Speaker Dependent Bottleneck Layer Training for Speaker Adaptation in Automatic Speech Recognition

Rama Doddipatla, Madina Hasan and Thomas Hain

The University of Sheffield, Sheffield, United Kingdom.

[r.doddipatla,m.hasan,t.hain]@sheffield.ac.uk

Abstract

Speaker adaptation of deep neural networks (DNN) is difficult, and most commonly performed by changes to the input of the DNNs. Here we propose to learn discriminative feature transformations to obtain speaker normalised bottleneck (BN) features. This is achieved by interpreting the final two hidden layers as speaker specific matrix transformations. The hidden layer weights are updated with data from a specific speaker to learn speaker-dependent discriminative feature transformations. Such simple implementation lends itself to rapid adaptation and flexibility to be used in Speaker Adaptive Training (SAT) frameworks. The performance of this approach is evaluated on a meeting recognition task, using the official NIST RT'07 and RT'09 evaluation test sets. Supervised adaptation of the BN layer shows similar performance to the application of supervised CMLLR as a global transformation, and the combination of these appears to be additive. In unsupervised mode, CMLLR adaptation only yields 3.4% and 2.5% relative word error rate (WER) improvement, on the RT'07 and RT'09 respectively, where the baselines include speaker based cepstral mean and variance normalisation. The combined CMLLR and BN layer speaker adaptation yields a relative WER gain of 4.5% and 4.2% respectively. SAT style BN layer adaptation is attempted and combined with conventional CMLLR SAT, to show that it provides a relative gain of 1.43% and 2.02% on the RT'07 and RT'09 data sets respectively when compared with CMLLR SAT. While the overall gain from BN layer adaptation is small, the results are found to be statistically significant on both the test sets.

Index Terms: Deep neural networks, bottleneck features, speaker adaptation, automatic speech recognition.

1. Introduction

Deep neural networks have advanced the current state-of-the-art in automatic speech recognition (ASR). They have been employed to model the distributions in the hybrid DNN-HMM systems as well as for extracting discriminative features for training GMM-HMM systems. A number of studies have successfully shown that DNN's can provide considerable gains in system performance when compared with the current state-of-the-art MFCC/PLP GMM-HMM systems [1, 2, 3, 4, 5, 6]. Though DNN's improve the acoustic modelling capabilities, they do not have an inherent mechanism to normalise speaker variability. Conventional approaches to speaker adaptation like MLLR [7], CMLLR [8, 9] and SAT [10], that have been widely proven to normalise speaker variability in the GMM-HMM frame work, can not be directly applied in the DNN frame work. This generated a wide interest in the community to investigate approaches to make the DNN's robust to speaker variabilities.

Speaker adaptation of DNN's is difficult and is usually performed by changes to the input of DNN's. In [11], a network configuration was proposed to tune the speaker parameters to a particular speaker by converting a supervised network to an unsupervised mode. In [12], a linear input network (LIN) is trained to map speaker dependent input vectors to become speaker independent. The network is trained by back propagating the error from the output layer. It was also proposed to use a parallel hidden network (PHN), where the weights of the hidden units were updated, keeping the rest of the parameters fixed during speaker adaptation. In [13], linear input network (LIN) and linear output network (LON) were studied in the hybrid NN/HMM systems, using them as discriminative feature transformations. In [14], linear hidden network (LHN) is proposed, with the assumption that outputs of an internal layer represent a projection of the input layer into a space where it should be easier to learn the classification. In [15], feature space discriminative linear regression (fDLR) was introduced for performing MLLR style speaker adaptation in DNN's. In [16], speaker adaptation is performed by considering the outputs from the top hidden layer as the observation vectors to the top level log-linear model and updating only the bias for each speaker. In [17], a separate phone and speaker networks are trained, whose outputs are fed to a third network for performing phone recognition. In [18, 19], a speaker code is learnt for each speaker and is updated while performing speaker adaptation. In [20], speaker identity vectors (i-vectors) are used in parallel with regular acoustic features for ASR and allow the network to learn the speaker characteristics. In [21], a speaker separation DNN is trained and the bottle neck features are concatenated with filter-bank features for training the final DNN.

This paper proposes to train speaker dependent discriminative feature transformations to derive speaker normalised bottleneck (BN) features, which are further used for training GMM-HMM systems. Instead of trying to modify the input to the DNN's, the final two hidden layers are interpreted as a matrix transformation. This study investigates whether updating the weights of this matrix with data from a specific speaker can facilitate us to learn discriminative speaker dependent feature transformations to normalise speaker variability. Such an implementation allows for rapid adaptation and flexibility to be used in speaker adaptive training (SAT) frameworks. The rest of the paper is organised as follows: First, we describe our experimental setup and the corpus used in our experiments. Then introduce the proposed approach to learning a speaker dependent bottleneck layers in DNN to derive speaker normalised features and present our results, observations and conclusions on rapid adaptation and SAT.

Table 1: *Corpus statistics used in our experiments.*

Corpora	Words	Hours	Speakers
AMI+AMIDA+ICSI	228689	165.81	1129
RT'07	37323	3.27	35
RT'09	41805	3.50	38

Table 2: *Baseline System Performance (%WER) on RT'07 and RT'09 data sets.*

%WER	RT'07	RT'09
Triphone - XWRD	38.7	42.4
+ HLDA	37.2	40.6
+ CMLLR	35.5	38.5

2. Experimental Setup

The acoustic models are trained using AMI [22], AMIDA [23] and ICSI [24] meeting corpora, which in total is about 165 hours of speech data. The official NIST RT'07 and RT'09 individual head microphone (IHM) data [25] are used for evaluating the ASR system performance. The corpus statistics are presented in Table. 1. The baseline experiments use 39 dimensional PLP features with cepstral mean and variance normalisation (CMVN) applied at speaker level. The baseline system is a cross-word triphone based system with 5686 tied states. HLDA [26] models are built by appending third order derivatives and projecting the PLP features from 52 to 39 dimensions. The system performance on the RT'07 and RT'09 datasets are presented in Table. 2.

2.1. Bottleneck Feature Extraction

DNN's are trained using the TNET toolkit [27]. The input to the DNN uses 31 adjacent frames of the log filter-bank outputs, which are concatenated and decorrelated with DCT to form a 368 dimensional feature vector. The filter-bank inputs are mean and variance normalised at the speaker level. Global mean and variance normalisation is performed on each dimension before feeding the input for training the DNN. For our experiments, we chose to use 4 hidden layers with each hidden layer having 1745 units. The bottleneck (BN) layer is placed just before the output layer and has 26 units. The configuration of the network is illustrated in Figure 1. We set aside 15% of the training data for cross validation and use the rest to train the DNN. The training automatically stops once the frame accuracy of the cross validation set falls below 0.1%.

The DNN's are trained on the triphone targets by performing forced alignment on the training data. The network is trained layer by layer. Once the training is done, BN features are extracted to train a conventional GMM-HMM system. Initial experiments were conducted to find the difference in ASR performance when used in combination with PLP features forming a 65 dimensional feature vector, or as stand alone features. Experiments were also conducted by increasing the size of the BN layer from 26 to 39 dimensions. These investigations are only performed on the RT'07 task to decide on the network configuration used for further experiments in the paper. The results are presented in Table. 3. It is observed that increasing the number of hidden layers improves the performance in all cases. The

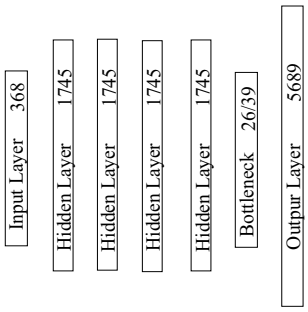

Figure 1: *DNN layer configuration used in our experiments*

Table 3: *System Performance (%WER) on RT'07 by changing the bottleneck layer configuration.*

%WER	PLP+BN-26D	BN-26D	BN-39D
1layer	34.2	36.6	35.0
2layer	30.8	31.2	30.5
3layer	30.1	29.5	29.9
4layer	30.3	29.7	29.7

improvements are noticeable with the first and second hidden layers and the performance saturates after the third hidden layer.

Comparing the performance of PLP+BN with BN-26D, it can be observed that, with increase in the number of hidden layers the difference in performance is very small and in fact after three hidden layers, BN-26D seem to perform better than PLP+BN. The performance is also compared with increasing the dimension of the BN layer to 39 dimensions and one can observe that after 4 four hidden layers, the performance is similar to using a BN of 26 dimensions. Based on these observations, all further experiments will be reported using 39 dimension BN features in this paper, as stand alone features for both RT'07 and RT'09 tasks.

Table 4, presents the results on both RT'07 and RT'09 data sets using the 39 dimensional BN features using a DNN trained with four hidden layers. The table also includes results using CMLLR in recognition using a single global transformation (1 FB) and four full block (4 FB) transformations. The transformations are estimated both in supervised (`sup`) and unsupervised (`unsup`) mode of adaptation for comparing with the results presented in later sections. In the following section, the proposed approach to train a speaker dependent BN layer is described for generating speaker dependent BN features.

3. Speaker Dependent Bottleneck Training

This section presents the proposed approach to perform speaker adaptation in DNN by training a speaker dependent BN layer for each speaker. Interpreting the final two hidden layers as a matrix, the paper proposes to update the weights between these layers using speaker specific data. Such a transformation enables us to derive speaker dependent BN features and allows to perform rapid speaker adaptation and provides flexibility to apply speaker adaptive training (SAT).

The idea to train a speaker specific BN is similar to ideas proposed in [14] to train a linear hidden network (LHN) or training a feature space discriminative transformation by considering the the outputs from the final hidden layer as the ob-

Table 4: *System Performance (%WER) on RT'07 and RT'09 data sets using the 39D bottleneck features using CMLLR and BN layer based speaker adaptation.*

%WER	RT'07	RT'09
BN-39D	29.7	31.3
+ CMLLR (1 FB) sup	28.1	29.4
+ CMLLR (4 FB)	26.8	27.9
+ BN Layer	28.7	30.0
+ CMLLR (1 FB) unsup	28.9	30.4
+ CMLLR (4 FB)	28.7	30.3
+ BN Layer	29.7	31.0

servation vectors to the top level log-linear model [16], both of which were proposed in the context of hybrid DNN-HMM frame work. In both these studies, it has been pointed out that special care should be taken while updating the hidden layer with larger number of units using limited amount of speaker data available for adaptation. In [14], a conservative training approach is proposed to compensate for lack of adaptation data for certain classes. In [16] only the bias component is updated rather than updating the entire weight matrix. In this paper, the size of the hidden layer prior to the BN layer is reduced, thereby reducing the number of parameters to estimated. Initial experiments were conducted without any changes to the size of the hidden layer prior to the bottleneck layer and use the configuration illustrated in Figure 1 for training speaker dependent BN layers. We call this "original network configuration".

3.1. Original Network Configuration

Speaker dependent BN layer training is performed by updating the weights between the final two hidden layers using data from a specific speaker and keeping the rest of the weights in the network unchanged. This means a separate BN layer is trained for each of test speakers to perform rapid adaptation. The results are presented in Table 4. Both supervised (sup) and unsupervised (unsup) update of the weights is performed to understand the proposed approach to speaker adaptation.

It can be observed that, the supervised update of the weights to train the speaker specific BN layer performs similar to the unsupervised CMLLR (4 FB). This means that, updating the BN layer on a speaker basis can be used for deriving speaker dependent BN features. For fair comparison, this performance should be compared with a single global transformation of CMLLR (1 FB), in which case there is a small improvement in performance. This gain can be attributed to the discriminative nature of updating the weights in DNN. But, the performance could not reach the supervised CMLLR adaptation using a global transform (1 FB). This indicates that the size of the hidden layer prior to BN layer might be too large to train a speaker dependent discriminative feature transformation, which requires the estimation of a matrix of size 1745x39.

On the other hand, performing an unsupervised update of the weights for training a speaker dependent BN layer did not provide any gain in performance. This behaviour can be attributed either to the huge size of the hidden layer or to the discriminative nature of training the speaker dependent BN layer. Discriminative approaches are known to be sensitive to recognition errors and might influence the training of the speaker dependent BN layers. In the next section, the size of the hidden

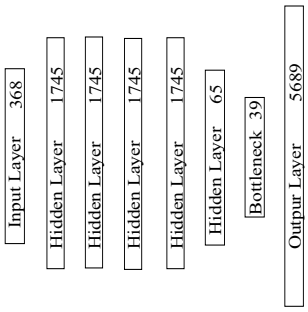

Figure 2: *DNN modified configuration to perform speaker dependent bottleneck layer.*

Table 5: *Baseline System Performance (%WER) on RT'07 and RT'09 using the modified DNN configuration using CMLLR and SAT based speaker adaptation.*

	RT'07	RT'09
BN-39D	29.1	31.3
+ CMLLR (1 FB) sup	27.6	29.5
+ CMLLR (4 FB)	26.6	28.2
+ CMLLR SAT (4 FB)	26.0	27.2
+ CMLLR (1 FB) unsup	28.2	30.5
+ CMLLR (4 FB)	28.1	30.5
+ CMLLR SAT (4 FB)	27.8	29.6

layer prior to the BN layer is reduced such that the number of parameters required to update are less. We call this "modified network configuration".

3.2. Modified Network Configuration

Based on the above observations, the network configuration was modified such that there are fewer parameters to be estimated while training speaker specific BN layers. The modified network configuration is illustrated in Figure 2. A new hidden layer is introduced after the fourth layer prior to the bottleneck layer, having a dimension of 65. This reduces the number of parameters to 65x39. This change in DNN configuration also changes our baseline results, which are presented in Table 5. It is observed that the baseline performance improves on the RT'07 test set, while there is no change on the RT'09 test set (compare with results in Table 4). The table also includes results using speaker adaptive training (SAT) and CMLLR adaptation only in test. The results include both supervised (sup) and unsupervised (unsup) modes of adaptation used for estimating the CMLLR transforms.

Table 6 presents the results using the proposed approach to train speaker dependent BN layers for deriving speaker normalised BN features on both RT'07 and RT'09 data sets. The analysis of results is presented based on the mode of adaptation used to update the network weights using speaker specific data. In all cases the results in Table 5 and Table 6 are compared.

3.2.1. Supervised Adaptation

One can observe that rapid adaptation of the test speaker using BN layer adaptation has a performance very close to the CMLLR estimated as global transformation (1 FB) in the su-

Table 6: *System performance (%WER) using the proposed speaker dependent BN layer adaptation for both RT'07 and RT'09 data sets.*

	RT'07	RT'09
BN-39D	29.1	31.3
+ BN Layer sup	27.5	29.7
+ BN Layer + CMLLR (4 FB)	26.3	27.8
+ BN Layer SAT + CMLLR (4 FB) SAT	25.7	26.9
+ BN Layer unsup	28.6	30.4
+ BN Layer + CMLLR (4 FB)	27.8	30.0
+ BN Layer SAT + CMLLR (4 FB) SAT	27.4	29.0

pervised mode on both the test sets. Updating the weights between the hidden layer and BN layer is equivalent to estimating a single matrix transformation. Therefore, provided the correct transcriptions, BN layer adaptation can perform similar to CMLLR estimated as a global transformation. Since, the derived speaker dependent BN features are further used in training a GMM-HMM system, CMLLR adaptation can be applied on top of the BN features. One can observe that combining these different approaches to speaker adaptation provides additive gains on both the test sets, indicating that they learn complementary speaker characteristics.

BN layer adaptation can be applied in training, similar to using CMLLR in speaker adaptive training (SAT). We perform adaptation on the training data using the proposed approach and call it BN Layer SAT. Conventional CMLLR SAT can still be performed on top of the proposed BN Layer SAT. One can observe that BN Layer SAT, when combined with CMLLR SAT improves the performance when compared with using only CMLLR SAT in Table 5 on both the test sets. Supervised adaptation assumes that the transcription of the test speech is known. This is not usually the case, so the performance of the proposed approach in unsupervised mode of adaptation are reported in the next section.

3.2.2. Unsupervised Adaptation

Unsupervised speaker dependent BN layer training uses the previous recognition outputs as the true transcripts for updating the network weights. One can observe that rapid adaptation of the network weights improves the performance over the baseline, a behaviour not visible in Table 4. This indicates that reducing the dimension of the hidden layer prior to the BN layer indeed helped in learning the speaker characteristics using limited adaptation data.

Comparing the results of BN layer adaptation with unsupervised CMLLR (1 FB), one can notice that RT'07 has slightly inferior performance while RT'09 has similar performance. These results seem to indicate that given partially correct transcriptions, BN layer adaptation can still try to perform similar to unsupervised CMLLR estimated as a global transformation. Further, BN layer adaptation when combined with CMLLR provides additive gains on both the test sets. This behaviour seems to be consistent both in supervised and unsupervised modes of adaptation. The combined result is also the we could achieve on both the test sets in unsupervised mode.

It is found that a relative gain of 3.4% on the RT'07 and 2.5% on the RT'09 data sets is obtained respectively, when only CMLLR adaptation is performed on the baseline system (from Table 5). Similarly, one can find that a relative gain of 4.5% on the RT'07 and 4.2% on the RT'09 data sets is obtained respectively using the proposed speaker dependent BN layer adaptation in combination with CMLLR over the baseline (from Table 6). While the overall gain using BN layer adaptation in unsupervised mode is small, the relative gains were found to be statistically significant on both the test sets. Applying BN Layer SAT and combining with CMLLR SAT further improved the performance, providing a relative gain of 1.43% on the RT'07 and 2.02% on the RT'09 data sets over the CMLLR SAT in Table 5. These are the best results that could be achieved in all the combinations on both the test sets. The relative gains were also found to be statistically significant. The statistical tests have been done using the NIST scoring toolkit [28] using the `sc_stats` tool for performing paired-comparison statistical significance tests.

4. Conclusion

Speaker adaptation in DNN is difficult and is often performed by applying changes to the input. In this paper, interpreting the weights between the two final hidden layers as a matrix, discriminative feature transformations were estimated by updating the weights with data from a specific speaker. This facilitates to train speaker dependent BN layers and in turn derive speaker normalised BN features. Following such an approach not only eliminates the need to apply changes to the input for DNN training, but also enables to perform rapid speaker adaptation and apply speaker adaptive training. The performance of the proposed approach is evaluated on a meeting recognition task, using the official NIST RT'07 and RT'09 data sets.

Using the original network configuration with a large hidden layer prior to the BN layer, supervised update of the weights showed promise, but unsupervised mode of adaptation did not provide any gains. Attributing this behaviour to the large size of the hidden layer prior to BN layer, further investigations were performed by reducing the size of the hidden layer prior to the BN layer from 1745 to 65.

It was found that supervised update of BN layer has similar performance to estimating CMLLR as a global transformation in supervised mode, indicating that BN layer adaptation is able to learn the speaker characteristics provided the correct transcriptions. Similar behaviour is also visible in unsupervised mode of adaptation on the RT'09 data set, indicating that even with errors in transcription, the proposed approach can still perform similar to CMLLR as a global transformation. More interestingly, BN layer adaptation when combined with CMLLR is shown to provide additive gains. This indicates that, they learn complementary characteristics of the speaker. Finally, it was also shown that BN layer adaptation can be applied similar to CMLLR in SAT on the training data, which we call as BN Layer SAT. This approach when combined with CMLLR SAT provided the best performance on both the test sets. Though the relative improvements in performance with BN layer adaptation seem to be small, they have be found to be statistically significant on both the test sets.

5. Acknowledgement

This work is in part supported by the EU FP7 DocuMeet Project `http://www.documeet.eu/the-project`.

6. References

[1] A. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in Proc. NIPS Workshop Deep Learning for Speech Recognition and Related Applications, 2009

[2] A. Mohamed, T. N. Sainath, G. E. Dahl, B. Ramabhadran, G. E. Hinton, and M. Picheny, "Deep belief networks using discriminative features for phone recognition," in Proc. ICASSP, 2011.

[3] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," IEEE Trans. Audio Speech Lang. Processing , vol. 20, no. 1, pp. 1422, Jan. 2012.

[4] F. Seide, G. Li, and D. Yu, "Conversational Speech Transcription Using Context-Dependent Deep Neural Networks," In Interspeech 2011.

[5] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," IEEE Trans. on Audio, Speech, and Language Processing, vol. 20, no. 1, pp. 3042, 2012.

[6] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition," IEEE Signal Processing Magazine, 2012.

[7] C. J. Leggetter and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171 – 185, 1995.

[8] V. V. Digalakis, D. Rtischev, and L. G. Neumeyer, "Speaker Adaptation using Constrained Estimation of Gaussian Mixtures," *IEEE Trans. on Speech and Audio Processing*, vol. 3, no. 5, pp. 357 –366, Sep. 1995.

[9] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[10] T. Anastasakos, J. Mcdonough, R. Schwartz, and J. Makhoul, "A Compact Model for Speaker-Adaptive Training," in *Proc. of ICSLP*, pp. 1137–1140, Oct. 1996.

[11] J. S. Bridle, and S. Cox, "RecNorm: Simultaneous Normalisation and Classification Applied to Speech Recognition," in NIPS, page 234-240, 1990.

[12] J. P. Neto, L. B. 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.

[13] B. Li, and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in INTERSPEECH, 2010.

[14] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," in ICASSP, 2006.

[15] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in ASRU, 2011.

[16] K. Yao, D. Yu F. Seide, H. Su, L. Deng and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in IEEE SLT, 2012.

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

[18] 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.

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

[20] G. Soan, H. Soltau, D. Nahamoo and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in ASRU 2013.

[21] Y. Liu, P. Zhang and T. Hain, "Using neural network front-ends on far field multiple microphones based speech recognition," to apper ICASSP 2014.

[22] J. Carletta, S. Ashby, S. Bourban, M. Guillemot, M. Kronenthal, G. Lathoud, M. Lincoln, I. McCowan, T. Hain, W. Kraaij, W. Post, J. Kadlec, P. Wellner, M. Flynn, and D. Reidsma, "The AMI meeting corpus," In Proc. MLMI05 , Edinburgh, 2005.

[23] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiát, D. A. van Leeuwen, M. Lincoln, V. Wan: The 2007 AMI(DA) System for Meeting Transcription. CLEAR 2007: 414-428

[24] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. "The ICSI meeting corpus," In ICASSP 2003.

[25] NIST: Rich transcription evaluations (2007 and 2009), `http://www.itl.nist.gov/iad/mig//tests/rt/`.

[26] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," Speech Communication, vol. 26, no. 4, pp. 283 297, 1998.

[27] TNet: Neural Network Trainer. http://speech.fit.vutbr.cz/software/neural-network-trainer-tnet

[28] NIST, "Speech recognition scoring toolkit (SCTK)," `http://www.nist.gov/speech/tools/`.