IMPROVED CONDITIONAL GENERATIVE ADVERSARIAL NET CLASSIFICATION FOR SPOKEN LANGUAGE RECOGNITION

Xiaoxiao Miao[1,2,3], *Ian McLoughlin*[1], *Shengyu Yao*[2,3], *Yonghong Yan*[2,3,4]

[1]School of Computing, The University of Kent, Medway, UK
[2]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics
[3]University of Chinese Academy of Sciences
[4]Xinjiang Key Laboratory of Minority Speech and Language Information Processing,
Xinjiang Technical Institute of Physics and Chemistry, Chinese Academy of Sciences

ABSTRACT

Recent research on generative adversarial nets (GAN) for language identification (LID) has shown promising results. In this paper, we further exploit the latent abilities of GAN networks to firstly combine them with deep neural network (DNN)-based i-vector approaches and then to improve the LID model using conditional generative adversarial net (cGAN) classification. First, phoneme dependent deep bottleneck features (DBF) combined with output posteriors of a pre-trained DNN for automatic speech recognition (ASR) are used to extract i-vectors in the normal way. These i-vectors are then classified using cGAN, and we show an effective method within the cGAN to optimize parameters by combining both language identification and verification signals as supervision. Results show firstly that cGAN methods can significantly outperform DBF DNN i-vector methods where 49-dimensional i-vectors are used, but not where 600-dimensional vectors are used. Secondly, training a cGAN discriminator network for direct classification has further benefit for low dimensional i-vectors as well as short utterances with high dimensional i-vectors. However, incorporating a dedicated discriminator network output layer for classification and optimizing both classification and verification loss brings benefits in all test cases.

***Index Terms*—** language identification, conditional generative adversarial net, deep bottleneck features, i-vector

1. INTRODUCTION

Language identification (LID) is an aspect of speech preprocessing typically followed by automatic speech recognition, or by language-specific post-processing. The main task in LID is to identify which language is being spoken using information extracted from the speech signal, and do so with speed and accuracy. Current mainstream systems are mainly i-vector based approaches [1], and these obtain state-of-the-art performance for LID. An i-vector is a low-dimensional representation of an arbitrary length utterance, and serves as a fixed-length feature vector to represents the useful information within that utterance.

Conventional Gaussian Mixture Model (GMM) i-vector-based system can be divided into two levels. At the front-end feature level, i-vectors are extracted from a GMM supervector [2]. Since i-vectors are learned in an unsupervised fashion without any specific label information, they need to be trained with language labels at the back-end. For example, using support vector machines (SVM), logistic regression classifiers and probabilistic linear discriminant analysis (PLDA). While traditional acoustic features can be used for classification, they are not particularly robust to noise, and performance in short-duration utterance test conditions or with highly confusable dialects is often poor.

Thanks to the development of Deep Nerural Networks (DNN) [3], DNN-based LID methods dramatically improved the performance of LID system. In the feature domain, some researchers [4, 5], used a phoneme dependent deep bottleneck feature (DBF) extractor, obtained from the lower layers of a deep bottleneck network (DBNs) that has been well trained for an automatic speech recognition (ASR) task. In the model domain, for both LID and speaker recognition, novel total variability (TV) modeling methods have been proposed based on phonetic-aware DNNs [6, 7], In these studies, instead of GMM posterior probability, DNN output posteriors are exploited to obtain sufficient statistics. Thus the DBF DNN i-vector [8, 9] has been proposed. This combines a DBF for extracting robust features with the posteriors of the DNN for improved model capability, obtaining more and better phoneme information for the TV modeling, further enhancing LID performance. These advances clearly demonstrate the relevance of phonetic-aware ASR-trained DNNs to LID,

Generative adversarial networks (GANs) [10] have recently become very popular for signal generation processing

Thanks to China Scholarship Council for funding.This work is partially supported by the National Key Research and Development Program (Nos. 2016YFB0801203, 2016YFB0801200).

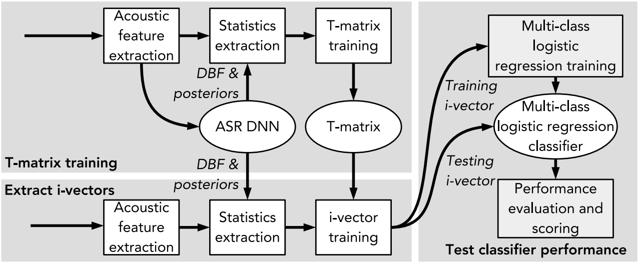

Acoustic feature extraction
Statistics extraction
T-matrix training
DBF & posteriors
ASR DNN
T-matrix
T-matrix training
Extract i-vectors
DBF & posteriors
Acoustic feature extraction
Statistics extraction
i-vector training
Multi-class logistic regression training
Training i-vector
Multi-class logistic regression classifier
Testing i-vector
Performance evaluation and scoring
Test classifier performance

Fig. 1. Block diagram of DBF i-vector baseline.

in areas such as image generation [11], image-to-image translation [12, 13, 14] and speech enhancement [15]. A GAN consists of a generator that produces fake data from noise, and a discriminator to distinguish between fake and real data. The training process for GANs involves updating the generator and discriminator in turn, causing the generated fake data to become more and more similar to the real data.

GANs are still uncommon in LID research, with the notable exception of Shen et.al [16] who studied and used conditional generative adversarial nets (cGAN) as a classifier for spoken language identification with limited training data. Compared to the GAN, instead of only inputting noise to the generator, his cGAN-based classifier used real data as conditional information, and for the output of the discriminator, language labels were also applied. In that paper, the cGAN was trained directly as a classifier, although most previous research on GANs was generative (and usually for images). While performance was good, Shen et.al used balanced dataset sizes. In fact LID systems often observe performance degradation with unbalanced training, however standard LID data sets are often unbalanced, especially due to data collection difficulties with resource-constrained languages. In this paper we therefore explore cGAN performance with unbalanced but standard training data (adopting the whole of NIST LRE07). We then propose two new approaches to the architecture. We combine DBF DNN i-vector and cGAN to build a new LID model. First, we use DBF DNN to extract i-vectors, and then the i-vectors are sent to cGAN for classification. We then take advantage of the cGAN to perform classification, while integrating the language identification and Fake/Real signal verification supervisory learning in the cGAN parameter optimization over two output layers, by minimizing two loss functions. The aim is overcome the challenge of effective generalization despite unbalanced training data. The paper is organized as follows: Section 2 describes the baseline systems and the proposed changes; Section 3 reports experimental results and then Section 4 concludes our work and discuss future issues.

2. METHODS

2.1. DBF DNN i-vector

The DBF DNN i-vector baseline that we adopt [8] needs a pre-trained DNN for an ASR task. Fig. 1 shows the overall procedure. The output of the ASR DNN is phoneme states for each input frame. We use the ASR DNN output posteriors and BNFs to extract i-vectors, then back-end classification using multi-class logistic regression training. From this structure, sufficient statistics can be computed from the BNFs and the posterior probabilities of the ASR DNN:

$$N_k(s) = \sum_{t=1}^{T_s} p(k \mid x_{s,t}, \phi) \tag{1}$$

$$F_k(s) = \sum_{t=1}^{T_s} p(k \mid x_{s,t}, \phi) y_{s,t} \tag{2}$$

$$S_k(s) = \sum_{t=1}^{T_s} p(k \mid x_{s,t}, \phi) y_{s,t} {y_{s,t}}^{\top} \tag{3}$$

Where ϕ represents the parameters of the ASR DNN, $p(k \mid)$ corresponds to k class posteriors from the ASR DNN, $x_{s,t}$ is the acoustic feature of the t-th frame of utterance s that has L frames, $y_{s,t}$ is the DBF vector from the t-th frame of utterance s. These sufficient statistics are all that are needed to train subspace T and extract the i-vector.

2.2. cGAN-classifier

The cGAN-classifier proposed by Shen et al. [16] comprises a discriminator D and a generator network, G. The inputs of the generator network are noise and conditional information in the form of real i-vectors (not class labels). Then the generator effectively transforms features from real input samples to generated features. These generated samples are then used for discriminator network optimization. The baseline discriminator has just one output layer comprising a single binary real/fake feature output node and remaining nodes corresponding to language label outputs.

2.3. Proposed system

Fig. 2 shows the framework of the combined DNN DBF and cGAN classifier. The front-end, depicted on the left, uses a single DNN trained for ASR to extract bottleneck features and estimate posteriors to extract i-vectors. The back-end, depicted on the right, is a cGAN-classifier similar to the system of Shen et al. described in Section 2.2, but with the main difference that our discriminator network has two separate output layers with separate loss functions (Shen et al. [16] allowed for the possibility of two separate loss functions, but their experiments used common features and a single loss).

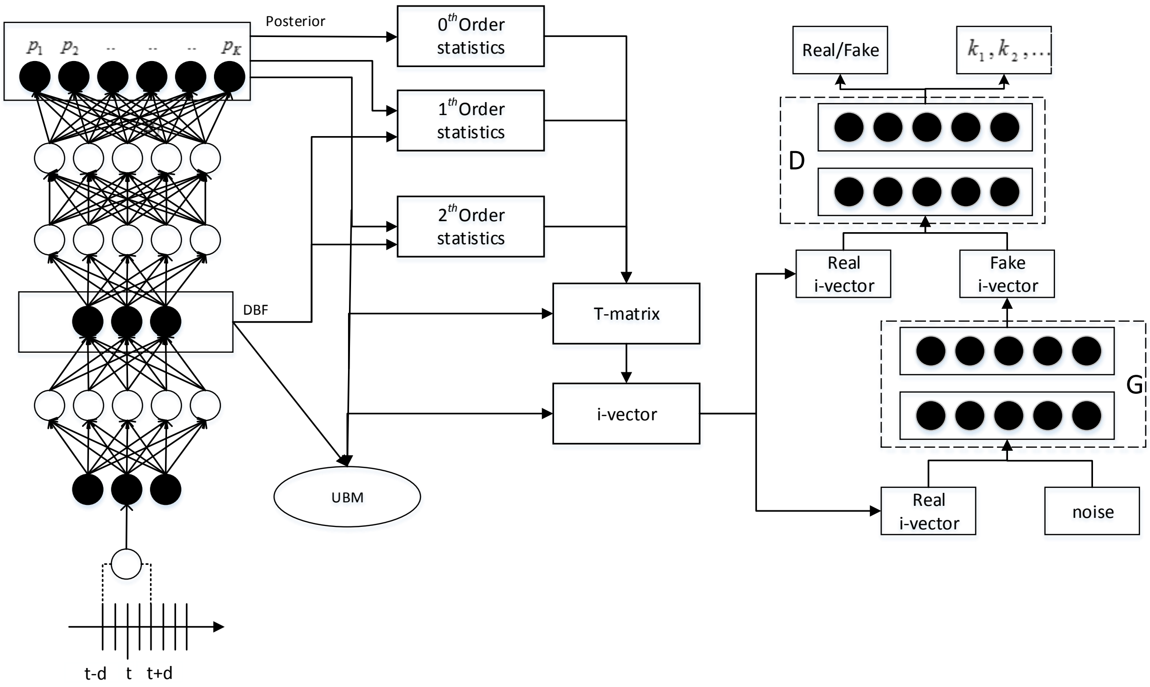


Fig. 2. Block diagram of the proposed LID system.

The two new output layers include one with sigmoid activation for Real/Fake signal verification, and one softmax to output categorical class labels (by contrast, the architecture of Shen et al. had a single softmax output layer). The aim is to separately optimize for both Real/Fake identification and class labels. The two losses are computed as follows;

$$\begin{aligned} \underset{G}{min}\,\underset{D}{max}\,V_v &= E[logD(c, G(z,c))] + \\ &\quad E[log(1 - D(G(z,c)))] \qquad (4) \\ \underset{G}{min}\,\underset{D}{max}\,V_i &= E[logD(k \mid c, G(z,c))] + \\ &\quad E[log(1 - D(k \mid G(z,c)))] \qquad (5) \end{aligned}$$

where V_v is the objective function of the Real/Fake signal verification, V_i is the objective function of the language identification. $G(z, c)$ represents the probability of the generator from noise z and real data c, $D(c, G(z, c))$ represents the probability of the discriminator from real data c and fake data $G(z, c)$. $D(k \mid c, G(z, c))$ means the k-th categorical probability of the discriminator from real data and fake data. The final objective function is defined as $V = V_v + V_i$.

3. EVALUATION

3.1. Database and Experimental Setup

3.1.1. Database

The ASR DNN is trained on roughly 1000 hours of clean English telephone speech from Fisher [17]. For the LID task, we conducted experiments using NIST LRE07 which is the closed-set language detection spanning 14 languages: Arabic (AR), Bengali (BE), English (EN), Farsi (FA), Russian (RU), German (GE), Hindustani (HI), Japanese (JA), Korean (KO), Mandarin (MA), Spanish (SP),Tamil (TA), Thai (TH) and Vietnamese (VI). The experiments used the LID training corpus including Callfriend datasets, LRE03, LRE05, SRE08 datasets, and development data for LRE07. The total training data is about 88822 utterances of length 120s or less. Specially, it should be noted that the training data set sizes are unequal, e.g. the English, Chinese, Spanish as the top three corpuses account for around 43%, 13%, 7% of the total data respectively, whereas the Bengali corpus makes up only about 0.5% of the total. The experimental LID test corpus is the NIST LRE07 test dataset separated into 30s, 10s and 3s conditions. Each condition has 2158 utterances.

3.1.2. Experimental Setup

A nine-layer ASR DNN is trained with cross entropy, from a 40×11 input layer (40-dimensional PLP features concatenation over a context of the current frame with the preceding and following 5 frames). Input is followed by linear discriminant analysis (LDA). The hidden layers have 3000 nodes followed by Pnorm nonlinear activation (with 300 nodes) and normalization, except that the bottleneck layer (the fourth hidden layer) has 390 nodes; the output of Pnorm is 39 dimensional and the BNFs are extracted from the subsequent normalization. The output layer has 5560 nodes and the dimension of the i-vector is 600. The experiments for ASR DNN and i-vector extraction are all carried out using Kaldi [18].

Table 1 details the configuration of the cGAN-classifier networks used in this work for the reduced 49 dimensional i-vector. This can be compared to the structures in [16, 19], particularly in the overall depth and the structure of the two discriminator output layers.

During training, the parameters of the discriminator network are trained, with all but the final layers shared between the Real/False output objective function and the classification objective function. During testing, we keep only the classification output; we then only use real i-vectors as input to the discriminator. The networks are trained using the adaptive subgradient method (Adagrad) optimizer, with a mini-batch size of 128 and learning rate of 0.0005 over 500 epochs. The cGAN parts of the investigation are performed using the Keras [20] open source toolkit.

3.2. Experimental results

Two sets of experiments were run on the LRE07 corpora, using 600-dimensional and 49-dimensional i-vectors respectively. For both sets of experiments, i-vectors were extracted using the DBF DNN i-vector method. The Kaldi recipe produced 600 dimensional i-vectors, reduced to 49-dimensions using LDA for the 49-dimensional evaluation. Table 2 summarizes the results for different dimensions. The first three rows of each table correspond to baseline systems; a logistic regression classifier, a cGAN followed by classifier and Network_D, which is the system described in [16] where the discriminator performs classification directly classifier. These both have single loss optimization. The next two sets of experiments show the performance of the cGAN-classifier and Network_D with two losses (indicated with subscript $_2$).

Table 1. The discriminator and generator networks used for cGAN-classifier, conv. is a convolutional layer, FC is a fully connected layer. The italicised outputs use tanh function.

Disciminator, D		**Generator, G**	
Layer config.	**Output**	**Layer config.**	**Output**
FC, real i-vector	*49*	FC, real i-vector	*49*
FC, fake/real i-vector	*49*	FC, noise	*100*
Merge FC1 & 2	*98*	Merge FC1 & 2	*149*
FC (1024)	1024	FC (1024)	1024
FC (128×7×7)	128-7-7	FC (128×7×7)	128-7-7
Reshape	128-7-7	Reshape	128-7-7
		Batch normalization	
3×3 conv. 128,	*128-7-7*	2×2 up-sample	128-14-14
FC (1024)	*1024*	5×5 conv. 64	*64-14-14*
Output 1: sigmoid(1)	15	2×2 up-sample	64-28-28
Output 2: softmax(14)		5×5 conv. 1	*1-28-28*
Output(15)	15	Output: FC(49)	*49*

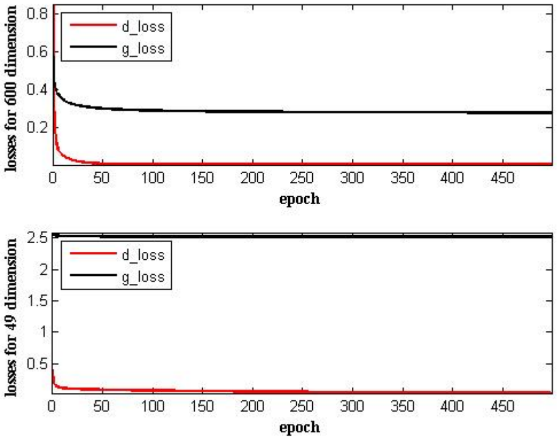


Fig. 3. The training loss for cGANs with 600- (top) and 49-dimensional (bottom) i-vectors.

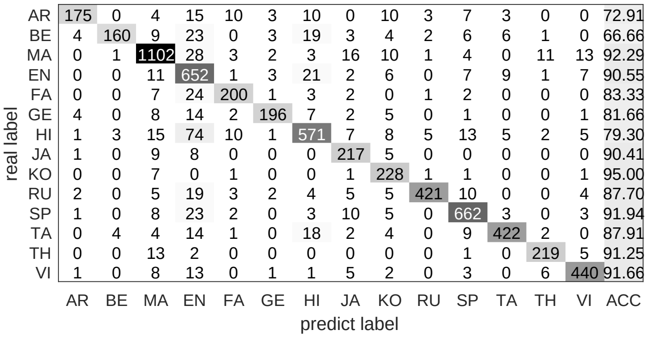
The results firstly indicate that Network_D is better than the cGAN-classifier, no matter what dimension is. The reason for this is based on the fact that the generator aims to generate fake features that have similar characteristics to the real features; the ideal situation being that the fake features are indistinguishable from true features, but cannot be better than true features. However, the generated features used as interference can improve the generalization of the discriminator.

Both network_D$_2$ and the cGAN-classifier$_2$ yield a gain over systems with one loss, for both sets of experiments. For the 49 dimension task, compared with the cGAN-classifier with one loss, cGAN-classifier$_2$ decreased EER at 30s, 10s, 3s test durations by 21.69%, 14.91%, 8%, respectively. Similarity, compared with the network_D one loss, network_D$_2$ decreased EERs at 30s, 10s, 3s test durations by 25%, 12.71%, 3.14%, respectively. The improvement was due to their being two losses. For the full 600 dimension task, we can see the same trend between having one loss and two losses. This shows that using two losses, including Real/Fake signal verification and language identification, is effective.

In comparison to the logistic regression classifier, for the 49 dimension task, network_D$_2$ decreased EERs at 30s, 10s, 3s test durations by 40%, 24.85%, 20.86%, respectively. Unfortunately, all GAN-based systems performed worse than the logistic regression classifier for the 600 dimension task. However, when the i-vectors are reduce from 600 to 49 dimension via LDA, the resulting low-dimensional i-vectors are more

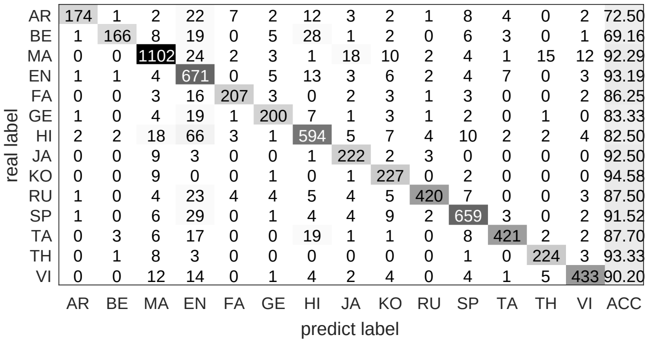
Table 2. performance results for 49 and 600 dimensional i-vector experiments.

Classifier	**No. of losses**	3s			10s			30s		
49 dimensional i-vector results:		**C_avg**	**EER**	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**	**ER**
Logistic regression classifier		16.82	10.88	28.68	5.47	3.38	9.64	1.46	1.15	2.78
cGAN-classifier	One	17.34	9.87	28.82	6.20	3.42	10.52	1.77	1.06	3.01
Network_D	One	16.00	9.22	26.51	5.35	3.05	8.62	1.42	0.84	2.36
cGAN-classifier$_2$	Two	15.15	9.08	25.02	5.12	2.91	8.02	1.57	0.83	2.55
Network_D$_2$	Two	15.13	8.61	25.44	4.68	2.54	7.41	1.36	0.69	2.09
600 dimensional i-vector results:		**C_avg**	**EER**	**ER**	**C_avg**	**EER**	**ER**	**C_avg**	**EER**	**ER**
Logistic regression classifier		14.24	7.73	23.73	4.94	2.54	8.06	1.44	0.78	2.46
cGAN-classifier	One	17.41	10.47	28.68	8.88	4.63	10.57	2.97	1.06	2.97
Network_D	One	17.58	9.82	28.17	8.78	4.21	13.44	2.90	1.52	4.87
cGAN-classifier$_2$	Two	16.91	10.19	28.08	8.29	4.35	13.62	3.21	1.85	5.65
Network_D$_2$	Two	16.78	9.87	27.99	7.15	4.07	11.91	2.56	1.62	4.26



real label \ predict label	AR	BE	MA	EN	FA	GE	HI	JA	KO	RU	SP	TA	TH	VI	ACC
AR	175	0	4	15	10	3	10	0	10	3	7	3	0	0	72.91
BE	4	160	9	23	0	3	19	3	4	2	6	6	1	0	66.66
MA	0	1	1102	28	3	2	3	16	10	1	4	0	11	13	92.29
EN	0	0	11	652	1	3	21	2	6	0	7	9	1	7	90.55
FA	0	0	7	24	200	1	3	2	0	1	2	0	0	0	83.33
GE	4	0	8	14	2	196	7	2	5	0	1	0	0	1	81.66
HI	1	3	15	74	10	1	571	7	8	5	13	5	2	5	79.30
JA	1	0	9	8	0	0	0	217	5	0	0	0	0	0	90.41
KO	0	0	7	0	1	0	0	1	228	1	1	0	0	1	95.00
RU	2	0	5	19	3	2	4	5	5	421	10	0	0	4	87.70
SP	1	0	8	23	2	0	3	10	5	0	662	3	0	3	91.94
TA	0	4	4	14	1	0	18	2	4	0	9	422	2	0	87.91
TH	0	0	13	2	0	0	0	0	0	0	1	0	219	5	91.25
VI	1	0	8	13	0	1	1	5	2	0	3	0	6	440	91.66

Fig. 4. LID confusion matrix for 49-dimension Network_D.



real label \ predict label	AR	BE	MA	EN	FA	GE	HI	JA	KO	RU	SP	TA	TH	VI	ACC
AR	174	1	2	22	7	2	12	3	2	1	8	4	0	2	72.50
BE	1	166	8	19	0	5	28	1	2	0	6	3	0	1	69.16
MA	0	0	1102	24	2	3	1	18	10	2	4	1	15	12	92.29
EN	1	1	4	671	0	5	13	3	6	2	4	7	0	3	93.19
FA	0	0	3	16	207	3	0	2	3	1	3	0	0	2	86.25
GE	1	0	4	19	1	200	7	1	3	1	2	0	1	0	83.33
HI	2	2	18	66	3	1	594	5	7	4	10	2	2	4	82.50
JA	0	0	9	3	0	0	1	222	2	3	0	0	0	0	92.50
KO	0	0	9	0	0	1	0	1	227	0	2	0	0	0	94.58
RU	1	0	4	23	4	4	5	4	5	420	7	0	0	3	87.50
SP	1	0	6	29	0	1	4	4	9	2	659	3	0	2	91.52
TA	0	3	6	17	0	0	19	1	1	0	8	421	2	2	87.70
TH	0	1	8	3	0	0	0	0	0	0	1	0	224	3	93.33
VI	0	0	12	14	0	1	4	2	4	0	4	1	5	433	90.20

Fig. 5. LID confusion matrix for 49-dimension Network_D$_2$.

compact and network_D$_2$ seems to be particularly effective when trained by the compact representation.

To further explore the effect of different dimensions on cGAN, we also plot the training losses in Fig. 3. The results show how the training losses of the discriminator and generator, for both 49 and 600 dimensional i-vectors, evolve over 500 epochs. The discriminator loss is consistently lower than that of the generator, which is reasonable since previous works showed that the discriminator converges much easier than generator. For the 49 dimension cGAN, generator losses change very little and are effectively flat after 200 epochs, although the discriminator losses continue to reduce. The generator losses also imply that i-vectors generated in the 49 dimensional system are less similar to real i-vectors than those in the 600 dimensional system. While this seems undesirable, it may actually have a beneficial effect in encouraging the generalization ability of the discriminator. This is something that may be interesting to explore further in future.

Finally, we can explore the LID confusion matrix for all 14 languages for Network_D with one loss and two losses in Figs. 4 and 5 respective. We can observe that the more data that a certain language has, the greater its LID accuracy tends to be. Comparing EN, MA and SP to BE, the effect is pronounced. This is unsurprising since the network better optimizes the first three languages during training. Conversely, smaller languages such as BE are less well trained. We can also see that network_D$_2$ achieves better accuracy than network_D for almost languages, but particularly for the smaller language sets like BE.

4. CONCLUSION AND FUTURE WORK

This paper has presented a new LID model that combines DBF DNN i-vector and cGAN approaches to optimize cGAN parameters by optimizing for both LID labels and Fake/Real signal verification supervision. This involves creating separate output layers within the discriminator which are optimzed according to separate loss functions. Experiments on the LRE07 dataset show that this new LID structure is effective.

The results of high dimensional i-vector tests in our experiments demonstrated comparatively degraded performance compared to the lower dimensional i-vector tests, therefore future work will explore the sensitivity of different dimensional i-vectors to the new LID model, as well the effect of generator accuracy on discriminator generalization ability. We also aim to explore whether the cGAN performance gain is obtained from similar information that is lost through the LDA dimensionality reduction operation, since this seems to be a reasonable explanation for the effect.

5. REFERENCES

[1] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas Reynolds, and Reda Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Twelfth annual conference of the international speech communication association*, 2011.

[2] William M Campbell, Douglas E Sturim, and Douglas A Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE signal processing letters*, vol. 13, no. 5, pp. 308–311, 2006.

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

[4] Bing Jiang, Yan Song, Si Wei, Meng-Ge Wang, Ian McLoughlin, and Li-Rong Dai, "Performance evaluation of deep bottleneck features for spoken language identification," in *Chinese Spoken Language Processing (ISCSLP), 2014 9th International Symposium on*. IEEE, 2014, pp. 143–147.

[5] Yan Song, Xinhai Hong, Bing Jiang, Ruilian Cui, Ian McLoughlin, and Li-Rong Dai, "Deep bottleneck network based i-vector representation for language identification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[6] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Mitchell 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.

[7] Patrick Kenny, Vishwa Gupta, Themos Stafylakis, P Ouellet, and J Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[8] Fred Richardson, Douglas Reynolds, and Najim Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[9] Fred Richardson, Douglas Reynolds, and Najim Dehak, "A unified deep neural network for speaker and language recognition," *arXiv preprint arXiv:1504.00923*, 2015.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[11] Guo-Jun Qi, "Loss-sensitive generative adversarial networks on Lipschitz densities," *arXiv preprint arXiv:1701.06264*, 2017.

[12] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[13] Alec Radford, Luke Metz, and Soumith Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[14] Junbo Zhao, Michael Mathieu, and Yann LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[15] Zeng-Xi Li, Li-Rong Dai, Yan Song, and Ian McLoughlin, "A conditional generative model for speech enhancement," *Circuits, Systems, and Signal Processing*, pp. 1–18, 2018.

[16] Peng Shen, Xugang Lu, Sheng Li, and Hisashi Kawai, "Conditional generative adversarial nets classifier for spoken language identification," in *Proc. of Interspeech*, 2017.

[17] Christopher Cieri, David Miller, and Kevin Walker, "The Fisher corpus: a resource for the next generations of speech-to-text.," in *LREC*, 2004, vol. 4, pp. 69–71.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The Kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[19] Jost Tobias Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *arXiv preprint arXiv:1511.06390*, 2015.

[20] François Chollet et al., "Keras," 2015.