Regularization of neural network model with distance metric learning for i-vector based spoken language identification

Xugang Lu[1] Peng Shen[1], Yu Tsao[2], Hisashi Kawai[1]

1. National Institute of Information and Communications Technology, Japan
2. Research Center for Information Technology Innovation, Academic Sinica, Taiwan

Abstract

The i-vector representation and modeling technique has been successfully applied in spoken language identification (SLI). The advantage of using the i-vector representation is that any speech utterance with a variable duration length can be represented as a fixed length vector. In modeling, a discriminative transform or classifier must be applied to emphasize the variations correlated to language identity since the i-vector representation encodes several types of the acoustic variations (e.g., speaker variation, transmission channel variation, etc.). Owing to the strong nonlinear discriminative power, the neural network model has been directly used to learn the mapping function between the i-vector representation and the language identity labels. In most studies, only the point-wise feature-label information is fed to the model for parameter learning that may result in model overfitting, particularly when with limited training data. In this study, we propose to integrate pair-wise distance metric learning as the regularization of model parameter optimization. In the representation space of nonlinear transforms in the hidden layers, a distance metric learning is explicitly designed to minimize the pair-wise intra-class variation and maximize the inter-class variation. Using the pair-wise distance metric learning, the i-vectors are transformed to a new feature space, wherein they are much more discriminative for samples belonging to different languages while being much more similar for samples belonging to the same language. We tested the algorithm on an SLI task, and obtained promising results, which outperformed conventional regularization methods.

Key words: Neural network model; cross entropy; pair-wise distance metric learning; spoken language identification.
PACS: 01.30.−y

Email address: `xugang.lu@nict.go.jp` (Xugang Lu[1]).

1 Introduction

Spoken language identification (SLI) is a technique to identify the language from spoken utterances. It is one of the most important processing steps in several speech applications, e.g., multi-lingual speech recognition, dialog, audio information retrieval, etc. (Ma et al, 2007a,b; Li et al, 2007, 2013). Conventional algorithms for the SLI include two steps-feature extraction and classification modeling. The i-vector representation and modeling technique is one of the state-of-the-art frameworks for SLI. It has been successfully applied in spoken language recognition (Dehak et al, 2011a; Li et al, 2007). The i-vector representation can be regarded as a middle-level representation between the Gaussian mixture model (GMM)-based super-vector and the short-time Mel-frequency cepstral coefficient (MFCC) frame-based feature representations. One of the advantages of using the i-vector representation is that acoustic variations of speech utterances with different time durations can be represented as fixed-length feature vectors. It is convenient to be applied to various standard pattern classification techniques, e.g., Gaussian mixture model (GMM), support vector machine (SVM), probabilistic linear discriminant analysis (PLDA), etc. (Dehak et al, 2011b; Prince et al, 2007).

The i-vector is a compact low-dimensional representation of acoustic variations. It is obtained through a total variability factor analysis of the supervector representations of GMM. In the representation, several types of acoustic variations or factors are encoded in the feature vectors (e.g., speaker, language and transmission channel variations), hence a discriminative transform must be applied to remove uncorrelated variations while emphasizing discriminative variations of different tasks (Sugiyama, 2006; Dehak et al, 2011b; Sadjadi et al, 2015; Shen et al, 2016). During an SLI task, a transform must be applied to emphasize the feature variations correlated to language identity. Conventionally, a linear discriminant analysis (LDA)-based transform is applied on the i-vectors to obtain the discriminative features for SLI. Improvements on the LDA, such as the nearest neighbor discriminant analysis (NNDA) and the local fisher discriminative analysis (LFDA), were also proposed for SLI tasks (Sadjadi et al, 2015; Sugiyama, 2006; Shen et al, 2016). Since acoustic variations in the i-vectors are entangled with complex nonlinearity and non-gaussian, it is difficult to use a linear transform to disentangle the variations. A nonlinear transform is preferred in order to extract better discriminative features for SLI. Although nonlinear feature extraction and classification can be obtained by setting the nonlinear kernel functions of SVM or distance metric, it is difficult to tune the parameters in a unified optimization framework with feature and classifier learning simultaneously. An artificial neural network model is a natural choice that unifies feature extraction and classification with nonlinear transforms. In addition, the model parameters can be learned efficiently using an error back propagation algorithm.

Neural network models (including their deep form, i.e., deep neural network (DNN) algorithms) have been shown to have dominant power for feature learning and classification in image processing and speech recognition (Hinton et al, 2012a; Yu et al, 2015; LeCun et al, 2015). Following the step of neural network acoustic modeling technique, they have also been used in speaker recognition and spoken language recognition (Montavon, 2009; Lopez-Moreno et al, 2014, 2016; Richardson et al, 2015a,b; Ranjan et al, 2016). Neural network modeling can automatically explore the nonlinear feature variations correlated to the classification task, and jointly learn the nonlinear transform and classification. In most studies that use neural network models for SLI, two modeling methods have been adopted (Richardson et al, 2015b). The first method, i.e., "indirect method", uses a neural network model as a front-end processing for feature extraction (e.g., bottleneck feature and i-vector extraction), and subsequently, models the extracted feature with another classifier. The second method, i.e., "direct method", uses a neural network model to directly learn the mapping function between the i-vector representation (or other acoustic feature representations) and their language identities (IDs). In this study, we focus on the "direct method" of using a neural network model for SLI, with the i-vector representation as the input to the neural network. The discriminative feature extraction from the i-vectors and the classification modeling are optimized in a unified framework.

In conventional neural network model learning algorithms for SLI, the model is trained to approximate a point-mapping function from the samples by feeding the feature label information to the model (hereafter named as point-wise training). It is based on the principle of minimizing an objective function measuring the difference between predicted language labels and true target labels (e.g., cross-entropy) (Montavon, 2009). In most model training, a large quantity of training samples is required. For example, in automatic speech recognition (ASR), a large quantity of training samples is provided for each class. Similarly, in neural network modeling for SLI, a large quantity of training data is required for each language (Lopez-Moreno et al, 2014, 2016; Richardson et al, 2015a). If training data is limited, e.g., the number of utterances for each language is hundreds, the neural network model is easily over-fitted to the training data set. The over-fitted model will lose its strong capacity in the classification task, which results in a bad performance with a test data set (i.e., it exhibits a weak generalization ability). This study attempts to deal with the problem of generalization that arises while using the neural network model in SLI with limited training data samples.

In order to improve the generalization ability of the model, many techniques have been proposed. Considering the lack of training data, an efficient way to augment the data (LeCun et al, 1998) is to artificially increases the training data samples by adding noise or distortions to the training data. Another efficient way is to impose constraints on the model parameter space, for example,

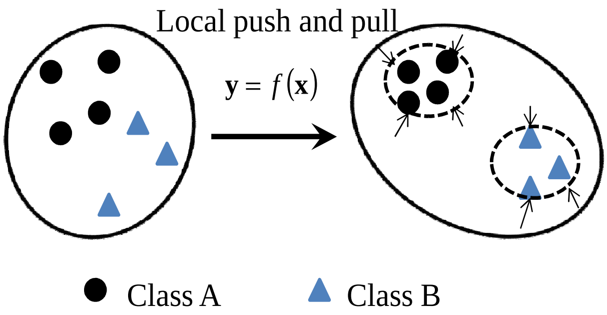


Fig. 1. Distance metric learning with local push and pull transforms to reduce intra-class variation and increase inter-class variation.

regularization of neural weights with smoothness (L_2) and sparseness (L_1) is widely used in neural network model learning to deal with the problem of overfitting (Bishop, 2006). The technique of "dropout" which randomly sets a portion of neural responses to zeros, has shown an impressive performance in recent times (Hinton et al, 2012b); further, it has also been used in neural network model training with limited training data for SLI (Ranjan et al, 2016). In this study, instead of using point-wise training in the parameter optimization of the neural network model, we investigate the possibility of exploring data geometric structure as a constraint (or regularization) in model parameter learning, and attempt to improve the generalization ability of the model.

Point-wise information (or structure) is widely used in supervised model training, however, other information in a training data set may also be used to provide auxiliary constraints for robust model training, e.g., training data geometric structure, pattern-clustering property, etc. In this study, besides using the feature label information of samples independently for supervised training of the neural network model, the similarity or distance measure of pair-wise samples is also taken into consideration. Learning that takes into account pair-wise distance measurements belongs to a large category of machine learning, i.e., the metric learning (Xing et al, 2002; Weinberger et al, 2006, 2009). In linear metric learning (Xing et al, 2002), for instance, learning a Mahalanobis distance metric in the input feature space is done in order to measure the similarity of a pair of input samples. The basic idea of metric learning is to learn a distance measurement by determining which samples belonging to the same category should be distributed in a neighboring space, but should be far away otherwise. This property is preferred for application in pattern discrimination. In the framework of the neural network model, nonlinear distance metric learning has been proposed for face recognition and re-identification (Guillaumin et al, 2009; Hu et al, 2015). In most metric learning studies, the discriminative transform is optimized explicitly, based on an objective function, which is supposed to reduce the intra-class variation, while increasing the inter-class variation. Fig. 1 gives an illustration of this process. As shown in this figure, a distance metric transform $f(\mathbf{x})$ should be learned to "pull" samples belonging to the same class to the neighboring space, while being learned to "push" samples belonging to a different class to a longer distance. The metric learning is widely applied for information retrieval and pattern classification. However, most of the metric learning algorithms are used to learn discriminative features for a linear classifier, e.g., support vector machine, and nearest neighbor classifier. In this study, we integrate this metric learning with a conventional framework of neural network model, and jointly learn the nonlinear transform and classifier in order to improve the generalization ability of the model for SLI.

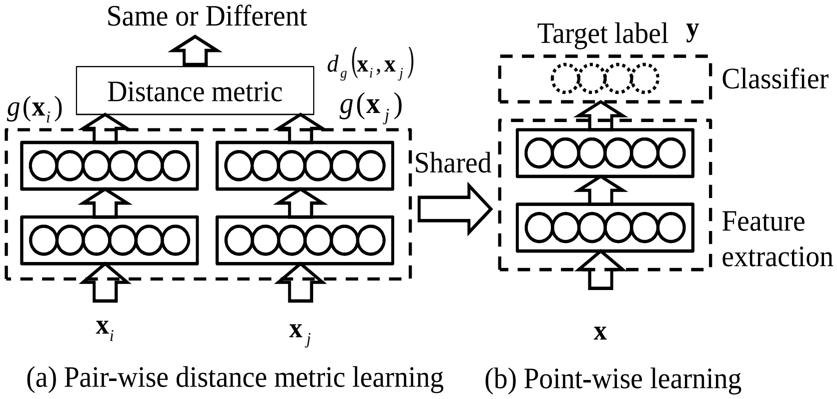
The main contribution of this paper is to integrate the pair-wise distance metric learning with the point-wise classifier learning in order to achieve a unified neural network model for spoken language classification. Based on this model, the conventional mapping function for classification is explicitly formulated as a combination of a feature extraction function and a classification function regularized with different objectives. By considering the pair-wise distance metric learning as an objective, the feature extraction function explicitly extracts a nonlinear discriminative feature, which is optimal for classification learning. Our spoken language identification experiments exhibited a promising performance.

The reminder of the paper is organized as follows. Section 2 introduces the framework of the neural network model that explicitly integrates pair-wise distance metric learning in the model parameter optimization. Section 3 describes the SLI experiments that were carried out. Finally, discussions are presented in Section 4, and the final conclusion is presented in Section 5.

2 Neural network model with pair-wise distance metric learning

In the “direct” neural network model of classification, a softmax layer is often stacked as a classifier layer on the hidden layers in order to achieve normalized probability. The model has two functions: one function provides the discriminative features via the nonlinear transforms of hidden layers; the other function acts as a classifier via the softmax layer. The feature learning and classification are coupled into a unified model, which learns an efficient feature representation and is suitable for classification. In most studies, each sample (in a mini-batch) is used independently to train model parameters with a unified objective function in order to achieve pattern discrimination. There is no structure constraint on the transform of the two samples in the hidden layers. In order to include a structure constraint in the framework of neural network model, we integrate pair-wise distance metric learning in the framework along with the explicit regularization of the feature transform function.

Fig. 2 shows the two explicitly coupled function modules of the framework of



(a) Pair-wise distance metric learning (b) Point-wise learning

Fig. 2. Neural network model learning with pair-wise distance metric constraint (a) pair-wise distance metric learning for feature extraction in hidden layers, (b) point-wise classifier learning by softmax layer.

neural network model-the feature transform (with pair-wise distance metric constraint $d_g(\cdot,\cdot)$) and the classification learning. In Fig. 2 (b), as conventionally used in neural network model learning, the feature-label map is directly learned by minimizing an objective function based on cross-entropy. In parameter optimization, there is no explicit constraint on the learning of the hidden layer features. In pair-wise distance metric learning (as in Fig.2 (a)), the transform functions (as $g(\cdot)$) via the hidden layers are constrained using a pair-wise loss function. As shown in the figure, two representations from the last hidden layers are obtained using a pair of input vectors. The two input vectors share the same model parameters, which are similar to those used in the Siamese network (Chopra et al, 2005). In the following part of the study, the point-wise feature-label learning and the pair-wise distance metric learning are introduced, respectively.

2.1 Point-wise learning for neural network model

Conventionally, the purpose of training the neural network model is to learn the input-target mapping function. The difference between the two coupled functions of feature and classifier processes is not considered. The training is performed point-wise as one input corresponds to one target output. For a neural network model with $K-1$ hidden layers, the output of a hidden layer is represented as

$$\mathbf{h}^k = \mathbf{f}^k\left(\mathbf{W}^k\mathbf{h}^{k-1} + \mathbf{b}^k\right), \tag{1}$$

where $k = 1, ..., K-1$, and $\mathbf{h}^0 = \mathbf{x}$ is the input layer with feature vector $\mathbf{x}$. Further, $\mathbf{W}^k$ and $\mathbf{b}^k$ are the neural weight matrix and bias of the k-th hidden layer, respectively. Furthermore, $\mathbf{f}^k(.)$ is a nonlinear active function (an element-wise transform), e.g., sigmoid function, tanh function, Rectified Linear Units (ReLU) (Nair et al, 2010), etc. In this study, a tanh function was used as

$$f^k(z) = \tanh(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)}. \tag{2}$$

Given an input feature vector $\mathbf{x}$, a predicted label is obtained from the final output layer which is a softmax layer with the transform as

$$\begin{aligned} \hat{y}_j &= p(y_j = 1|\mathbf{x}, \mathbf{W}, \mathbf{b}) \\ &= \frac{\exp\left(\mathbf{W}_j^K \mathbf{h}_j^{K-1} + \mathbf{b}_j^K\right)}{\sum\limits_{i=1}^{\#Class} \exp\left(\mathbf{W}_i^K \mathbf{h}_i^{K-1} + \mathbf{b}_i^K\right)}, \end{aligned} \tag{3}$$

where y_j is the output of the j-th neuron in the softmax layer and "$\#Class$" is the total number of classes. For learning the model parameters, an objective function defined as the cross-entropy (CE) between the predicted and true target labels is used as

$$l(\Theta) \triangleq \sum_{m=1}^{\#Sample} CE(\mathbf{y}_m, \hat{\mathbf{y}}_m) = -\sum_{m=1}^{\#Samples} \sum_{n=1}^{\#Class} y_{m,n} \log \hat{y}_{m,n}, \tag{4}$$

where $\hat{y}_{m,n}$ and $y_{m,n}$ are the probabilities of the predicted and true targets, which are represented as the scalar elements of the vectors of $\hat{\mathbf{y}}_m$ and $\mathbf{y}_m$, respectively. Furthermore, m and n are the indexes of sample and class number, respectively, and "$\#Samples$" is the total number of training samples.

The method of learning the model parameters is based on the minimization of this cross-entropy based objective function (equation (3)) with parameter regularization performed on a training data set as follows:

$$\begin{aligned} \Theta^* &= \arg\min_{\Theta} C(\Theta) \\ C(\Theta) &= l(\Theta) + \lambda R(\Theta), \end{aligned} \tag{5}$$

where $\Theta = \left\{\mathbf{W}^k, \mathbf{b}^k, k = 1, 2, ..., K\right\}$ is the model parameter set with neural weight matrix $\mathbf{W}^k$ and bias $\mathbf{b}^k$. Further, λ is a regularization coefficient to control the tradeoff between the cross-entropy based loss and parameter regularization $R(\Theta)$. In most studies, parameter regularization is defined as smoothness or sparseness of the model parameter space (e.g., either L_2 or L_1 regularization) which has been shown to improve the generalization ability of the model.

The stochastic gradient descent (SGD) algorithm is used in learning (Bottou, 2012). From the transforms and objective function (equations (1)-(5)), it is evident that the learning tries to find a local optimal solution to approximate the feature-label mapping of the training data set. In order to find a better solution, there must be constraints on the learned transform functions. In this study, a pair-wise distance metric learning is explicitly designed on the representation space explored in the hidden layers. This is equivalent to imposing constraints on the feature transform function.

2.2 Nonlinear pair-wise distance metric learning

The pair-wise distance metric learning is intended to determine a transform function via the reduction of intra-class variation and increase of inter-class variation (Xing et al, 2002), i.e., the two samples should be close if they belong to the same class, but far otherwise. In our model framework, we explicitly add this property to control the nonlinear transform function realized by the hidden layers.

Considering a pair of samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in a transform space using the corresponding hidden layer outputs $\mathbf{f}\left(\mathbf{x}_i\right)$ and $\mathbf{f}\left(\mathbf{x}_j\right)$, the distance between them can be defined as a Euclidean distance or cosine distance. As widely used in most studies for i-vector based spoken language recognition, the cosine distance metric is also used in this paper defined as

$$d_{\mathbf{f}}\left(\mathbf{x}_i, \mathbf{x}_j\right) = \frac{\mathbf{f}\left(\mathbf{x}_i\right)^T \mathbf{f}\left(\mathbf{x}_j\right)}{\left\|\mathbf{f}\left(\mathbf{x}_i\right)\right\|_2^{\frac{1}{2}} * \left\|\mathbf{f}\left(\mathbf{x}_j\right)\right\|_2^{\frac{1}{2}}}. \tag{6}$$

This metric measures the angle between two vectors, which is a similarity measure widely used in vector space modeling (VSM) (Sidorov et al, 2014). It has a maximum value of 1 (angle 0) and a minimum value of -1 (angle π). Therefore, the values of pair-wise distance are distributed between $[-1, 1]$.

For the sake of convenience, given a training data set with a feature vector $\mathbf{x}_i$ and a label vector $\mathbf{y}_i$ (a one-hot encoding vector), and $i = 1, 2, ...$, we define two data sets of pair-wise samples, S and D, as follows:

$$\begin{aligned} S &= \left\{\left(\mathbf{x}_i, \mathbf{x}_j\right) | \forall \mathbf{y}_i = \mathbf{y}_j\right\} \\ D &= \left\{\left(\mathbf{x}_i, \mathbf{x}_j\right) | \forall \mathbf{y}_i \neq \mathbf{y}_j\right\}, \end{aligned} \tag{7}$$

i.e., data sets S and D consist of pair-wise data samples belonging to the same and different classes, respectively.

Based on the basic principle of metric learning, two loss functions are defined on the pair-wise data sets in a transform space $\mathbf{f}(.)$ as follows:

$$\begin{aligned} J_{\text{Intra}}(\Theta) &\triangleq \tfrac{1}{\#S} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in S} \left(d_{\mathbf{f}}(\mathbf{x}_i,\mathbf{x}_j) - 1\right)^2 \\ J_{\text{Inter}}(\Theta) &\triangleq \tfrac{1}{\#D} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in D} \left(d_{\mathbf{f}}(\mathbf{x}_i,\mathbf{x}_j) + 1\right)^2, \end{aligned} \tag{8}$$

where "$\#S$" and "$\#D$" represent the number of sample pairs in sets S and D, respectively. In equation (8), minimizing the value of $J_{\text{Intra}}(\Theta)$ could decrease the pair-wise intra-class variation, and minimizing the value of $J_{\text{Inter}}(\Theta)$ could increase the pair-wise inter-class variation.

Considering the tradeoff between the robustness and discrimination, we formulate the objective function for pair-wise metric learning as follows:

$$J(\Theta) = J_{\text{Intra}}(\Theta) + \alpha J_{\text{Inter}}(\Theta), \tag{9}$$

where α controls the tradeoff between the two above mentioned losses. For equal weighting of each pair-wise loss, the metric learning is based on minimizing the following objective function:

$$J(\Theta) = \frac{1}{\#\{S \cup D\}} \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\{S\cup D\}} \left(d_{\mathbf{f}}(\mathbf{x}_i,\mathbf{x}_j) - t_{i,j}\right)^2, \tag{10}$$

where "$\#\{S \cup D\}$" is number of sample pairs in sets S and D together, and the pair-wise label $t_{i,j}$ is defined as:

$$t_{i,j} = \begin{cases} 1, \forall(\mathbf{x}_i,\mathbf{x}_j) \in S \\ -1, \forall(\mathbf{x}_i,\mathbf{x}_j) \in D \end{cases} \tag{11}$$

2.3 Regularization with pair-wise distance metric learning

There are two strategies for integrating the pair-wise distance metric learning into the neural network model during parameter learning. One method is to use it as a regularization term in the cross-entropy based objective function (s-trategy I); whereas, the other method is to use it as a pre-training criterion for the initialization of model parameters, and subsequently fine-tune the model with an error back propagation (BP) algorithm based on the cross-entropy based objective function (strategy II). The two strategies are introduced in detail as follows.

2.3.1 Combination of point-wise and pair-wise objective functions (strategy I)

In this strategy, the pair-wise distance metric loss and the point-wise cross-entropy loss are combined into a single objective function. For the sake of convenience, we assume the parameter set related to the feature extraction (via the hidden layers) and the classifier (via softmax layer) as $\Theta = \{\Theta_F, \Theta_C\}$. Further, Θ_F represents the entire model parameter set (neural connection weights and bias) except the softmax layer, and Θ_C is the model parameter set that represents only the softmax layer. The parameters are obtained as

$$\begin{aligned} &\{\Theta_F^*, \Theta_C^*\} = \arg \min_{\{\Theta_F, \Theta_C\}} M\left(\Theta_F, \Theta_C\right) \\ &\text{with } M\left(\Theta_F, \Theta_C\right) = l\left(\Theta_F, \Theta_C\right) + \gamma J\left(\Theta_F\right) + \lambda R\left(\Theta_F, \Theta_C\right). \end{aligned} \tag{12}$$

In this objective function, the first term $l\left(\Theta_F, \Theta_C\right)$ is the point-wise cross-entropy between the predicted and true labels (as defined in equation (4)) and the second term $J\left(\Theta_F\right)$ is the objective function for the distance metric learning (as defined in equation (10)). Further, γ controls the tradeoff between the two losses. Furthermore, λ and $R\left(\Theta_F, \Theta_C\right)$ have been defined in equation (5).

The feed forward transform is expressed as a composition function $\mathbf{g}_C \circ \mathbf{g}_F : \mathbf{x} \rightarrow \mathbf{c}$, where $\mathbf{g}_F : \mathbf{x} \rightarrow \mathbf{z}$, and $\mathbf{g}_C : \mathbf{z} \rightarrow \mathbf{c}$ are the feature transform and classifier transform, respectively, and $\circ$ is a composition operator. Based on the chain rule, their gradients are calculated as follows:

$$\begin{aligned} &\nabla\Theta_F = \tfrac{\partial M}{\partial \Theta_F} = \tfrac{\partial l}{\partial \mathbf{g}_C} \tfrac{\partial \mathbf{g}_C}{\partial \mathbf{g}_F} \tfrac{\partial \mathbf{g}_F}{\partial \Theta_F} + \gamma \tfrac{\partial J}{\partial \mathbf{g}_F} \tfrac{\partial \mathbf{g}_F}{\partial \Theta_F} + \lambda \tfrac{\partial R}{\partial \Theta_F} \\ &\nabla\Theta_C = \tfrac{\partial M}{\partial \Theta_C} = \tfrac{\partial l}{\partial \mathbf{g}_C} \tfrac{\partial \mathbf{g}_C}{\partial \Theta_C} + \lambda \tfrac{\partial R}{\partial \Theta_C}. \end{aligned} \tag{13}$$

From these equations, it is evident that the feature transform and classifier transform for obtaining cross-entropy loss are involved in the calculation of gradient, during the learning of the feature metric parameter Θ_F. In addition, the gradient is explicitly regularized with the pair-wise loss function $J\left(\Theta_F\right)$, which is a function of feature transform. However, during the learning of the classifier layer parameter Θ_C, the gradient is calculated using the derivative of the cross-entropy with respect to the classifier transform only.

Combining the point-wise and pair-wise training, the calculation was synchronized in each mini-batch, where the objective function (cross-entropy) was accumulated from the samples. In pair-wise learning, the corresponding pair-wise samples were collected from each mini-batch, and the objective function (i.e., the pair-wise distance metric loss) was estimated for each mini-batch.

2.3.2 Pair-wise distance metric learning as pretraining for model parameter initialization (strategy II)

In order to deal with the overfitting problem, it is recommended to store enough feature information during the early stages of neural network model learning for the sake of robustness. Based on this consideration, the pair-wise distance metric learning is used as a pre-training for the initialization of model parameters. In contrast to multi-category classification tasks, the distance metric learning is a much simpler task (which includes a binary classification task as an inter-class or intra-class decision). During the pre-training, the pair-wise distance metric learning is performed layer-by-layer. In our implementation, the inter-class and intra-class labels were extracted based on class clustering information (as shown in equation (11). The optimization criterion is based on equation (10), which measures the mean square error of the cosine distance (or similarity) metric. As a result of this pre-training, a softmax layer is stacked onto the pre-trained hidden layers for fine-tuning. Since the initialized model parameters are well constrained, it is possible to find a good solution for the model parameters with fine-tuning.

3 Experiments

In this section, we test the proposed algorithm using an SLI task. A data set from NIST i-vector challenge 2015 for SLI is used in this paper (NIST SLI data, 2015). Fifty languages are included in the training data set, and each language has 300 samples (and an i-vector with 400 dimensions for each utterance). In our study, 250 i-vectors from the training set for each language were randomly chosen for training, and the remaining 50 i-vectors were used for validation. In the originally released test set, there are 6500 test samples (as i-vectors). Among them, 1500 samples are out-of-set class (OOS) samples, i.e., the language types are not included in the 50 languages used in training (Tong et al, 2016). Special algorithms were developed to deal with the OOS problem (Lee et al, 2016; Sun et al, 2016), including a novel OOS detection scheme developed by (Sun et al, 2016). However, in this work, we did not focus on the solution to the OOS problem. Therefore, in the released test set, only the in-set class i-vectors were chosen. Finally, 5000 test i-vectors were used including 100 samples for each language.

In the original evaluation criterion, the cost function is defined as

$$\text{cost} = \frac{(1 - p_{OOS})}{K} \times \sum_{k=1}^{K} p_{error}\,(k) + p_{OOS} \times p_{error}\,(OOS)\,, \tag{14}$$

where $p_{error}(k)$ and $p_{error}(OOS)$ are the error rates of test set (in-set classes) and OOS set, respectively. In the original test set that includes the OOS, $K = 50$ and $p_{OOS} = 0.23$. In this paper, $p_{OOS} = 0$ since the OOS problem was not dealt with in our study.

3.1 Conventional models

In order to choose a baseline for comparison, we first conducted experiments using several conventional algorithms. The performances are shown in table 1. In this table, "COSINE" and "LDA+COSINE" represent the methods the use a COSINE similarity measure as a classifier and a raw i-vector of 400 dimensions, and its dimension-reduced feature of 49 dimensions via linear discriminative analysis (LDA) as inputs, respectively. Further, "LINSVM" and "LDA+LINSVM" represent the methods that use linear kernel SVM (LINSVM) as a classifier and raw a i-vector of 400 dimensions, and its LDA dimension-reduced feature as inputs, respectively. From this table, it is evident that the LINSVM with LDA shows the best performances for feature extraction.

Table 1
Performance of conventional algorithms (identification error rate in %)

Model	Train	Validation	Test
COSINE	10.93	17.24	17.86
LDA+COSINE	10.90	17.44	18.20
LINSVM	5.66	16.44	16.68
LDA+LINSVM	8.61	16.04	**16.44**

We further examined a nonlinear kernel function based SVM, and showed the results in table 2 as "LDA+RBFSVM". This method represents the radial basis function (RBF) kernel SVM based classifier with LDA for feature extraction. Comparing the results of "LDA+LINSVM" in table 1, we can see that nonlinear kernel SVM shows little improvement on the test set (although there was a large improvement on training set). The parameters used in RBFSVM are difficult to tune as compared to the LINSVM. In our implementation, the SVM toolbox (SVM Tool, 2015) was used. The optimal model parameters were obtained based on a grid search with cross-validation. In the following experiments, LDA+LINSVM is selected as the baseline for comparison.

Table 2
Performance of SVM classifier with nonlinear function kernel (identification error rate in %)

Model	Train	Validation	Test
LDA+RBFSVM	2.91	15.78	16.40

3.2 Neural network models for SLI

In building the neural network models for SLI, two types of architectures were used as follows: 400-M*512-50, where 400 is the number of dimensions of the input i-vector, M represents the hidden layers with 512 neurons for each layer, and there are 50 neurons in the output layer corresponding to 50 language IDs. For M=1,2, we obtained two models named M1 and M2 for short, henceforth. The network was first pre-trained layer-wise as a restricted Boltzmann machine (RBM) with a contrastive divergence algorithm (Hinton, 2010). While fine-tuning using the stochastic gradient descendent algorithm, the mini-batch size was 128, and the learning rate was 0.001.

In order to improve the generalization ability of the model, two regularization methods were implemented for comparison-one method is the parameter L_2 regularization (Bishop, 2006), and the other is the dropout (DP) regularization. These two methods are widely used in the field of deep learning in order to improve the generalization ability; In particularly, the DP regularization shows an impressive performance in dealing with the overfitting problem (Hinton et al, 2012b). The regularization parameters were fine-tuned to obtain the best performance with a validation data set. In the L_2 regularization used in equation (5), λ was set to 0.001. In the dropout regularization, the dropout probabilities for the input and hidden layer output were set to 0.3 and 0.5, respectively. The model used for the test set was obtained until the best performance w.r.t. a validation data set was obtained in a total of 500 epoches. The results are shown in Table 3.

Table 3
Performance of point-wise training for neural network model systems (identification error rate in %)

Model	Train	Validation	Test
M1_L_2	2.61	17.04	17.62
M2_L_2	1.31	17.92	18.48
M1_DP	6.40	15.41	16.36
M2_DP	3.31	15.25	**15.78**

In this table, “M1_L_2” and “M2_L_2” represent the two models M1 and M2 with L_2 regularization, respectively. "M1_DP" and "M2_DP" are the two models M1 and M2 with dropout regularization, respectively. From this table, it is evident that the performance of linear model in this task (performance of "LDA+LINSVM" as showed in table 1.) is much better than that of the plain neural network models regularized with the L_2 technique. The dropout regularization showed a large improvement as compared to the L_2 regularized models. For the neural network model, after adding more hidden layers, there is no significant increase in the performance w.r.t. the test data set although the training error continuously decreased. Further, we test the effect of adding pair-wise metric learning to the two models for parameter learning.

3.3 Combination with pair-wise distance metric learning in model training

We implemented the pair-wise metric learning for the optimization of model parameters, and tested the performance of the two combination strategies introduced in sections 2.3.1 and 2.3.2. All our implementations are in theano (theano, ver0.8.2) and lasagne (lasagne, ver0.2).

3.3.1 As a regularization in cross-entropy based objective function

We first implemented strategy I as introduced in section 2.3.1. In this strategy, the pair-wise distance metric based loss is included as a regularization term in optimizing the cross-entropy based objective function. During the implementation, equations (10) and (11) were used. As shown in equation (12), the tradeoff between losses of feature metric learning and classifier learning is controlled by changing the regularization parameter γ. We first determine the performance when the regularization parameter is varied for metric learning, and the results are shown in Table 4 (for model M2). From the results, it is evident that by varying the regularization of the feature metric loss, the identification error for the training data set increased, but the performance

Table 4
Performance of neural network model system with pair-wise distance metric learning with varying regularization coefficients (identification error rate in %)

Coef γ	Train	Validation	Test
0.001	1.76	17.29	17.56
0.005	2.68	15.50	16.58
0.01	2.37	13.68	**15.43**
0.03	4.05	14.82	16.12
0.05	4.59	14.87	16.70

w.r.t. the validation and test data sets were improved significantly. When γ is approximately 0.01, we obtained the best performance for the test data set. These results suggest that using the pair-wise distance metric learning in model regularization improved the generalization ability of the learned model.

By adjusting the regularization parameter γ for the two models M1 and M2 (to approximately 0.01), we obtained the best performance of each model, and showed the results in Table 5. In this table, "M1_DM" and "M2_DM" represent models M1 and M2 with the pair-wise distance metric learning as a regularization. Comparing the results in tables 5 and 3, we find that all the models benefit from the pair-wise distance metric learning showing significant improvements in performance.

Table 5
Performance of neural network systems with pair-wise distance metric learning (identification error rate in %)

Model	Train	Validation	Test
M1_DM	3.20	14.40	15.94
M2_DM	2.37	13.68	**15.43**

3.3.2 As a pre-training for initialization of neural network model parameters

Instead of using the pair-wise distance metric based loss as a regularization term in the optimization of the objective function, we use it as pre-training for the initialization of model parameters. The basic assumption behind this decision is that the model parameters for feature extraction can be initialized in a "good" position. This "good" position is guaranteed by adding constraints on the hidden layer transform space using the pair-wise distance metric based loss function. The features extracted by hidden layers are much more suitable for classification, after fine-tuning them with cross-entropy based classifier learning. In order to investigate the contribution of the metric learning as pre-training in feature extraction, we compare three model initialization algorithms: using pair-wise distance metric learning for model initialization (DM4Init) (based on minimizing the objective function defined in equation (10)); using RBM pretraining for model parameter initialization (RBM4Init); and using Glorot method for initializing the model parameters (Glorot4Init) (Glorot et al, 2010). After model initialization, a softmax layer was stacked and exclusively fine-tuned. In fine tuning the softmax layer, the cross-entropy based objective function was used (based on equation (5)). The results for model M2 are shown in Table 6. In this table, "M2_DM4Init_softmax", "M2_RBM4Init_softmax" and "M2_Glorot4Init_softmax" represent M2 model with DM4Init, RBM4Init, and Glorot4Init for initialization, respectively, and only the softmax layer is fine-tuned. From this table, it is evident that

Table 6
Performance of only tuning the softmax layer with different model initialization algorithms (identification error rate in %)

Model	Train	Validation	Test
M2_DM4Init_softmax	6.93	15.42	**16.00**
M2_RBM4Init_softmax	12.70	22.00	22.28
M2_Glorot4Init_softmax	19.67	31.23	27.48

pre-training (either DM4Init or RBM4Init) is more important than random feature projection. The distance metric learning efficiently extracts discriminative features, which help to improve the performance significantly as compared to the RBM method for model parameter initialization.

3.4 Combination of pair-wise distance metric learning with dropout

Dropout regularization is a universal method, which can be easily applied to model parameter learning with any types of regularized objective functions. To investigate whether our proposed pair-wise distance metric based regularization has a complementary effect in improving the generalization ability when combined with the dropout regularization, we implemented the dropout regularization using the SGD-based optimization on our proposed regularized objective function (equation (12)) (wherein the regularization parameter $\gamma = 0.01$ was used). The results are shown in Table 7. In this table, "M1_DM_DP" and "M2_DM_DP" represent the combination of dropout regularization with pair-wise distance metric learning as regularization in models M1 and M2, respectively. Comparing with the results in Tables 3, 5 and 7, it is evident that applying both of these methods resulted in a better performance.

Table 7
Performance of neural network model systems with combination of pair-wise metric learning and dropout regularization (identification error rate in %)

Model	Train	Validation	Test
M1_DM_DP	4.30	15.47	15.54
M2_DM_DP	3.39	14.45	**15.14**

4 Discussion

In order to extract discriminative features from the i-vectors for SLI, a discriminative transform is required to explore variations in the i-vectors correlated to language IDs. Further, LDA is one of the most efficient methods. It can be regarded as a distance metric learning, which explores the linear and Gaussian variations of acoustic signals correlated to language IDs. The distance metric learning based on the neural network model can explore acoustic variations entangled with complex non-Gaussian and nonlinearity. In most studies, usually the metric learning is directly used to improve the k-nearest neighbor (KNN)-based classification. We performed additional experiments to determine the performance of metric learning with a KNN-based classification on the test set. The results are shown in table 8. In this table, "i-vector" represents the raw i-vector feature (400 dimensions), "i-vector+LDA" is the representation of i-vector with the LDA transform (49 dimensions), and "i-vector+DeepMetric" is feature extraction based on DNN (containing 2 hidden layers with 512 neurons for each) only with pair-wise distance metric learning (and no fine-tuning of the softmax layer). "KNN (5)", "KNN(10)", "KNN(50)", and "KNN(100)" represent the KNN with 5, 10, 50, and 100 nearest neighbor samples in decision-making, respectively. From the table, it is evident that LDA significantly improves the discrimination of class clustering. The deep metric learning obtains a large gain over the LDA.

Table 8
Performance of metric learning in KNN-based classification (identification error rate in %)

Model	KNN (5)	KNN(10)	KNN(50)	KNN(100)
i-vector	63.48	60.88	59.64	61.12
i-vector+LDA	23.94	22.52	20.66	20.48
i-vector+DeepMetric	16.92	16.90	16.94	17.34

In the i-vector extraction, it is supposed that the utterance duration should be long enough. For short utterances, the variations in the i-vectors may be due to unbalanced phone realizations and not accurate statistics estimation. Large variations will degrade the performance of SLI. We further checked the performance w.r.t. the samples grouped according to utterance durations. The results of the validation and test sets are shown in table 9. In this table, we compared two methods; "LDA+LINSVM" and "M2_DM" (distance metric regularization on two hidden layers and one softmax layer of the neural network). Four groups of utterances with time durations of $(0s, 3s]$, $(3s, 10s]$, $(10s, 30s]$, and $(30s, \inf)$ are considered. From this table, it is evident that the main improvements occur for utterances smaller than $3s$ and larger than $30s$.

Table 9
Performance comparison according to utterance length (identification error rate in %)

Set	Model	$(0s, 3s]$	$(3s, 10s]$	$(10s, 30s]$	$(30s, \inf)$
Valid	LDA+LINSVM	64.71	34.02	15.65	6.04
	M2_DM	58.82	34.53	14.15	5.10
Test	LDA+LINSVM	67.44	36.86	15.24	7.20
	M2_DM	62.79	36.82	14.35	6.28

As showed in Table 1, for the COSINE distance based classification, using a raw i-vector feature yields a better performance than the LDA dimension reduced feature. However, for the SVM based classification, it is important to obtain a compact feature set using the LDA for a better performance. On one side, reducing the dimensions of input features using the LDA helps reduce the number of model parameters and consequently helps to overcome the overfitting problem; on the other hand, reducing the dimensions may result in the loss of useful information, which could have been explored by nonlinear distance metric learning in order to improve the performance. We examine whether the LDA is helpful or not in improving the neural network based classification with and without pair-wise distance metric learning. The results are shown in table 10. From this table, it is evident that using the LDA as a front-end for the reduction of dimensions ("LDA+M2_L_2") improves the performance (on comparison with the results of "M2_L_2" in Table 3). When the model includes the pair-wise distance metric learning regularization, the performance was further improved ("LDA+M2_DM"). However, compared to the results of "M2_DM" in table 5, the full length i-vector is still preferred as the input.

Table 10
Performance of i-vector with LDA for dimension reduction for neural network model systems (identification error rate in %)

Model	Train	Validation	Test
LDA+M2_L_2	9.18	16.25	16.92
LDA+M2_DM	8.59	15.46	16.30

5 Conclusion

In this study, we proposed a pair-wise distance metric learning algorithm for a neural network based language identification. The metric learning imposed constraints on the feature representation space using a certain geometric structure, i.e., by decreasing the pair-wise intra-class variation while increasing the inter-class variation. These constraints aided in improving the generalization ability of the model. We have tested the pair-wise distance metric learning for the cross-entropy based training of the neural network model, and it showed an encouraging improvement. Although current training and testing data corpus is based on small data sets, it is possible to extend our work to large training and testing sets, since pair-wise distance metrics can be more accurately described using a large number of samples. One of the tasks for the future will be to test this idea on a SLI task using large training and test data sets.

In this paper, the distance metric is defined as a cosine distance between two feature vectors (samples). In the definition, the intra-class vectors were pushed to be near (similar) with cosine angle 0 while the inter-class vectors were pushed to be far away with cosine angle π. We have also examined a case where the inter-class vectors were pushed to be orthogonal (with a cosine angle $\pi/2$). The performance was a little worse compared to our current setting. Besides the cosine distance metric, other distance metrics may be used to define the geometric structure of the representation space, e.g. Euclidian distance metric, Riemannian distance metric (Belkin et al, 2003), etc. Instead of using the pair-wise distance metric, other high-order geometric structures of representation space can also be explored as the regularization for parameter learning, e.g., geometric structures of triple samples or local neighborhood samples. In the future, we will further investigate the introduction of new geometric structures to the parameter space for a more efficient model learning.

References

Nair, V., Hinton, G., 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proc. of the 27th International Conference on Machine Learning, 807-814.

Belkin, M., Niyogi, P., 2003. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373-1396.

Bishop, C. M., 2006. Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York.

Bottou, L., 2012. Stochastic Gradient Descent Tricks. In book of Neural Networks Tricks of the Trade (2nd ed.), Springer, 421-436.

Chopra, S., Hadsell, R., LeCun, Y., 2005. Learning a similarity metric discriminatively with application to face verification. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 1, 539-546.

Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011a. Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19, 788-798.

Dehak, N., Torres-Carrasquillo, P., Reynolds, D., Dehak, R., 2011b. Language Recognition via I-vectors and Dimensionality Reduction. In Proc. of Interspeech, 857-560.

Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS) 9, 249-256.

Gonzalez-Dominguez, J., Lopez-Moreno, I., Sak, H., et al., 2014. Automatic language identification using long short-term memory recurrent neural networks. In Proc. of Interspeech 2014, 2155-2159.

Guillaumin, M., Verbeek, J., Schmid, C., 2009. Is that you? Metric learning approaches for face identification. In Proc. of the IEEE 12th International Conference on Computer Vision, 498-505.

Hinton, G., 2010. A Practical Guide to Training Restricted Boltzmann Machines. UTML TR 2010-003, University of Toronto.

Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T., Kingsbury, B., 2012a. Deep Neural Networks for Acoustic Modeling in Speech Recognition: The shared views of four research groups. IEEE Signal Processing Magazine 29 (6), 82-97.

Hinton, G., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2012b. Improving neural networks by preventing co-adaptation of feature detector. arXiv preprint arXiv:1207.0580.

Hu, J., Lu, J., Tan, Y., 2015. Deep Transfer Metric Learning. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 325-333.

Lasagne, http://lasagne.readthedocs.io/en/latest/index.html

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. of the IEEE 86 (11), 2278-2324.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436-444.

Lee, K., Li, H., Deng, L., et al., 2016. The 2015 NIST Language Recognition Evaluation: the Shared View of I2R, Fantastic4 and SingaMS. In Proc. of Interspeech, 3211-3215.

Tong, A., Greenberg, C., Martin, A., et al., 2016. Summary of the 2015 NIST Language Recognition i-Vector Machine Learning Challenge. In Proc. of Odyssey, 297-302.

Li, H., Ma, B., Lee, K., 2013. Spoken Language Recognition: From Fundamentals to Practice. Proceedings of the IEEE 101 (5), 1136-1159.

Li, H., Ma, B., Lee, C., 2007. A Vector Space Modeling Approach to Spoken Language Identification. IEEE Transactions on Audio, Speech and Language Processing 15 (1), 271-284.

Lopez-Moreno, L., Gonzalez-Dominguez, J., Plchot, O., Martinez, D., Gonzalez-Rodriguez, J., Moreno, P., 2016. Automatic language identification using deep neural networks. In Proc. of ICASSP 2016, 5337-5341.

Ma, B., Tong, R., Li, H., 2007a. Discriminative Vector for Spoken Language Recognition. In Proc. of ICASSP IV, 1001-1004.

Ma, B., Li, H., Tong, R., 2007b. Spoken Language Recognition with Ensemble Classifiers. IEEE Transactions on Audio, Speech and Language Processing 15 (7), 2053-2062.

Montavon, G., 2009. Deep Learning for Spoken Language Identification. In NIPS workshop on Deep Learning for Speech Recognition and Related Applications.

NIST, https://lre.nist.gov

Prince, S., Elder, J., 2007. Probabilistic linear discriminant analysis for inferences about identity. IEEE International Conference on Computer Vision, 1-8.

Ranjan, S., Yu, C., Zhang, C., Kelly, F., Hansen, J., 2016. Language recognition using deep neural networks with very limited training data. In Proc. of ICASSP, 5830-5834.

Richardson, R., Reynolds, D., Dehak, N., 2015a. A Unified Deep Neural Network for Speaker and Language Recognition. In Proc. of Interspeech, 1146-1150.

Richardson, R., Reynolds, D., Dehak, N., 2015b. Deep Neural Network Approaches to Speaker and Language Recognition. IEEE Signal Processing Letters 22 (10), 1671-1675.

Sadjadi, S. O., Pelecanos, J. W., Ganapathy, S., 2015. Nearest neighbor discriminant analysis for language recognition. In Proc. of ICASSP, 4205-4209.

Shen, P., Lu, X., Liu, L., Kawai, H., 2016. Local Fisher discrimiant analysis for spoken language identification. In Proc. of ICASSP, 5825-5829.

Sidorov, G., Gelbukh, A., Gomez-Adorno, H., Pinto, D., 2014. Soft Similarity and Soft Cosine Measure: Similarity of Features in Vector Space Model. Computacion y Sistemas 18 (3), 491-504.

http://www.csie.ntu.edu.tw/ cjlin/libsvm/

Sugiyama, M., 2006. Local Fisher discriminant analysis for supervised dimensionality reduction. In Proc. of ICML, 905-912.

Sun, H., Nguyen, T., Wang, G., Lee, K., Ma, B., Li, H., 2016. I2R Submission to the 2015 NIST Language Recognition I-vector Challenge. In Proc. of Odyssey, 311-318.

theano, http://deeplearning.net/software/theano/

Weinberger, K., Blitzer, J., Saul, L., 2006. Distance Metric Learning for Large Margin Nearest Neighbor Classification. Advances in Neural Information Processing Systems 18, 1473-1480.

Weinberger, K., Saul, L., 2009. Distance Metric Learning for Large Margin Classificatio. Journal of Machine Learning Research 10, 207-244.

Xing, E., Ng, A., Jordan, M., Russell, R., 2002. Distance Metric Learning, with application to Clustering with side-information. Advances in Neural Information Processing Systems 16, 521-528.

Yu, D., Deng, L., 2015. Automatic Speech Recognition A Deep Learning Approach, Springer-Verlag London.