TANDEM-Bottleneck Feature Combination using Hierarchical Deep Neural Networks

Mirco Ravanelli [1*], Van Hai Do[2*], Adam Janin[3]

[1]Fondazione Bruno Kessler, Trento, Italy
[2]School of Computer Engineering, Nanyang Technological University, Singapore
[3]International Computer Science Institute, Berkeley, USA

mravanelli@fbk.eu, dova001@@i2r.a-star.edu.sg, janin@icsi.berkeley.edu

Abstract

To improve speech recognition performance, a combination between TANDEM and bottleneck Deep Neural Networks (DNN) is investigated. In particular, exploiting a feature combination performed by means of a multi-stream hierarchical processing, we show a performance improvement by combining the same input features processed by different neural networks.

The experiments are based on the spontaneous telephone recordings of the Cantonese IARPA Babel corpus using both standard MFCCs and Gabor as input features.

Index Terms: Deep Neural Networks, TANDEM feature, bottleneck feature.

1. Introduction

Despite the remarkable progress of the last decade, Automatic Speech Recognition (ASR) is still far from the human level performance, especially in noisy environments and in presence of spontaneous speech. One promising research direction recently arisen is represented by the so called Deep Neural Networks (DNN) [1], usually formed by feed-forward Neural Networks (NN) with many hidden layers properly introduced inside the speech recognition process [2, 3]. Interesting features of both NNs and DNNs are the inherent discriminative nature, a fewer model assumption about the input distribution, the non-linearity of the classification and the flexibility of merging multiple input streams. Beside DNNs directly inheriting all the advantages of standard NNs, a deeper architecture allows a more complex non-linear transformation able to significantly increase the representational power of the network.

Although the theoretical benefits of DNNs have been known for many decades [4], their practical use was limited to shallow architectures, since NNs with many hidden layers were very difficult to train, as weights to layers far from the targets tended to remain fairly constant during training. Nevertheless, some noteworthy algorithmic progress, such as the introduction of an unsupervised greedy layer-wise training method based on Restricted Boltzmann Machines (RBM) [5], has renewed interest in their use.

Although NNs and DNNs can be used in many ways inside a speech recognition system, the most popular approaches can be divided into two main categories: hybrid approaches and TANDEM approaches. Hybrid architectures [6] exploit a discriminative trained Multi-Layer Perceptron (MLP) to estimate HMM state posterior probabilities instead of using a standard Gaussian Mixture Model (GMM). TANDEM approaches [7], on the other hand, augment the input to a GMM-HMM system with features derived from a neural network based transformation. Typically TANDEM techniques are based on a MLP estimating the posterior probability over a set of target phones and, after a logarithmic smoothing followed by a linear feature reduction step, such processed posteriors are directly used as feature for a GMM-HMM speech recognizer. Other interesting TANDEM-like systems are TRAPS [8] and HATs [9] techniques, which aim to extract long-term information about phones from critical bands. Another recent progress is represented by the bottleneck approach [10], which can be considered as an architecture modification of the typical TANDEM speech recognition. Bottleneck features are based on a MLP with several hidden layers in which one has a small number of neurons compared to size of the other layers. The processing performed by such a neural network can be interpreted as a non-linear NN based dimensionality reduction. The advantage is that the feature reduction is achieved implicitly, without using any linear techniques like Linear Discriminant Analysis (LDA) or Principal Component Analysis (PCA). A DNN extension of bottleneck features has been explored in [11], while a hierarchical bottleneck architecture was firstly explored in [12, 13, 14] to account for long-term speech modulations.

One limitation of many past DNN-related works could be the prevalent use of single input feature streams, while the exploration of DNN based features combination schemes still remain an under-explored research direction to investigate. We indeed believe that complementary information, crucial to improve ASR performance, could be generated by not only different input features [15, 16], but even by different neural network architectures. Although the first one is an intuitive and straightforward peculiarity, already explored in many past works, the second one still represents a largely under-investigated direction to investigate. In the present work, we propose a feature transformation and combination scheme based on a Multi-Stream Hierarchical Deep Neural Network able to efficiently merge TANDEM and bottleneck features streams. The proposed architecture, which can be considered as an extension of the standard Hierarchical Bottleneck approaches [12, 13, 14, 17], has been alternatively fed by Mel-Frequency Cepstral Coefficients (MFCCs) and Gabor features [18]. The results show that, for both input features, TANDEM and bottleneck streams provide partially not-redundant information which can be exploited to increase the ASR performance.

The rest of the paper is organized as follows. Section 2 discusses the proposed architecture. Experimental setup and experimental results are respectively reported in section 3 and 4, while section 5 concludes the paper.

[*] This work was performed while the authors were visiting ICSI

978-1-4799-4219-0/14/$31.00 ©2014 IEEE

2. Proposed Architecture

The method proposed here aims at providing a powerful feature transformation and feature combination module able to exploit the potential complementary information generated by two different DNN architectures. The proposed Multi-Stream Hierarchical DNN solution, depicted in Fig.1, has initially been fed by MFCCs. To make this work stronger and more meaningful, the same architecture has later been fed by Gabor features. Although many features have been proposed in literature, we preferred to focus on such input streams because of their expected complementary behaviour [19], due to the different extraction processes. Since Cantonese is a tonal language, pitch (F0) and Probability of Voicing (PoV) were also added to each input stream.

As shown in Fig.1, the input features are simultaneously processed by a TANDEM (a) and a bottleneck (b) NNs. The choice of such architectures could be ascribed to the expected not-redundant information: posterior features represent speech at a phone state level, while bottleneck features represent data in a more compact way.

The later level of the hierarchy, formed by the bottleneck NN (c), performs a non-linear TANDEM-bottleneck stream combination and a non-linear dimensionality reduction. Together with this, a long-term analysis of the input features is also achieved by selecting several sparse input frames. The DNN derived features are finally concatenated with the standard MFCCs to generate the features used for speech recognition training and test.

In the proposed solution, we preferred to use a deeper architecture for the first level neural networks (a) and (b), since we believe they play a crucial role in the overall feature extraction process. Nevertheless, also the neural network (c) plays a remarkable role, but since it performs a simpler task (the merger operates on already processed input streams), a shallower architecture should be sufficient. For this reason, we chose to take advantage of RBM pre-training only over the first level NNs.

The reminder of this section firstly analyzes the input features (sub-sections 2.1, 2.2, 2.3), while the role of TANDEM (a) and bottleneck (b) NNs is deepened in sub-sections 2.4 and 2.5. The effectiveness of the RBM pre-training is examined in 2.6, while the feature combination is discussed in 2.7.

2.1. Cepstral Features

From the audio files, Vocal Tract Length Normalization (VTLN) is applied to the log-power spectrum, estimating the warping factors using a text-independent GMM classifier trained on the acoustic training corpus. The warped power spectrum is computed every 10 ms over a Hamming window of 25 ms and, after the integration of the spectrum by 23 triangular Mel filters, their logarithm is computed. 13 MFCCs are then selected, after the DCT based de-correlation.

2.2. Gabor Features

Gabor are auditory inspired features computed by convolving the log mel spectrogram of speech with a set of 2-D modulation filters, with varying extents and tuned to different rates and directions [18].

In the context of this paper, we used a set of 59 Gabor filters firstly proposed in [20] and further considered in [21]. The adopted filter bank parameters (e.g., filter spacing, lowest and highest modulation frequencies in time and frequency), were empirically determined based on a speech recognition task and were directly inherited from [21]. For each feature frame, each 2-D Gabor filter was convolved with a set of 23 log mel spectrum frequency bands, with frequencies ranging from 64 to 4000 Hz. A vector of $59 * 23 = 1357$ represents the initial feature dimensions. The initial dimensionality was reduced to 449 through a selective sampling of the filter outputs aiming at decreasing the redundancy of the selected features. This procedure, described in [21], helps in significantly increasing the ASR performance with Gabor features.

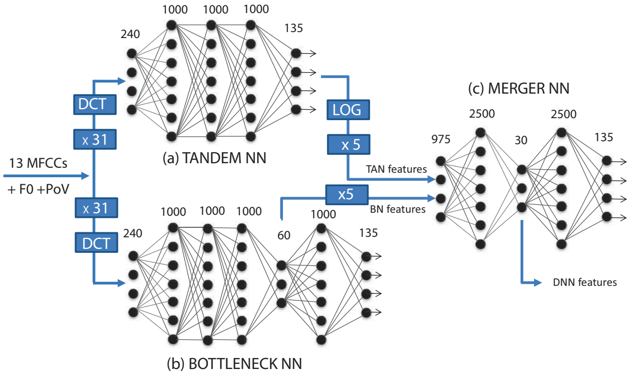

Figure 1: The proposed Multi-Stream Hierarchical DNN architecture for MFCCs input features.

2.3. Pitch and Probability of Voicing

Since Cantonese is a tonal language, the pitch plays an important role. In this work, we extract pitch (F0) and probability-of-voicing (PoV) using Columbia's sub-band autocorrelation classification (SAcC) [22]. The algorithm estimates such features by means of a MLP classifier trained on the principal components of the sub-band autocorrelations. F0 and PoV are simply pasted with both MFCCs and Gabor features to respectively form a 15 and 451 dimensional vector.

2.4. TANDEM Features

The TANDEM approach [7] is a data processing paradigm in which a MLP is used to estimate the posterior probabilities of each sub-word class given the data. In order to better fit the subsequent GMM, the probabilities are usually logarithmized and de-correlated by Principal Components Analysis (PCA) or Linear Discriminant Analysis (LDA), which also can provide dimensionality reduction. In addition, to take into account the dynamic nature of speech, context windows are often applied to the original input streams after a mean and variance normalization step. These features, derived by a non-linear transformation, are known in literature as posterior or TANDEM features. Although many published results show contradictory performance trends over different tasks and corpora [7, 10], the performance of such features could be in general similar or sometimes even below that of standard mel-cepstral coefficients. The main advantage is that posterior observations exhibit a large amount of not-redundant information with standard MFCCs, making convenient a combination between them. Another advantage is that TANDEM systems are fully compatible with standard GMM-HMM ASR systems, which still represent the dominant paradigm despite the growing interest towards DNN based hybrid approaches [3]. A prominent limitation could be ascribed to the increased risk of data over-fitting: since the same labels are typically used for both neural network and GMM training,

a possible lack of generalization can occur.
In this work, the linear feature reduction step is replaced by the non-linear processing performed by the bottleneck NN (c).Furthermore, a context of 31 frames has been selected for the MFCCs-fed network (a). The total input dimensionality of such a NN is hence $15 * 31 = 465$, reduced to 240 by means of a DCT transformation. For the Gabor-fed MLP, the role of the context window was investigated under some in-depth analysis not reported here, showing no benefit deriving from the use of such context information. On the other hand, Gabor features inherently have a context description since a set of 2-D spectro-temporal filters are exploited in the feature extraction process. The total input dimensionality, considering also pitch (F0) and Probability of Voicing (PoV), is in this case 451. Finally, the output dimensionality of all the NNs of the proposed architecture is 135, corresponding to all the tonal phones considered for Cantonese.

2.5. Bottleneck Features

The bottleneck approach, introduced by Grézl et al [10], can be interpreted as a non-linear dimensionality reduction. Bottleneck features are indeed based on an MLP in which one of the internal layers has a small number of hidden units, relative to the size of the other hidden layers. This layer creates a constraint in the network able to generate compressed features forcing the dimensionality reduction. The bottleneck features tend to outperform the posterior probabilities [10, 23] and, similarly to TANDEM posteriors, such features are usually concatenated with MFCCs. One advantage of such approach is that, together with a feature transformation, a non-linear feature reduction able to outperform linear methods such as LDA, HLDA and PCA is implicitly performed.
A disadvantage is ascribable to the increased risk of data underfitting, since some information could be lost inside the bottleneck layer. In the proposed architecture, the bottleneck MLP (b) has been fed with the same input features of the TANDEM NN (a), considering also the same context window. A linear activation function for the bottleneck layer has been adopted since, as first observed in [24], this provides a slight improvement in ASR performance.

2.6. Restricted Boltzmann Machine Pre-Training

The idea behind pre-training is to replace random initialization of the parameters with a smarter and more convenient weight initialization, without which it is difficult to usefully employ more than one or two hidden layers using back-propagation training. A number of pre-training techniques have been explored previously, including both discriminative approaches [25, 26] and unsupervised methods based on a stack of Restricted Boltzmann Machines (RBM) [5]. We have chosen to use the unsupervised approach, which was successfully used to improve speech recognition performance for both TANDEM [27] and bottleneck approach [11] as well as for hybrid ASR [3]. In this method, an efficient greedy layer-wise technique independently trains each adjacent pair of NN layers as an RBM, providing better initial neural network weights. The derived weights are directly used to initialize the MLP and, by means of fine-tuning phase carried out using the standard back-propagation algorithm, a joint optimization of all the layers is performed.

2.7. Feature Combination

The proposed DNNs architecture achieves a features combination exploiting the bottleneck NN (c). The role of such a NN is similar to the one fulfilled by the standard hierarchical bottleneck features [12, 13, 14], where a second bottleneck MLP performs a non-linear dimensionality reduction as well as a long-term analysis of the speech. Contrary to the past works, in this case the long-term modulations are jointly analyzed for both TANDEM and bottleneck streams since a concatenation between such features has been previously performed.
The long-term MLP (c) adopted here is based on a 5 frames sparse context windows, sampling the combined input at the positions: -10, -5, 0, +5, +10. Considering that, for both MFCCs and Gabor fed NNs, the output dimensionality of TANDEM (a) and bottleneck (b) features is respectively 135 and 60, the total input dimensionality of (c) is $(135 + 60) * 5 = 975$. The output features are extracted from the 30-dimensional bottleneck layer. The 30-dimensional DNN features derived by the NN (c) are finally concatenated with 13 standard MFCCs to generate the set of 43 observations used for speech recognition training and test.

3. Experimental Setup

3.1. Corpus Description

In this work, we used the spontaneous telephone recordings of the IARPA Babel Program Cantonese language collection (full-language pack, release babel101b-v0.4c). The actual speech portion used for NNs and ASR training is about 80 hours for training, while 10 hours are used for test. From the training data, a small cross validation set (12 hours) has been derived for neural network training purposes.

3.2. Neural Network Training

The neural network training and pre-training phases are based on the GPU version of the TNet toolkit [28]. During the pre-training phase, the weights between the first two hidden layers are initialized by a RBM (Gaussian-Bernoulli), using a learning rate of 0.005 with 10 pre-training epochs. For the remaining RBMs (Bernoulli-Bernoulli), we used a learning rate of 0.05 with 5 pre-training epochs. The fine-tuning phase is performed by a stochastic gradient descent optimizing cross-entropy loss function. The learning rate is kept fixed at 0.005 as long as the single epoch increment in cross-validation frame accuracy is higher than 0.5%. For the subsequent epochs, the learning rate is being halved until the cross-validation increment of the accuracy is less than the stopping threshold of 0.1%.

3.3. Recognition system

We used a standard GMM-HMM speech recognizer, trained and tested with HTK [29]. The acoustic model contains 5k context-dependent tied states with 16 Gaussian components each. Every phone is modeled by a three state left-to-right continuous density HMM with diagonal covariance matrix. The mixtures are initialized with a single Gaussian per state, and subsequently more components are progressively allocated by splitting the Gaussians at every Baum-Welch iteration until the pre-determined total number of components is reached. We also estimated a tri-gram language model on the training transcripts, applying Kneser-Ney smoothing. The vocabulary size is 19k, OOV rate is 2.6% while the perplexity, computed over test transcriptions, is 110. In the test phase, a multi-pass Viterbi decoding of each input signal is adopted.

4. Experimental Results

First, we carried out some experiments to find a reasonable NN architecture for both MFCCs and Gabor fed networks. The role of RBM pre-training was also investigated. Later we focused on the neural network combination, emphasizing the performance obtained by merging TANDEM-bottleneck architectures. Since Cantonese is a character based language, all the results shown in the present section are reported in terms of Character Error Rate (CER%).

The baseline considered here relies on a feature vector composed by 13 MFCCs, 13 first derivatives, 13 second derivatives, pitch (F0), and PoV. Acoustic and language models, as well as decoding procedure are the same used for the proposed techniques. The baseline CER(%) result is 50.1%. Note that the system performance is affected by both a poor signal to noise ratio of the recorded signals and by the spontaneous and conversational nature of the telephone speech, which make this transcription task very challenging.

4.1. Hidden Layer Optimization

For the reasons explained in section 2, we preferred to focus on the optimization of the TANDEM (a) and bottleneck (b) NNs, while the architecture of the merger (c) is kept fixed. In table 1, are reported the results obtained progressively increasing (from 1 to 3) the number of hidden layers of the neural networks (a) and (b) for both MFCCs and Gabor input features. Each added layer is composed by 1000 neurons. For the MLP (b), each layer is added before the bottleneck. The DNN-derived features, later concatenated with 13 MFCCs, are respectively taken from the 30-dimensional bottleneck networks (c), alternatively enabling only TANDEM or bottleneck input streams.

Input	**NN**	**1**		**2**		**3**	
Features	**Arch.**	**RND**	**PT**	**RND**	**PT**	**RND**	**PT**
MFCCs	*TAN*	48.5	48.3	47.5	46.9	47.5	46.5
MFCCs	*BN*	48.4	48.2	47.3	46.6	47.0	46.1
GABOR	*TAN*	50.1	49.9	48.5	48.0	48.1	47.3
GABOR	*BN*	49.7	49.9	47.8	47.5	47.3	46.8

Table 1: CER(%) obtained progressively adding 1000 neurons hidden layers. The column RND refers to a random initialization of the weights, PT column shows the results with RBM pre-training, while TAN and BN respectively refers to TANDEM and bottleneck streams.

Note that both MFCCs and Gabor based MLPs significantly outperform the baseline system achieving a relative improvement of respectively 8% and 6.6%. Another observation is that the performance increment achieved when increasing the number of hidden layers. For instance, from 1 to 3 pre-bottleneck hidden layers we reached a relative improvement of 4.4% for MFCCs and 6.2% for Gabor features. Furthermore, bottleneck features slightly outperforms TANDEM features, confirming the trend firstly observed in [10]. An interesting role is also played by the pre-training techniques. Although RBM pre-training is in general helpful in all the architectures tried, it seems to be more effective in case of deeper neural networks, where a smart initialization is crucial to find a good training solution.

4.2. Feature Combination Performance

Table 2 reports the performance achieved when concatenating TANDEM and bottleneck streams. Rows “TAN” and “BN” respectively refers to the case where only TANDEM or bottleneck streams are enabled, while row “TAN+BN” shows the performance archived by the combination of such features. Column “MFCCs+GAB” shows the results obtained when an additional merger (an MLP with the same architecture of NN (c)) has been placed in cascade to (c) in order to combine the DNN streams derived by both MFCCS and Gabor input features.

NN	**Input features**		
Arch.	**MFCCs**	**GAB**	**MFCCs+GAB**
TAN	46.5	47.3	45.7
BN	46.1	46.8	45.4
TAN+BN	45.4	46.2	45.0

Table 2: CER(%) obtained combining TANDEM and bottleneck feature streams.

The proposed combination scheme leads to an increased level of performance, proving that TANDEM and bottleneck streams generate partially not redundant information even when the NNs are fed by the same input features. This performance improvement, is significant and coherent for both MFCCs and Gabor input features. We believe that the partially not redundant information provided by the different architectures could be ascribe to the different type of processing performed by the two NN. The TANDEM MLP, in effect, generates information at a phone-state level and, in spite of the log-smoothing, the data-set tends to be over-fitted while bottleneck features, representing information in a more compact way, could potentially under-fit the training-set. The merger, which combines such features, tries, in some way, to reconstruct the information lost in the bottleneck MLP exploiting the information derived by the TANDEM stream. A further improvement is also achieved, as expected, by merging together MFCCs and Gabor streams.

5. Conclusions

In this paper, a combination between TANDEM and bottleneck Deep Neural Networks (DNN) is explored. Results show that, using a hierarchical processing to combine different neural network architectures, leads to an improvement in the ASR performance even when the same input features are adopted. This improvement, which is consistent for both MFCCs and Gabor input features, showed for the first time that is possible to obtain partially not redundant information not only from different input features (as already discussed in many past works) but even from different neural network architectures.

6. Acknowledgment

We would like to thank Prof. Nelson Morgan for helpful comments on this paper. This work is supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory.contract number W911NF-12-C-0014. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

7. References

[1] N. Morgan, “Deep and wide: Multiple layers in automatic speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 7–13, 2012.

[2] A. Mohamed, G. Dahl, and G. E. Hinton, “Deep Belief Networks for phone recognition,” in *NIPS*, 2009.

[3] 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*, vol. 20, no. 1, pp. 30–42, 2012.

[4] C.M Bishop, “Neural Networks for Pattern Recognition,” in *Oxford University Press*, 1995.

[5] G.E. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[6] H. Bourlard and N. Morgan, “Continuous speech recognition by connectionist statistical methods,” *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 893–909, 1993.

[7] H. Hermansky, D. Ellis, and S. Sharma, “Tandem connectionist feature extraction for conventional HMM systems,” in *ICASSP*, 2000, pp. 1635–1638.

[8] H. Hermansky and S. Sharma, “TRAPS-Classifiers of temporal patterns,” in *ICSLP*, 1998, pp. 1003–1006.

[9] B.Y. Chen, “Learning discriminant narrow-band temporal patters for automatic recognition of conversational telephone speech,” in *Ph.D. dissertation, Univ. of California, Berkeley*, 2005.

[10] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, “Probabilistic and bottle-neck features,” in *ICASSP*, 2007, pp. 757–760.

[11] D. Yu and Michael L. Seltzer, “Improved Bottleneck Features Using Pretrained Deep Neural Networks,” in *INTERSPEECH*, 2011, pp. 237–240.

[12] F. Grézl, M. Karafiát, and L. Brurget, “Investigation into Bottle-Neck Features for Meeting Speech Recognition,” in *INTERSPEECH*, 2009, pp. 2947–2950.

[13] F. Grézl and M. Karafiát, “Hierarchical Neural Net Architectures for Feature Extraction in ASR,” in *INTERSPEECH*, 2010, pp. 1201–1204.

[14] C. Plahl, R. Schluter, and H. Ney, “Hierarchical Bottle-Neck Features for LVCSR,” in *INTERSPEECH*, 2010.

[15] P. Zhou, L. Dai, Q. Liu, and H. Jiang, “Combining Information from Multi-Stream Features Using Deep Neural Network in Speech Recognition,” in *ICSP*, 2012, pp. 557–561.

[16] C. Plahl, M. Kozielski, R. Schluter, and H. Ney, “Feature Combination and Stacking of Recurrent and Non-Recurrent Neural Network for LVCSR,” in *ICASSP*, 2013.

[17] M. Ravanelli, B. Elizalde, K. Ni, and G. Friedland, “Audio Concept Classification with Hierarchical Deep Neural Networks,” in *EUSIPCO, 2014*.

[18] M. Kleinschmidt, “Methods for capturing spectro-temporal modulations in automatic speech recognition,” *Acta Acustica united with Acustica*, vol. 88, no. 3, pp. 416–422, 2002.

[19] M. Kleinschmidt and D. Gelbart, “Improving word accuracy with Gabor feature extraction,” in *ICSLP*, 2002, pp. 25–28.

[20] M.R. Schadler, B.T. Meyer, and B. Kollmeier, “Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition,” *Journal of the Acoustical Society of America*, vol. 131, no. 5, pp. 4134–4152, 2012.

[21] B.T. Meyer, S. Ravuri, M.R. Schadler, and N. Morgan, “Comparing Different Flavors of Spectro-Temporal Features for ASR,” in *INTERSPEECH*, 2011, pp. 1269–1272.

[22] B. Lee and D. Ellis, “Noise robust pitch tracking using subband autocorrelation classification (SAcC),” in *INTERSPEECH*, 2012.

[23] F. Valente, M.M. Doss, and W. Wang, “Analysis and comparison of recent MLP features for LVCSR systems,” in *INTERSPEECH*, 2011, pp. 1245–1248.

[24] F. Grézl and P. Fousek, “Optimizing bottle-neck features for LVCSR,” in *ICASSP*, 2008, pp. 4729–4732.

[25] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy Layer-Wise Training of Deep Networks,” in *NIPS*, 2006.

[26] F. Seide, G. Li, X. Chen, and D. Yu, “Feature Engineering in Context-Dependent Deep Neural Networks for Conversational Speech Transcription,” in *ASRU*, 2011.

[27] O. Vinyals and S. V. Ravuri, “Comparing Multilayer Perceptron to Deep Belief Network Tandem Features for Robust ASR,” in *ICASSP*, 2011, pp. 4596–4599.

[28] K. Veselý, L. Burget, and F. Grézl, “Parallel Training of Neural Networks for Speech Recognition,” in *INTERSPEECH*, 2010, pp. 439–446.

[29] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, “The HTK Book Version 3.0,” in *Cambridge*, 2000.