ADAPTING END-TO-END NEURAL SPEAKER VERIFICATION TO NEW LANGUAGES AND RECORDING CONDITIONS WITH ADVERSARIAL TRAINING

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

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

ABSTRACT

In this article we propose a novel approach for adapting speaker embeddings to new domains based on adversarial training of neural networks. We apply our embeddings to the task of text-independent speaker verification, a challenging, real-world problem in biometric security. We further the development of end-to-end speaker embedding models by combing a novel 1-dimensional, self-attentive residual network, an angular margin loss function and adversarial training strategy. Our model is able to learn extremely compact, 64-dimensional speaker embeddings that deliver competitive performance on a number of popular datasets using simple cosine distance scoring. One the NIST-SRE 2016 task we are able to beat a strong i-vector baseline, while on the Speakers in the Wild task our model was able to outperform both i-vector and x-vector baselines, showing an absolute improvement of 2.19% over the latter. Additionally, we show that the integration of adversarial training consistently leads to a significant improvement over an unadapted model.

Index Terms— Speaker Verification, Adversarial Training , Domain Adaptation, End-to-End

1. INTRODUCTION

Text-Independent Speaker Verification systems are binary classifiers that given two recordings answer the question:
Are the people speaking in the two recordings the same person?

The answer is typically delivered in the form of a scalar value or verification score. Verification scores can be formulated as a likelihood ratio, as in the popular i-vector/PLDA approach [1, 2]. An alternate approach is to use simple distance metrics like mean-squared error or cosine distance. Verification models that can be scored like this typically need to optimize the distance metric itself, i.e. they are optimized end-to-end. While contrastive loss based end-to-end face verification models have shown state-of-the-art performance [3], their adoption in the speaker verification community has not been widespread due to the difficulties associated with training such models.

State-of-the-art speaker verification systems follow the same recipe as i-vector systems by using a LDA/PLDA classifier, but replace the i-vector extractor with a Deep Neural Network (DNN) feature extractor [4]. The DNN embedding model is trained by minimizing the cross-entropy loss over speakers in the training data. While cross-entropy minimization is simpler than optimizing contrastive losses, the nature of the verification problem makes learning a good DNN embedding model challenging. This is evidenced by the Kaldi x-vector recipe, which we use as one of the baseline systems in this work. The recipe involves extensive data preparation, followed by a multi-GPU training strategy that involves a sophisticated model averaging technique combined with a natural gradient variant of SGD [4]. Replicating the performance of x-vectors with conventional first order optimizers is non-trivial [5].

In this article we present Domain Adversarial Neural Speaker Embeddings (DANSE) for text-independent speaker verification. We make the following contributions:

- We propose a novel architecture for extracting neural speaker embeddings based on a 1-dimensional residual network and a self-attention model. The model can be trained using a simple data sampling strategy and using traditional first order optimizers.

- We show that the DANSE model can be optimized end-to-end to learn extremely compact (64-dimensional) embeddings that deliver competitive speaker verification performance using simple cosine scoring.

- Finally, we propose to integrate adversarial training into part of learning a speaker embedding model, in order to learn domain invariant features [6].

Modern speaker verification datasets like NIST-SRE 2016 and Speakers in the Wild (SITW) are challenging because in-domain or target data is not available for training verification systems [7, 8]. This leads to a domain shift between training and test datasets, which in turn degrades performance. Our key insight in this work is that verification performance can be improved significantly by encouraging the speaker embedding model to learn domain invariant features. We achieve
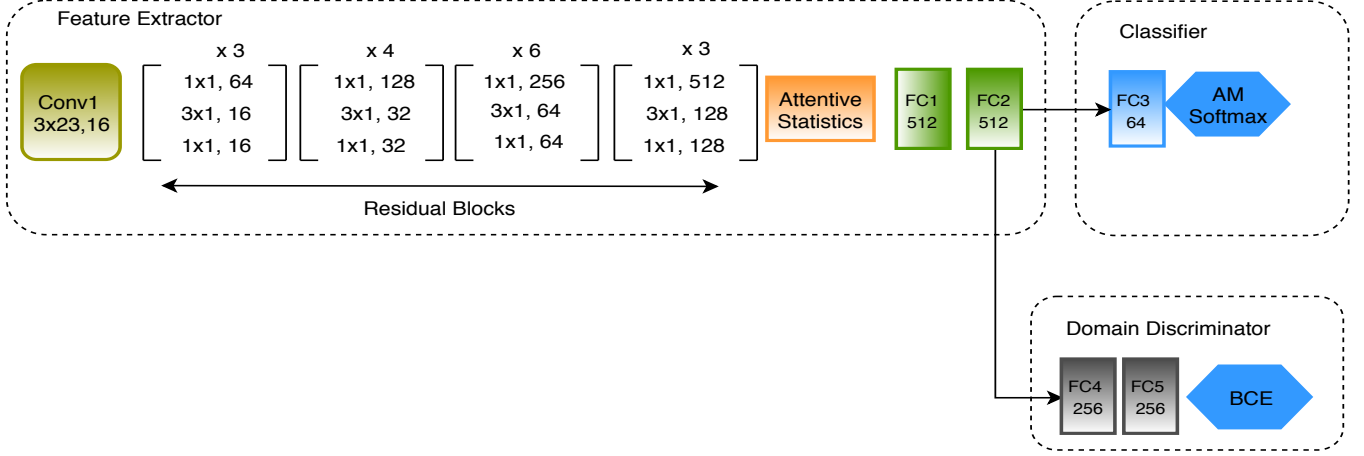
Fig. 1. Domain Adversarial Neural Speaker Embedding Model

this through Domain Adversarial Training (DAT) using the framework of Gradient Reversal [9]. This allows us to learn domain invariant speaker embeddings using a small amount of unlabelled, target domain data. The gradient reversal model has been applied previously to the speaker verification problem in the i-vector domain [6], and our work extends this model to work directly with speech features (MFCC, Spectrogram etc).

One of the main objectives of this work is to show that domain robust speaker recognizers can be learned end-to-end - a task which has proved especially challenging on the NIST datasets. Our experiments suggest that the main requirement for learning such models is to optimize for *similarity*, which we achieve by using a margin based loss function. Additionally, we make the use of a data sampling strategy that is easy to implement. We believe that this is an important factor as it encourages replication and further improvements to the model. One advantage of end-to-end optimization is that the learned speaker embeddings are more inherently discriminative. Intuitively, such embeddings are likely to be more beneficial when used as components or conditional inputs to other speech processing applications [10, 11].

2. LEARNING DOMAIN INVARIANT SPEAKER EMBEDDINGS

2.1. Feature Extractor

The first step for learning discriminative speaker embeddings is to learn a mapping $F(X_s) \longrightarrow \mathbf{f}$, $\mathbf{f} \in R^D$ from a sequence of speech frames from speaker s to a D-dimensional feature vector $\mathbf{f}$. $F(X)$ can be implemented using a variety of neural network architectures [4, 12, 13, 14]. In this work we use a deep residual network (ResNet) as our feature extractor [15]. Motivated by the fact that speech is translation invariant along the time-axis only, we propose to build our model

using 1-dimensional convolutional layers. The ResNet architecture allows us to train much deeper networks, and leverage the greater representational capacity afforded by these models. The first convolutional layer utilizes a $3XF$ filter, where F is the dimension of the frequency axis. The residual blocks are followed by an attentive statistics pooling layer (described in next section) and two fully connected layers. In total the feature extractor consists of 52 layers.

2.2. Self-Attentive Speaker Statistics

Self-Attention models are an active area of research in the speaker verification community [14, 16, 17]. Intuitively, such models allow the network to focus on fragments of speech that are more speaker discriminative. The attention layer computes a scalar weight corresponding to each time-step t:

$$e_t = \gtrsim^T f(\mathbf{W}h_t + \mathbf{b}) + k \qquad (1)$$

These weights are then normalized, $\alpha_t = softmax(e_t)$, to give them a probabilistic interpretation. We use the attention model proposed in [16], which extends attention to the mean as well as standard deviation:

$$\tilde{\boldsymbol{\mu}} = \sum_{t}^{T} \alpha_t \mathbf{h}_t \qquad (2)$$

$$\tilde{\boldsymbol{\sigma}} = \sum_{t}^{T} \alpha_t \mathbf{h}_t \odot \mathbf{h}_t - \tilde{\boldsymbol{\mu}} \odot \tilde{\boldsymbol{\mu}} \qquad (3)$$

In this work we apply a self attention model on convolutional feature maps, as indicated in Fig. 1. The last residual block outputs a tensor of size $nBXnFxT$, where nB is the batch size, nF is the number of filters and T is time. The input to the attention layer, h_t, is a nF dimensional vector.

2.3. Classifier

The classifier block, $C(\mathbf{f}, \theta_y)$, is arguably the key component of the model, as it is responsible for learning speaker discriminative features. This is especially challenging for verification models as we would like to optimize for similarity. The most natural way to achieve this is by using a triplet loss, however these models are known to be difficult to train. Recently, loss functions based on angular and additive margins have been introduced which are able to outperform triplet models [18, 19, 20]. The Additive Margin Softmax (AM-Softmax) loss function is one such algorithm with an intuitive interpretation.

$$
\begin{aligned}
L_{AMS} &= -\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{s.(cos\theta_{y_i}-m)}}{e^{cos\theta_{s.(y_i}-m)}+\sum_{j\neq y_i}e^{s.(cos\theta_j)}}\\
&= -\frac{1}{n}\sum_{i=1}^{n}\log\frac{e^{s.(\mathbf{W}^T\boldsymbol{f_i}-m)}}{e^{s.(\mathbf{W}^T\boldsymbol{f_i}-m)}+\sum_{j\neq y_i}e^{s.(\mathbf{W}^T\boldsymbol{f_j})}}
\end{aligned} \quad (4)
$$

The loss retains the structural form of the popular Softmax loss, but transforms it into a large-margin classifier. $\mathbf{W}^T$ and $\mathbf{f}_i$ are the normalized weight vector and speaker embedding respectively. This normalization is crucial, as it forces the classification only depends on the cosine. The margin m insures that correct classes are more similar than incorrect classes by a given threshold, and the the scale factor $\int$ helps with model converge faster.

2.4. Domain Adversarial Training

So far we have covered the feature extractor $F(X; \theta_f)$ and classifier $C(\mathbf{f}, \theta_y)$ part of our proposed model. In order to encourage our model to learn a symmetric feature space, we augment our network with a domain discriminator $D(\mathbf{f}, \theta_d) \longrightarrow d_i$. The discriminator takes features from both the source and target data and outputs the posterior probability that an input feature belongs to the target domain.

$$
\begin{aligned}
E(\theta_f, \theta_y, \theta_d) &= \sum_{\substack{i=1;\\d_i=0}}^{N} L_y(C(F(X_i;\theta_f);\theta_y), y_i)\\
&\quad -\lambda \sum_{i=1...N} L_d(D(F(X_i;\theta_f);\theta_d), d_i)
\end{aligned}
$$
$$(2)$$

Where $\mathbf{L_y}$ is the AM-Softmax loss described in section. and $\mathbf{L_d}$ is the the binary cross-entropy loss. The objective of domain adversarial training is to learn parameters $\theta_f, \theta_y, \theta_d$ that deliver a saddle point of the functional (2):

$$
(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \hat{\theta}_d) \quad (5)
$$

$$
\hat{\theta}_d = \arg\max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d) \quad (6)
$$

At the saddle point, the parameters of the domain classifier θ_d minimize the domain classification loss, while the parameters θ_y of the speaker classifier minimize the label prediction loss. The feature mapping parameters θ_f minimize the label prediction loss - so the features are discriminative, while maximizing the domain classification loss - so the features are domain invariant. The parameter λ controls the trade-off between the two objectives [9]. A saddle point of (5)-(6) can be found using backpropagation:

$$
\theta_f \longleftarrow \theta_f - \mu_1\left(\frac{\partial L_y^i}{\partial \theta_f} - \lambda\frac{\partial L_d^i}{\partial \theta_d}\right) \quad (7)
$$

$$
\theta_y \longleftarrow \theta_y - \mu_2\frac{\partial L_y^i}{\partial \theta_y} \quad (8)
$$

$$
\theta_d \longleftarrow \theta_d - \mu_3\frac{\partial L_d^i}{\partial \theta_d} \quad (9)
$$

Where μ_1, μ_2 and μ_3 are learning rates.

The negative coefficient in eq. (7) induces a reverse gradient that maximizes L_d^i and makes the features from the source domain similar to those from the target domain. The implementation of the gradient reversal layer is conceptually simple - it acts as the identity transformation during forward propagation, and multiplies the gradient by $-\lambda$ during backpropagation.

3. EXPERIMENTAL SETUP

Training Data All our systems are trained using data from previous NIST-SRE evaluations (2004-2010) and Switchboard Cellular audio for training the proposed DANSE model as well as the x-vector and i-vector baseline systems. For speech features extracted 23-dimensional MFCC features from the training set, which mean variance normalization. The baseline i-vector and x-vector systems were trained using the recipes provided with Kaldi. We note that the performance of our baseline systems can be further improved by adapting the PLDA classifier to the target domain [4].

Model: The feature extractor consists of $3XF$ input convolutional layer followed by 4 residual blocks [3,4,6,3], consisting of 48 layers. This is followed by an attentive statistics layer and 2 fully connected layers. The classifier consists of a one hidden layer and the AM-Softmax output layer. The Domain Discriminator consists of 2 hidden layers of 256 units each and the binary cross-entropy (BCE) output layer. We use Exponential Linear Unit (ELU) activations and batch-normalization on all layers of the network.

Optimization: We start by pre-training the feature extractor using standard cross-entropy training. Cross-entropy pre-training is carried out using the RMSprop optimizer with a

learning rate (lr) of 0.001. This learning rate is annealed by a factor of 0.1 after epochs 4 & 8.

For training the DANSE model we found it beneficial to optimize the feature extractor, classifier and domain discriminator with different optimizers. The classifier is trained using RMSprop with $lr = 0.003$, while the domain discriminator and feature extractor are trained using SGD with $lr = 0.001$. In all models the NIST data is given label '0' and the unlabelled target data is given the label '1' for discriminator training. We measure model performance on a held out validation set to determine when to stop training. The m and s parameters of the AM-Softmax loss are set to 0.6 and 30 respectively in all our experiments, while the Gradient Reversal scaling coefficient λ is fixed at 3.0.

Data Sampling: We use an extremely simple approach for sampling data during training. We sample random chunks of audio (3-8 seconds) from each recording in the training set. We sample each recording 10 times to define an epoch. For each mini-batch of source data, we randomly sample (with repetition) a mini-batch from the unlabelled adaptation data for adversarial training.

Speaker Verification: At test time we discard the domain discriminator branch of the model, as it is not needed for extracting embeddings. Extraction is done by performing a forward pass on the full recording, and using the 64-dimensional $fc3$ layer as our speaker embeddings. Verification trials are scored using cosine distance. Verification performance is reported in terms of Equal Error Rate (EER).

4. RESULTS

NIST-SRE 2016: The 2016 edition of the NIST evaluation presented a new set of challenges as compared to previous years. The evaluation data consists of Cantonese and Tagalog speakers. The change in language introduces a shift between the data distributions of the training (source) and evaluation data.

Adaptation Data: NIST also provides 2272 recordings of unlabelled, in-domain, target data for adapting verification systems.

Table 1. compares the performance of the proposed DANSE model with the baseline i-vector and x-vector systems. The DANSE outperforms the i-vector system, showing a 2.6% relative improvement in terms of the pooled EER. DANSE performs at the level of x-vectors + PLDA, however we are unable to match the full x-vector + LDA/PLDA recipe. We also see that DANSE outperforms the un-adapted AM-Softmax model by a large margin, indicating the advantage of adversarial training.

SPEAKERS IN THE WILD (SITW): The SITW database provides a large collection of real-world data with speech

Model	Classifier	Cantonese	Tagalog	Pooled
i-vector	PLDA	9.51	17.61	13.65
x-vector	COSINE	36.44	41.07	38.69
x-vector	**LDA/PLDA**	**7.52**	**15.96**	**11.73**
x-vector	PLDA	7.99	18.46	13.32
AMS	COSINE	11.44	21.22	16.28
DANSE	COSINE	8.84	17.87	13.29

Table 1. Performance of different speaker verification systems on NIST-SRE 2016

from individuals across a wide array of challenging acoustic and environmental conditions. The audio is extracted from open-source video, and while consisting of English speakers (like the training data) there is still a distribution shift due to the difference in the microphones used.

Adaptation Data: We use a small random selection of 3000 recordings from the VoxCeleb dataset [21] as adaptation data.

Model	i-vector	x-vector	AMS	DANSE
EER	11.47	10.51	9.87	**8.32**

Table 2. Performance of different speaker verification systems on SITW

From Table 2. we see that the DANSE model displays the strongest performance on the SITW dataset, showing a 2.19% absolute improvement over the x-vector baseline. Comparing the performance of our model with and without adversarial adaptation, once again we see a clear advantage for the former, with DANSE outperforming the un-adapted AM-Softmax model by 1.5%.

5. CONCLUSION

In this work we we presented a novel framework for learning domain-invariant speaker embeddings in an end-to-end fashion. By combining a powerful deep feature extractor, an end-to-end loss function and most importantly, domain adversarial training we are able to learn extremely compact speaker embeddings that deliver robust verification performance. Our main finding is that through appropriate optimization, such models can be learning using a relatively simple training and data sampling strategy. In our parallel work, we extend the framework presented here to incorporate ideas from the literature of generative adversarial networks, which yields further performance improvements [22].

6. REFERENCES

[1] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis

for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[2] Patrick Kenny, "Bayesian speaker verification with heavy-tailed priors," *Proc. Speaker Odyssey 2010*, pp. 3588–3592, 2010.

[3] Florian Schroff, Dmitry Kalenichenko, and James. Philbin, "Facenet: A unified embedding for face recognition and clustering.," *IEEE conference on computer vision and pattern recognition, In Proceedings of*, pp. 815–823, 2015.

[4] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *Submitted to ICASSP*, 2018.

[5] Sergey Novoselov, Andrey Shulipa, Ivan Kremnev, Alexandr Kozlov, and Vadim. Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition.," *Proc. Speaker Odyssey 2018*, 2018.

[6] Qing Wang, Wei Rao, Sining Sun, Leib Xie, Eng Siong Chng, and Haizhou Li, "Unsupervised domain adaptation via domain adversarial training for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4889–4893.

[7] Seyed Omid Sadjadi, Timothée Kheyrkhah, Audrey Tong, Craig Greenberg, Elliot Singer Reynolds, Lisa Mason, and Jaime Hernandez-Cordero, "The 2016 nist speaker recognition evaluation," in *Proc. Interspeech*, 2017, pp. 1353–1357.

[8] Mitchell McLaren, Luciana Ferrer, Diego Castan, and Aaron Lawson, "The speakers in the wild (sitw) speaker recognition database.," in *Interspeech*, 2016, pp. 818–822.

[9] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *arXiv preprint arXiv:1409.7495*, 2014.

[10] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

[11] Quan Wang, Hannah Muckenhirn, Kevin Wilson, Prashant Sridhar, Zelin Wu, John Hershey, Rif A Saurous, Ron J Weiss, Ye Jia, and Ignacio Lopez Moreno, "Voicefilter: Targeted voice separation by speaker-conditioned spectrogram masking," *arXiv preprint arXiv:1810.04826*, 2018.

[12] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, 2017.

[13] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[14] Gautam Bhattacharya, Md Jahangir Alam, Vishwa Gupta, and Patrick Kenny, "Deeply fused speaker embeddings for text-independent speaker verification," *Proc. Interspeech 2018*, pp. 3588–3592, 2018.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition.," *IEEE conference on computer vision and pattern recognition, In Proceedings of*, pp. 770–778, 2016.

[16] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding.," *ArXiv e-prints, abs/1803.10963*, 2018.

[17] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey, "Self-attentive speaker embeddings for text-independent speaker verification," *Proc. Interspeech 2018*, pp. 3573–3577, 2018.

[18] Wang Feng, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification.," *IEEE Signal Processing Letters 25, no. 7*, pp. 926–930, 2018.

[19] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[20] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," *arXiv preprint arXiv:1801.07698*, 2018.

[21] Arsha Nagrani, Joon Son Chung, and Zisserman Gupta, "Voxceleb: a large-scale speaker identification dataset.," *Proc. Interspeech 2017*, pp. 358–362, 2017.

[22] Gautam Bhattacharya, Jahangir Alam, and Patrick Kenny, "Generative adversarial speaker embedding networks for domain robust end-to-end speaker verification," in *Acoustics Speech and Signal Processing (ICASSP), 2019 IEEE International Conference on*. IEEE, 2019.