IMPROVING DEEP NEURAL NETWORKS USING STATE PROJECTION VECTORS OF SUBSPACE GAUSSIAN MIXTURE MODEL AS…

Conference Paper · December 2014

DOI: 10.13140/2.1.2191.4884

CITATIONS
0

READS
240

2 authors, including:

Murali Karthick Baskar
Brno University of Technology
7 PUBLICATIONS **3** CITATIONS

SEE PROFILE

IMPROVING DEEP NEURAL NETWORKS USING STATE PROJECTION VECTORS OF SUBSPACE GAUSSIAN MIXTURE MODEL AS FEATURES

Murali Karthick B [*] *and S. Umesh*

Department of Electrical Engineering
Indian Institute of Technology - Madras
{ee13s010, umeshs}@ee.iitm.ac.in

ABSTRACT

Recent advancement in deep neural network (DNN) has surpassed the conventional hidden Markov model-Gaussian mixture model (HMM-GMM) framework due to its efficient training procedure. Providing better phonetic context information in the input gives improved performance for DNN. The state projection vectors (state specific vectors) in subspace Gaussian mixture model (SGMM) captures the phonetic information in low dimensional vector space. In this paper, we propose to use state specific vectors of SGMM as features thereby providing additional phonetic information for the DNN framework. To each observation vector in the train data, the corresponding state specific vectors of SGMM are aligned to form the state specific vector feature set. Linear discriminant analysis (LDA) feature set are formed by applying LDA to the training data. Since bottleneck features are efficient in extracting useful discriminative information for the phonemes, LDA feature set and state specific vector feature set are converted to bottleneck features. These bottleneck features of both feature sets act as input features to train a single DNN framework. Relative improvement of 8.8% for TIMIT database (core test set) and 9.7% for WSJ corpus is obtained by using the state specific vector bottleneck feature set when compared to the DNN trained only with LDA bottleneck feature set. Also training Deep belief network - DNN (DBN-DNN) using the proposed feature set attains a WER of **20.46%** on TIMIT core test set proving the effectiveness of our method. The state specific vectors while acting as features, provide additional useful information related to phoneme variation. Thus by combining it with LDA bottleneck features improved performance is obtained using the DNN framework.

Index Terms— Deep neural network, SGMM, bottleneck features, state specific vectors

1. INTRODUCTION

In recent years, advent of DNN has shown improved performance in the ASR systems. The feature information is utilized efficiently by the DNN framework providing better acoustic models. DNN can model both uncorrelated features like Mel scale filter cepstral coefficients (MFCC) and correlated filter-bank features. These characteristics of DNN inspire us to proceed further investigation in the front-end processing, to obtain efficient features for the DNN framework. In [1], MFCC features were projected to higher dimensions by splicing, then dimensionality reducing is performed using linear discriminant analysis (LDA) and further decorrelated using maximum likelihood linear transformation (MLLT). Introduction of bottleneck layer in DNN framework leads to better classification. Thus features (bottleneck features) are extracted from this layer producing discriminable information [2]. Bottleneck features are enhanced in [3] using stacked auto encoders and DBN systems. Bottleneck features models both GMM & DNN effectively by representing the data in efficient manner [4].

Speaker normalization over features is yet another approach to enhance DNN. This aspect is considered in [5] using i-vectors which carry information about speakers in a low dimensional vector. These vectors were appended to filter-bank / MFCC features to perform speaker normalization as well as phone classification simultaneously.

SGMM [8] is a recently proposed acoustic modeling technique which uses state projection vectors or state specific vectors (SSVs) to estimate parameters for each state. These vectors are claimed to contain phonetic variability of speech in a low dimensional vector. Intuitively, each SSV represents a particular point in phonetic space while i-vectors capture information about a particular speaker. DNN obtains speaker characteristics from i-vector to perform speaker normalization. A similar approach can be used in DNN to provide phonetic information to DNN using SSVs for improved classification.

This motivated us to provide phonetic information using SSVs to model DNN, instead of using i-vectors. SSVs are fed to DNN to provide additional information about phonetic-context. Appending SSV to input features in a similar way as appending i-vectors to input features [5] did not give appreciable results.

Considering the above constraints in using SSVs as features, we have proposed an approach to introduce SSVs as complementary features along with input (LDA / feature space maximum likelihood linear regression (fMLLR)) features to DNN. In our approach, input features and SSV features are fed into a DNN containing bottleneck, to obtain bottleneck features separately for both feature set. During training, SSV bottleneck features and input bottleneck features are fed to a single DNN simultaneously, thereby optimizing the tied state classification in the output layer. This method is an efficient way of fine-tuning DNN by providing better contextual information.

Experiments performed using TIMIT [6] and WSJ [7] substantiates our hypothesis by giving improved performance compared to DNN trained with input (LDA / fMLLR) features. The efficacy of our method is also tested for DBN-DNN model using TIMIT dataset. A competitive performance of **20.46%** WER is obtained using TIMIT core test set, proving the effectiveness of our method .

The rest of this paper is organized as follows. Section 2 explains about the concept of SGMM, motivation in using state specific vectors as features and its importance. Section 3 describes about the

[*] Thanks to Sriranjani for her helpful comments and suggestions

experimental study done to validate our method. Section 4 discusses the inferences and interpretations of the obtained results.

2. STATE SPECIFIC VECTOR (SSV) FEATURES

2.1. SGMM

SGMM [8] is a HMM based system, in which the state dependent parameters (mean, covariance and mixture weights) are not estimated independently. Instead, we use a globally shared low dimensional subspace S from which these models are trained. $\mathbf{M}_i$ is the mean subspace from which the mean of the GMM $\boldsymbol{\mu}_{ji}$ can be obtained using the state specific vector $\mathbf{v}_j$. The mixture weights ω_{ji} can be obtained from the weight vector $\mathbf{w}_i$ using $\mathbf{v}_j$. The basic expression of SGMM is given as:

$$p(x/j) = \sum_{i=1}^{I} \omega_{ji}\, \mathcal{N}(x; \boldsymbol{\mu}_{ji},\ \boldsymbol{\Sigma}_i) \tag{1}$$

$$\boldsymbol{\mu}_{ji} = \mathbf{M}_i\, \mathbf{v}_j \tag{2}$$

$$\omega_{ji} = \frac{exp\,(\mathbf{w}_i^T\, \mathbf{v}_j\,)}{\sum_{i'}^{I} exp\,(\mathbf{w}_{i'}^T\, \mathbf{v}_j\,)} \tag{3}$$

where $p(x/j)$ is the likelihood of observation vector $x \in \mathbb{R}^{\mathbb{D}}$ in D dimensional space. Each GMM has the same number of mixture components I for all states. The parameters $\mathbf{M}_i$, $\boldsymbol{\Sigma}_i$(covariance) and $\mathbf{w}_i$ are shared across all the states ($\Sigma_{ji} = \Sigma_i$) as mentioned above in equations 2 and 3.

2.2. Importance of State Specific vectors (SSVs)

In SGMM, each tied state distribution is represented as a low-dimensional vector ($\mathbf{v}_j$), which characterizes the co-ordinates of that subspace. The mean ($\boldsymbol{\mu}_{ji}$) and weight parameter (ω_{ji}) for each state j is obtained by projecting these vectors using projection matrix $\mathbf{M}_i$.

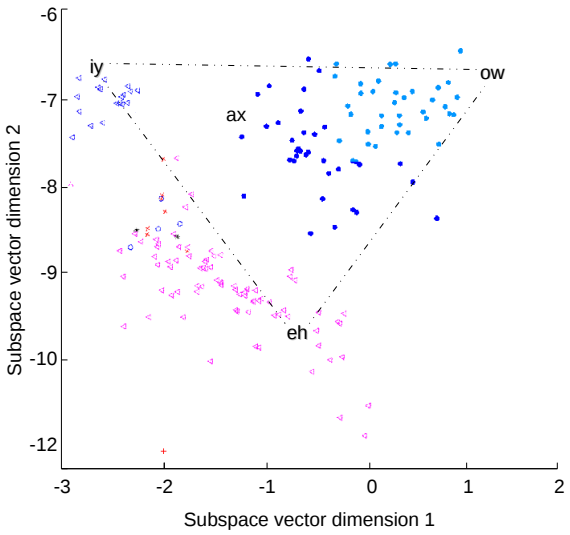

Fig. 1. Scatter plot of the 1st and 2nd dimension of state specific vectors for TIMIT

In [9], it is stated that SSVs bear a lot of information about phonetic context. A two dimensional plot of SSVs shown in figure 1 which portrays the center phones of each tied-state are positioned in their corresponding location as in vowel triangle forming small clusters. This closely relates to the characterization of articulatory information in features. This indicates that SSVs containing significant phonetic information, can be used as features.

2.3. State Specific vector feature extraction

Initially a SGMM is built using input features to obtain state specific vector for each tied-state. Since we expect a single SSV mapped to each tied-state, SGMM is trained without substates. The train data is then aligned frame wise to its corresponding tied-state label using SGMM. Each aligned state label is then mapped to their corresponding SSV, resulting in the formation of SSV feature set.

Figure 2 shows the block diagram implementation of the proposed approach. In the figure 2 the input (LDA or fMLLR features) are passed on SGMM training block to obtain state specific vectors. These vectors are then converted to SSV features and then passed to a proposed DNN framework to train the model by using SSV as complementary features.

2.3.1. *Constraints in using SSV along with features*

1. Similar approach as in [5], was experimented by appending SSV features along with input features for each corresponding frame. This provides an input data with increased dimensions and is used for training DNN. Same method is used to modify test data by getting reference transcription from first pass decode of SGMM. This method failed to give better performance. Thus SSV cant be used in similar way as i-vectors are used while training DNN.
2. SSV has to be post-processed for giving discriminative information about the phonetic-context.

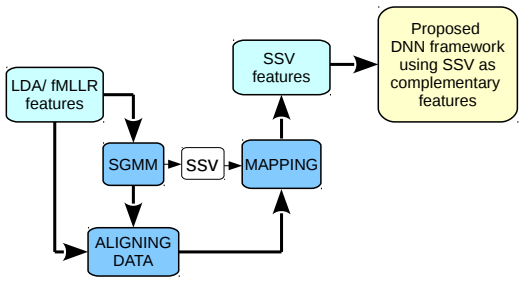

Fig. 2. Overall framework of proposed approach

2.4. Proposed Framework Description

The above constraints are handled using our proposed method to use SSV as complementary features to train DNN.

Figure 3 shows the proposed neural network framework for training $SSV - BN$ as complementary features. As shown in figure 3, both input (LDA or fMLLR) features and their corresponding SSV features (SSV_{LDA}/SSV_{fMLLR}). Neural network containing bottleneck layer (b1 & b2) are trained with input features and their SSV features to get their corresponding bottleneck features. These features are then combined into a single feature list and sent to the final DNN system to obtain improved acoustic model.

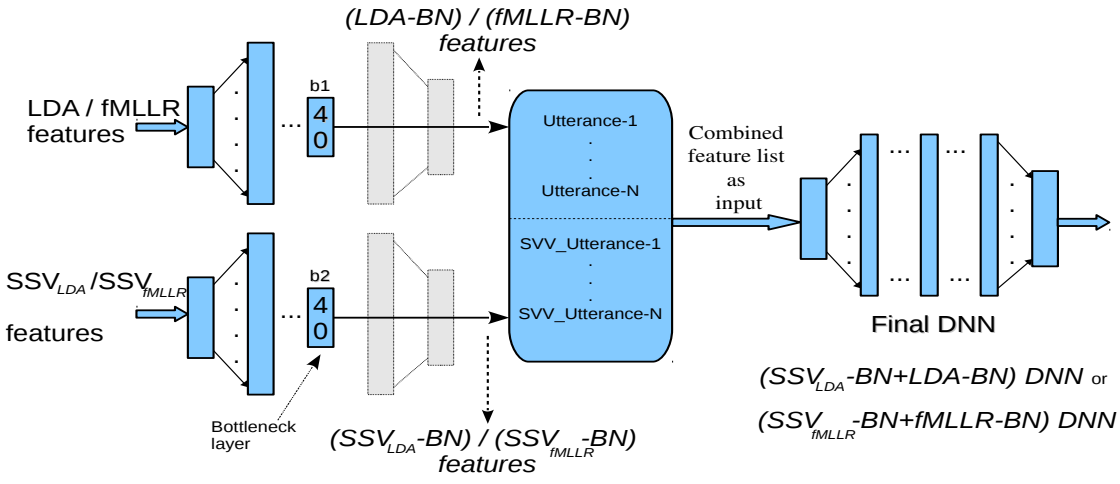

Fig. 3. DNN framework using SSV as complementary features

1. The initial SGMM formed using input (LDA / fMLLR) features create corresponding set of SSV features (SSV_{LDA}/ SSV_{fMLLR}). They are fed into DNN having bottleneck layer to obtain ($SSV_{LDA} - BN/SSV_{fMLLR} - BN$). They are mentioned generally as ($SSV - BN$) features in this paper. The bottleneck layer b2 discriminates the state specific information to obtain $SSV - BN$ features. These features provide better discriminative information about each tied-state.
2. Similarly, input features are passed to the bottleneck layer b1 to extract input bottleneck features i.e., ($LDA-BN/fMLLR-BN$) features.
3. The utterance index of each $SSV - BN$ feature is renamed and included in the feature set as a separate utterance along with the input bottleneck features. This approach helps in training each class label with phonetic information obtained from both $SSV-BN$ features and input bottleneck features.
4. The combined feature set containing both $SSV - BN$ features and input bottleneck features forms our proposed set of features used for DNN training.
5. Reference class labels to train final DNN are obtained from the GMM-HMM model trained only using input features.
6. The final DNN is now trained using the reference class labels and our proposed feature set to obtain the improved acoustic models.
7. During decoding, first pass transcription is not needed since SSV are not included in test data. Thus our proposed method handles the constraint mentioned in 2.3.1.
8. The main advantage of our system is that it provides additional phonetic context information into the system leading to better classification.

3. EXPERIMENTAL STUDY

The experiments are performed using Kaldi Speech recognition toolkit [10].

3.0.1. Speech Corpus

TIMIT [6] and Wall street journal (WSJ) [7] corpus are used for our experiments.

TIMIT: 3.1 hours (3696 utterances) of train data and 0.1 hours (192 utterances) of test data were used for our experiments. 1 silence phone and 38 voiced phones are used while training and testing. A phone based bi-gram language model is used for phone decoding.

WSJ: To make tuning faster WSJ0 SI-84 (84 speakers/ 7240 utterances) is used as train data [7]. Evaluation is done using Nov 92 DARPA evaluation test set for 5k closed vocabulary data using bi-gram language model.

3.0.2. Front-end processing

Input speech is windowed using 25ms window with an overlap of 15ms to extract 13 dimensional MFCC features from the speech signal. c_0, velocity, acceleration coefficients are appended to form 39 dimensional features for training the basic GMM-HMM model. Splicing is done over the 13 dimensional MFCC features to get 117 dimensions and then reduced to 40 dimensions to obtain LDA features. The LDA features are then transformed using fMLLR to obtain fMLLR features.

3.0.3. Baseline DNN system

Conventional GMM-HMM model is built by following the Kaldi recipe for both datasets. MLLT is applied over the LDA features to obtain a LDA+MLLT model. SAT based on fMLLR is applied over LDA+MLLT to obtain LDA+MLLT+SAT model. The number of tied-states and Gaussian mixtures are used from Kaldi recipe, i.e., TIMIT and WSJ uses 1905 states and 2055 states for LDA+MLLT model, 1954 and 2027 for LDA+MLLT+SAT model.

Two different types of features are involved in DNN training:

1. A baseline DNN system built using LDA features with class labels obtained from LDA+MLLT model
2. Baseline model built using fMLLR features with class labels for speech frames obtained from LDA+MLLT+SAT system.

Table 1 shows the configuration of baseline DNN of TIMIT and WSJ dataset using two types of features. The input contains 360 dimen-

sions for all cases, due to the splice width of 9 over 40 dimensional input features. Experiments are performed with network containing "$tanh$" activation function in the hidden layers and a softmax component at the output layer. Input layer and hidden layers contain affine transform pre-conditioner for de-correlating the provided input. In case of no improvement in frame classification accuracy the learning rate is reduced by half after each iteration.

For TIMIT, fixed minibatch size of 128 and a initial learning rate of 0.015 is kept constant for 15 epochs. These parameters are tuned as per Dan's DNN implementation in Kaldi [10]. For WSJ0-SI84 dataset, 0.005 is initial learning rate and final rate is 0.0005 with a minibatch size of 256 was used. No momentum or regularization is provided in both datasets

Table 1. Baseline system details

Configuration	TIMIT		WSJ	
	LDA	fMLLR	LDA	fMLLR
Input dimension (units)	360	360	360	360
No of Hidden layers	2	2	6	6
Hidden dimension (units)	300	300	512	512
Output dimension (units)	1905	1954	2055	2027
# Parameters (millions)	1.2	1.7	5.4	5.2

3.0.4. Generating state specific vectors

In our experiment, SGMM training involves usage of two feature types (LDA or fMLLR) for both TIMIT and WSJ datasets.

SGMM is trained using LDA or fMLLR features along with the alignment information obtained from LDA+MLLT model or LDA+MLLT+SAT model. In TIMIT, the number of tied-states in SGMM using LDA+MLLT model is 2340 and 2531 in case of LDA+MLLT+SAT model. SGMM on WSJ has 2686 tied-states using LDA+MLLT model and 2817 in case of LDA+MLLT+SAT model. The substate creation is disabled to acquire single state specific vector for each tied-state. The Gaussian components are fixed at 400 for both cases. The phone space dimension is empirically found as 40 for each state specific vector. The training phase involves estimation of state specific vectors and thus state specific vectors using LDA features (SSV_{LDA}) and fMLLR features (SSV_{fMLLR}) are obtained. in SSV_{LDA} or SSV_{fMLLR}.

3.0.5. DNN with bottleneck

Bottleneck features are extracted using DNN having a bottleneck layer of 40 dimensions in between two hidden layers, which is chosen empirically. The configuration of DNN having bottleneck layer is: 2 hidden layers each of 1024 dimensions for TIMIT, and 3 hidden layers with 1024 dimensions for WSJ. These networks are trained using input features (LDA / fMLLR) to obtain input bottleneck features ($LDA-BN/fMLLR-BN$). The same configuration is used for (SSV_{LDA}/SSV_{fMLLR}) features to obtain $SSV-BN$ features i.e. ($SSV_{LDA}-BN/SSV_{fMLLR}-BN$) features

3.0.6. Final DNN training

Different utterance id is assigned to each $SSV_{LDA}-BN$ feature and combined with $LDA-BN$ feature set. This modified feature set is used to train the final DNN. Fine tuning is done by varying the hidden layer count and its dimensions. ($SSV_{LDA}-BN+LDA-BN$) DNN is trained with 5 hidden layers each of 300 units for TIMIT and 7 hidden layers each containing 1024 units for WSJ. Increase in network configuration is noted due to increase in data size (double the data size). Network is also trained only using $LDA-BN$ features to obtain $LDA-BN$ DNN. These models are needed to compare the result of proposed method. The ($SSV_{fMLLR}-BN+fMLLR-BN$) DNN and $fMLLR-BN$ DNN are also obtained by following the similar procedure in building models using LDA features.

4. RESULTS & DISCUSSION

The Task id is assigned in the table 2 for easy reference to the DNN models in this paper. Task C and D represents our proposed method which uses ($SSV_{LDA}-BN+LDA-BN$) and ($SSV_{fMLLR}-BN+fMLLR-BN$) features.

TIMIT: Task C achieves a WER of 22.1 % i.e., **8.4 %** relative improvement compared to Task A. In case of Task D the error rate reduces to 21.9 % i.e., ***8.8%*** relative improvement compared to Task B. The $SSV_{LDA}-BN$ or $SSV_{fMLLR}-BN$ features show a consistent improvement in performance along with both type of feature sets.

WSJ: The Task D model achieves **10.5 %** relative improvement over Task B. A relative improvement of **9.7 %** is obtained by Task C over Task A.

The proposed methods in Task C and D show consistent improvement for LDA and fMLLR features. This is due to the increase in frame classification accuracy during each iteration due to the efficient classification of class labels.

Figure 4 shows the increment in performance for Task C and Task D when the number of hidden layers were increased.

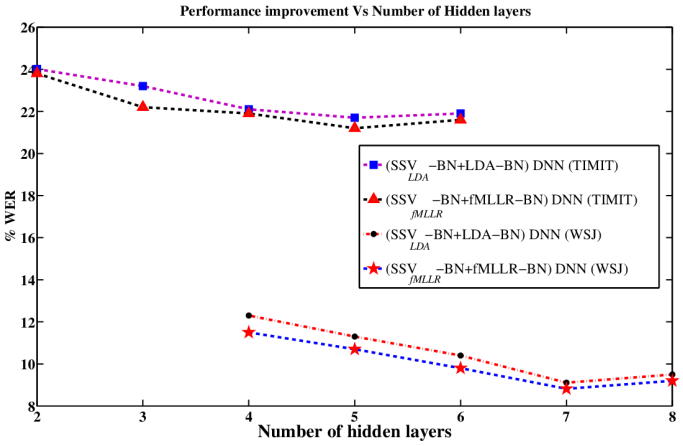

Fig. 4. Performance variation by tuning the number of hidden layers for TIMIT & WSJ in DNN

Results of TIMIT baseline system using fMLLR features is replicated to match Kaldi baseline results. The results obtained for WSJ using DNN can't be compared with Kaldi results since the Kaldi DNN is trained over 284 speaker train data while we train using 84 speaker train data. Table 2 shows the result obtained using the proposed framework for TIMIT and WSJ.

4.1. DBN-DNN using TIMIT

Experiments on deep belief network - DNN (DBN-DNN) are performed only for TIMIT dataset. The Baseline DBN-DNN model is

built by following the Kaldi recipe. The major difference between the previous DNN experiments and DBN-DNN is the process of pre-training done using DBN.

***DBN training*:** Layer-wise pre-training is done through a stack of six restricted Boltzmann machines (RBM). The momentum is increased from 0.5 to 0.9 for better convergence by decreasing the learning rate.

Table 2. Comparison of % WER for various DNN trained on TIMIT and WSJ corpus using $LDA-BN$ features, $SSV-BN$ features and fMLLR features on 192 utterances core test set and Nov'92-5k evaluation set

Task	Models	TIMIT		WSJ	
		%WER	% RI	%WER	% RI
	Baseline DNN (LDA)	25.63	-	11.43	-
	Baseline DNN (SAT+fMLLR)	24.84	-	10.39	-
A	$LDA-BN$ DNN	24.13	-	10.09	-
B	$fMLLR-BN$ DNN	24.02	-	9.85	-
C	$(SSV_{LDA}-BN$ $+LDA-BN)$ DNN	**22.1**	*8.4*	**9.11**	*9.7*
D	$(SSV_{fMLLR}-BN$ $+LDA-BN)$ DNN	**21.9**	*8.8*	**8.82**	*10.5*

% RI - % Relative Improvement

DBN-DNN training: The train dataset is now split into train set (90%) and validation set (10%). 40 dimensional input data is spliced with a splice width of 11 to obtain 440 dimensional input. Each frame is now trained using cross-entropy error criterion. Classification of each frame to a tied-state label is performed during this stage. DBN-DNN training is done using the stochastic gradient descent (SGD) approach. Learning rate of 0.008, minibatch size of 256 is used and no momentum or regularizer is set. The six trained RBMs with their determined weights are used as hidden layers for DNN training. Baseline model is trained by keeping 6 RBMs as hidden layers and the proposed model is trained using 7 hidden layers (6 RBMs + one extra hidden layer).

Table 3. Comparison of % WER for DBN-DNN trained on TIMIT using and fMLLR features on core test set

Models	Config	% WER	% RI
Baseline (SAT+fMLLR)	6 hl, 1024 ns	21.43	-
fMLLR-BN DNN	6 hl, 1024 ns	21.36	0.3
SSV-BN + fMLLR-BN DNN	7 hl, 1024 ns	**20.46**	4.2

hl - hidden layers, ns - nodes, RI - Relative Improvement

Table 3 shows the performance of DBN-DNN on TIMIT dataset. The experiments are performed using $fMLLR-BN$ features. The $(SSV_{fMLLR}-BN+fMLLR-BN)$ DNN model achieves **20.46 %** WER i.e., **4.2 %** relative improvement compared to $fMLLR-BN$ DNN model. The baseline result 21.43% is replicated to match standard Kaldi result on TIMIT dataset. The proposed method achieves a relative improvement of 3.9% over the standard baseline model.

This performance improvement is due to increase in frame classification accuracy. i.e., Baseline model's average frame classification accuracy is 78.7% and proposed method $(SSV_{fMLLR}-BN+fMLLR-BN)$ achieves 80.76%. This shows that, the proposed approach works better on DBN-DNN for TIMIT as shown in Table 2.

Due to lack of computational resources DBN-DNN training on WSJ0-SI84 is skipped. However the same steps used in TIMIT can be followed for WSJ for performing DBN-DNN training.

5. CONCLUSION & FUTURE WORK

In this paper, the problem in using state specific vectors as features directly or along with other feature type is handled. A novel approach has been proposed to handle this issue using state specific vector of SGMM as complementary features for training DNN and DBN-DNN. This provides better classification performance when compared to DNN trained only with input (LDA / fMLLR) features. This statement is supported by performing various experiments on WSJ and TIMIT datasets. The proposed system was found to give consistent improvement for both LDA and fMLLR features. The main advantage of our method is by proving state specific vectors as features, additional phonetic context information related to each tied-state is provided to the system. This helps in improving the discriminative capacity of the system, thereby improving the performance of the system. This work can be further extended for other datasets to obtain better performance.

6. REFERENCES

[1] S. P. Rath, D. Povey, K. Veselỳ, and J. Cernockỳ, "Improved feature processing for deep neural networks.," in *INTERSPEECH*, pp. 109–113, 2013.

[2] F. Grezl, M. Karafiát, S. Kontár, and J. Cernockỳ, "Probabilistic and bottle-neck features for lvcsr of meetings.," in *ICASSP (4)*, pp. 757–760, 2007.

[3] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4153–4156, IEEE, 2012.

[4] F. Grézl, M. Karafiát, and K. Veselỳ, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, Florence, Italy*, 2014.

[5] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pp. 55–59, IEEE, 2013.

[6] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, p. 27403, 1993.

[7] D. B. Paul and J. M. Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*, pp. 357–362, Association for Computational Linguistics, 1992.

[8] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow,

R. C. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model-a structured model for speech recognition," *Computer Speech & Language*, vol. 25, pp. 404–439, Apr. 2011.

[9] S. H. Ghalehjegh and R. C. Rose, "Phonetic subspace adaptation for automatic speech recognition.," in *ICASSP*, pp. 7937–7941, 2013.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, *et al.*, "The kaldi speech recognition toolkit," in *Proc. ASRU*, pp. 1–4, 2011.