# DEEP NEURAL NETWORK TRAINED WITH SPEAKER REPRESENTATION FOR SPEAKER NORMALIZATION

*Yun Tang*[1]*, Aanchan Mohan*[2*]*, Richard C. Rose*[2]*, Chengyuan Ma*[1]

[1]Nuance Communications
[2]McGill University
{yun.tang,chengyuan.ma}@nuance.com, aanchan.mohan@mail.mcgill.ca, rose@ece.mcgill.ca

## ABSTRACT

A method for speaker normalization in deep neural network (DNN) based discriminative feature estimation for automatic speech recognition (ASR) is presented. This method is applied in the context of a DNN configured for auto-encoder based low dimensional bottleneck (AE-BN) feature extraction where the derived features are used as input to a continuous Gaussian density hidden Markov model (HMM/GMM) based ASR decoder. While AE-BN features are known to provide significant reduction in ASR word error rate (WER) with respect to more conventional spectral magnitude based features, there is no general agreement on how these networks can reduce the impact of speaker variability by incorporating prior knowledge of the speaker. An approach is presented in this paper where spectrum based DNN inputs are augmented with speaker inputs that are derived from separate regression based speaker transformations. It is shown the proposed method could reduce the WER by 3% relative to the best speaker adapted AE-BN CDHMM system.

***Index Terms***— Neural networks, speaker adaptation, speaker normalization

## 1. INTRODUCTION

DNNs applied to acoustic modeling have advanced the state of the art in many different ASR task domains[1]. They have been employed for representing local distributions in hybrid hidden Markov model / neural network (HMM/NN) based ASR and for discriminative feature extraction [2, 3]. The issue addressed in this work is how DNN based models can be normalized using limited amounts of adaptation data to minimize the impact of speaker variability.

This paper investigates an approach for generating DNN based speaker adaptive discriminative features. These adapted features are used as input to a continuous Gaussian mixture hidden Markov model (HMM/GMM) based ASR system. One important aspect of the approach is that it provides a mechanism for incorporating well known regression based speaker adaptation techniques, such as maximum likelihood linear regression (MLLR) [4] and constrained MLLR (CMLLR) [5], to provide speaker information for estimating parameters in DNN based feature analysis. It is well known that DNN based features can provide a significant reduction in ASR WER compared to the traditional features such as mel-frequency cepstrum coefficients (MFCCs). However, it is also true that this improvement in WER is less significant if standard adaptation techniques like MLLR or CMLLR are applied in HMM/GMM based ASR. This emphasizes the importance of developing effective adaptation scenarios for DNN based feature analysis.

Given an utterance from a particular speaker, the approach for speaker adaptive DNNs presented in Section 3 relies on two sets of input activations for each analysis frame. The first set of activations, updated at the frame rate, are derived from the MFCC features. The second set of activations, held fixed as a representation of the speaker, are a set of speaker parameters estimated using data from that speaker. In this work, these speaker parameters are derived from regression based transformations, estimated as described in Section 3 using CMLLR, from the available data in utterances taken from that speaker. Hence, prior knowledge of speaker characteristics are provided in the form of the parameters of these transforms as inputs to the DNN both in DNN training and in estimating posterior features during recognition. One weakness of the proposed method is that it assumes that enrollment data is available from each speaker for estimating the speaker parameters, making it difficult to apply in scenarios where this data is not available for some speakers. Mixed mode training is introduced in Section 4 as a partial solution to this problem. Experiments were also conducted using MLLR transforms as speaker representation and a similar gain to CMLLR transforms is observed. Detailed results are not reported in this paper.

It is helpful to consider this approach in the context of two adaptive discriminative feature analysis methods that have recently been proposed in the literature. These two methods are referred to here as the speaker factor [6] and the speaker code [7, 8] methods. The first method was proposed by Ferras and Bourlard as a neural network approach for factorizing speaker and phonetic information [6]. This involves building two bottleneck DNNs that share common input layers. The first is trained as a phone classifier and the other is trained as a speaker classifier. The outputs from the bottleneck layers of the two DNNs are used as input features for a final phone recognition system. It is argued here that parametric speaker representations used in Section 3 have the potential for incorporating more prior speaker information in DNN training than is possible in [6], when the data for estimating these representations is available.

The second method, proposed by Abdel-Hamid and Jiang, allows for the encoding of speaker information at the input of the DNN [7, 8]. This is done by including speaker normalization hidden layers as well as a speaker representation, or speaker code, preceding the input layer of a speaker independent DNN and updating all of these parameters through backpropagation training. The proposed approach differs from the previous work in that the speaker representation is obtained from regression based parametric model parameters that are used as input activations to the DNN for utterances from a given speaker. This provides a means for incorporat-

*The author performed this work during an internship at Nuance Communications, Montreal, Canada. Financial support was provided jointly by Nuance Communications, MITACS-Accelerate,Canada and the FQRNT.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

ing prior knowledge of speaker characteristics as well as a mechanism for reducing the complexity of simultaneous training of the prepended speaker layers and the speaker representations.

The paper is organized as follows. A brief description of DNN based feature analysis is given in Section 2. The approach for speaker adaptive DNNs using parametric speaker representations is presented in Section 3. An experimental study performed to evaluate this approach on a large vocabulary English speech task is described in Section 4. Summary and conclusions are provided in Section 5.

## 2. DNN BASED FEATURE ANALYSIS

The use of generative pre-training and large datasets in neural network training have enabled the use of many hidden layers in deep neural networks (DNNs) for ASR acoustic modelling. In addition, the use of rectified linear units (ReLU), which are activation functions of the form $f(z) = \max(0, z)$, has been shown to decrease training time and improve classification performance in a number of tasks [9, 10, 11]. DNNs with ReLU activation functions are used for all the networks in this work. $Soft-max$ activation functions defined as

$$p_{i,x_t} = \frac{\exp(z^L_{i,x_t})}{\sum_k \exp(z^L_{k,x_t})} \quad (1)$$

are used in the final network layer to model the posterior probability for class $i$ given input vector $x_t$. In Equation 1, $z^L_{i,x_t}$ and $p^L_{i,x_t}$ are the input and output for the $i$th neuron at layer $L$ given input vector $x_t$, respectively. DNN parameters are typically trained by maximizing the cross entropy

$$E = \sum_t \sum_i \hat{p}_{i,x_t} \log p_{i,x_t} \quad (2)$$

where $\hat{p}_{i,x_t}$ is the target probability, which is equal to 1 if $i$ is the target label and equal to 0 otherwise. In this work, the classes are defined as the states of the context clustered HMMs.

DNNs configured with a low dimensional bottleneck middle layer have been shown to provide improved ASR performance when compared to other discriminative feature extraction approaches for ASR [3]. The activations of the bottleneck layer in these networks are used as feature vectors input to HMM/GMM based recognition. Further improvements have been obtained by first performing DNN training without a bottleneck layer and then training a second auto-encoder neural network with a bottleneck middle layer [3] on top of the original DNN. This two step training of auto-encoder bottleneck (AE-BN) features is employed here for extracting discriminative features for HMM/GMM recognition.

The AE-BN training can be summarized as follows. First, a DNN model is trained according to the cross entropy based optimization criterion in Equation 2. Second, a new auto-encoder DNN with a bottleneck layer is trained on the top of the first DNN with the last output layer dropped. An extra soft-max output is calculated from the output of the first DNN before applying the non-linearity at the last hidden layer. The parameters of the second AE-BN network are optimized by the cross-entropy cost between the soft-max output of the network and the extra soft-max output of the first DNN.

It is important to note here that our proposed method for speaker normalization is not limited to the use of the AE-BN configuration of the DNN. While the results of this experimental study mainly utilize the AE-BN configuration of the DNN, in general it is applicable to other DNN configurations as well. For example, we observed similar gains with a more general configuration of the bottle-neck DNN system with the proposed method.

## 3. SPEAKER REPRESENTATION NORMALIZED DNN

This section describes a procedure for training DNNs in a speaker adaptive mode using parametric speaker representations. Speaker representations, derived from an auxiliary GMM model serve as an additional input to the first layer along with frame based speech features. This is described in detail in Section 3.1. A comparison of the proposed method to well known methods of neural network adaptation is described in Section 3.2.

### 3.1. Speaker information as extra DNN input

The block diagram in Figure 1, shows an outline of a framework to train a DNN with a speaker representation. The input at the first layer of the DNN is the concatention of two sets of activations. The first, labelled as speech input, corresponds to frame based spectral features and are updated for each frame. The second, labelled as speaker input in the figure, characterizes an individual speaker and is held fixed for the duration of that speaker's utterances. More formally this can be expressed as,

$$z^1 = (w_1)^T v_d + (w_s)^T v_s + b_1. \quad (3)$$

Here $v_d$ denotes the speech input vector and $w_1$ is the corresponding weight matrix. Similarly, $v_s$ is the speaker input in the first layer and $w_s$ is the speaker weight matrix. The vector $b_1$ is the bias vector for the first layer. The vector $z^1$ is used to denote the collective set of activations which are inputs to the neurons in the first layer.

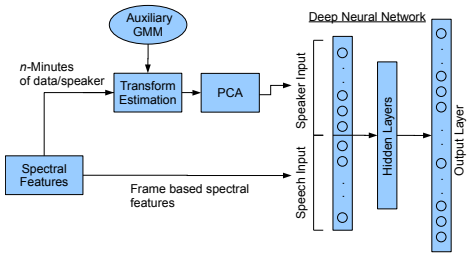


**Fig. 1**. Framework to build DNN with speaker representation.

The choice of speaker representation in this work is a vectorized CMLLR transform. The process of obtaining speaker inputs is shown in the top half of the figure. The regression class based CMLLR transforms are derived from an auxiliary GMM model. This auxiliary GMM model is trained using the maximum-likelihood criterion on MFCC based spectral features. The vectorized transforms are projected to a lower dimension using a principal components analysis (PCA) transformation. The PCA transformation matrix is trained on all of the training speakers' vectorized transforms. The principal components thus obtained were chosen so as to preserve 95% of the variance in the speaker data.

### 3.2. Speaker information as a speaker dependent bias

The DNN trained with a speaker representation that was proposed in the previous section can also be interpreted as adapting the bias term towards target speaker in the input layer. Hence, Equation 3 can be re-written as:

$$z^1 = (w_1)^T v_d + b_s \quad (4)$$

where $b_s = (w_s)^T v_s + b_1$. This perspective of viewing speaker adaptation in the DNN could potentially reduce computational costs

for speaker adaptation during test. This mode of rapid speaker adaptation during ASR is yet to be implemented in our current system.

It is also useful to compare the proposed method in the context of the more well-known adaptation methods which work by adjusting weights toward adaptation data, such as Linear Input Network(LIN) [12] and Linear Hidden Network (LHN) [13]. There are two advantages for the proposed method with respect to these methods. First, an explicit speaker representation is used from the beginning of DNN training, whereas the other methods only do adaptation after the speaker independent DNN model is built. Thus, the proposed method allows the DNN to learn from the input speaker representation in the spirit of speaker adaptive training(SAT) [14]. This approach removes speaker variability coded in speech spectral features, thereby giving an improved speaker independent DNN. Such a DNN can better focus on intra-speaker variability and good performance can be expected. Second, $v_s$ is estimated through traditional speaker adaptation algorithms in HMM/GMM systems. There are many adaptation methods to choose from, with adaptation data varying from one utterance to several hours. So DNN adaptation based on the proposed method could be used with different adaptation scenarios.

## 4. EXPERIMENTAL STUDY

An experimental study is performed to evaluate the impact of the speaker normalized discriminative feature extraction approach presented in Section 3 on ASR performance. In addition to speaker normalized feature extraction, MLLR/CMLLR based speaker adaptation is performed in the HMM/GMM ASR system. As a point of reference, this performance is compared to the WER obtained using MLLR/CMLLR based speaker adaptation applied in HMM/GMM ASR without speaker normalized discriminative feature extraction.

### 4.1. Task Domain and Feature Extraction

The experimental study is conducted on a proprietary data set consisting of English language speech. The speech corpus consists of thousands of hours of speech collected under relatively clean acoustic conditions. All systems are trained from data consisting of approximately 10,000 speakers with an average of approximately 10 minutes of speech data per speaker. The evaluation corpus consists of hundreds of speakers with approximately 1 hour of test utterances per speaker and approximately 4 minutes of enrollment data per speaker. The enrollment data in the evaluation set is used for training the CMLLR based speaker vectors as described in Section 3.

The generation of speaker normalized discriminative features for the AE-BN system in Figure 1, in both training and evaluation, consists of the following components. First, spectral features are extracted for input to both the AE-BN network and the CMLLR transform estimation from the auxiliary GMM shown in Figure 1. The spectral features consist of 12 dimension MFCCs with appended first, second, and third difference coefficients. Vocal tract length normalization (VTLN) is also performend. These 48 dimensional feature vectors are used as input to CMLLR transformation estimation from the auxiliary GMM. The spectral feature vectors provided to the DNN consist of concatenated vectors from five surrounding frames resulting in a dimensionality of 528 (48*11).

The second component of feature generation is estimation of the speaker vector input to the DNN shown in Figure 1. Mutliple regression class CMLLR transform matrices have been investigated as possible speaker representations. Transform estimation in Figure 1 involves estimating CMLLR transformations for two regression classes. The actual speaker vectors are obtained from these matrices using PCA to obtain a low dimensional speaker activation vector by transforming from (48 X 48 X 2) dimensions to a 1416 dimensional speaker vector.

The third component to feature generation is the AE-BN network shown in Figure 1. The input activations consist of the concatenated spectral and speaker vectors described above. A DNN, consisting of 7 layers, is trained in the first step of the procedure described in Section 2. The first 5 hidden layers of this network contain 1,000 nodes and the softmax output layer contains 3500 output targets where the target classes correspond to the context dependent states in the HMM/GMM acoustic model. The AE-BN, trained in the 2nd step, has 5-layers where the layers contain 1000, 500, 40, 500 and 1000 nodes respectively. Weights and biases in both steps are optimized using the standard backpropagation algorithm using Gnumpy [15] and Cudamat [16] packages on a GPU server. The 40 dimensional output of the AE-BN network is taken from the 40 node bottleneck layer.

The final component of the approach shown in Figure 1 is the use of the speaker normalized discriminative features for ASR decoding and training. The dimensionality of the 40 dimensional bottleneck features is reduced to 32 using a heteroscedastic linear discriminate analysis (HLDA) transformation. These transformed features are input to the maximum likelihood trained HMM/GMM based ASR decoder.

### 4.2. Evaluation of Speaker Normalized Discriminative Features

In this section, the performance for several different definitions of AE-BN based discriminitive features are evaluated including the speaker normalized configuration described in Section 4.1. These configurations will be evaluated on the test set described in Section 4.1 in terms of the WERs obtained for the HMM/GMM ASR system using the discriminative features as input. MLLR and CMLLR based speaker adaptation is also performed separately for the HMM/GMM ASR system using the same enrollment data that is used for estimating the speaker vectors in the speaker normalized AE-BN. Therefore, the results reported in this section for AE-BN features represent improvements made to the best performing speaker adaptive HMM/GMM ASR systems which employ speaker adaptation during recognition. CMLLR adaptation for the HMM/GMM recognizer is performed using a single regression class and MLLR adaptation is performed using 156 regression classes.

Table 1 shows the WER obtained from the tandem configuration of the AE-BN discriminative feature analysis and the HMM/GMM ASR decoder. In this table The first row of the table, labeled "MFCC", refers to the baseline condition where HMM/GMM acoustic models are trained using MFCC features. "DNN$_s$", "DNN" and "DNN$_n$" represent three different definitions of AE-BN discriminative features. "DNN$_s$" corresponds to the speaker normalized AE-BN configuration described in Section 4.1. "DNN" and "DNN$_n$" share a configuration similar to the AE-BN network of "DNN$_s$", except for the fact that they do not have a speaker input like the network "DNN$_s$" does. The networks "DNN" and "DNN$_n$" differ in the kind of features presented at the input layer. The features input to the "DNN$_n$" consist of MFCC features that are pre-transformed with speaker-specific regression-class based CMLLR transforms. The comparison of "DNN$_s$" and "DNN$_n$" illustrates the advantage of using the vectorized CMLLR speaker information as a specific input to the DNN ("DNN$_s$") rather than just performing CMLLR transformation of the features ("DNN$_n$"). The column labeled "None" in Table 1 displays the WERs obtained when no adaptation is ap-

**Table 1**. Comparison of HMM/GMMs with different DNN features.

| Model | AE-BN config. | | HMM/GMM Adaptation | | |
|---|---|---|---|---|---|
| | feature norm. | speaker input | None | CMLLR | MLLR |
| MFCC | ✗ | ✗ | 10.76 | 9.99 | 9.44 |
| DNN | ✗ | ✗ | 9.23 | 8.71 | 8.32 |
| $\text{DNN}_n$ | ✓ | ✗ | *9.02* | 8.75 | 8.34 |
| **$\text{DNN}_s$** | ✗ | ✓ | *8.57* | 8.39 | 8.08 |

plied to the HMM/GMM ASR system. The results in this column demonstrate the impact on WER of performing speaker normalized discriminative feature extraction while presenting the features to the unadapted ASR system.

The following discussion compares the different feaure extraction and speaker adaptation scenarios in terms of their relative impact on WER. It is clear from Table 1 that in general, the HMM/GMM with DNN features are significantly better than the MFCC baseline. Comparing the $1^{st}$ and $2^{nd}$ rows of Table 1, a 14.22% WER reduction (WERR) is observed when DNN AE-BN features are used instead of MFCC features when no adaptation is applied. The gain becomes smaller if speaker adaptation is applied during ASR. The WERR is 12.81% if CMLLR adaptation is performed ($5^{th}$ col. of Table 1) and 11.86% if MLLR adaptation is performed ($6^{th}$ col. of Table 1) in HMM/GMM ASR.

Comparing the $2^{nd}$ and $3^{rd}$ rows of Table 1, a second observation that can be drawn is that the DNN trained with features pre-transformed with the CMLLR ("$\text{DNN}_n$") has a 2.28% WERR compared to the "DNN" baseline if no adaptation applied on the HMM/GMM. This is smaller than the WERR achieved by "$\text{DNN}_s$"(7.16% compared to the "DNN" baseline), in which vectorized CMLLR transforms are used as speaker representation. Furthermore, if speaker adaptation is applied to the HMM/GMM ASR system, there is only a very small reduction in WER for the "$\text{DNN}_n$" features with respect to the "DNN" baseline. This observation agrees with results previously reported in [17, 18].

Finally, the results in Table 1 demonstrate that speaker normalized DNNs result in an overall reduced WER when speaker adaptation is performed for HMM/GMM ASR. The WER for the "$\text{DNN}_s$" is 3.67% smaller than the corresponding "DNN" baseline for the CMLLR adapted ASR system. Furthermore, a 2.87% WERR is achieved when the HMM/GMM is adapted using multiple MLLR transforms.

### 4.3. HMM/GMM trained with mixed mode

A potential drawback of the proposed method is that a faithful speaker representation is required at the DNN input during test. Adequate amounts of adaptation data might be required to estimate the speaker information(CMLLR transformations via the auxilary model in this work, for example). If no adaptation/enrollment data is available, identity transforms could be used in the place of speaker representation. However, this introduces a train-test mismatch. The mismatch can be alleviated by building a HMM/GMM using the so-called "mixed-mode" training scenario.

In this training scenario, the training data set is divided into two parts. The first part of the training set consists of the AE-BN features extracted using the real speaker representation (vectorized CMLLR transform in this work). The second part of the training data consists of features extracted with the AE-BN DNN but with the corresponding speaker information replaced with "'faked' identity transforms. The DNN used for this "mix-mode" feature extraction is the same as the DNN used for the "$\text{DNN}_s$" case in the previous section. In this study, a new HMM/GMM is trained in mixed mode with 50% of the speakers using their true speaker representation and the other 50% of the speakers in the training set using identity transforms.

**Table 2**. HMM/GMM trained with mixed mode

| Model | no enrollment | 4min enrollment |
|---|---|---|
| DNN | 9.23 | 8.32 |
| $\text{DNN}_s$ | 9.80 | 8.08 |
| $\text{DNN}_s$(mix) | 9.54 | 8.01 |

The results for this experimental study are shown in Table 2. The table displays results for two enrollment scenarios - "No enrollment" and "4 min" depending on the amount of adaptation/enrollment data available per test speaker. The "no enrollment" case denotes tests without enrollment data where the test features were extracted so that the speaker input was set to a "faked" identity transform. "$\text{DNN}_s$(mix)" in this table denotes the use of the "mix-mode" feature set for training. Results for this training scenario are compared to the "DNN" and "$\text{DNN}_s$" results that were reported in Table 1.

Looking at the 1st column of Table 2, "$\text{DNN}_s$" is 6.13% worse compared to the corresponding DNN results due to mismatch of training and test scenarios when no enrollment data is available. The gap is reduced to 3.41% when HMM/GMM is trained with mixed mode. On the other hand, when adaptation data is available ("4 min. enrollment"), mixed mode trained model ("$\text{DNN}_s$(mix)") achieves similar results if not better as the one with true speaker representation only model ("$\text{DNN}_s$"). In this test scenario, MLLR is employed on the HMM/GMMs during ASR with 4 min. enrollment data. The same enrollment data is used for DNN feature normalization when test features were generated for the "$\text{DNN}_s$" and "$\text{DNN}_s$(mix)" cases.

## 5. CONCLUSIONS

A speaker normalization procedure for discriminative feature estimation has been presented in the context of an auto-encoder bottleneck (AE-BN) DNN front-end for ASR. Speaker normalization is performed by augmenting spectrum based DNN inputs with speaker inputs that are derived from separate regression based speaker transformations. It was argued that, in theory, these speaker normalized AE-BN DNNs should provide an efficient characterization of both intra-speaker and inter-speaker variability. The proposed method is evaluated using a tandem configuration where speaker normalized AE-BN features are input to an HMM/GMM ASR system. The speaker normalized AE-BN features were shown to reduce WER by approximately 3% relative to AE-BN features using no speaker normalization.

Mixed-mode training of the HMM/GMM ASR system is investigated to deal with scenarios where enrollment data may not be available for some speakers during recognition. This HMM/GMM training mode for ASR involves using features in training that are obtained from a mixture of both speaker normalized DNNs and from DNNs trained using "degenerate" speaker representations. The mixed-mode trained HMM/GMM recognizer was shown to be more robust to lack of a speaker dependent enrollment data in recognition. WER was reduced by 2% with respect to normal training when no enrollment data was available during testing. Similar WER reduction was was achieved using mixed-mode training even when speaker dependent enrollment data was available.

## 6. ACKNOWLEDGEMENTS

The authors would like to thank colleagues from Nuance for productive discussions and help with building DNN baseline.

## 7. REFERENCES


[1] G. Hinton, L. Deng, D. Yu, et al., "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, November 2012.

[2] Daniel P. W. Ellis Hynek Hermansky and Sangita Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ICASSP*, 2000, pp. 1635–1638.

[3] T. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *ICASSP*, 2012, pp. 4153–4156.

[4] C.Leggetter and P.Woodland, "Flexible speaker adaptation using maximum likelihood linear regression," in *the ARPA Spoken Language Technology Workshop*, 1995, pp. 104–109.

[5] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[6] M. Ferras and H. Bourlard, "MLP-based factor analysis for tandem speech recognition," in *ICASSP*, 2013.

[7] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *ICASSP*, 2013.

[8] O. Abdel-Hamid and H. Jiang, "Rapid and effective speaker adaptation of convolutional neural network based models for speech recognition," in *INTERSPEECH*, 2013.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks,," in *NIPS*, 2012.

[10] G. Dahl, T. Sainath, and G. Hinton, "Improving deep neural network for LVCSR using recified linear units and dropout," in *ICASSP*, 2013.

[11] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," in *ICASSP*, 2013.

[12] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker adaptation for hybrid HMM-ANN continuous speech recognition system," in *Eurospeech*, 1995, pp. 2171–2174.

[13] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. de Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

[14] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *ICSLP*, 1996, vol. 2, pp. 1137–1140 vol.2.

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

[16] Volodymyr Mnih, "Cudamat: a CUDA-based matrix class for python," Tech. Rep., University of Toronto, Department of Computer Science, 2009.

[17] Z. Tuske, C. Plahl, and R. Schluter, "A study on speaker normalized MLP features in LVCSR," in *Interspeech*, Auguest 2011, pp. 1089–1092.

[18] Y. Wang and M. J.F. Gales, "Tandem system adaptation using multiple linear feature transforms," in *ICASSP*, 2013.