Speaker Verification using DNN

Þorsteinn Daði Gunnarsson
tdgu@kth.se

Þór Stefánsson
thorstef@kth.se

Abstract

A recent method using deep neural networks (DNNs) for speaker verification(SV), the d-vector, has shown promising results. In this paper we use the d-vector for semi text-independent SV. A DNN is trained to classify speakers. The last hidden layer is then used to create feature vectors for enrollment and evaluation. The resulting model shows promising performance, down to 5% EER, increasing as more utterances are used for enrollment.

1 Introduction

Speaker verification (SV) is the task of verifying whether or not a user is who he claims to be based on the user's voice. It can be used in security systems. SV is generally either text dependent (TD) or text independent (TI). In TD-SV, a user is verified based on a fixed password or utterance. In TI-SV the user is verified based on his voice independent of what he says. In this study we use the TIDIGITS dataset which is composed of utterances that only contain digits between zero and nine, so we are somewhere in between TD and TI verification.

SV can typically be split into three phases (10). Our implementation follows mostly the d-vector approach described in (10):

1) A background model is made from training data. In our case the model is a deep neural network (DNN) trained with supervised learning where the correct output is one of the speakers in the training data. This method is also used in (3; 5; 10).

2) New speakers are enrolled by finding their voice characteristics. In our case this involves looking at the values in the last hidden layer of the neural network. The enrolled speakers typically are not a part of the training data.

3) Test utterances are compared against the background model and the claimed identity model. A decision is made to verify or deny the identity claim based on a threshold.

Other popular background models include Gaussian mixture model-universal background model (GMM-UBM) (9) and Joint Factor Analysis (JFA) models (4). A common model for SV is based on i-vectors (1) and Probabilistic Linear Discriminant Analysis (PLDA)(8). The i-vector approach is currently the state of the art method in speaker verification. This method has been improved by using DNN acoustic models such as in (2).

We follow mostly the method described in (10). In (10) the d-vector approach is compared to a baseline i-vector approach. The results show that the d-vector outperforms the i-vector at low false rejection probability. Furthermore, a sum fusion of the two approaches performs much better than using only the i-vector approach. This method has further been developed by adding phone dependent DNN structure (6; 7) where the former uses semi text-independent speaker verification.

2 Method

We plan to adapt the method used in (10) for semi text-independent speaker verification. The method consists of two steps. First a background model is created by training a DNN to classify different speakers. Secondly the last layer, the output layer, of the resulting DNN is removed and the output from the last hidden layer used to produce feature vectors for both enrollment and verification. This method relies on that the DNN can generalize different sounds (utterances) from a speaker into a vector representation that is different for each subsequent speaker.

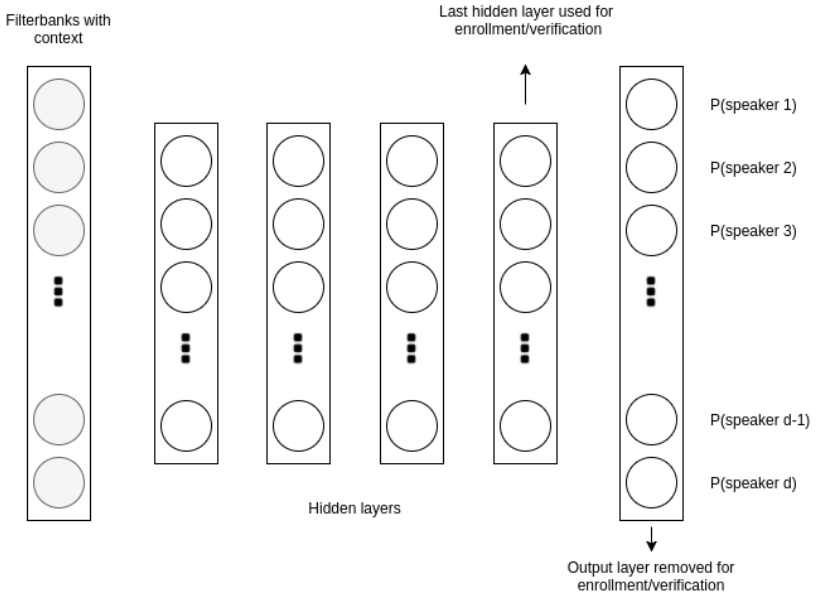


Figure 1: DNN for speaker verification.

2.1 Enrollment

For enrollment we use a set of utterances $U_s = \{u_1, u_2, ..., u_n\}$ belonging to speaker s. Each utterance is fed into the network. The output of the last hidden layer is extracted and averaged cell-wise over all frames in each utterance u_i creating a feature vector the same size as the last hidden layer in the DNN. We refer to this vector as the d-vector for the utterance u. The final feature vector for speaker s is the average d-vector in the set U_s. Thus if utterance u_i has frames $w_{i1}, w_{i2}, ..., w_{im}$ and $D(w)$ represents the last hidden layer of the DNN for frame w, the feature vector S for speaker s is:

$$S = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{m}\sum_{j=1}^{m}D(w_{ij})\right]$$

2.2 Verification

Verification is performed by matching the d-vector for the spoken utterance to the average d-vector of the user. The new utterance is processed the same way as utterances used for enrollment. A binary result is obtained by measuring the distance between the d-vectors and comparing to a threshold. If now U represents the d-vector of length K of a test utterance, and S is the feature vector for the claimed speaker, the result is:

$$\frac{1}{K}\sum_{k=1}^{K}(U(k) - S(k))^2 < threshold$$

3 Experiments

3.1 Dataset

We use the TIDIGIT dataset for our experiments. The dataset consists of 77 spoken digit sequences for 326 different speakers. The dataset was split into training, validation (development) and test sets with even distribution of male and female speakers. Training and validation sets share speakers while the test set has no common speakers with the other sets.

The training and validation set was used to train a DNN while the test set was used for enrollment and evaluation. For enrollment n utterances from the test set were chosen for each speaker. The remaining utterances were then used to evaluate the model.

3.2 Features

The features are 40 dimensional filter bank energy features with a window duration of $20ms$ and frame period of $10ms$ normalized over each set.

3.3 Neural Network Architecture

The trained DNN has 4 hidden layers with 256 cells each. The input of the network includes a context of 30 frames backwards and 10 frames forward resulting in a 1640 dimensional input vector. The training vector has $d = 112$ dimensions where d corresponds to the number of speakers in the training and validation data sets. A 50% dropout factor is applied to the two final layers. The dropout factor is expected to reduce overfitting and increase the generalization of the model.

The network was trained to classify the speakers in the training set and is then used to extract features for new speakers that are used for the verification.

The performance of the DNN is not strictly important for the experiment however the training results are disclosed in table 1.

Table 1: Statistics for the trained DNN.

Total epochs	Training error	Validation error
22	7.76%	16.35%

3.4 Verification

For verification we find the Euclidean distance between the d-vector of the spoken utterance and the user's feature vector. Other distance methods like cosine distance were not considered in this project but are also applicable for experimentation. The distance is compared to a threshold to get either a positive or negative response.

Table 2: EER results using different number of utterances for enrollment.

	Number of utterances in enrollment				
#	1	4	7	10	20
EER	10.3%	5.8%	5.1%	5.5%	5.1%

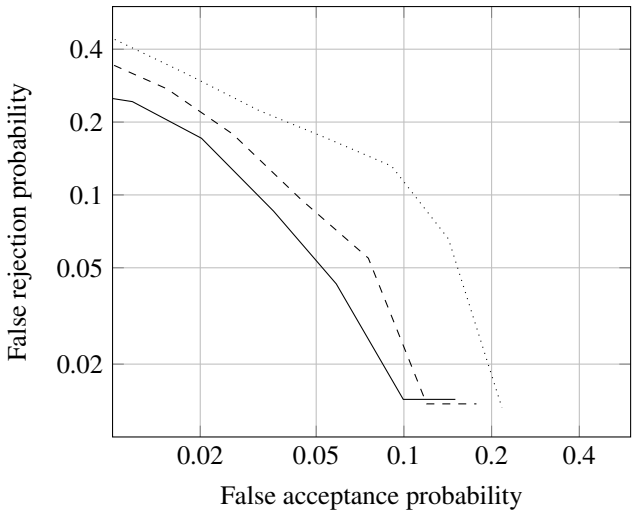
4 Results

The resulting model has an equal error rate (EER) as shown in table 2. The EER is the error at the point where the false acceptance rate and false rejection rate are the same. Lowering the threshold would increase the false rejection rate while decreasing the false acceptance rate. Increasing the threshold would have the exact opposite effect. As we can see in table 2 the EER gets lower as we

increase the number of utterances for enrollment up to 7 utterances. After that, the EER does not seem to improve.

Figure 2 shows the detection error tradeoff (DET) curve for the models using one, four and seven utterances for enrollment. The DET curve shows how the false rejection rate versus the false acceptance rate tradeoff develops as the threshold is increased. There we can see clearer how more utterances used for enrollment increase the performance of the model. Furthermore the figure shows a high drop in false rejection probability as the false acceptance probability increases.

Figure 2: DET curve with 1 (dotted line), 4 (dashed line) and 7 (solid line) utterances for enrollment



5 Discussion and Conclusions

In this project we created a DNN based semi text-independent speaker verification model. The DNN is trained to classify speakers and then used to extract user specific features. The results show great promise for this method in semi text-independent speaker verification with relatively few utterances needed for enrollment. Not every speaker had the same utterances for enrollment. This possibly made some enrollment vectors a better representation of the speaker than other because of different phonetic distributions in the enrollment utterances. We don't see this as a problem for our results since randomization would even this out. However it would be interesting to see how different accumulations of utterances effect the outcome of the model. That is for example if a even distribution of phonemes would give a better representation than an uneven one. Further studies might include experimenting with different architectures for the background model and how its error affects the enrollment and verification process. The data set included 11 words and 24 phonemes. Improvements could be made by creating different d-vectors for each phoneme and coupling the model with an automatic speech recognizer for enrollment and validation. This might permit the model to do *true* text-independent speaker verification where all phonemes are included.

References

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

[2] Garcia-Romero, D., Zhang, X., McCree, A. & Povey, D. (2014) Improving speaker recognition performance in the domain adaptation challenge using deep neural networks. *IEEE Spoken Language Technology Workshop (SLT)*, pp. 378–383.

[3] Ghalehjegh, S. & Rose, R. (2015) Deep bottleneck features for i-vector based text-independent speaker verification. *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). pp. 555–560.*

[4] Kenny, P., Boulianne, G., Ouellet, P. & Dumouchel, P. (2007) Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing.* vol. 15, pp. 1435–1447.

[5] Konig, Y., Heck, L., Weintraub, M. & Sonmez, K. (1998) Nonlinear discriminant feature extraction for robust text-independent speaker recognition. *Proc. RLA2C, ESCA workshop on Speaker Recognition and its Commercial and Forensic Applications*

[6] Li, L., Wang, D., Zhang, Z. & Zheng, T.F. (2015) Deep speaker vectors for semi text-independent speaker verification," *Computer Science*

[7] Li, L., Lin, Y., Zhang, Z. & Wang, D. (2015) Improved deep speaker feature learning for text-dependent speaker recognition. *Proceedings of APSIPA Annual Summit and Conference*

[8] Prince, S. & Elder, J. (2007) Probabilistic linear discriminant analysis for inferences about identity. *IEEE 11th International Conference on Computer Vision*, pp. 1–8.

[9] Reynolds, D., Quatieri, T.F. & Dunn, R.B. (2000) Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41.

[10] Variani, E., Lei, X., McDermott, E., Moreno, I.L. & Gonzalez-Dominguez, J. (2014) Deep neural networks for small footprint text-dependent speaker verification. *IEEE International Conference on Acoustic, Speech and Signal Processing.*