Manifold Regularized Deep Neural Networks

Vikrant Singh Tomar, Richard C. Rose

Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

Abstract

Deep neural networks (DNNs) have been successfully applied to a variety of automatic speech recognition (ASR) tasks, both in discriminative feature extraction and hybrid acoustic modeling scenarios. The development of improved loss functions and regularization approaches have resulted in consistent reductions in ASR word error rates (WERs). This paper presents a manifold learning based regularization framework for DNN training. The associated techniques attempt to preserve the underlying low dimensional manifold based relationships amongst speech feature vectors as part of the optimization procedure for estimating network parameters. This is achieved by imposing manifold based locality preserving constraints on the outputs of the network. The techniques are presented in the context of a bottleneck DNN architecture for feature extraction in a tandem configuration. The ASR WER obtained using these networks is evaluated on a speech-in-noise task and compared to that obtained using DNN-bottleneck networks trained without manifold constraints.

Index Terms: manifold learning, deep neural networks, speech recognition, tandem feature extraction

1. Introduction

Deep neural networks (DNNs) have recently been widely applied to acoustic modeling in automatic speech recognition (ASR) [1, 2]. These applications involve DNNs being used for both discriminative feature extraction and for discriminative estimation of observation probabilities in ASR systems. In feature extraction, a DNN is used in tandem with a separate Gaussian mixture based hidden Markov model (GMM-HMMs) ASR system [3, 4]. The HMM observation probabilities are obtained by mapping posterior probabilities from a DNN to state level observation probabilities in a hybrid DNN-HMM ASR system [1, 5].

While the DNN parameters in both of these configurations can be trained using error back-propagation (EBP), there are two issues relating to the associated learning algorithms that have recently received considerable attention in the literature. The first is the negative impact resulting from local minima in the error surface associated with EBP optimization. There have been a number of approaches that have been presented for reducing the impact of these local minima. These include restricted Boltzmann machines (RBMs) based generative pre-training approaches [5], layer-by-layer discriminative pre-training approaches [6], and various regularization procedures including, for example, the use of drop-out training and the use of rectified linear units (ReLUs) in place sigmoid based network nonlinearities [7]. The second issue is the impact of the local structure of the feature space on EBP optimization, including the features provided to the input of the DNN as well as features produced at the output of hidden layers of the DNN. There is empirical evidence that propagation of information through a DNN is well behaved when the feature space can be characterized as having strong local structure embedded within a high dimensional space [8]. Furthermore, denoising auto-encoders have been described as a mechanism for learning a low-dimensional manifold based representation of the training data, albeit without explicitly imposing any such constraints [9, 10].

This paper presents a manifold learning approach to regularizing DNN back-propagation training. Local relationships among speech feature vectors along a low dimensional manifold are preserved under this approach by using a manifold regularized optimization criteria for estimating network parameters. This is achieved by imposing manifold based locality preserving constraints on the outputs of the network. The notion of discriminative manifold learning, which maximizes separability between classes along a manifold, is also investigated. The techniques are presented in the context of a bottleneck DNN architecture for feature extraction in a tandem ASR configuration. The bottleneck features are then used to train a GMM-HMM system.

This class of approaches relies on the assumption that speech features lying on a low-dimensional manifold embedded in a high-dimensional acoustic space can be characterized by their local neighborhood relationships [11–13]. In discriminative manifold learning, it is assumed that the separability between classes for speech feature vectors on this manifold can also be maintained by exploiting local neighborhood relationships [14–16]. The use of discriminative manifold learning algorithms has been investigated in a number of applications including the estimation of manifold constrained linear dimensionality reducing feature space transformations [14, 17, 18]. While manifold learning approaches are known to require relatively high computational complexity, methods for approximate nearest neighbor computation have been investigated for making these approaches more efficient [19, 20].

There have been several recent efforts to apply manifold based constraints for regularization in semi-supervised learning scenarios. A framework for manifold regularization was introduced in [21] and applied to estimating the parameters of a regularized least squares classifier. Manifold regularization in single hidden layer multilayer perceptrons for a phone classification task is investigated in [22]. Various aspects of manifold based semi-supervised embedding is applied to deep learning for a hand-written character recognition task in [23]. While the manifold based regularization approach presented in this paper is related to these efforts, the methods presented in Section 2 rely on supervised training of the manifold related parameters.

The authors thank Nuance Foundation for providing financial support for this project. The experiments were conducted on the Compute-Canada and Calcul-Quebec supercomputing clusters. The authors are thankful to the consortium for providing the access and support.

In other related work, deep recurrent neural networks based methods have been explored to preserve temporal correlation structure in the speech sequence data [24].

2. Application of Manifold learning to Deep Neural Networks

This section summarizes the discriminative manifold learning framework and its incorporation into DNN training. First, a brief introduction to discriminative manifold learning is provided in Section 2.1. Then, Section 2.2 introduces the incorporation of these manifold learning techniques as a regularization framework in DNN training using EBP. A detailed discussion of discriminative manifold learning techniques as applied to feature analysis in ASR can be found in [14].

2.1. Discriminative Manifold Learning

Manifold learning methods assume that speech features lie on or close the surface of a low-dimensional manifold embedded in the high-dimensional acoustic space. This assumption is supported by the argument that speech is produced by the movement of loosely constrained articulators [13, 25]. Motivated by this, manifold learning can be used to constrain acoustic modeling algorithms using the local relationships among feature vectors that are characterized by these manifolds. This is usually realized by embedding the feature vectors into one or more graphs [18]. An optimality criterion is then formulated that includes the preservation of manifold based relationships.

Given a set of feature vectors, $\boldsymbol{X}$, a graph, $\mathcal{G} = \{\boldsymbol{X}, \boldsymbol{\Omega}\}$, is used to characterize the manifold based relationships among feature vectors. Here, $\boldsymbol{\Omega} = [\omega_{ij}]$ is a matrix containing the weights over edges connecting graph nodes and is referred to as the affinity matrix. The weight, ω_{ij}, on an edge connecting two nodes, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, provides a measure of closeness between the feature vectors. These weights govern various characteristics of a graph, including structure, connectivity and compactness. The graph based relationships are usually characterized using the Euclidean distance based Gaussian heat kernel as

$$\omega_{ij} = \begin{cases} exp\left(\frac{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{\rho}\right) & ; \quad e(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \\ 0 & ; \quad \text{Otherwise} \end{cases}, \quad (1)$$

where ρ is the kernel scale parameter and the function $e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ indicates whether $\boldsymbol{x}_i$ lies in the near neighborhood of $\boldsymbol{x}_j$. Closeness to a vector $\boldsymbol{x}_i$ can be determined either by membership in its k-nearest neighborhood (kNN) or distance to $\boldsymbol{x}_i$ within some finite radius r.

For a graph $\mathcal{G} = \{\boldsymbol{X}, \boldsymbol{\Omega}\}$, a measure of the graph's scatter for a mapping $f : \boldsymbol{x} \rightarrow \boldsymbol{y}$ can be defined as

$$F_{\mathcal{G}}(\boldsymbol{Y}) = \sum_{i,j} ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2 \omega_{ij}. \quad (2)$$

Eq. (2) provides a measure of compactness over the graph nodes. An algorithm designed to minimize this measure will constrain the outputs, $\boldsymbol{y}$'s, to preserve the manifold based local relationships defined by the affinity weight matrix $\boldsymbol{\Omega}$.

It is shown in [14, 18] that the manifold based framework can be further extended by adding a discriminative measure that penalizes the local relationships between feature vectors not belonging to the same class. These methods attempt to preserve the within-class manifold based local structure, while at the same time discriminate among classes along the manifold. To this end, the feature vectors are embedded into separate intrinsic and penalty graphs. The intrinsic graph, $\mathcal{G}_{int} = \{\boldsymbol{X}, \boldsymbol{\Omega}_{int}\}$, characterizes the within-class manifold based relationships among the feature vectors. The penalty graph, $\mathcal{G}_{pen} = \{\boldsymbol{X}, \boldsymbol{\Omega}_{pen}\}$, characterizes the relationships among feature vectors belonging to different classes. The elements of the intrinsic and penalty graph affinity matrices are defined as

$$\omega_{ij}^{int} = \begin{cases} exp\left(\frac{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{\rho}\right) & ; \quad C(\boldsymbol{x}_i) = C(\boldsymbol{x}_j), e(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \\ 0 & ; \quad \text{Otherwise} \end{cases} \quad (3)$$

and

$$\omega_{ij}^{pen} = \begin{cases} exp\left(\frac{-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{\rho}\right) & ; \quad C(\boldsymbol{x}_i) \neq C(\boldsymbol{x}_j), e(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1 \\ 0 & ; \quad \text{Otherwise} \end{cases} \quad (4)$$

where $C(\boldsymbol{x}_i)$ refers to the class or label of vector $\boldsymbol{x}_i$.

In discriminative manifold learning, the objective function is designed to minimize the scatter of the intrinsic graph, while at the same time maximizing the scatter of the penalty graph. This can be achieved by minimizing the difference of the intrinsic and penalty scatter measures,

$$\begin{aligned} F_{\mathcal{G}_{diff}}(\boldsymbol{Y}) &= \sum_{i,j} ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2 \omega_{ij}^{int} - \sum_{i,j} ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2 \omega_{ij}^{pen}, \\ &= \sum_{i,j} ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2 \omega_{ij}^{diff}, \end{aligned} \quad (5)$$

where $\omega_{ij}^{diff} = \omega_{ij}^{int} - \omega_{ij}^{pen}$.

Eq. (5) provides a measure of locality preservation and class separation over the graphs. The contribution of the discriminative constraints is twofold. The intrinsic graph, characterized by the weights in Eq. (3), only preserves the manifold based local relationships among feature vectors belonging to the same class. This is contrary to the affinity matrix used in conventional manifold learning as given for example in Eq. (1), where classes of feature vectors are not taken into account. In addition, the penalty graph, characterized by the weights in Eq. (4), penalizes the relationships among feature vectors not belonging to the same class. This approach has been shown to increase robustness of manifold based techniques [14, 18]. Motivated by these results, this paper applies discriminative manifold constraints to DNN training. In Section 2.2, the expression in Eq. (5) is applied as a regularization term in the objective function of DNN parameter estimation.

2.2. Manifold Regularized Deep Neural Networks

A DNN is a feed-forward artificial neural network with multiple hidden layers. If $f : \boldsymbol{x} \rightarrow \boldsymbol{y}$ defines the mapping from the inputs, $\boldsymbol{x}_i, i = 1 \ldots N$, to the outputs, $\boldsymbol{y}_i$, the network is trained to find the optimal set of weights, $\boldsymbol{W}$, to minimize a loss, $V(\boldsymbol{x}_i, \boldsymbol{t}_i, f)$, between the outputs of the network and the targets, $\boldsymbol{t}_i$,

$$\mathcal{F}(\boldsymbol{W}) = \frac{1}{N} \sum_{i=1}^{N} V(\boldsymbol{x}_i, \boldsymbol{t}_i, f). \quad (6)$$

The weights of the network are updated by multiple iterations of stochastic gradient descent method over the training set. A layer-by-layer pre-training is sometimes used to improve the convergence behavior of EBP training in DNNs [5, 26].

An extension of DNN training is proposed that incorporates the discriminative manifold learning based constraints

discussed in Section 2.1. Due to their layered architectures, DNNs are thought to be capable of learning complex non-linear relationships between feature vectors. Thus, if DNNs are able to learn underlying manifold based non-linear relationships among data vectors while learning to discriminate between speech classes, one can hope to achieve better ASR models. To this end, incorporating a manifold learning based regularization term in the DNN's objective function may be expected to improve performance. For example, a modified objective function investigated in this work is,

$$\mathcal{F}(\boldsymbol{W};\boldsymbol{Y}) = \frac{1}{N}\sum_{i=1}^{N}\left\{ V(\boldsymbol{x}_i,\boldsymbol{t}_i,f) + \frac{1}{k}\gamma\sum_{j=1}^{2k}||\boldsymbol{y}_i-\boldsymbol{y}_j||^2\omega_{ij}^{diff}\right\}, \tag{7}$$

where γ is the regularization trade-off parameter, and k denotes the number of nearest neighbors associated with each feature vector and is used in populating the intrinsic and penalty matrices, $\boldsymbol{\Omega}_{int}$ and $\boldsymbol{\Omega}_{pen}$. The objective function in Eq. (7) imposes manifold based constraints on the outputs of the DNN. For each feature vector $\boldsymbol{x}_i$ of the network, $2k$ neighbors of x_i, k for the intrinsic graph and k for the penalty graph, are used to enforce the manifold based constraints.

Note that Eq. (7) takes very similar form to Eq. (6), and thus the weights of the network can be updated in a similar fashion using the EBP and gradient descent,

$$\nabla_{\boldsymbol{W}}\mathcal{F}(\boldsymbol{W};\boldsymbol{Y}) = \sum_{i}^{N}\frac{\partial\mathcal{F}(\boldsymbol{W};\boldsymbol{Y})}{\partial\boldsymbol{y}_i}\frac{\partial\boldsymbol{y}_i}{\partial\boldsymbol{W}}, \tag{8}$$

where $\nabla_{\boldsymbol{W}}\mathcal{F}(\boldsymbol{W};\boldsymbol{Y})$ is the gradient of the objective function with respect to the DNN weight matrix, $\boldsymbol{W}$.

3. Experimental Study

This section describes the experimental study performed to evaluate the impact of manifold constrained training of a bottleneck tandem DNN on ASR WER. The speech corpus and neural network configurations are described in Section 3.1 followed by the ASR WER results in Section 3.2. Section 3.3 provides a discussion on the computational complexity of these techniques, and the configuration of the systems and infrastructure used for the experiments. The importance of including the penalty graph in the manifold based regularization framework is discussed in Section 3.4.

3.1. Task Domain and System Configuration

The experiments in this work are conducted on the Aurora-2 corpus. The Aurora-2 mixed-condition set is used for training [27] . This set represents a simulated speech-in-noise task created by adding four different noise types to connected digit utterances spoken in a quiet environment. There are a total for 8440 noise-corrupted utterances collected from 55 male and 55 female speakers in the training set. For baseline system, the standard Aurora-2 ASR configuration specified in [27] is used. This corresponds to using 13-dimensional Mel filtered cepstrum coefficients (MFCCs) with first and second order differences as features. The ASR system is trained using whole word continuous density HMM (CDHMM) models with 16 states per word-model, 3 states for the silence model and 1 state for the short pause model. There were 11 word-based CDHMM models, and 180 states in total. Each state was modeled by a mixture of 3 Gaussians. The CDHMM states, obtained by a single pass force-alignment using the baseline CDHMM system, are used as class labels for supervision. The test set consists of features corrupted by four different noise types, namely subway, babble, car and exhibition hall, at signal-to-noise ratios (SNR) ranging from 5 to 20 dB, and clean speech. There are 1001 utterances in each noise-corrupted subset.

For all DNN experiments, the networks contain 5 hidden layers. At the input layer, 9 context frames of 13-dimensional MFCC features are concatenated to obtain a 117 dimensional input layer. The first four hidden layers have 1024 units each, the fifth hidden layer is the bottleneck layer with only 40 units. The number of units in the output layer is equal to the number of CDHMM states for Aurora-2. All hidden layers use the sigmoid function as units, whereas the output layer units use the soft-max nonlinearity. After training, the output features are taken from the bottleneck layer. Finally, a 40 to 39 dimension de-correlating feature transformation using principal component analysis (PCA) is performed on the bottleneck features before feeding those into the GMM-HMM ASR system trained using maximum-likelihood criteria.

Two separate DNNs are trained for evaluating the impact of manifold regularization on the ASR performance of the output features. The first is a DNN without any regularization. This is referred to as the baseline-DNN in the rest of the paper. The second is the manifold regularized DNN (MRDNN). The loss, $V(\boldsymbol{x}_i,\boldsymbol{t}_i,f)$, in objective functions in Eq. (6) and Eq. (7) is set to the cross-entropy loss between the outputs and targets of the network. Both of the networks are trained for 20 epochs over the training data. The starting learning rate is set to 0.05 and decreased exponentially with each epoch. For populating the manifold learning based affinity matrices as per Eq. (3) and Eq. (4), the Gaussian kernel scale factor, ρ, is taken to be 1000 and 3000 for the intrinsic and penalty graph respectively. These values are empirically derived on a development set, and are taken from previous work [15, 28]. The number of nearest neighbors, k, is also empirically derived and set to 30 for both intrinsic and penalty graphs.

3.2. ASR Results

ASR WER results for the Aurora-2 speech-in-noise task is given in Table 1. Four different tables are presented for four noise types, namely subway, babble, car and exhibition hall. For each noise type, ASR results at five different noise levels ranging from 20dB to 5dB SNR and clean speech are given. Each row of these tables presents ASR WERs obtained using a particular feature type. The first row, labeled "MFCC", presents the ASR WER for the baseline MFCC features. The second row, labeled "DNN", presents the ASR WERs for features obtained from a bottleneck DNN (baseline-DNN). The last row, labeled "MRDNN", presents the WERs when features are obtained from a manifold regularized bottleneck DNN. For all these cases, the GMM-HMM and DNN systems follow the configuration described in the Section 3.1.

Two major observations can be made from Table 1. The first is that all DNN derived features provide significant reductions in ASR WERs when compared to the MFCC features. The second observation is made by comparing the performance of DNN and MRDNN features. It is clear that the MRDNN features provide consistent improvements over features obtained from DNNs trained without manifold regularization. The relative WER improvements range from 8 to 38%.

Note that RBM based generative pre-training was also investigated to initialize the weights of the baseline-DNN system. Surprisingly, however, the pre-training did not provide

Table 1: WER for mixed noise training and noisy testing on Aurora-2 speech corpus for MFCC, DNN, and manifold regularized DNN (MRDNN) features. The best performance has been highlighted for each noise type per SNR level.

Noise	Technique	SNR (dB)				
		clean	20	15	10	5
Subway	MFCC	1.76	2.99	4.00	6.21	11.89
	DNN	0.98	1.17	1.91	3.13	6.42
	MRDNN	**0.83**	**0.95**	**1.63**	**2.55**	**5.89**
Babble	MFCC	1.83	3.31	4.37	7.97	18.06
	DNN	0.97	1.21	1.55	2.86	7.97
	MRDNN	**0.60**	**1.00**	**1.27**	**2.63**	**7.13**
Car	MFCC	1.99	2.77	3.36	5.45	12.31
	DNN	1.10	1.10	1.88	2.98	7.12
	MRDNN	**0.84**	**0.95**	**1.43**	**2.77**	**6.20**
Exhibition	MFCC	1.89	3.34	3.83	6.64	12.72
	DNN	0.89	1.18	2.13	3.70	9.10
	MRDNN	**0.56**	**1.02**	1.73	**2.72**	**7.68**

additional gains in the ASR WER performance. The consistent reduction in WERs using MRDNN as compared to the baseline-DNN suggests that the manifold regularization might present an alternative to other pre-training and regularization schemes.

3.3. Computational Complexity and Infrastructure

One drawback of manifold learning algorithms is the added computational complexity. This computational cost is two-fold for manifold regularized DNNs. The first source of this cost is the estimation of the weights of the affinity matrices in Eq. (3) and Eq. (4). The second source is forward propagation of $2k$ neighbors for each feature vector during EBP training. In previous work, the authors have demonstrated that computational complexity related to the calculation of the nearest neighbors search can be reduced by an order of magnitude using locality sensitive hashing for approximate nearest neighbor search without sacrificing ASR performance [19, 20]. The computational cost associated with EBP can be managed using various parallelization techniques for DNN training [2]. It should be noted that this added computational complexity only affects the training of the network, and has no impact during test phase when transforming the data using a trained network for recognition.

In this work, the networks are trained on Nvidia Tesla K20 graphics cards using python based tools developed on numpy, gnumpy and cudamat libraries [29, 30]. The manifold regularized DNNs took 11.5 hours in training for 20 epochs as compared to 2 hours for the DNNs trained without manifold constraints for the same number of training epochs.

3.4. Importance of the Penalty Graph

The manifold based constraints given in Eq. (5) and Eq. (7) have two components, namely intrinsic and penalty, for preserving within manifold nonlinear relationships between feature vectors and for penalizing between manifold relationships, respectively. The penalty graph component is responsible for enforcing the discrimination between classes along the manifold. In previous work, where these discriminative manifold learning methods are used for feature space transformation, inclusion of the penalty graph is found to be important [14, 15, 28]. DNNs, however, are inherently discriminative, hence a natural question arises if the penalty graph is adding any robustness to the manifold regularized DNNs.

To evaluate the importance of the penalty graph term in the manifold regularized DNN framework, a separate set of experiments are conducted by only including the components corresponding to the intrinsic graph. The gains achieved by including the penalty graph are found to be inconsistent for the different conditions present in the Aurora-2 dataset. Thus, it is not clear if there is any advantage to including the penalty graph term.

4. Discussion and Future Work

The study presented here raises a number of questions. The first question is whether the relative reductions in WER obtained using manifold regularized training generalize to other task domains. Preliminary experiments on the large vocabulary Aurora-4 task [31] have shown a 12.1% relative reduction in WER using a MRDNN in a bottleneck tandem network under noise-free conditions.

Another question is how manifold regularization compares to other techniques that have been used to regularize DNN training, for example, L2-regularization, and ReLU+dropout [7, 32]. Preliminary experiments have been conducted to compare the ASR WER performance of MRDNN with a DNN trained with L2-regularization constraints on its weights. On the Aurora-2 task, the L2-regularized DNN reduced the average WER for clean testing conditions to 0.92 as compared to 0.99 for the baseline-DNN. However, the performance is still significantly worse than that of the MRDNN, for which the average WER for clean test conditions is 0.79. The use of L2-regularization in combination with the manifold regularization for DNN training is a topic for future investigation.

Preliminary experiments are also performed for investigating the impact of using ReLU units instead of the sigmoid units for DNNs on the Aurora-2 task. In line with other research [33], the experiments show faster convergence of the networks and small gains in ASR WER performance when ReLU units are used in the baseline-DNN system. Future work will investigate the use of ReLU units in the manifold regularized DNN framework. Comparison of the manifold regularization for DNNs with dropout based regularization is also a topic of future research.

5. Conclusions

This paper has presented a technique to incorporate discriminative manifold learning based constraints into DNN training for acoustic feature extraction. The manifold based constraints are applied to DNN training in order to preserve the underlying low-dimensional manifold based relationships between feature vectors while minimizing the cross-entropy loss between network outputs and targets. The ASR performance of features obtained using these networks is evaluated on a speech-in-noise task and is compared to that obtained using bottleneck DNN networks trained without manifold constraints. The proposed approaches have shown relative WER reduction ranging from 8 to 32% as compared to an unregularized DNN. Preliminary work comparing the performance of the manifold regularized DNN framework with other regularization techniques have also been discussed. The consistent gains in ASR performance with MRDNNs might suggest the manifold regularization as an alternative to other regularization schemes.

Errata: In the published version the time taken in training MRDNNs is giving wrong. Later we developed a more efficient implementation. For the Aurora-2 set, each epoch of DNN training took 240 seconds and MRDNN training took 1200 seconds.

6. References

[1] G. Hinton, L. Deng, and D. Yu, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, pp. 1–27, 2012.

[2] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.-r. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, Dec. 2011, pp. 30–35.

[3] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ICASSP*, 2000, pp. 1–4.

[4] F. Grézl and M. Karafiát, "Probabilistic and bottle-neck features for LVCSR of meetings," *ICASSP*, 2007.

[5] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-Dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Transactions on Audio, Speech and Language processing*, pp. 1–13, 2012.

[6] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in Context-Dependent Deep Neural Networks for conversational speech transcription," in *IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, Dec. 2011, pp. 24–29.

[7] G. Dahl, T. Sainath, and G. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," *ICASSP*, 2013.

[8] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," *Neural Networks*, pp. 6–9, 2012.

[9] P. Vincent, H. Larochelle, and I. Lajoie, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[10] D. Erhan, P. Manzagol, and Y. Bengio, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in *Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR Workshop and Conference Proceedings*, vol. 5, 2009, pp. 153–160.

[11] X. He and P. Niyogi, "Locality preserving projections," in *Neural Information Processing Systems (NIPS)*, 2002.

[12] H. Tang, S. M. Chu, and T. S. Huang, "Spherical discriminant analysis in semi-supervised speaker clustering," *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers on - NAACL '09*, no. June, p. 57, 2009.

[13] A. Jansen and P. Niyogi, "A geometric perspective on speech sounds," University of Chicago, Tech. Rep., 2005.

[14] V. S. Tomar and R. C. Rose, "A family of discriminative manifold learning algorithms and their application to speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language processing*, vol. 22, no. 1, pp. 161–171, 2014.

[15] ——, "Application of a locality preserving discriminant analysis approach to ASR," in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*. Montreal, QC, Canada: IEEE, Jul. 2012, pp. 103–107.

[16] ——, "A correlational discriminant approach to feature extraction for robust speech recognition," in *Interspeech*, Portland, OR, USA, 2012.

[17] D. Cai, X. He, K. Zhou, and J. Han, "Locality sensitive discriminant analysis," in *International Joint Conferences on Artificial Intelligence*, no. 60633070, 2007, pp. 708–713.

[18] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction." *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[19] V. S. Tomar and R. C. Rose, "Efficient manifold learning for speech recognition using locality sensitive hashing," in *ICASSP: IEEE International Conference on Acoustics Speech and Signal Processing*, Vancouver, BC, Canada, 2013.

[20] ——, "Locality Sensitive Hashing for Fast Computation of Correlational Manifold Learning based Feature space Transformations," in *Interspeech*, Lyon, France, 2013, pp. 2–6.

[21] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold Regularization : A Geometric Framework for Learning from Labeled and Unlabeled Examples," *Journal of Machine Learning*, vol. 1, pp. 1–36, 2006.

[22] A. Subramanya and J. Bilmes, "The semi-supervised switchboard transcription project." *Interspeech*, 2009.

[23] J. Weston, F. Ratle, and R. Collobert, "Deep learning via semi-supervised embedding," *Proceedings of the 25th international conference on Machine learning - ICML '08*, pp. 1168–1175, 2008.

[24] J. Chen and L. Deng, "A Primal-Dual Method for Training Recurrent Neural Networks Constrained by the Echo-State Property," *Proc. ICLR*, pp. 1–17, 2014.

[25] A. Jansen and P. Niyogi, "Intrinsic Fourier analysis on the manifold of speech sounds," in *ICASSP: IEEE International Conference on Acoustics Speech and Signal Processing*, 2006.

[26] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," *Proc. ICASSP*, 2013.

[27] H. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," *Speech Recognition: Challenges for the*, 2000.

[28] V. S. Tomar and R. C. Rose, "Noise aware manifold learning for robust speech recognition," in *ICASSP: IEEE International Conference on Acoustics Speech and Signal Processing*, Vancouver, BC, Canada, 2013.

[29] V. Mnih, "CUDAMat : a CUDA-based matrix class for Python," Department of Computer Science, University of Toronto, Tech. Rep., 2009.

[30] T. Tieleman, "Gnumpy : an easy way to use GPU boards in Python," Department of Computer Science, University of Toronto, Tech. Rep., 2010.

[31] N. Parihar and J. Picone, "Aurora Working Group : DSR Front End LVCSR Evaluation," European Telecommunications Standards Institute, Tech. Rep., 2002.

[32] L. Deng, J. Li, J. Huang, K. Yao, and D. Yu, "Recent advances in deep learning for speech research at Microsoft," *ICASSP 2013*, pp. 0–4, 2013.

[33] M. Zeiler, M. Ranzato, R. Monga, and M. Mao, "On Rectified Linear Units for Speech Processing," pp. 3–7, 2013.