International Journal of Engineering Research in Electronic and Communication Engineering (IJERECE) Vol 3, Issue 5, May 2016

An Investigation of End-To-End Speaker Recognition Using Deep Neural Networks

[1] Lakhsmi HR, [2] Sivanand Achanta [3] Suryakanth V Gangashetty [4] R Kumaraswamy
[1] lakshmiranganath84@gmail.com [2] sivanand.ag@research.iiit.ac.in [3] svg@iiit.ac.in [4]hyrkswamy@gmail.com

Abstract: **State-of-the-art automatic speaker recognition (SR) has been dominated by Gaussian mixture model-universal background model (GMM-UBM) based i-vector feature extraction methods. Although these systems are robust, extraction of ivectors is very time consuming and a separate classifier needs to be trained for decision making in the end. Inorder to alleviate the above disadvantages, in this paper we propose to use deep neural networks for end-to-end speaker recognition. We perform several experiments to determine the best suited architecture, the hyper-parameter tuning algorithm and the initialization scheme for SR task. The proposed method combines feature extraction and classification step, and is of very low foot print. Through objective metric (equal error rate) we show that the proposed method outperforms the GMM-UBM conventional system**

I. INTRODUCTION

Automatic speaker recognition refers to the task of identifying the speaker in a given speech utterance. This system has many applications ranging from e-commerce, forensics and law-enforcement [1]. Depending on whether the user is asked to speak a fixed prompt (like for instance "Ok Google") or an unrestricted speech, speaker recognition (SR) can be classified into text-dependent or text-independent speaker recognition respectively. In this paper, we will focus on the more general text-independent SR system, although the techniques discussed will apply equally well to the former case. Most of the successful approaches to speaker recognition are based on generative models like Gaussian mixture model-universal background model (GMM-UBM) [2] and its successive refinements like using support vector machines as backend [3], factor analysis techniques like i-vector approach [4] [5] [6].

The current state of the art approach, which uses the vector for speaker recognition proceeds in two stages. Firstly, an utterance level feature is extracted using factor analysis (called front-end) and then a classifier is trained (called backend) discriminatively to improve the speaker recognition [7]. This method has shown to be very robust against the channel variations and the session variations [5]. Another task that is closely related to speaker recognition is identifying the language in the spoken utterance or language identification (LI) and most of these methods have been successfully applied for LI task as well [8] [9]. However, this method requires high computational power to extract i-vectors for the test utterances during verification time. There have been attempts to reduce the computational cost of i-vector feature extraction [10]. Another approach would be to merge both the feature extraction and decision making step into a single step and thereby reducing the computational over-head.

II. RELATION TO PRIOR WORK

Deep learning is a recent machine learning methodology used to learn task-specific features by discriminative learning [11]. Given the recent success of deep learning methods in automatic speech recognition [12] [13], some researchers have explored usage of deep learning in speaker and language recognition tasks [14] [15] [16] [17]. There are two ways in which one can use deep learning for SR or LR tasks, one as a feature extractor and the other as a classifier. Various researchers have tried using deep belief networks (DBN), and deep neural networks with bottleneck (BN) layer for extracting features from the speech signal and then classifying using the standard probabilistic linear discriminate analysis (PLDA) classifier. These groups reported improvements in the performance compared to GMM-UBM based approaches. Neural networks in the auto-associative mode have been explored earlier also as an alternative to GMM for speaker verification [18] [19]. Some of the current studies can be seen as an extension of these earlier works in the light of latest advances in deep learning [14] [15] [20].

On the other-hand, there have been fewer attempts (very recent) in using deep neural network architectures for both learning features and discriminating speakers simultaneously, in [21] [22], convolution neural networks (CNN) are used for end to end language recognition. In this paper, similar in spirit to these recent attempts, we propose to explore DNNs for the SR task in an end-to-end fashion where feature extraction and classification are jointly performed. This not only reduces the computational over-head of feature extraction, but also makes the model viable to be embedded into a low foot-print system.

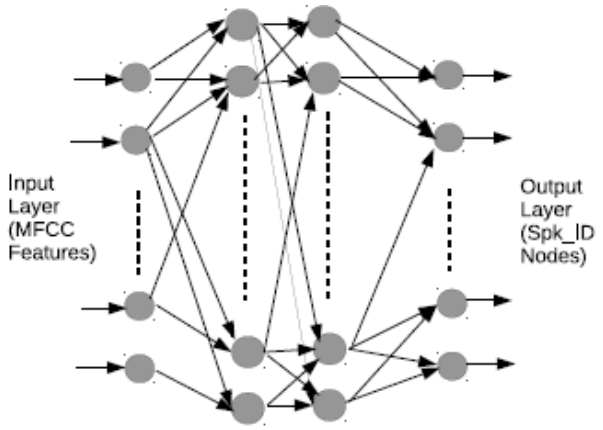

Fig. 1: DNN based SR architecture

III. DEEP NEURAL NETWORKS FOR SR

In this section, we will present the details of our proposed method for speaker recognition. The basic architecture of deep neural network for speaker recognition is shown in Fig. 1. Here, a closed set of N speakers in the database are considered, consequently the task of speaker recognition is to assign one of these identities to a spoken utterance given at the test time based on a similarity measure. It boils down to a multiclass classification problem having N classes. So a neural network as shown in Fig. 1 with a soft max layer having N nodes can be trained to minimize cross-entropy loss function for this multiclass classification problem.

However, neural networks with more than one hidden layer they are found to be difficult to optimize [23]. After initial success of pre-training scheme for training deep neural networks [24], investigations like [25], confirmed that the difficulty in training is most likely due to poor initialization schemes used earlier. Most recently there has been plethora of papers on better schemes for initializations [26] [27] [28]. In [27], it has been indicated that random initialization with orthogonality constraints are probably better than other schemes.

On the other hand, there have also been modifications to the non-linearity used which have been shown to improve the performance of deep networks in various application domains [29] [30]. Improvements over using better first-order gradient descent techniques as in [31] [32] [25], instead of naïve stochastic gradient descent with classical momentum (SGDCM) have also led to better optimum and faster training of the networks. Keeping these various developments in view, we investigated how to train deep neural networks for SR task. Especially the effect of depth and width of neural network architecture, hyper-parameter optimization techniques and initialization techniques has been explored in this work.

Below we briefly present the various initialization schemes and hyper-parameter optimization schemes implemented. The implementation is made available online 1.

3.1. Normalized initialization (NI)

This initialization scheme was proposed in [23] for training deep neural nets. Nin and N out are the number of nodes/units in the current layer and the next layer respectively.

$$\sigma = \sqrt{\frac{6}{(N_{in} + N_{out})}} \tag{1}$$

$$W = U(-\sigma, +\sigma) \tag{2}$$

Where U(□a; a) is the uniform distribution in the interval (□a; a).

3.2. Random walk initialization (RW)

This recently proposed initialization [26], assumes that weights are drawn randomly from a zero-mean Gaussian distribution and have to be scaled by a factor g to make sure that gradients don't explode or vanish after depth D. The scale factor g is different for different non-linearity.

$$g_{linear} = \exp\left(\frac{1}{2N}\right) \tag{3}$$

$$g_{ReLU} = \sqrt{2}\exp\left(\frac{1.2}{(max(N,6) - 2.4)}\right) \tag{4}$$

$$W = g * \mathcal{N}\left(0, \frac{1}{N}\right) \tag{5}$$

Where N is the number of nodes in the current layer.

3.3. SGD-CM

The most widely used stochastic gradient descent rule is given below. A momentum term is added with a momentum factor _. This term essentially adds a scaled version of previous gradients to the current gradient.

$$\Delta\theta_t = -\eta g_t + \mu\Delta\theta_{t-1} \tag{6}$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \tag{7}$$

Where _t refers to the parameters, gt is the gradient and the _ is the learning rate. It is often very difficult to set learning rate and momentum factor parameters of SGD-CM update rule and is usually set by trial and error. Also in this rule learning rate for every parameter is constant and fixed, most of the times it is desirable to have a parameter-wise learning rate without increasing computation and storage.

3.4. ADAM

Adaptive moments (ADAM) is a recently proposed update rule [32], which alleviates the problem of

having to manually fine tune learning rate hyper-parameter. This method proposes a simple per-parameter learning rate using running averages of first and second-order moments of gradients.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \quad (8)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \quad (9)$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (10)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (11)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (12)$$

Where _1; _2; _ are the hyper-parameters in this case. These hyper-parameters are shown to be less sensitive than the _and _ in SGD-CM as an adaptation procedure is in-built in the update steps.

IV. EXPERIMENTAL DETAILS AND RESULTS

For our experiments we use TIMIT database. A subset of 462 speakers is taken. Each speaker has 10 spoken utterances. We split the total data of speaker as 8/1/1 for training/ validation/testing respectively. There are total of 3696 wave files for training, 462 each for testing and validation. Static Mel-frequency costrel coefficients (MFCC'S) are used as acoustic features for both baseline and proposed SR systems.

The dimensionality of MFCC'S used is 13. As our baseline system we trained a GMM-UBM system using open-source software toolkit [33]. In the GMM-UBM system we have varied the number of mixture components from 8 to 128 in the increasing powers of 2. The EER results for the baseline are shown in Table. 1. The DET curve for the best GMMUBM system can be seen in Fig. 3. The EER decreases as the number of mixtures is increased but the rate of decrease in the EER seems to be saturating as we increase the number of mixtures from 64 to 128.

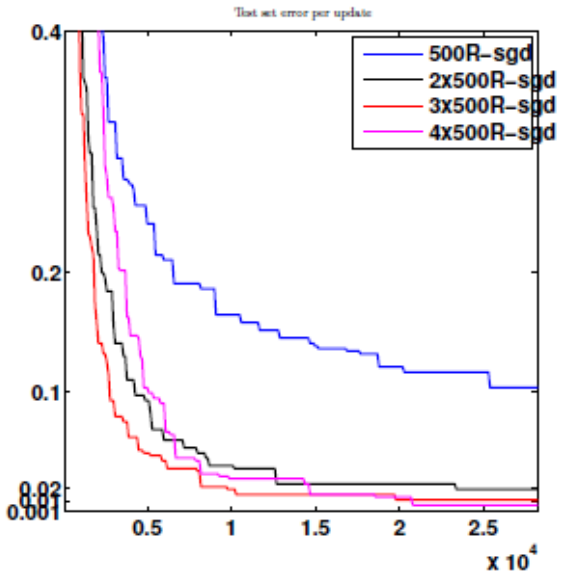

Fig. 2: figure Test error curves with NI initialization

Table 1: Performance of baseline GMM-UBM system

# mixtures	8	16	32	64	128
EER (in %)	2.2640	1.4891	1.01	0.7576	0.6364

For DNN, we trained networks of different architectures. The depth of architectures was varied from 1 to 4 layers and the breadth was also varied per layer with either 500 or 1000 units in each layer. Each DNN was trained for 50 epochs. Input layer is linear and has 13 nodes, the output layer is a Soft max layer and the number of nodes was set to the number of speakers in the database (in this case it was 462). The cross-entropy loss function was minimized using a mini batch SGD algorithm. The mini batch size was set to 1000. All the DNNs were trained using two hyper-parameter learning algorithms one with naive SGD-CM and the other with more recent ADAM [32] method.

The test set error (misclassification in %) per update step is plotted for each of the architectures in Fig. 2. It can be seen from the test error plots that deeper architectures perform better than the shallower counter parts (a similar observation was made with 1000 units width and hence the results are not reported here to avoid redundancy).

The EER is reported in the Table. 2. DNN with 4 hidden layers, with REL U non-linearity using simple ADAM (_ = 0:001) 0.8658 EER was achieved. After this, the system has been retrained with ADAM (_ = 0:00001) optimizer which Outperformed our baseline GMM-UBM system as shown un der the ADAM (_ = 0:001) + ADAM (_ = 0:00001) column in Table. 2. The DET curve for the best GMM-UBM system can be seen in Fig. 4.

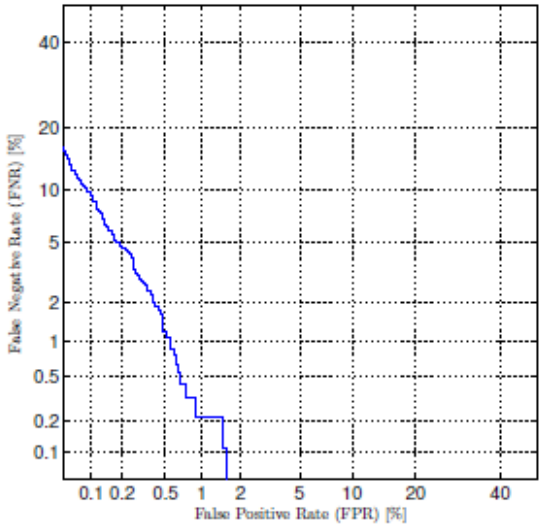

Fig. 3: figure DET curve for best GMM-UBM system

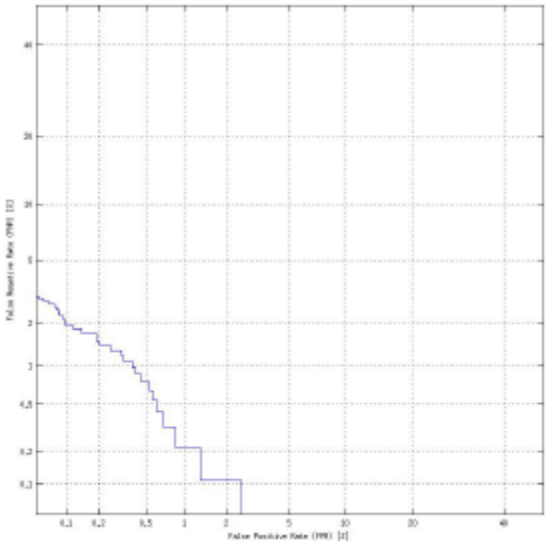

Fig. 4: figure DET curve for best DNN system.

The above results are with NI initialization scheme, up to two layers of depth similar results were observed with RW initialization scheme. However with more than two hidden layers there seemed to be vanishing gradients problem with this initialization. We have not compared with the state-of-the-art i-vector based SR system because of two reasons, (1) the objective of this study is investigate the use of deep neural nets for end to-end SR and not to provide the best possible system for SR and (2) the proposed system architecture is different from the state-of-the-art architecture of front-end and back-end and hence the comparison may not be straightforward.

Table 2: Performance proposed DNN based system

Architecture/EER(in %)	ADAM	ADAM + ADAM
1000R	2.4171	-
4x1000R	0.8658	**0.564**

V. CONCLUSIONS AND FUTURE WORK

It can be seen from the current study that a deep neural network with sufficient depth can outperform traditional SR systems based on GMMs. The resulting system can be easily incorporated on to the low-foot print devices because of the very low computation complexity during the forward pass and the storage capacity for the network parameters. The current study can be extended in many ways. Especially from the input feature dimension, we can use more robust features like contextual MFCC'S rather than current frame alone. Also, bottle neck features can be appended to the capstan features. Source features have been shown to have complementary information to that of systems features. Applying voice activity detection also might improve the results. From the model perspective, a more robust dropout based training can be used so that it can act as a better regularize that the simple l2 weight decay used here. We look forward to incorporate these in our further studies on deep learning for SR task. An investigation into how recurrent neural networks [34] and DNN can be combined to overcome the frame based training can be made.

VI. ACKNOWLEDGEMENTS

Thanks to the members of speech and vision lab for comments and proof-reading.

REFERENCES

[1] J. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review," Signal Processing Magazine, IEEE, vol. 32, no. 6, pp. 74–99, Nov 2015.

[2] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted gaussian mixture models," Digital signal processing, vol. 10, no. 1, pp. 19–41, 2000.

[3] William M Campbell, Douglas E Sturim, and Douglas A Reynolds, "Support vector machines using gmm super vectors for speaker verification," Signal Processing Letters, IEEE, vol. 13, no. 5, pp. 308–311, 2006.

[4] Najim Dehak, Patrick Kenny, R´eda 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.

[5] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Speaker and session variability in gmm-based speaker verification," Audio, Speech, and Language Processing, IEEE Transactions on, vol. 15, no. 4, pp. 1448–1460, 2007.

[6] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," Audio, Speech, and Language Processing, IEEE Transactions on, vol. 15, no. 4, pp. 1435–1447, 2007.

[7] Shou-Chun Yin, Richard Rose, and Patrick Kenny, "A joint factor analysis approach to progressive model adaptation in text-independent speaker verification," Audio, Speech, and Language Processing, IEEE Transactions on, vol. 15, no. 7, pp. 1999–2010, 2007.

[8] Fred Richardson, Douglas Reynolds, and Najim Dehak, "A unified deep neural network for speaker and language recognition," in Proc. of Interspeech, 2015, pp. 1146–1150.

[9] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas A Reynolds, and Reda Dehak, "Language recognition via ivectors and dimensionality reduction.," in INTERSPEECH. Citeseer, 2011, pp. 857–860.

[10] Patrick Kenny, "A small footprint i-vector extractor," in Odyssey 2012-The Speaker and Language Recognition Workshop, 2012.

[11] Yoshua Bengio, "Learning deep architectures for ai," Foundations and trendsR in Machine Learning, vol. 2, no. 1, pp. 1–127, 2009.

[12] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," Signal Processing Magazine, IEEE, vol. 29, no. 6, pp. 82–97, 2012.

[13] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," Signal Processing Magazine, 2012.

[14] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Jorge 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.

[15] 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.

[16] Yan Song, Ruilian Cui, Xinhai Hong, Ian Mcloughlin, Jiong Shi, and Lirong Dai, "Improved language identification using deep bottleneck network," in Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015, pp. 4200–4204.

[17] Pavel Matejka, Le Zhang, Tim Ng, HS Mallidi, Ondrej Glembek, Jeff Ma, and Bing Zhang, "Neural network bottleneck features for language identification," Proc. IEEE Odyssey, pp. 299–304, 2014.

[18] B Yegnanarayana and S Prahallad Kishore, "Aann: an alternative to gmm for pattern recognition," Neural Networks, vol. 15, no. 3, pp. 459–469, 2002.

[19] SP Kishore, B Yegnanarayana, and Suryakanth V Gangashetty, "Online text-independent speaker verification system using autoassociative neural network models," in Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on. IEEE, 2001, vol. 2, pp. 1548–1553.

[20] Sriram Ganapathy, Kyu Han, Samuel Thomas, Mohamed Omar, Maarten Van Segbroeck, and Shrikanth S Narayanan, "Robust language identification using convolutional neural network features," in INTERSPEECH.

[21] Yun Lei, Luciana Ferrer, Aaron Lawson, Mitchell McLaren, and Nicolas Scheffer, "Application of convolutional neural networks to language identification in noisy conditions," Proc. Odyssey-14, Joensuu, Finland, 2014.

[22] Ignacio Lopez-Moreno, Jorge Gonzalez-Dominguez, Oldrich Plchot, David Martinez, Joaquin Gonzalez-Rodriguez, and Pablo Moreno, "Automatic language identification using deep neural networks," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 5337–5341.

[23] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural

networks," in International conference on artificial intelligence and statistics, 2010, pp. 249–256

[24] Geoffrey E Hinton and Ruslan R Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504–507, 2006.

[25] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in Proc. of ICML, 2013, pp. 1139–1147.

[26] David Sussillo, "Random walks: Training very deep nonlinear feed-forward networks with smart initialization," arXiv preprint arXiv:1412.6558, 2014.

[27] Andrew M Saxe, James L McClelland, and Surya Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," arXiv preprint arXiv:1312.6120, 2013.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing humanlevel performance on imagenet classification," arXiv preprint arXiv:1502.01852, 2015.

[29] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th International Conference on Machine Learning (ICML- 10), 2010, pp. 807–814.

[30] Matthew D Zeiler et al., "On rectified linear units for speech processing," in Proc. of ICASSP, 2013, pp. 3517–3521.

[31] Matthew D Zeiler, "ADADELTA: an adaptive learning rate method," arXiv preprint arXiv: 1212.5701, 2012.

[32] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," Proc. of the 3rd International Conference on Learning Representations (ICLR), 2014.

[33] Seyed Omid Sadjadi, Malcolm Slaney, and Larry Heck, "Msr identity toolbox v1.0: A matlab toolbox for speakerrecognition research," Speech and Language Processing Technical Committee Newsletter, November 2013.

[34] Sivanand Achanta, Tejas Godambe, and Suryakanth V Gangashetty, "An investigation of recurrent neural network architectures for statistical parametric speech synthesis," in Proc. Of Interspaced, 2015, pp. 2524–2528.