DISCRIMINATIVE AUTOENCODERS FOR SPEAKER VERIFICATION

Hung-Shin Lee[1,2] *Yu-Ding Lu*[3] *Chin-Cheng Hsu*[2] *Yu Tsao*[3] *Hsin-Min Wang*[2] *Shyh-Kang Jeng*[1]

[1] Department of Electrical Engineering, National Taiwan University, Taiwan
[2] Institute of Information Science, Academia Sinica, Taiwan
[3] Research Center for Information Technology Innovation, Academia Sinica, Taiwan

ABSTRACT

This paper presents a learning and scoring framework based on neural networks for speaker verification. The framework employs an autoencoder as its primary structure while three factors are jointly considered in the objective function for speaker discrimination. The first one, relating to the sample reconstruction error, makes the structure essentially a generative model, which benefits to learn most salient and useful properties of the data. Functioning in the middlemost hidden layer, the other two attempt to ensure that utterances spoken by the same speaker are mapped into similar identity codes in the speaker discriminative subspace, where the dispersion of all identity codes are maximized to some extent so as to avoid the effect of over-concentration. Finally, the decision score of each utterance pair is simply computed by cosine similarity of their identity codes. Dealing with utterances represented by i-vectors, the results of experiments conducted on the male portion of the core task in the NIST 2010 Speaker Recognition Evaluation (SRE) significantly demonstrate the merits of our approach over the conventional PLDA method.

Index Terms— autoencoders, speaker verification, discriminative training, neural networks, PLDA

1 Introduction

Even though the methodology of deep neural networks (DNNs) has seemingly become more and more popular in the field of speaker recognition and obtained some gains in performance [1, 2, 3, 4, 5, 6], either total variability modeling (i-vector) [7] or probabilistic linear discriminant analysis (PLDA) [8], as well as their modifications, are still indispensable and robust ingredients in most of current speaker verification systems. The aim of *i-vector* is to represent variable-length speech signals by fixed-size vectorial tokens while the session/channel variabilities induced by various sources are compensated and the speaker characteristics are abundantly and ulteriorly preserved [9]. Given two i-vectors, the task of *PLDA* is to linearly discriminate between speakers in a low-rank subspace and give a reasonable metric to measure their decision score in a probabilistic sense [10, 11].

Actually, going through the latest three ICASSP proceedings (2014-16), more than three-fourths of papers dealing with speaker verification use PLDA as one of their scoring backends, among which there are much fewer efforts to either develop their own competitive algorithms for backend speaker discrimination or make some improvements to PLDA by standing on its shoulders. For example, Rohdin *et al.* gave a discriminative PLDA training algorithm, where some constraints are imposed on the derivation of the speaker variability matrix [12]. Similar to the work by Lee *et al.* [13], Cumani and Laface employed pairwise support vector machines (SVMs) to efficiently classify pairs of i-vectors as belonging or not to the same speaker even with large-scale datasets [14]. Nautsch *et al.* proposed a PLDA-alike approach with restricted Boltzmann machines (RBM), which aims at suppressing channel effects and recovering speaker-discriminative information on a small dataset [15]. Most recently, Heigold *et al.* used DNNs and long short term memory (LSTM) to represent utterances and directly map each trial set of utterances to a decision score for verification [16].

In this paper, we replace the role of PLDA with an *autoencoder* and tweak its objectives for speaker discrimination. The autoencoder is a symmetric neural network that is trained to approximately copy its input to the output [17]. Besides the reconstruction error, which makes the autoencoder analogous to a generative model that benefits to unsupervisedly learn most salient and useful properties of the data, two more objective functions are concerned in our proposed framework. They attempt to ensure that utterances spoken by the same speaker would have similar identity codes (i-codes) in the speaker-discriminative subspace represented by the middlemost hidden layer, where the scatterness of all i-codes are also maximized to some extent to avoid the effect of over-concentration. Finally, the decision score of each utterance pair is simply computed by cosine similarity of their i-codes. To our best knowledge, despite the autoencoder has been widely applied to many speech processing tasks, such as speech enhancement [18, 19], acoustic novelty detection [20], and reverberant speech recognition [21], much less papers used it directly for speaker recognition.

Most important of all, our contributions are two-fold. First, we present a kind of neural network-based discriminant analysis, which consumedly and nonlinearly extends the capability of PLDA. Second, our proposed model is immune to

computational intractability, e.g., matrix inversion when the training set becomes very large [8, 11].

Our proposed framework, its objectives and analogy with PLDA, and its realization and evaluation, are given and reported in the remainder of this paper.

2 Objectives

2.1 Probabilistic linear discriminant analysis

The PLDA model assumes that the j-th utterance (i-vector) of the ith speaker is described by the following process [11]:

$$\mathbf{x}_{ij} = \underbrace{\boldsymbol{\mu} + \mathbf{F}\mathbf{h}_i}_{\text{Identity}} + \underbrace{\mathbf{G}\mathbf{w}_{ij} + \boldsymbol{\epsilon}_{ij}}_{\text{Noise}}, \tag{1}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\epsilon}_{ij}$ denote the global mean and the residual following a Gaussian distribution with zero mean and diagonal covariance $\boldsymbol{\Sigma}$, respectively. By (1), each $\mathbf{x}_{ij}$ is factorized into two parts. In the identity part, the matrix $\mathbf{F}$ denotes the subspace, where the utterances that belong to the same speaker would have the same projective location or speaker identity, characterized by a hidden variable $\mathbf{h}_i$. As for the noise part, all other information irrelevant to speaker discrimination is thrown into the subspace $\mathbf{G}$ and locates in a noise factor $\mathbf{w}_{ij}$. Both $\mathbf{h}_i$ and $\mathbf{w}_{ij}$ are standard Gaussian distributed.

The model parameters $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{F}, \mathbf{G}\}$, can be estimated by the EM algorithm, and the likelihood of a set of i-vectors can be used as the decision score for speaker verification.

2.2 Three PLDA-inspired objective functions

By the framework of PLDA, we have two kinds of sights. First, PLDA is a generative model, equipped with an inference procedure for computing the distribution of latent variables given an observation, and providing a generative procedure for stochastically generating a copy of an observation. Second, given two utterances $\mathbf{x}_{11}$ and $\mathbf{x}_{21}$ that belong to different speakers, PLDA seems *not* to explicitly ensure that their speaker identities $\mathbf{h}_1$ and $\mathbf{h}_2$ will be consumedly different. This might increase the number of false alarms in verification tasks. Therefore, we design three kinds of objective functions for speaker discrimination in order to take the essence of PLDA and make up for its deficiency.

2.2.1 The reconstruction error

Suppose our proposed model $\mathcal{M}$ contains a pair of deterministic mappings $f(\cdot)$ and $g(\cdot)$, which are responsible for latent variable inference and observation generation in the terminology of Bayesian inference, respectively. Given a set of training data $\mathcal{X}$ ready to go through the inference-generation process by $\mathcal{X} \xrightarrow{f} \mathcal{H} \xrightarrow{g} \mathcal{X}$, where $\mathcal{H}$ is the internal representations residing in a latent subspace, the average reconstruction error based on the residual sum of squares between $\mathbf{x} \in \mathcal{X}$ and its reconstruction $\mathbf{y} = g(f(\mathbf{x}))$ is given by

$$F_r(\mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} \|\mathbf{y} - \mathbf{x}\|_2^2, \tag{2}$$

where $\|\cdot\|_2$ is the 2-norm operator and $|\mathcal{X}|$ is the sample size.

Like a copy machine, $\mathcal{M}$ is usually restricted in ways that allow it *to copy only approximately, and to copy only input that resembles the training data. Because the model is forced to prioritize which aspects of the input should be copied, it often learns useful properties of the data.*

2.2.2 The speaker identity loss

According to the above italic description quoted from [17], we can suppose $\mathcal{H}$ contains salient and useful features, certainly, including untreated speaker characteristics, by minimizing (2). Without loss of generality, we split $\mathcal{H}$ into two parts, $\mathcal{H}_s$ and $\mathcal{H}_n$, where $\mathcal{H}_s$ is supposed to contain all of information for speaker discrimination, and $\mathcal{H}_n$ possesses the residual content. Suppose $\mathcal{H}_s = \{\mathcal{H}_{s1}, \ldots, \mathcal{H}_{sm}\}$, where $\mathcal{H}_{si}$ corresponds to those training data that belong to the i-th speaker of m training speakers, and the loss function related to speaker identity is described as follows:

$$F_s(\mathcal{H}_s) = \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{|\mathcal{H}_{si}|} \sum_{\mathbf{h} \in \mathcal{H}_{si}} \left\|\mathbf{h} - \overline{\mathbf{h}}_{si}\right\|_2^2 \right), \tag{3}$$

where $\overline{\mathbf{h}}_{si}$ denotes the empirical mean vector of $\mathcal{H}_{si}$. Apparently, the term in parentheses in (3) measures the average within-speaker *compactness*. Therefore, to minimize (3) implies to increase the similarity score between two utterances that belong to the same speaker if the score metric is 2-norm-related. In this way, the number of false rejects in verification tasks will conceivably decrease to some extent.

2.2.3 The internal dispersion

The last objective function to be minimized is about the dispersion of *total* internal representations, which is given by

$$F_d(\mathcal{H}_s) = -\frac{1}{|\mathcal{H}_s|} \sum_{\mathbf{h} \in \mathcal{H}_s} \left\|\mathbf{h} - \overline{\mathbf{h}}_s\right\|_2^2, \tag{4}$$

where $\overline{\mathbf{h}}_s$ denotes the empirical mean vector of $\mathcal{H}_s$. There are two reasons to support the necessity of (4). First, the minimum of $F_s(\mathcal{H}_s)$ in (3) might naturally become *zero* when the model makes all of $\mathbf{h} \in \mathcal{H}_s$ turn out to be the same. Second, analogically speaking, the goal of linear discriminant analysis (LDA) is to maximize the Rayleigh quotient given by the *total* scatterness over the within-class scatterness [22]. The expression is equivalent to the traditional attempt to maximize the between-class scatterness in the whitened space [23]. Therefore, the cooperation of (3) and (4) can keep a distance

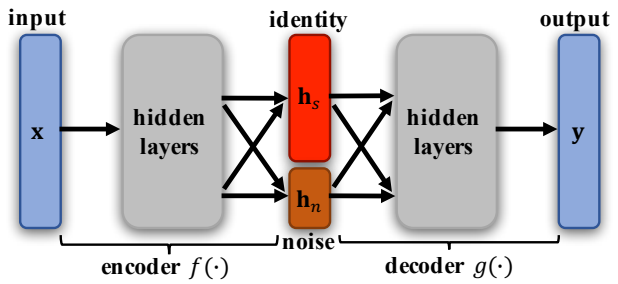

Fig. 1. The architecture of DCAE.

between utterances that belong to different speakers so as to reduce the errors caused by false alarms.

Finally, by combining (2), (3), and (4), the new goal of the training process is to find an optimal $\mathcal{M}$ by minimizing

$$F_r(\mathcal{X}) + \alpha\left(\beta F_s(\mathcal{H}_s) + (1-\beta)F_d(\mathcal{H}_s)\right) + \lambda\|\mathcal{M}\|_2^2, \quad (5)$$

where $\alpha > 0$ controls the relative importance of the last two objective functions to the reconstruction error, $\beta \in [0, 1]$ adjusts the ratios between the speaker identity loss and the internal dispersion that we want to emphasis, and λ is a regularization parameter that controls the complexity of the model. The treatment of α and β is akin to that in ElasticNet [24].

3 Realizations

We use an artificial neural network-based autoencoder, called the discriminative autoencoder (DCAE), to realize $\mathcal{M}$ along with the objective function (5). In contrast with another work that trains the autoencoder in a discriminative way in [25], the cost function of DCAE is not only determined on the output layer but also highly affected by the middlemost code, i.e., the internal representation. The detailed structure and definition of the autoencoder can be referred to [26], [27], and [17].

The architecture of DCAE is depicted in Figure 1. It maps an input $\mathbf{x}$ to an output $\mathbf{y}$ through an internal representation or code $\mathbf{h}$, which is split into $\mathbf{h}_s$ and $\mathbf{h}_n$ as described in Section 2.2.2. The encoder $f(\cdot)$ and the decoder $g(\cdot)$ are built up of full-connected hidden layers and their corresponding weights and biases. The activation flows through the hidden layers and the code layer by means of the hyperbolic tangent function, until it reaches the final layer and linearly gives the output.

To train DCAE, a back-propagation process is implemented from the output layer down through the whole DCAE to adjust all parameters. The gradient of each parameter in $\mathcal{M}$ can be easily derived by partial differentiation on the cost function (5), so that the model can be iteratively updated by using an optimizer based on gradient descent.

To generate the decision score $s_{1,2}$ of $\mathbf{x}_1$ and $\mathbf{x}_2$ for speaker verification, we simply use the cosine similarity of their i-codes, i.e., $(\mathbf{h}_{s1} \cdot \mathbf{h}_{s2}) / \|\mathbf{h}_{s1}\| \|\mathbf{h}_{s2}\|$.

4 Experiments and Results

4.1 Experiment setup

All the experiments were carried out on the male portion of the core task in NIST SRE-10 (core-core/condition-5) for evaluation and NIST SRE-08 (short2-short3/condition-6) for model validation based on the equal error rate (EER), where each session is an excerpt of five-minutes telephone speech [28, 29]. With the frame length of 25 ms and the frame shift of 10 ms, speech parameters were represented by a 60-dimensional feature vector of Mel-frequency cepstral coefficients (MFCCs) with first and second derivatives appended using a 2-frame window, followed by data distribution-based feature warping with a 300-frame window in order to compensate for the effects of environmental mismatch [30].

A gender-dependent UBM consisting of 1,024 Gaussian components with diagonal covariance matrices, the total variability model (for i-vectors) with rank 400, the PLDA model, as well as our proposed DCAE model, were trained with 8,511 utterances spoken by 413 speakers, drawn from NIST SRE-04 and SRE-05, Switchboard II-Phase 1, 2 and 3, and Switchboard Cellular Part 1 and 2. The i-vectors were length-normalized prior to PLDA and DCAE training [31].

For DCAE, we set the tunable numbers of nodes in the identity and noise layers to 300 and 100, respectively. The number of hidden nodes is 400 for all hidden layers. Initial weights are uniformly sampled by the Glorot process that is fit for the $\tanh$ activation function [32]. The adaptive gradient algorithm adopted to update the model parameters is AdaGrad, which scales the learning rate by dividing with the square root of accumulated squared gradients [33].

4.2 Results compared with baselines

The DET curves with respect to various baselines and our proposed methods are shown in Figure 2, where "PLDA-300" stands for the PLDA model with $\mathbf{F}$ of rank 300 in (1), "cosine" means the cosine kernel presented in [7], and "DCAE-1" represents the DCAE model with one hidden layer in both encoder and decoder parts. Obviously, our proposed method performs much better than PLDA whatever the cost parameters are. Table 1 also shows that, compared with PLDA-250, DCAE-0 achieves 36% and 24% relative improvements in EER and normalized minimum detection cost (NMDC).

On the other side, it can be seen that DCAE-1 and DCAE-2 do not outperform DCAE-0 as expected, although their results are acceptable while compared with PLDA. Actually, for the sake of convenience, all of the parameters for training the structure of DCAE with hidden layers are copied from the well-tuned parameters based on DCAE-0. Besides, the training set with less than 10,000 samples dose not seem to be enough for a more complicated structure.

Moreover, to demonstrate the effectiveness of speaker discrimination of DCAE, we arbitrarily single out two sets of

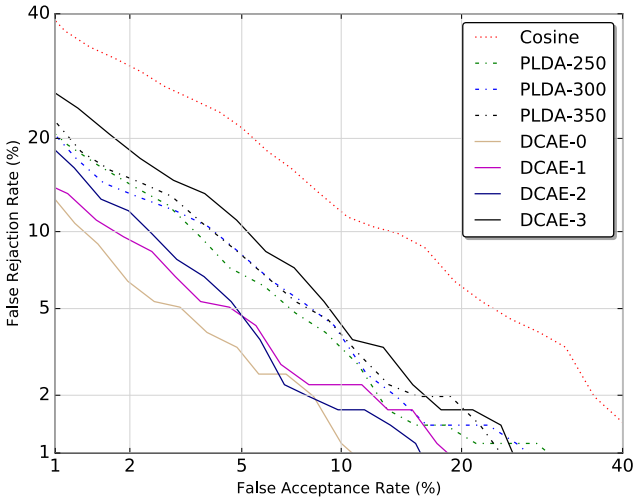

Fig. 2. DET curves of DCAE, PLDA, and cosine for SRE-10.

Table 1. Results for SRE-10. The percentages are the relative improvements over PLDA-250.

Method	EER	NMDC
Cosine	10.99	0.47
PLDA-250	6.20	0.29
DCAE-0	**3.94 (36%)**	**0.22 (24%)**

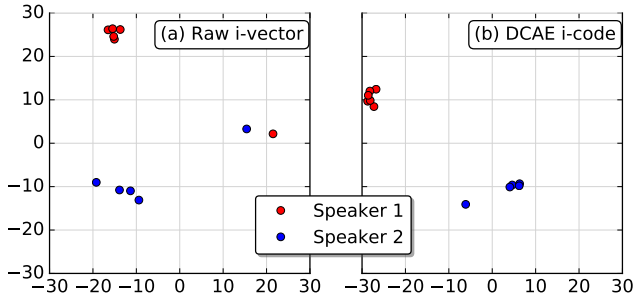

Fig. 3. Scatter plot of the results by t-SNE for SRE-10.

sessions from SRE-10, which belong to two different speakers, to be visualized by t-Distributed Stochastic Neighbor Embedding (t-SNE) [34]. Figure 3 illustrates the results that t-SNE maps the corresponding 400-dimensional i-vectors and the 300-dimensional i-codes into a 2-dimensional plane.

4.3 Observations on DCAE training

We made some observations from the training process of DCAE-0. First, *ceteris paribus*, we recorded the validation results in each training epoch to see the trend of different learning rates, batch sizes, and L2 weight regularization penalties (i.e., λ in (5)). From Figure 4, it can be seen that, in most cases, the best parameter usually occurs within 10 training epochs. This property for DCAE training makes the tuning of hyper-parameters much more tractable.

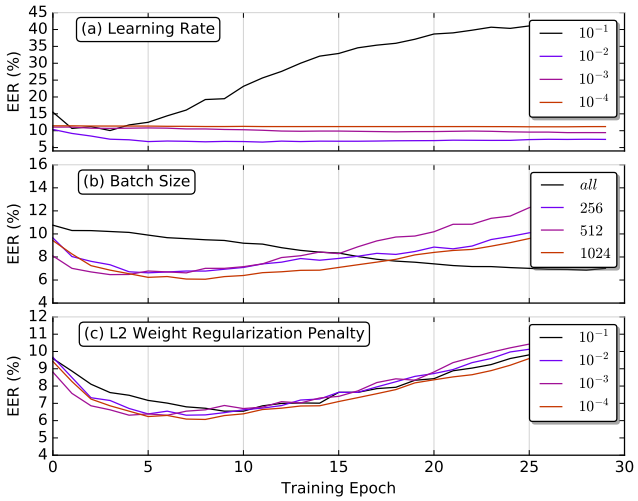

Fig. 4. EERs of SRE-08 for various settings of hyper-parameters with respect to the training epoch in DCAE.

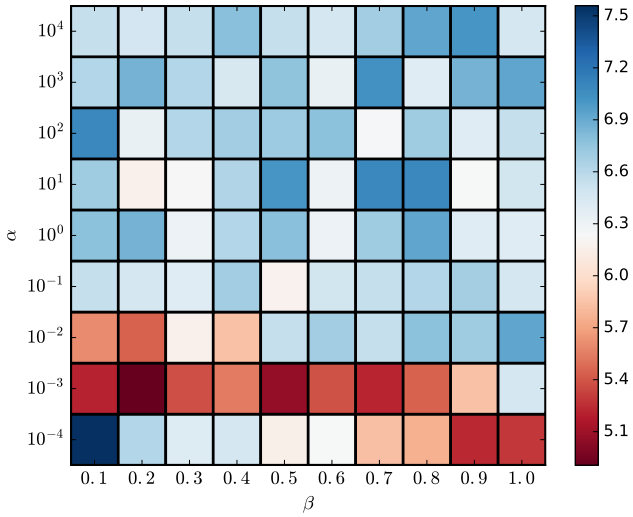

Fig. 5. Heat map generated from validation results reflecting EERs of SRE-08 in various settings of α and β in (5).

Second, we were also interested in the relationship among the three objective functions proposed in Section 2. As depicted in Figure 5, most of lower EERs occur when α, the weightiness of speaker identity loss and internal dispersion, is relatively small. This implies that the underestimation of the generative aspect of DCAE dose not necessarily help increase the discriminative power of the learning machine.

5 Conclusions

In this paper, we have proposed a framework based on three kinds of objective functions, which cooperate to make our model more like a generative model that possesses discriminative power for speaker verification. The framework has been realized by a neural network-based autoencoder, where the implementation is tractable for big data. The experiment results demonstrated the potential of the framework.

6 References

[1] T. Yamada *et al.*, "Improvement of distant-talking speaker identification using bottleneck features of DNN," in *Proc. Interspeech*, 2013.

[2] A. K. Sarkar *et al.*, "Combination of cepstral and phonetically discriminative features for speaker verification," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1040–1044, 2014.

[3] Y. Lei *et al.*, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *Proc. ICASSP*, 2014.

[4] P. Kenny *et al.*, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Proc. IEEE Odyssey*, 2014.

[5] E. Variani *et al.*, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014.

[6] F. Richardson *et al.*, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1671–1675, 2015.

[7] N. Dehak *et al.*, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 4, pp. 788–798, 2011.

[8] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," in *Proc. IEEE Odyssey*, 2010.

[9] H.-S. Lee *et al.*, "Clustering-based i-vector formulation for speaker recognition," in *Proc. Interspeech*, 2014.

[10] P. Li *et al.*, "Probabilistic models for inference about identity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 144–157, 2012.

[11] L. El Shafey *et al.*, "A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1788–1794, 2013.

[12] J. Rohdin *et al.*, "Constrained discriminative PLDA training for speaker verification," in *Proc. ICASSP*, 2014.

[13] H.-S. Lee *et al.*, "Speaker verification using kernel-based binary classifiers with binary operation derived features," in *Proc. ICASSP*, 2014.

[14] S. Cumani and P. Laface, "Training pairwise support vector machines with large scale datasets," in *Proc. ICASSP*, 2014.

[15] A. Nautsch *et al.*, "Towards PLDA-RBM based speaker recognition in mobile environment: Designing stacked/deep PLDA-RBM systems," in *Proc. ICASSP*, 2016.

[16] G. Heigold *et al.*, "End-to-end text-dependent speaker verification," in *Proc. ICASSP*, 2016.

[17] I. Goodfellow *et al.*, *Deep Learning*. The MIT Press, 2016.

[18] S. Araki *et al.*, "Exploring multi-channel features for denoising-autoencoder-based speech enhancement," in *Proc. ICASSP*, 2015.

[19] O. Plchot *et al.*, "Audio enhancing with DNN autoencoder for speaker recognition," in *Proc. ICASSP*, 2016.

[20] E. Marchi *et al.*, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional LSTM neural networks," in *Proc. ICASSP*, 2015.

[21] M. Mimura *et al.*, "Deep autoencoders augmented with phone-class feature for reverberant speech recognition," in *Proc. ICASSP*, 2015.

[22] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Academic Press, 1990.

[23] H.-S. Lee and B. Chen, "Linear discriminant feature extraction using weighted classification confusion information," in *Proc. Interspeech*, 2008.

[24] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 67, no. 2, pp. 301–320, 2005.

[25] S. Razakarivony and F. Jurie, "Discriminative autoencoders for small targets detection," in *Proc. ICPR*, 2014.

[26] Y. Bengio, "Learning deep architectures for AI," *FNT in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[27] P. Vincent *et al.*, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learning Research*, vol. 11, pp. 3371–3408, 2010.

[28] "The NIST year 2010 speaker recognition evaluation plan," NIST, 2010.

[29] "The NIST year 2008 speaker recognition evaluation plan," NIST, 2008.

[30] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Proc. IEEE Odyssey*, 2001.

[31] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," *Proc. Interspeech*, 2011.

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. AISTATS*, 2010.

[33] J. C. Duchi *et al.*, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learning Research*, vol. 12, pp. 2121–2159, 2011.

[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learning Research*, vol. 9, pp. 2579–2605, 2008.

Acknowledgment

This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant: MOST 105-2221-E-001-012-MY3.