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

Alternative Approaches to Neural Network based Speaker Verification

Anna Silnova, Lukáš Burget, Jan Černocký

Brno University of Technology, Czech Republic
{isilnova,burget,cernocky}@fit.vutbr.cz

Abstract

Just like in other areas of automatic speech processing, feature extraction based on bottleneck neural networks was recently found very effective for the speaker verification task. However, better results are usually reported with more complex neural network architectures (e.g. stacked bottlenecks), which are difficult to reproduce. In this work, we experiment with the so called deep features, which are based on a simple feed-forward neural network architecture. We study various forms of applying deep features to i-vector/PDA based speaker verification. With proper settings, better verification performance can be obtained by means of this simple architecture as compared to the more elaborate bottleneck features. Also, we further experiment with multi-task training, where the neural network is trained for both speaker recognition and senone recognition objectives. Results indicate that, with a careful weighting of the two objectives, multi-task training can result in significantly better performing deep features.

Index Terms: automatic speaker recognition, deep neural networks, bottleneck features

1. Introduction

Often, the problem of *automatic speaker recognition* is formulated as answering the question of whether two audio segments were uttered by the same speaker or by two different speakers. For several years *i-vector/PLDA* approach [1, 2] has been the state-of-the-art method for text-independent speaker verification. Recently, several methods making use of the popular *artificial neural networks* have been introduced, providing significant improvements to the i-vector/PLDA paradigm [3, 4, 5]. Feature extraction based on *bottleneck* (BN) NNs was recently found very effective for the speaker verification task [5], just like in other areas of automatic speech processing [6]. A BN NN has one "bottleneck" hidden layer with output dimensionality significantly lower than the other layers. The network learns to compress high dimensional NN inputs into low dimensional vectors of BN layer activations while preserving the information relevant for the task that the whole network is trained for. The activations of the BN layer (BN features) can be used as low dimensional features. In the case of speaker recognition, bottleneck features are extracted frame-by-frame from raw spectral features and used as the input for an i-vector/PLDA system.

For speaker recognition, the intuitive choice would be to train BN NN for the speaker classification task. Indeed, such training should help to preserve the speaker related information in the bottleneck layer. However, such strategy was never successful. Instead, current state-of-the-art systems use BN networks trained for frame-by-frame senone classification (i.e. ASR-like task) [5]. It might seem a counterintuitive choice since such network should suppress speaker information, and emphasize information useful for ASR. However, this method provides significant gains compared to the standard i-vector/PLDA approach and, currently, yields the best results reported on English data. The success of ASR trained bottleneck features can be explained by the sensible clustering of the acoustic space when GMM-UBM is trained on top of BN features. In this work, we delve into multi-task training, where the NNs are trained for both speaker recognition and senone recognition objectives. The motivation is to keep the ASR-like training, which has proved useful, while also encouraging the NNs to preserve more speaker related information.

In our previous works on BN features for speaker recognition [7], we have used more elaborate BN architectures, the so-called *stacked bottlenecks* (SBN), which originally proved to be very effective for the ASR task [6]. SBNs use not a single BN network but a cascade of two such networks. The BN features are extracted from the first network and then stacked in time and used as an input to the second one. Finally, the activations of the bottleneck layer of the second network are used as the final feature vectors. In this work, we provide the results of both techniques, BN and SBN features, and show that stacked bottleneck method does not bring any significant improvement with regard to the simpler BN architecture.

Even for the simple BN architecture, finding the best configuration might be difficult and expensive. One has to decide on the size of the neural network, position of the BN layer, size of the BN layer (i.e. dimensionality of the extracted features), etc. In order to test the different configurations, a separate NN needs to be trained for each of them. Moreover, optimization of parameters is more difficult with the BN architecture, where a different learning rate needs to be usually carefully selected for the weights forming the bottleneck. Recently, an alternative feature extraction method based on NN without any bottleneck was introduced. The so-called *deep features* are simply the activations of some high-dimensional hidden layer postprocessed by a standard dimensionality reduction technique such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA). This way, features of different dimensionalities derived from different hidden layers can be extracted from a single NN. In [8], these new features were successfully applied for text-dependent speaker recognition, which inspired us to test their effectiveness also for the text-independent task.

In this paper, we experiment with the standard i-vector/PLDA system trained on the different NN based features. The results reported on female part of NIST SRE 2010, condition 5 (English telephone data) show that the deep features are able to outperform BN or SBN. Further improvements in speaker verification performance can be obtained by means of the multi-task training of the NNs.

2. Deep features

The extraction of deep features is similar to the one of the BN features. First, the NN needs to be trained for senone classification (and also speaker classification in the case of multi-task

http://dx.doi.org/10.21437/Interspeech.2017-1062

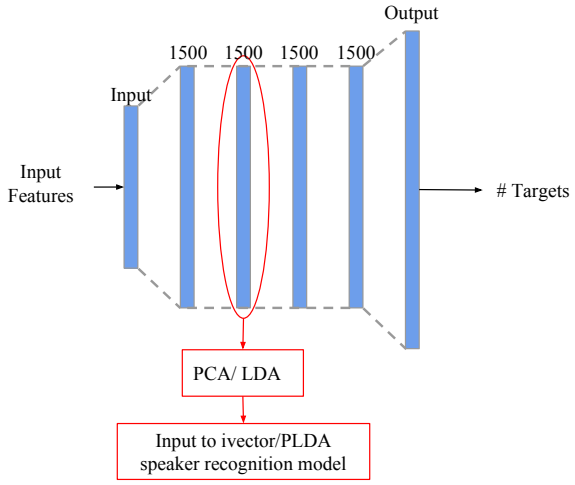

Figure 1: *Example of a neural network used for deep feature extraction.*

training). The network does not have to have any BN layer. Figure 1 shows the NN architecture for the deep feature extraction as used in our experiments.

To form the input to the network, 60-dimensional spectral features are used (20 MFCCs, including C_0, augmented with their Δ and $\Delta\Delta$ features). These features are short-term mean and variance normalized [9] over a 3 second floating window. For each frame, a window of 31 frames around the current frame (i.e. +/- 15 frames) is considered. In this window, the temporal trajectory of each feature coefficient is weighted by a Hamming window and projected into first 6 DCT bases (including C_0) [10]. This results in a $6 \times 60 = 360$-dimensional input to the NN for each frame.

We use a NN with 4 hidden layers each consisting of 1500 sigmoid units. The NN has 2423-dimensional softmax output corresponding to senone (triphone tied-state) targets. The frame-by-frame triphone tied-state labels were obtained using force alignment with a pre-trained GMM-HMM ASR system. The 2423 triphone states were obtained using decision tree state clustering during the ASR system training.

In the case of multi-task training, the network has an additional 1307-dimensional softmax output, which predicts speaker targets. In other words, all the weights in the NN are shared for solving both tasks except for the weights to the two softmax outputs, which are task specific. The objective function for the multi-task training is a weighted sum of two cross-entropy objectives corresponding to the two individual tasks. With the stochastic gradient descent training, this corresponds to updating NN parameters using two gradients (one for each objective) and two corresponding learning rates defining the relative importance of the two objectives. For optimizing the two objectives, we use two different databases each annotated only for one of the tasks (see section 3.1). Therefore, for every update of the NN parameters, two equal mini-batches are used (one from each database).

Once the NN is trained, the input spectral features are frame-by-frame propagated through the network. From a selected hidden layer, the vectors of activations (after the non-linearity) are reduced in dimensionality using PCA or LDA (with speakers as classes) and taken as deep features. Finally, the obtained frame-by-frame features are used as the input to the standard i-vector/PLDA based speaker verification system.

3. Experimental Setup

3.1. Datasets and performance metrics

For the (single) task of senone classification, the NNs are trained on Fisher English parts 1 and 2 datasets containing approximately 1700 hours of transcribed English speech. Unfortunately, only one or two recordings are available for each speaker in the Fisher dataset, which makes it unsuitable for speaker classification task training. Therefore, for the multi-task training, we additionally used Switchboard 2 dataset, which has many recordings per speaker. We discard all the speakers with less than 10 utterances and train the NNs on the data from the remaining 1307 speakers. One utterance from each speaker is left out for cross-validation. However, no transcriptions are available for Switchboard 2. Therefore, we keep Fisher English for the senone classification task in the multi-task training.

The i-vector/PLDA speaker recognition system is trained on PRISM dataset [11], containing Fisher parts 1 and 2, Switchboard 2, 3 and Switchboard cellphone phases. Also, Mixer datasets are added to the training. We evaluate the performance on female part of NIST Speaker Recognition Evaluation (SRE) 2010, condition 5 which consists of English telephone data [12]. As evaluation metrics, we use the *equal error rate* (EER, in %) and the *minimum detection cost functions* (minDCF_{08} and minDCF_{10}) as they were defined in evaluation plans of NIST SRE 2008 and 2010 [13, 12].

3.2. I-vector/PLDA back-end

For all of our experiments, we train the same i-vector/PLDA systems, which only differ in the used input features. For fast turnaround of experiments, we report most of our results for scaled down systems, where GMM-UBM models have only 512 Gaussian components and i-vectors have 400 dimensions. The dimensionality of i-vectors is further reduced to 250 using LDA. Finally, i-vectors are normalized using length normalization followed by global mean and variance normalizations. A PLDA model is used to obtain log-likelihood ratio speaker verification scores for each pair of i-vectors forming a trial. At the end of the paper, we also present selected results with the full-sized systems (UBMs with 2048 Gaussians, 600-dimensional i-vectors).

4. Experiments

4.1. Deep features for speaker verification

Table 1 compares the performances obtained with the conventional MFCC features, BN features and various configurations of deep features (DF). To extract the BN features, we use our standard NN configuration, which is the same as for the deep features (see Figure 1) except that the third 'bottleneck' layer has 80 neurons (rather than 1500). Also, the bottleneck layer uses linear activations (i.e does not use the sigmoid non-linearity).

In Table 1, the column labeled as *dim.* corresponds to the feature dimensionality. In the case of deep features, it is the dimensionality obtained after applying PCA or LDA. Figures show that deep features outperform the MFCC baseline in all cases and provide a performance similar or better than the BN features. Note that the deep features used here are extracted from the last (forth) hidden layer of the network trained in the multi-task fashion.

As the results indicate, both dimensionality reduction approaches provide similar performance with PCA having the advantage that no class labels are necessary for its estimation.

Table 1: *Performance of MFCC, BN and deep features (extracted from the 4th layer of the multi-task trained network) obtained with the scaled-down system on NIST SRE 2010, condition 5, female part.*

Features		dim.	EER,%	minDCF_{08}	minDCF_{10}
MFCC	-	60	2.68	0.133	0.517
BN	-	80	1.99	0.085	0.328
DF	LDA	80	1.96	0.096	0.378
DF	LDA	100	1.95	0.088	0.360
DF	LDA	200	1.58	0.074	0.324
DF	PCA	80	1.94	0.086	0.312
DF	PCA	100	1.92	0.081	0.330
DF	PCA	200	1.77	0.074	0.285
DF	PCA	300	1.68	0.072	0.291

When using the deep features it is important to choose the target dimensionality for the dimensionality reduction. We started with 80, the optimal dimensionality for the BN features [7]. Further, we performed the comparison of different dimensionalities of deep features. We found that 80 dimensions is not the optimal choice. Generally, larger feature vectors result in better speaker recognition performance. We increased our feature vectors up to 300 dimensions for PCA, which yielded the best performance for some of the metrics. However, the improvement is already relatively small and not worth of the increased computational complexity and memory requirements.

4.2. Layer used for the feature extraction

In Table 2, we compare deep features extracted from different hidden layers of the same neural network. For this experiment, the NN is trained in the multi-task fashion and the dimensionality of the extracted feature vectors is reduced by PCA to 80. The results indicate that, in the case of the deep features, the layers closer to the output produce better features for speaker recognition. This trend is different from the one reported in [14] for BN features, where better performance was obtained with BN layers closer to the NN input.

Table 2: *Comparison of 80-dimensional deep features extracted from different layers of a multi-task trained NN. Results are reported for the scaled-down system on NIST SRE 2010, condition 5, female part.*

layer	EER,%	minDCF_{08}	minDCF_{10}
2	2.72	0.127	0.393
3	2.21	0.095	0.321
4	1.94	0.086	0.312

4.3. Multi-task training

As mentioned before, excellent speaker recognition performance can be obtained with NN based features when the NN is trained for the senone classification task. In contrast, attempts to train the NN for the speaker recognition task have generally failed [15]. In our experiments with multi-task training, we train the NN for both tasks at the same time. However, the relative weight of the two tasks has to be carefully selected. Intuitively, the "successful" senone classification task should be weighted as the more important one. Technically, we use two different learning rates for the two multi-task objectives. Table 3 shows results obtained with different learning rate values. First, we fix the learning rate for the senone classification task to 0.004 (the optimal value for single-task training), while we sweep the learning rate for the speaker classification task over a range from 0 to 0.02 (0 in the last line means senone classification task only). The best performance was achieved when the learning rate for speaker classification is in the range from 0.0001 to 0.0004, which is an order of magnitude lower than the learning rate for senone classification. The other choices, including the single-task training, resulted in a significant degradation in performance. Next, we fix the learning rate for speaker classification task to 0.0002 and we vary the learning rate for the senone classification task. We see that the training is not very sensitive to the exact value of the learning rate as long as it stays high compared to the learning rate for the speaker recognition task.

Table 3: *Performance of deep features depending on learning rates of two tasks in multi-task training. EER,% for NIST SRE10, condition 5, female part*

learning rate for speaker \ senone	0.001	0.002	0.004	0.008
0.02			4.65	
0.002			2.28	
0.0004			1.66	
0.0002	1.78	1.83	1.77	1.75
0.0001			1.73	
0.00004			2.04	
0.00002			2.06	
0.000002			2.02	
0			2.11	

4.4. Full-scale systems

Table 4 presents selected results for full-size systems, which are based on 600-dimensional i-vectors extracted using UBM with 2048 Gaussian components. All the systems differ only in the used input features. The systems use the same BN or deep features as used in the previous experiments with the scaled-down systems. For comparison, we also include results obtained with the more elaborate SBN features, which were the features of our choice in our previous works (e.g. [7]).

For the deep features, we use the configuration that was found optimal in the previous experiments: deep features are extracted from the last layer of the multi-task trained neural network. The learning rates were set to 0.004 and 0.0002 for the senone and speaker classification tasks, respectively. We applied PCA dimensionality reduction to reduce the size of the deep features to 200.

Results reveal that SBN features are outperformed by both, BN and deep features by a wide margin (i.e. 20% and 25% relative improvement in terms of EER, respectively). The deep features turned out to be the best performing NN based features.

It has been shown [7] that the performance of i-vector/PLDA systems can be improved by concatenating BN features with the original MFCCs. Therefore, we also report re-

Table 4: *Comparison of the performance of BN, stacked BN and 200-dimensional deep features (extracted from the 4th layer of the multi-task network) on NIST SRE 2010, condition 5, female part. Large-scale system: UBM of 2048 Gaussians, 600-dimensional i-vectors*

features	EER,%	minDCF_{08}	minDCF_{10}
BN[16]	1.62	0.065	0.220
SBN[7]	2.02	0.077	0.222
DF	1.52	0.062	0.210
BN+MFCC[16]	0.96	0.042	0.146
SBN+MFCC[7]	0.93	0.041	0.140
DF+MFCC	1.32	0.053	0.189

sults for the different sets of features concatenated with MFCCs. Unfortunately, the best "stand alone" performing deep features attain the worst performance when concatenated with MFCCs, meaning that both sets of features are not as complementary as the others. The SBN+MFCC based system outperforms BN+MFCC based one by a small margin (i.e. 3% relative improvement in terms of EER). Still, the attained gain is not significant enough when considering how computationally expensive the SBN features are, compared to the simple BN architecture.

5. Conclusions

We have shown that the conceptually simpler deep features can be a good alternative to BN features. For larger feature dimensionalities, the deep features have the potential to outperform BN features. Besides, deep features are more experiment-friendly as, unlike with BN features, it is not necessary to retrain the neural network to experiment with feature dimensionality, or with the position of the layer used for the feature extraction.

The results presented on NIST SRE 2010 dataset indicate that: deep features are more informative when extracted from the layer closer to the output of the network. Both PCA and LDA projected deep features perform similarly, therefore, PCA seems a better choice as it does not require any class labels for training. Generally, deep features of higher dimensionality perform better, while the improvements will not always compensate for the higher computational costs. And multi-task training can be beneficial with careful tuning of learning rates for each task.

Unfortunately, deep features seem to be less complementary to raw MFCCs than BN features. The improvements from concatenating deep features with MFCCs are much smaller than what was previously observed for the BN features.

6. Acknowledgements

The work was supported by Czech Ministry of Interior project No. VI20152020025 "DRAPAK", European Union's Horizon 2020 project No. 645523 BISON, by Google research award, Grant Agency of the Czech Republic project No. GJ17-23870Y, and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602".

7. References

[1] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Tansactions on audio, speech, and language processing*, vol. 19, pp. 788–798, 2011.

[2] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Odyssey 2010: The Speaker and Language Recognition Workshop, Brno, Czech Republic, June 28 - July 1, 2010*, 2010, p. 14.

[3] F. Richardson, D. A. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," *CoRR*, vol. abs/1504.00923, 2015. [Online]. Available: http://arxiv.org/abs/1504.00923

[4] Y. Tian, M. Cai, L. He, and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification," in *Proceedings of Interspeech 2015*. International Speech Communication Association, 2015, pp. 1151–1155.

[5] M. McLaren, Y. Lei, and L. Ferrer, "Advances in deep neural network approaches to speaker recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, 2015, pp. 4814–4818.

[6] F. Grézl and M. Karafiát, "Hierarchical neural net architectures for feature extraction in asr," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[7] P. Matějka, O. Glembek, O. Novotnỳ, O. Plchot, F. Grézl, L. Burget, and J. H. Cernockỳ, "Analysis of dnn approaches to speaker identification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5100–5104.

[8] T. Fu, Y. Qian, Y. Liu, and K. Yu, "Tandem deep features for text-dependent speaker verification." in *INTERSPEECH*, 2014, pp. 1327–1331.

[9] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, pp. 133 –147, 1998.

[10] M. Karafiát, F. Grézl, K. Veselý, M. Hannemann, I. Szőke, and J. Černocký, "But 2014 babel system: Analysis of adaptation in nn based systems," in *Proceedings of Interspeech 2014*. International Speech Communication Association, 2014, pp. 3002–3006.

[11] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matějka, O. Plchot *et al.*, "Promoting robustness for speaker modeling in the community: the prism evaluation set," in *Proceedings of NIST 2011 workshop*. Citeseer, 2011.

[12] NIST, "The nist year 2010 speaker recognition evaluation plan," www.itl.nist.gov/iad/mig/tests/sre/2010/NIST_SRE10_evalplan.r6.pdf, 2010.

[13] ——, "The nist year 2008 speaker recognition evaluation plan," www.itl.nist.gov/iad/mig/tests/sre/2008/sre08_evalplan_release4.pdf, 2008.

[14] M. McLaren, L. Ferrer, and A. Lawson, "Exploring the role of phonetic bottleneck features for speaker and language recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5575–5579.

[15] S. Yaman, J. Pelecanos, and R. Sarikaya, "Bottleneck features for speaker recognition," in *Odyssey 2012-The Speaker and Language Recognition Workshop*, 2012.

[16] A. Lozano-Diez, A. Silnova, P. Matejka, O. Glembek, O. Plchot, J. Pešán, L. Burget, and J. Gonzalez-Rodriguez, "Analysis and optimization of bottleneck features for speaker recognition," in *Odyssey 2016: The Speaker and Language Recognition Workshop*, 2016, pp. 21–24.