Improved feature processing for Deep Neural Networks

Shakti P. Rath [1,2], *Daniel Povey* [3], *Karel Veselý* [1] *and Jan "Honza" Černocký* [1]

[1]Brno University of Technology, Speech@FIT, Božetěchova 2, Brno, Czech Republic.
[2]Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, UK.
[3]Center for Language and Speech Processing, Johns Hopkins University, USA.
rath@fit.vutbr.cz, dpovey@gmail.com, iveselyk@fit.vutbr.cz, cernocky@fit.vutbr.cz

Abstract

In this paper, we investigate alternative ways of processing MFCC-based features to use as the input to Deep Neural Networks (DNNs). Our baseline is a conventional feature pipeline that involves splicing the 13-dimensional front-end MFCCs across 9 frames, followed by applying LDA to reduce the dimension to 40 and then further decorrelation using MLLT. Confirming the results of other groups, we show that speaker adaptation applied on the top of these features using feature-space MLLR is helpful. The fact that the number of parameters of a DNN is not strongly sensitive to the input feature dimension (unlike GMM-based systems) motivated us to investigate ways to increase the dimension of the features. In this paper, we investigate several approaches to derive higher-dimensional features and verify their performance with DNN. Our best result is obtained from splicing our baseline 40-dimensional speaker adapted features again across 9 frames, followed by reducing the dimension to 200 or 300 using another LDA. Our final result is about 3% absolute better than our best GMM system, which is a discriminatively trained model.

1. Introduction

The recent success of Deep Neural Network (DNN) has revolutionized automatic speech recognition systems. In this "hybrid" frame-work, an artificial neural network (ANN) is trained to output hidden Markov model (HMM) context-dependent state-level posterior probabilities [1, 2]. The posteriors are converted into quasi-likelihoods by dividing by the prior of the states, which are then used with an HMM as a replacement for the Gaussian mixture model (GMM) likelihoods.

The purpose of this paper is to investigate better features to use as the input to the DNN. Our baseline features are the conventional speaker-adapted 40-dimensional features, which are generated using a setup tuned for the optimal performance with the traditional GMM-based acoustic models. Although we obtained good results using the baseline features, we were interested to investigate ways to increase the dimensionality of the feature vectors beyond the baseline case. This is motivated by the fact that the number of parameters in a DNN does not increase very much when we increase the input dimension, while otherwise leaving the model topology fixed. Hence, DNNs by design are less vulnerable to the un-reliable parameter estimation problem when the dimension of input features is high. Note that this is not the case with HMM/GMMs, where even a small increase in the dimensionality would greatly increase the number of acoustic parameters (means and co-variances); this makes the GMM-based acoustic models subject to the estimation problem, which may cause performance degradation when the dimensionality is high. The optimum choice for input dimension for GMM systems is widely believed to be about 40.

Our baseline features (shown in Figure 1, $d = 40$) are obtained as follows. The 13-dimensional Mel-frequency cepstral coefficient (MFCC) [3] features are spliced in time taking a context size of 9 frames (i.e., $\pm$ 4), followed by de-correlation and dimensionality reduction to 40 using linear discriminant analysis (LDA) [4]. The resulting features are further de-correlated using maximum likelihood linear transform (MLLT) [5], which is also known as global semi-tied covariance (STC) [6]. This is followed by speaker normalization using feature-space maximum likelihood linear regression (fMLLR), also known as constrained MLLR (CMLLR) [7]. The fMLLR in our baseline case has 40×41 parameters and is estimated using the GMM-based system applying speaker adaptive training (SAT) [8, 7] [1].

We investigated the following four ways to increase the dimension, d, of the features beyond 40:

- Type-I : By including additional rows of the LDA matrix beyond 40 (Section 3.1, Figure 1, $d > 40$).
- Type II : Keeping the dimension of the fMLLR transforms 40×41, and passing some of the dimensions "rejected" by LDA, while bypassing MLLT and fMLLR (Section 3.2, Figure 2).
- Type III : Splicing the (baseline) 40-dimensional speaker adapted features again across several frames (Section 3.3, Figure 3).
- Type IV : Splicing the (baseline) 40-dimensional speaker adapted features across several frames, and again decorrelating and performing dimensionality reduction using another LDA (Section 3.4, Figure 3).

The above features are used as the input to the DNN. Consistent improvements in the recognition performance is observed with all four types of features in comparison to the baseline 40-dimensional features. Our best results are obtained with Type-IV features. On the other hand, as expected, we observe that the

S. P. Rath was supported by "Detonation" project within SoMoPro - a program co-financed by South-Moravian region and EC under FP7 project No. 229603. The work was also partly supported by Technology Agency of the Czech Republic grant No. TA01011328, Czech Ministry of Education project No. MSM0021630528, and by European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070).

D. Povey was supported by DARPA BOLT contract No. HR0011-12-C-0015, IARPA BABEL contract No. W911NF-12-C-0015, and the Human Language Technology Center of Excellence. 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, DARPA/DoD, or the U.S. Government.

[1]The baseline recipe is the Kaldi system described in [9].

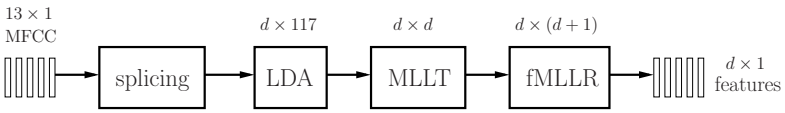

Figure 1: Generation of our baseline/Type I features

performance of GMM-based systems usually deteriorates with the investigated features.

The rest of the paper is organized as follows. In Section 2, we describe our DNN training setup. In Section 3, we provide details of the four types of features that we investigated. In Section 4, we discuss our experimental setup, and present the results in Section 5. Finally, we conclude in Section 6.

2. Our DNN training setup

Most of the details of our DNN setup are based on [10]. The neural networks had 4 hidden layers. The output layer is a softmax layer, and the outputs represent the log-posterior of the output labels, which correspond to context-dependent HMM states (there were about 2600 states in our experiments). The input features are either the standard 40-dimensional features in the baseline case, or various higher-dimensional features that we describe in this paper. The number of neurons in the hidden layer is the same for all hidden layers, and is computed in order to give a specified total number of DNN parameters (typically in the millions, e.g. 10 million for a large system trained on 100 hours of data). The nonlinearities in the hidden layers are sigmoid functions whose range is between zero and one. The objective function is the cross-entropy criterion, i.e. for each frame, the log-probability of the correct class. The alignment of context-dependent states to frames derives from the GMM baseline systems and is left fixed during training.

The connection weights were randomly initialized with a normal distribution multiplied by 0.1, and the biases of the sigmoid units were initialized by sampling uniformly from the interval [-4.1,-3.9][2]. The learning rate was decided by the "newbob" algorithm: for the first epoch, we used 0.008 as the learning rate, and this was kept fixed as long as the increment in cross-validation frame accuracy in a single epoch was higher than 0.5%. For the subsequent epochs, the learning rate was halved; this was repeated until the increase in cross-validation accuracy per epoch is less than a stopping threshold, of 0.1%. The weights are updated using mini-batches of size 256 frames; the gradients are summed over each mini-batch.

For these experiments we used conventional CPUs rather than GPUs, with the matrix operations parallelized over multiple cores (between 4 and 20) using Intel's MKL implementation of BLAS. Training on 109 hours of Switchboard telephone speech data took about a week for the sizes of network we used (around 10 million parameters).

3. Investigated Features

3.1. Baseline/Type-I features

Figure 1 shows the generation of Type-I features. The dimension of the final features supplied as the input to the DNN is denoted as d. The baseline features correspond to d=40. The features are derived by processing the conventional 13-dimensional MFCCs. The steps are as follows:

- Cepstral mean subtraction is applied on a per speaker basis.
- The resulting 13-dimensional features are spliced across ± 4 frames to produce 117 dimensional vectors.
- Then LDA [4] is used to reduce the dimensionality to d. The context-dependent HMM states are used as classes for the LDA estimation.
- We apply MLLT [12] (also known as global STC [6]). It is a feature orthogonalizing transform that makes the features more accurately modeled by diagonal-covariance Gaussians.
- Then, global fMLLR [7] (also known as global CMLLR) is applied to normalize inter-speaker variability.

In our experiments fMLLR is applied both during training and test, which is known as SAT. In some cases, the results are also shown when it is applied only during test.

3.2. Type-II features

The main concern with our Type-I features is that as we increase the dimension of the features, we also (quadratically) increase the number of parameters in the fMLLR transforms. As a consequence the speaker-specific data might become in-sufficient for reliable estimation of the parameters when d becomes large (e.g., 80 or more). In addition, Type-I features require training of the HMM/GMMs in the higher dimensional space which can be problematic. Our Type-II features (Figure 2) are designed to avoid the above problems by applying speaker adaptation to only the first 40 coefficients of the LDA features, and passing some of the remaining dimensions directly to the neural network while bypassing MLLT and fMLLR. It also avoids the training of the HMM/GMMs in the higher-dimensional space.

3.3. Type-III features

Another way to increase the dimension of the features, while keeping the dimension of fMLLR matrices 40×41, is to splice the baseline 40-dimensional speaker adapted features again across time and use them as the input to the DNN (Figure 3). The Type-III features are most closely related to the previous work in this area [13, 11].

3.4. Type-IV features

The Type-IV features (Figure 3) consist of our baseline 40-dimensional speaker adapted features that have been spliced again, followed by de-correlation and dimensionality reduction using another LDA. We use a variable window size in this case (typically ± 4 frames) and the LDA is estimated using the state alignments obtained from the baseline SAT model.

We do not believe that the dimensionality reduction provided by this LDA is something very useful; rather the whitening effect on the features will be favorable for the DNN training. The LDA would work as a pre-conditioner of the data, making it possible to set higher learning rates leading to a faster learning, especially when pre-training is not used.

4. Experimental setup

The experimental results are reported with the acoustic models trained on a 109-hour subset of the Switchboard Part I training

[2] It has been found that where training data is plentiful, pre-training does not seem to be necessary [11] and conventional random initialization [1] will suffice. In this work we do not use pre-training.

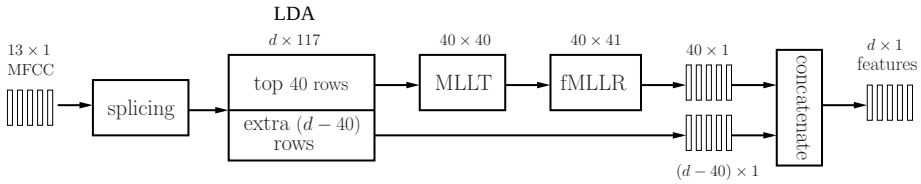

Figure 2: Type-II features: using extra rows of LDA matrix.

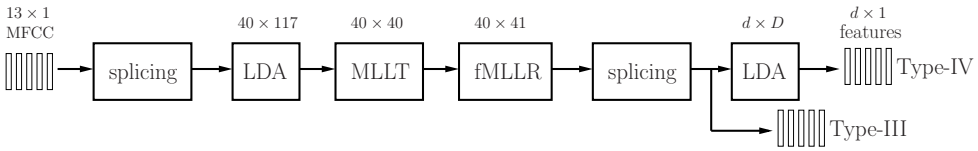

Figure 3: Type-III and IV features: splicing speaker-adapted features (Type-III), followed by de-correlation using LDA (Type-IV).

Table 1: WER (%) with GMM system using baseline features. The results are shown on Hub5'00-SWB and Hub5'00 (shown in brackets) test sets.

Type of feature	WER (%)
LDA+MLLT (no adaptation)	34.6 (42.5)
+fMLLR in test time	26.9 (34.4)
+fMLLR train/test (SAT)	25.6 (32.7)
+fBMMI+BMMI	**21.6** (**29.2**)

Table 2: WER (%) with GMM using baseline/Type I features. Results are shown on Hub5'00-SWB and (Hub5'00) test sets.

d	LDA+MLLT (un-adapted)	+fMLLR test	+fMLLR train/test (SAT)
40	34.6 (42.5)	26.9 (34.4)	25.6 (32.7)
60	36.1 (42.3)	27.0 (34.3)	24.9 (32.2)
80	36.2 (43.2)	27.2 (34.8)	25.3 (32.6)
100	38.5 (44.4)	28.8 (36.2)	26.1 (33.9)

Table 3: WER (%) with GMM using Type-II and IV features.

dimension of feature (d)	feature Type-II	feature Type-IV	
	WER (%)	context length	WER (%)
40	25.6 (32.7)	5	27.4 (35.0)
60	25.8 (33.7)	5	27.8 (35.3)
80	26.7 (34.4)	9	29.0 (36.3)
100	27.3 (34.9)	9	29.7 (37.1)

set (the total training data is 318 hours). The subset contains data from 1351 speakers. We used a separate 5.3 hour development set for cross-validation for the neural network training – it is used to set the learning rates and to decide when to terminate the training. The tri-gram language model was trained on the Switchboard Part I transcripts.

The baseline HMM/GMM system is trained using the Kaldi [9] example scripts for Switchboard. The sequence of systems that we build for the HMM/GMM baseline is: (i) monophone system, (ii) triphone system with MFCC+Δ + $\Delta\Delta$ features, (iii) triphone system with LDA+MLLT, (iv) triphone system with LDA+MLLT+SAT (v) discriminative training of the above system using first feature-space boosted MMI (fBMMI) and then model-space boosted MMI. Note that the fBMMI is similar to the form of fMPE described in [14], but uses the objective function of boosted MMI (BMMI) [15] instead of that of MPE.

For the DNNs trained using fMLLR features, we used the decision tree and state alignments from the GMM-based LDA+MLLT+SAT system as the supervision for training. The fMLLR transforms of the training/test speakers are taken from the same GMM system. Similarly, for DNNs trained using unadapted features (i.e. LDA+MLLT), the decision tree and alignments are obtained from the LDA+MLLT GMM system. The decision tree in both cases had about 2600 leaves, which was optimized for the GMM system. In all experiments, unless otherwise stated, the total number of parameters in the neural networks was about 8 million. Our DNNs had 4 hidden layers; this leads to hidden layers with around 1200 nodes in each.

Test was conducted on the eval2000 test set, also known as Hub5'00, which has 3.72 hours of speech. Note that in [13] the results are reported only on the Switchboard subset (Hub5'00-SWB) of Hub5'00 test set, excluding data from the Callhome subset. In this paper, the results are presented on both sets, with an emphasis given to the Hub5'00-SWB subset. The results on Hub5'00 are shown in brackets in all Tables.

The best word error rate (WER) we report on Hub5'00-SWB is 18.8%, while the authors of [13] report 15.2% on the same test data. The major differences in the experimental setup are that we used a 109 hour subset of Switchboard Part I for training, whereas the full 318 hours of data has been used in [13]; we tested with a language model trained only on the Switchboard Part I transcripts and used the 30k-word lexicon supplied with the Mississippi State transcripts, whereas 2000 hours of Fisher transcripts interpolated with a written-text language model, and a 58k-word lexicon were used in [13]. It is possible that there might be other differences involved that are specific to the Switchboard recipe, but in general, we find that Kaldi is competitive with other systems. So far as acoustic modeling is concerned, we believe that we are comparing with a reasonable baseline.

5. Experimental results

5.1. Results with GMM systems

Table 1 shows the baseline results with various GMM-based systems. The best result is provided by the discriminatively (fBMMI+BMMI) trained GMMs. The results of GMMs with Type-I and Type-II/Type-IV features are presented in Tables 2 and 3, respectively. We note that the WERs with these features are usually worse than the results given by the baseline features.

We do not present the results of discriminative training over the non-baseline features as they were usually worse. The WERs with Type-III features were worse than Type-IV features and are not presented.

Table 4: WER (%) with DNN using baseline/Type I features

d	LDA+MLLT (un-adapted)	+fMLLR test	+fMLLR train/test (SAT)
40	25.3 (32.6)	22.9 (29.4)	22.0 (28.4)
60	23.4 (30.6)	21.6 (28.0)	19.7 (26.5)
80	23.4 (30.1)	21.5 (27.7)	**19.5** (**26.1**)
100	22.9 (29.9)	21.2 (27.4)	19.8 (26.2)
117	23.4 (30.4)	21.7 (28.0)	20.0 (26.4)

Table 5: WER (%) with DNN using Type-III features.

dimension of feature (d)	context length for splicing	feature Type-III
40	no splicing	22.0 (28.4)
200	5 frames	19.7 (26.0)
440	11 frames	**19.7** (**25.8**)

5.2. Results with DNNs

5.2.1. Baseline/Type-I features

Table 4 shows results with the baseline/Type I features. The experiments are conducted in three ways: without speaker adaptation, speaker adaptation only during test, and speaker adaptive training (i.e. SAT). We note that a substantial improvement is obtained by speaker adaptation applied only during test, and a further improvement from SAT. Our overall best result with Type-I feature is 19.5% (26.1% on Hub5'00), which is given by the 80 dimensional features, using SAT. The relative improvements obtained by selecting the optimal dimensions over the baseline feature are 10.5%, 8.0%, 12.8%, that correspond to the three columns of Table 4, respectively.

We note from the experiments that simply increasing the feature dimension by including extra rows of LDA can be quite useful. Confirming the results of [13], we conclude that the speaker adapted features generated using fMLLR can be used as the input to DNNs with good advantage. However, it is also observed that the performance of this type of feature degrades as d becomes large, i.e., $d \geq 100$. The main reason is that the size of the fMLLR transforms becomes too large (more than 10,000 parameters) for reliable estimation of the parameters from the limited speaker-specific data. For instance, on average there was about 3 minutes of data from each speaker in the test set.

5.2.2. Type-II features

The results with the Type-II features are presented in Table 6. Note that in this case the size of fMLLRs is kept fixed at 40×41. We can see that this type of feature helps to reduce the WER compared to the baseline case as we increase the feature-space dimension – the best WER being given by 117 dimensional features, which is 20.1% (26.5% on Hub5'00). In addition, unlike the Type-I features, the performance does not degrade even when the dimension is very large. Hence, Type-II processing is a suitable way to increase the input dimension, while ensuring robustness to speaker adaptation.

We note, however, that the best result with Type-II features is worse than Type-I features (Table 4) that gives 19.5% (26.1% with Hub5'00) as the best WER. We believe that this would still not hold true if there was only a small amount of adaptation data available from the speakers, as in this case the estimated transforms for Type-I features would be poor.

5.2.3. Type-III features

The WERs with Type-III configuration are shown in Table 5. This is the type of features investigated by others in this area [13, 11]. Such features are also expected to provide robustness to speaker adaptation as the dimension in which adaptation is carried out is only 40. The best result in this configuration is obtained by splicing the frames with context lengths of 11 (or 5), which is 19.7% WER (25.8% on Hub5'00). We also note that on the Hub5'00-SWB set the performance of Type-I features is slightly better than the Type-III features, i.e., 19.5% WER compared to 19.7%, respectively.

Table 6: WER (%) with DNN using Type-II and IV features.

dimension of feature (d)	feature Type-II	feature Type-IV	
	WER (%)	context length	WER (%)
40	22.0 (28.4)	5	21.5 (28.0)
60	20.6 (26.8)	5	20.3 (26.7)
80	20.3 (26.5)	9	19.7 (26.0)
100	20.4 (26.5)	9	19.4 (25.7)
117	**20.1** (**26.5**)	-	-
200	-	9	**19.0** (**25.4**)
300	-	9	19.3 (25.4)
400	-	11	19.3 (25.6)
	With increased #parameters (12 million vs. 8)		
200	-	9	**18.8 (25.1)**

5.2.4. Type-IV features

The lowest WER is achieved with the Type-IV feature processing. Although we did not try all possible configurations, the best result among the experiments we conducted is obtained with a context length of 9, i.e. ± 4 frames. It gives a further 0.7% absolute reduction in WER compared to the lowest WER given by Type-III features (Table 5), i.e., from 19.7% to 19.0%, which is a 3.7% relative reduction.

We were able to get a further improvement by training a DNN with more parameters (12 million rather than 8), which improved the performance to 18.8%.

5.3. Comparison with GMM-based system

If we compare with GMM-based systems, our best DNN is substantially better than our best GMM system (SAT+fMMI+BMMI), i.e., a reduction in WER from 21.6% to 18.8% on Hub5'00-SWB, which is a 14.9% relative reduction, and from 29.2% to 25.1% on Hub5'00, which is a relative reduction of 16.3%. This is in the same ballpark as the improvement we see in [13], when comparing similar techniques. The "best" results from their GMM-based system, which included *only* model-space discriminative training, was 20.4% WER on Hub5'00-SWB, and the best WER with their DNN system was 16.3%, which is 20.0% relative improvement.

6. Conclusions and further work

In this paper, we explored various methods of providing higher-dimensional features to DNNs, while still applying speaker adaptation with fMLLR of low dimensionality. We found the Type-IV feature to be the most useful one among all. We were also able to show a substantial reduction in WER compared to our best (single system) WER using GMMs and discriminative training. Our results are consistent with the previous work reported in the literature in that we get similar improvements when we compare with similar baselines.

Further work that we would like to do in this area includes: testing whether initial MFCCs of dimension larger than 13, or an initial LDA dimension higher than 40, or an initial context window size larger than ± 4, would help as the input to DNNs.

7. References

[1] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[2] G. Hinton and L. Deng et. al, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82 –97, Nov. 2012.

[3] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuous Spoken Sentences," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. ASSP-28, no. 4, pp. 357–366, August 1980.

[4] R. O. Duda, P. E. Hart, and David G. Stork, "Pattern classification," in *Wiley*, November 2000.

[5] R. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. IEEE ICASSP*, 1998, vol. 2, pp. 661–664.

[6] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Trans. Speech and Audio Proc.*, vol. 7, no. 3, pp. 272–281, May 1999.

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

[8] S. Matsoukas, R. Schwartz, H. Jin, and L. Nguyen, "Practical implementations of speaker-adaptive training," in *DARPA Speech Recognition Workshop*, 1997.

[9] D. Povey, A. Ghoshal, et al., "The Kaldi Speech Recognition Toolkit," in *Proc. of IEEE ASRU*, 2011.

[10] K. Vesely, M. Karafiat, and F. Grezl, "Convolutive bottleneck network features for LVCSR," in *Proc. of IEEE ASRU*, 2011, pp. 42–47.

[11] N. Jaitly, P. Nguyen, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," *Interspeech*, 2012.

[12] R. A. Gopinath, "Maximum Likelihood Modeling with Gaussian Distribution for Classification," in *Proc. of ICASSP*, Sydney, 1998.

[13] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. of IEEE ASRU*, Dec. 2011, pp. 24–29.

[14] D. Povey, "Improvements to fMPE for discriminative training of features," in *Proc. of Interspeech*, 2005.

[15] D. Povey, D. Kanevsky, et al., "Boosted MMI for model and feature-space discriminative training," in *Proc. of IEEE ICASSP*, 2008.