speaker Embeddings for Short-Duration Speaker Verification

Gautam Bhattacharya[1,2], *Jahangir Alam*[2], *Patrick Kenny*[2]

[1]McGill University, Montreal, Canada
[2]Computer Research Institute of Montreal, Montreal, Canada

gautam.bhattacharya@mail.mcgill.ca, jahangir.alam@crim.ca, patrick.kenny@crim.ca

Abstract

The performance of a state-of-the-art speaker verification system is severely degraded when it is presented with trial recordings of short duration. In this work we propose to use deep neural networks to learn short-duration speaker embeddings. We focus on the 5s-5s condition, wherein both sides of a verification trial are 5 seconds long. In our previous work we established that learning a non-linear mapping from i-vectors to speaker labels is beneficial for speaker verification [1]. In this work we take the idea of learning a speaker classifier one step further - we apply deep neural networks directly to time-frequency speech representations. We propose two feedforward network architectures for this task. Our best model is based on a deep convolutional architecture wherein recordings are treated as images. From our experimental findings we advocate treating utterances as images or 'speaker snapshots, much like in face recognition. Our convolutional speaker embeddings perform significantly better than i-vectors when scoring is done using cosine distance, where the relative improvement is 23.5%. The proposed deep embeddings combined with cosine distance also outperform a state-of-the-art i-vector verification system by 1%, providing further empirical evidence in favor of our learned speaker features.

Index Terms: speaker recognition, convolutional neural networks, deep learning, i-vectors

1. Introduction

Text-Independent speaker verification considers the problem of verifying a speakers identity given two sides of a verification trial. The trials are audio recordings of arbitrary duration, and their phonetic content is unconstrained. The problem can be broken into two parts. First we need a way to model or represent speakers. The classical approach involves modeling speakers as Gaussian mixtures (GMM) via adaptation of a Universal Background Model (UBM) [2]. The GMM-UBM paradigm was followed successively by joint factor analysis (JFA) and the well-known i-vector speaker representation [3, 4].
The second part of the problem involves compensating for the degradation caused by the recording channel. These types of degradations are called channel effects, and can degrade the performance of a speaker verification system significantly. Channel compensation is essential while using both JFA and i-vectors. JFA attempts to model both the speaker and the channel simultaneously, while in the case of i-vectors, state-of-the-art performance can be obtained by combining them with a classifier like probabilistic linear discriminant analysis (PLDA). PLDA explicitly models the speaker and channel separately and thus performs channel compensation as part of verification [5].

key feature of the i-vector approach is that the algorithm reduces speech recordings of arbitrary duration to a low-dimensional, fixed-length vectors. The i-vector representation constrains the speaker and channel variability in a recording to reside in this low-dimensional space. One of the shortcoming of this approach is its lack of robustness to short-duration recordings. This shortcoming was highlighted in the 2016 NIST-SRE evaluation, where the performance of the winning system was significantly worse than in previous evaluations. It should be noted that the the 2016 NIST evaluation posed several new challenges, including short and variable duration recordings, and domain mismatch between training and test data.

In this work we focus on the problem of text-independent speaker verification, on recordings of short duration. We employ deep neural networks to learn speaker embeddings. The embeddings are extracted for a deep network that is trained to classify speakers given 5 seconds of speech. We chose 5 seconds because this is a challenging duration for i-vectors to model. In our experiments we show that the performance of PLDA degrades dramatically when evaluated on i-vectors extracted from 5 second long recordings. All the networks presented in this work can be extended to arbitrary and variable durations, however we reserve those experiments for future work.

We report speaker verification performance on female part of the NIST-SRE 2010 test set. We select the first 5 and first 10 seconds respectively to create two modified test sets for the 5s-5s and 10s-10s experiments. We present a deep convolutional network (convnet) architecture that draws inspiration from face recognition. We also present a fully-connected architecture that is adapted for sequential data. The performance of this model is also competitive, and the network architecture can easily be extended to sequences of variable length. Based on our experimental results, we advocate the view of treating a time-frequency representation of speech like an image. Where each 'image is 5 seconds long and 40 filter-banks wide. In the 5s-5s case, the proposed deep convnet speaker embeddings outperform i-vectors by 23.5% (with cosine scoring) and 6.5% (with PLDA scoring).

The remainder of the paper is organized as follows. In the next section we analyze the problem of modeling speakers with neural networks. We also provide details of the deep network architectures used in this work. This is followed by a section describing our experiments and results. We conclude with a discussion about our findings and directions we hope to pursue in future work.

2. Modeling Speakers with Deep Networks

Recently there have been several efforts to use deep neural network to learn speaker embeddings [6, 7, 8]. However, most these works have targeted text-dependent speaker verification. A notable exception is the work in [9]. The authors train a deep network in a end-to-end fashion using a loss function inspired

http://dx.doi.org/10.21437/Interspeech.2017-1575

by PLDA. While their model performed extremely well on a proprietary dataset, they did not report results on a dataset that is publicly available.

In this work we are interested in developing speaker recognition models that do not use information from an automatic speech recognition (ASR) system. Such 'ASR-free systems have several potential advantages. Apart from being a more elegant solution from a computational standpoint, finding an ASR system suited to our specific speaker recognition needs is not always feasible. The most recent NIST evaluation is a good example of this point, where the languages of the test data were not English. Drawing comparisons to speech recognition, we note that text-independent speaker recognition poses a related but different set of challenges. We are less interested in information at the frame level and more concerned with extracting information at the utterance or segment level. This implies that in order for a neural network to learn something meaningful about the identity of a speaker, it needs to see a large enough context. Consequently, all the deep network models presented in this works are trained on 5-second long snippets of speech.

In this context we argue that recognizing speakers has more in common with recognizing faces than recognizing speech. Indeed many ideas from face recognition have been successfully ported to speaker recognition, including the current state-of-the art PLDA model [10].

2.1. Learning a Speaker Classifier

We consider the problem of training a deep network ϕ to recognize $N = 4032$ unique individuals, setup as a N-ways classification problem. For each training time-frequency image, $s_t, t = 1, 2, ..., N$ the network outputs a score vector $x_t = W\phi(s_t) + b \in \mathbb{R}$ by means of a fully-connected output layer containing N linear predictors $W \in \mathbb{R}^{ND}$,$b \in \mathbb{R}^N$, one per identity/speaker. These scores are compared to the ground-truth speaker labels $l_t \in 1, ..., N$ by computing the empirical softmax log-loss $E(\phi) = \sum_t log(e^{<e_{l_t}, x_t>} / \sum_{q=1}^{N} e^{<e_q, x_t>})$, where e_l denotes the one-hot vector of class l. After learning, the classifier layer (W, b) can be removed and the score vectors $\phi(s_t)$ can be used for speaker verification using cosine distance to compare them.
Face recognition research has shown that verification scores can be improved significantly by tuning them for verification in Euclidean space using a triplet loss [11]. However our initial experiments with a triplet loss were not successful and we chose to work with only networks trained using the cross-entropy loss function.

2.2. Deep Convnet Speaker Embeddings

The convolutional network we settled on is inspired by the popular VGGnet architecture [12]. This model has been successfully adapted for speech recognition [13], as well as face recognition [11]. We made certain modifications to the basic network structure based the large size of the 'images that we wish to process (500x40). Specifically, we use large convolutional kernels in the first convolutional block of the network. We also pool more aggressively over the temporal dimension of our images as it is much larger than the feature dimension. We use the same conv-conv-pool structure of the original VGGnet. We refer to this structure as a convolutional block. With the exception of the first block, all other convolutional blocks use (3x3) small filters followed by (3x2) max-pooling. The first convolutional block uses (7x7) large filters and is followed by (2x2) max-pooling. Empirically we found that larger receptive fields in the lowest layers of the network improved performance significantly. The last two layers are fully connected and feed into the output softmax layer. A complete specification of the net-work is shown in figure 1. All convolutions are of stride 1. We use the RELU non-linearity in all the hidden layers of the net-work. For regularization we use batch normalization [14] in all the hidden layers, and dropout [15] in the fully-connected layers.

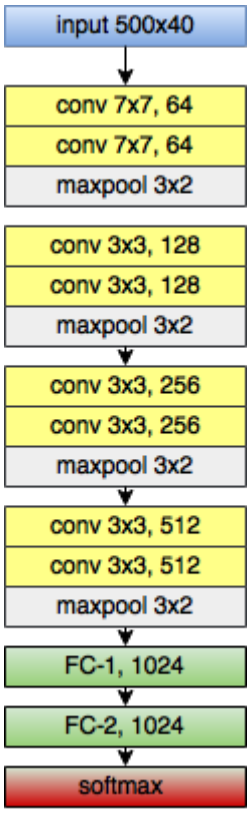

Figure 1: *Figure 1: 10-layer Deep Convolutional Network*

After learning the classifier, we treat the activations of the last fully-connected layer as our speaker embedding. These embeddings can be directly scored using cosine distance, or via PLDA-based scoring.

2.3. Fully-Connected Attention Embeddings

In order to use a fully connected network to process a sequence of frames, we need to make a simple modification while forward propagating the sequence through the network [7]. The idea is to forward propagate the entire sequence of 500 frames through the network, one-by-one, and then aggregate this information into a sequence-level feature before passing it to the output layer. In [7] the authors simply averaged the hidden activations to obtain a single feature.
In this work we propose to aggregate the hidden activations of the network using a feedforward attention model [16]. In our experiments we show that an attention based embedding works better than an averaged embedding.

Consider a sequence of speech frames $X = x_1, x_2, ..., x_N$. Forward propagating this sequence through the network, h = f(x), produces a sequence of hidden embeddings $H = h_1, h_2, ..., h_N$. Where f is a non-linear function.

The attention model is a small neural network that assigns an unnormalized score to each hidden embeddings. These scores are then normalized using the softmax function and used to compute a weighted average of the hidden embeddings.

$$s = [g(h1), g(h2), ..., g(hN)]$$
$$s_{norm} = softmax(s)$$
$$h_{attention} = \sum s_{norm_i}.h_i$$

Where f and g are non-linear functions. The utterance or segment-level feature $h_{attention}$ is in turn passed to the output softmax layer, which outputs a probability distribution over speakers. After learning is complete, we treat $h_{attention}$ as our speaker embedding for verification.

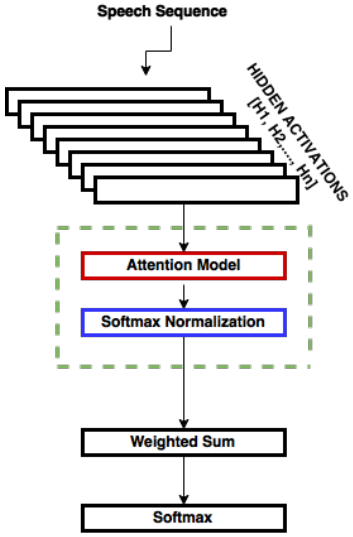

Figure 2: *Figure 2: Fully-Connected Attention Network*

In order to produce the hidden activation $[h_1, h_2, ..., h_N]$, we use a fully-connected network with 2 hidden layers and 2048 hidden units each. Leaky-RELU activations were used in both the layers. The output of the second hidden layer is processed by the attention layer that produces a 2048-dimensional segment-level feature via a weighted sum. We use the $tanh$ non-linearity in the attention layer.

3. Experiments and Results

In this section we present out experimental setup, as well as details related to the classifiers, front-end features and neural network training. We follow this with the results of our experiments on 5-second and 10-second test recordings.

3.1. Experimental Setup

We report speaker verification performance on the female portion of the NIST-SRE 2010 test set. We make use of data from previous NIST evaluations (2004-08) and a portion of the Switchboard dataset for training both our deep networks and baseline i-vector/PLDA systems. The training set consisted of 4032 unique speakers (male and female). Consequently our models are gender independent. Speaker verification performance is measured in terms of equal error rate (EER).

3.2. Speaker Verification

In this study we compare the performance of the proposed deep speaker embeddings against i-vectors. In order to keep the comparison fair, we score all the speaker features using the same classifiers, namely, cosine distance and PLDA. All the PLDA models are trained using short duration i-vectors and deep embeddings. When training PLDA models with deep embeddings, we reduce the dimensionality of the embeddings to 600 using PCA.

The i-vector extractor used in this work is trained using sufficient statistics collected from a 2048 component GMM-UBM. The i-vector extractor is trained using 60 dimensional mel-frequency cepstral coefficient (MFCC) speech features, while all the neural networks in this work were trained on 40 dimensional log filter-bank features. We also note that the i-vector extractor is trained on full length recordings. Short duration i-vectors are extracted from the first n frames frames of the test data. Where n is the duration of recordings being considered (in ms). Deep speaker embeddings are extracted from the same data.

3.3. Network Training Details

As mentioned previously, all the deep networks used in this work are trained on 5-second snippets of speech. We chop up the recordings in the training set to 5-second long chunks. We only use speakers who have 5 recordings or more. This results in a training set of approximately 2 million data points. A sub-set of this data was used to tune network hyper-parameters and to determine the threshold for early-stopping. We used the Adam optimizer with a learning rate of 0.0005 in all our models. We also decayed the learning rate based on validation set performance.

3.4. 5s-5s Experiments

In this work we focus on the case where both the enrollment and test recordings of a verification trial are of short duration. In this section we consider the case where the test recordings are only 5 seconds long. Speaker verification performance is severely degraded at this timescale, with i-vector/PLDA system produces error rates of 24.78%. This is compared to an error rate of 2.48% in the case of full-duration i-vectors.

Table 1: *speaker verification Results: 5s-5s*

Model	EER(%)
ivectors + cosine	31.1
Feedforward (mean) + cosine	33.2
Feedforward (Attention) + cosine	31.1
Convnet + cosine	**23.73**
ivectors + PLDA	24.78
Feedforward (Attention) + PLDA	25.11
Convnet + PLDA	**23.16**

Table 1 compares the performance of the proposed deep speaker embeddings against i-vectors. We see that both the deep speaker embeddings perform favorably compared to the baseline. The fully-connected attention model was able to roughly match the performance of i-vectors while outperforming the embeddings obtained by averaging the hidden activations of the network. The best speaker verification performance was shown by the

convolutional speaker embeddings that improved on the performance of i-vectors by 6.5% and 23.5%, in the case of PLDA and cosine distance respectively.

3.5. 10s-10s Experiments

We were also interested to see how well the deep convnet speaker embeddings would perform on a different time-scale. we employed a sliding-window approach without overlap to extract speaker embeddings from 10-seconds of speech . This amounts to extracting two embeddings, which we average to obtain a single feature. We note that we do not expect the CNN to do as well as in the 5s-5s case. This is because the network has not seen recordings longer than 5 seconds during training, and our sliding-window+averaging approach is suboptimal. Given the results presented in the previous section, we only compare the CNN speaker embeddings with i-vectors on the 10s-10s task.

Table 2: *speaker verification Results: 10s-10s*

Model	EER(%)
ivectors + cosine	25.66
Convnet + cosine	**20.82**
ivectors + PLDA	**17.44**
Convnet + PLDA	17.51

We see that the performance of 10 second Convnet embeddings improves speaker verification performance from 23.73% to 20.82% compared to embeddings extracted from 5 second long recordings. We believe that this performance could be further improved by exposing the network to 10-second chunks of speech during training. Interestingly the performance of the 5-second convnet embeddings is better than that of 10-second i-vectors when cosine distance is used for scoring.

In the 10s-10s condition we see that PLDA modeling produces more of a boost to speaker verification performance than in the 5 second case. We see that the performance of the convnet embeddings is marginally worse than that of i-vectors. However the difference is negligible. We also tried to combine the PLDA scores of the two systems via simple averaging, however this did not yield significant improvement.

4. Discussion

From our experiments we have seen that deep neural network based speaker embeddings are competitive with i-vectors in the case of short-duration text-independent speaker verification. A finding that we found surprising was the fact that the performance of the convnet embeddings was far superior to that of the fully-connected attention embeddings. We have conducted some preliminary experiments on a smaller training set of variable duration, on which the fully-connected attention model does slightly better than the convnet. We hypothesize that this is perhaps because the model learns to focus its attention differently, depending on the length of the recording. For the convolutional network, the solution is less elegant. Variable duration recordings were accommodated by zero-padding all inputs to the same length.

Another potential reason for the somewhat disappointing performance of the feedforward-attention embeddings is the lack of network depth compared to the convnet. The fully-connected attention network we used had only 2 hidden layers. This essentially amounts to the classifier (although it is twice as wide), i.e. the top two fully-connected layers of the convnet model. These hidden-layers interact directly with the speech input. The proposed deep convnet model in comparison has 8 convolutional (and pooling) layers that interact with the input before feeding into the fully connected layers. Our results clearly indicate that having a deep convolutional feature extractor before the classifier is beneficial.

Another interesting feature of the proposed speaker embeddings is their behavior when combined with PLDA. When i-vectors are used with PLDA, they are usually pre-processed via mean-centering and length normalization. However, we found that mean-centering our deep embeddings lead to a degradation in performance and we only perform length normalization.

5. Conclusions

In this work we proposed deep neural network based speaker embeddings using convolutional and fully connected networks.We showed that a deep convnet speaker classifier can be used to learn robust, short-duration speaker embeddings. Using a simple cosine scoring strategy these embeddings outperform an i-vector/PLDA baseline on a NIST-SRE 2010 test set consisting of 5-second long recordings. Combing these embeddings with PLDA is also fruitful, showing a relative improvement of 6.5% over i-vector PLDA. In the case of 10-second recordings we see that PLDA modeling does indeed help improve the speaker verification performance of i-vectors. However, this is not the case for 5-second i-vectors. We believe that this vulnerability of PLDA motivates the development of systems that can be used end-to-end (i.e. with cosine scoring), as well as the development of new classifiers. We also hypothesize that the performance of convnet based speaker embeddings would improve if the network were exposed to 10-second long recordings during training.

In future works we would like to extend the models presented here to deal with recordings of variable duration. One possible strategy might be to combine the convnet and attention frameworks. We also note that the networks may be further optimized, for example, by considering alternate loss functions. In future work we would like to explore a triplet-loss which directly optimizes a distance metric.

6. References

[1] G. Bhattacharya, J. Alam, P. Kenny, and V. Gupta, "Modelling speaker and channel variability using deep neural networks for robust speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

[2] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[3] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[4] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

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

[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

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

[8] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," *arXiv preprint arXiv:1701.00562*, 2017.

[9] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification."

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

[11] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition." in *BMVC*, vol. 1, no. 3, 2015, p. 6.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[13] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4955–4959.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[16] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015.