INTERSPEECH 2009 BRIGHTON

Investigation into bottle-neck features for meeting speech recognition

František Grézl, Martin Karafiát and Lukáš Burget

Speech@FIT, Brno University of Technology, Brno, Czech Republic

{grezl,karafiat,burget}@fit.vutbr.cz

Abstract

This work investigates into recently proposed Bottle-Neck features for ASR. The bottle-neck ANN structure is imported into Split Context architecture gaining significant WER reduction. Further, Universal Context architecture was developed which simplifies the system by using only one universal ANN for all temporal splits. Significant WER reduction can be obtained by applying fMPE on top of our BN features as a technique for discriminative feature extraction and further gain is also obtained by retraining model parameters using MPE criterion. The results are reported on meeting data from RT07 evaluation.

Index Terms: Bottle-neck, ANN architecture, features, LVCSR

1. Introduction

The possibility of obtaining features for standard Gaussian mixture model (GMM) based HMM recognition system from neural network has been studied for several years. In the beginning, Hermansky [1] proposed the *tandem* feature extraction in which posterior probability estimates obtained from artificial neural network (ANN) are modified to create an input to standard GMM-HMM recognizer.

Although *probabilistic features* have not reached the performance of standard MFCC or PLP features, they exhibit great complementarity to them. This encouraging property led to research addressing three parts of ANN: input features, ANN structure, and output classes.

As ANN *input features*, standard PLPs or MFCCs, or more innovative features, such as TRAPs [2] and their modifications (e.g. [3]), were used. The question of *ANN structure* was usually approached by combination of several smaller ANNs. As examples of this effort, *Tonotopic Multi-layered Perceptron* [4] and *Split Context* ANN architecture [5] should be named. As *output classes*, phoneme units were used at the beginning. However, using sub-phoneme classes such as phoneme states as ANN targets was more successful [5].

Thanks to these efforts, the probabilistic features soon became part of the state-of-the-art LVCSR systems [6, 7]. Nevertheless, they still did not themselves reach the performance of standard features.

2. Bottle-Neck features

2.1. Overview

The recently proposed *Bottle-Neck* features [8] are also obtained as a product of ANN, but they are not derived from the class posteriors. To obtain Bottle-Neck (BN) features, five layer ANN with narrow – bottle-neck – middle layer is used and the features are based on linear outputs of the neurons in the bottle-neck layer. These features significantly and consistently outperform the probabilistic features and reach the same or even better performance than standard features.

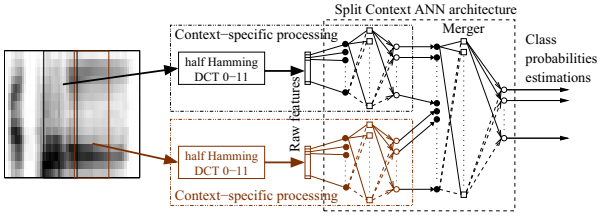

Figure 1: Block diagram of Split Context ANN architecture.

The advantage of the bottle-neck approach is its simplicity, as the training of ANN is done in the same way as for probabilistic features, only more layers are used. Moreover, the size of feature vector is independent on the number of ANN output classes, which allows for the use of phoneme states as ANN targets without the need of cruel dimensionality reduction.

In the ongoing research [9], BN features were derived from different input ("raw") features and used in several LVCSR tasks. The structure of ANN and its training targets were investigated together with the use of deltas. BN features outperformed the standard features in all tasks.

2.2. Split Context ANN Architecture

The bottle-neck approach can be introduced back into ANN architecture, similarly to HATS [10]. The advantage of bottle-neck is greater modeling power compared to only one ANN layer used in HATS. The architecture of great interest for BN features is the Split Context (SC) ANN, suggested in [5], which systematically outperformed a single ANN.

In SC ANN, a block of input vectors is split into left and right contexts of the current frame[1]. Each context block is classified separately and the resulting posterior estimates are fed into a merger ANN to obtain the final classification. The scheme of split context architecture is depicted in Fig. 1.

Since the probabilistic features obtained by this ANN architecture outperformed features obtained from one ANN, we expected the same behavior also on BN features. The first step is to use BN ANN as the merger. But taking into account that for GMM-HMM the BN outputs form better features than probability estimates, it is reasonable to expect that they will be also better input features for the merger ANN. This hypothesis is supported by better performance of HATS over classical TRAP approach [10]. BN ANNs are therefore used also as context-specific classifiers and BN outputs (without any post-processing) are used as merger inputs.

[1] Schwarz's paper [5] contains a justification of this approach comparing splitting context to breaking N-grams in language modeling.

2.2.1. Extensions of Split Context

The possible splits of the block of parameters presented at the input of a classifier were examined up to five temporal splits in [11]. The authors reported improved performance of the system with increasing number of splits.

If three splits are considered, then the merger input $\mathbf{V}$ can be written as $\mathbf{V}_t = [\mathbf{Y}_L, \mathbf{Y}_C, \mathbf{Y}_R]$ where $\mathbf{Y}$ are outputs of left, center and right context-specific ANNs each covering $1/3$ of input block $\mathbf{X}_t = [\mathbf{x}_{t-cont} \dots \mathbf{x}_t \dots \mathbf{x}_{t+cont}]$. These context-specific ANNs are trained with respect to the label associated with center frame $\mathbf{x}_t$ of the input block. The left and right context splits therefore do not contain this labeled vector and corresponding ANNs are forced to focus on the information carried by coarticualtion.

2.3. Universal context

In the split context approach, the context-specific ANNs are trained to output the label associated with the center frame $\mathbf{x}_t$ of the whole input block, $\mathbf{X}_t = [\mathbf{x}_{t-cont} \dots \mathbf{x}_t \dots \mathbf{x}_{t+cont}]$. This might be quite far from the input vectors covered by the given ANN. This way of training of context-specific nets seems to make sense when the probabilities are used at the input of merger ANN that classifies the central frame $\mathbf{x}_t$ of the whole input block. But the ANN is capable of more complex operation than simple "assembly" of partial probability estimates. Especially when working with BN outputs, the concept of merging the partial probability estimates into final ones naturally disappears.

Thus we may think of parameters on the merger input $\mathbf{V}_t$ as about another representation of underlying speech signal represented by $\mathbf{X}_t$. The fact, that this information was obtained by three different ANNs with respect to classification of a certain frame does not play a role. We may as well use the same ANN to obtain BN outputs from all three context splits. Then for the three-split system $\mathbf{V}$ becomes $\mathbf{V}_t = [\mathbf{Z}_{t-k}, \mathbf{Z}_t, \mathbf{Z}_{t+k}]$, where $\mathbf{Z}$ is output of context-independent – *universal* – ANN. This universal ANN covers smaller block of input parameters $\mathbf{X}_{t_U} = [\mathbf{x}_{t_U-U_{cont}} \dots \mathbf{x}_{t_U} \dots \mathbf{x}_{t_U+U_{cont}}]$, where U_{cont} is the context of the input block respective to its center frame $\mathbf{x}_{t_U}$.

The universal ANN is trained with respect to its central frame $\mathbf{x}_{t_U}$. The k is shift of t_U against t. If three splits are considered, the input to merger $\mathbf{V}_t$ is created by sampling the outputs of universal ANN at times $t,\ t-k,\ t+k$, but more samples and even a sampling that is asymmetrical with regard to t can be considered.

The universal ANN will convey maximum information about its center frame $\mathbf{x}_{t_U}$ in its BN outputs. If there is a useful information for classification of the central frame of the whole input block $\mathbf{X}_t$ in BN representation of farther context splits, the merger should be able to extract and use it. The scheme of this idea is depicted in Fig. 2. This approach will be called Universal Context (UC) because only one – universal – ANN is used to extract parameters from context splits.

By replacing context-specific ANNs by a general one, significant simplification was achieved: it is obvious, that the ANN does not have to be in the system several times. Instead, processing of the smaller – contextual – block is done frame by frame and stacked, and only desired frames are taken to form merger input. The number of trainable parameters in the system is therefore reduced, allowing for training of larger ANNs to reach the same number of trainable parameters in whole architecture. The stacking of context-independent ANN outputs

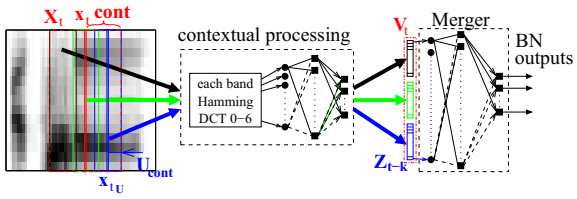

Figure 2: Block diagram of Universal Context approach.

is also convenient when experimenting with different numbers of temporal splits.

3. Experiments

3.1. Experimental setup

Our system is based on AMI-LVCSR system used in NIST RT'07 evaluation [7] which is quite complex system running in many passes. For detailed analysis of the novel features, we stopped the process after the first decoding pass and estimation of VTLN warping factor. The system was simplified by omitting the constrained MLLR adaptation and lattice generation followed by four-gram Language Model (LM) expansion, and full decoding using bi-gram LM was done instead. The LM scale factor and the word insertion penalty were tuned for the best WER.

The task is to recognize meeting speech recordings as defined by NIST RT'07 evaluations. The independent head set microphone (IHM) condition with reference segmentation was used in our experiments.

The training set consists of the complete NIST, ISL, AMI and ICSI meeting data – about 180 hours.

Mel-PLP features appended with derivatives Δ, Δ^2 and Δ^3, are transformed by HLDA to 39 dimensional vector. The HLDA considers each Gaussian component as a class. Resulting parameters are mean- and variance-normalized per speaker and are used as standard features (further denoted as HLDA-PLP). Cross-word tied-states triphone GMM-HMMs models were trained by Maximum Likelihood (ML). The model contains 5600 tied states with 18 mixture components per state. The performance of this baseline is given in Tab. 1.

The systems for different BN features were trained by single pass retraining from HLDA-PLP baseline system. Next, 18 maximum likelihood iterations followed to better settle new HMMs in the new feature space.

Feature concatenation is quite common for probabilistic features, so it was also tested for BN features. The results are reported for BN features separately, in concatenation with HLDA-PLP and for BN features appended with their first derivatives (denoted by _D) – these derivatives help overcome the HMM assumption of frame independence and significantly improve system performance [9].

3.2. BN feature extraction

The raw features are based on 23 short-term mel-scaled log-energies normalized by VTLN and speaker-based mean and variance normalization. The input block to ANN contains 31 frames of these energies. This is kept constant over all experiments. For the baseline BN features, the processing continues by weighting the energy trajectories (TRAPs) by Hamming window and projection on first 16 Discrete Cosine Transform (DCT) bases including the DC component. These raw features

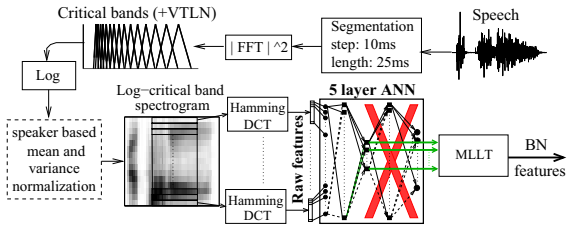

Figure 3: Block diagram of Bottle-Neck feature extraction.

HLDA-PLP	36.0
baseline BN	33.3
baseline BN_D	32.3
HLDA-PLP + baseline BN	31.7
SC-M BN	32.2
SC-M BN_D	30.5
HLDA-PLP + SC-M BN	30.6

Table 1: WER [%] of PLP features, baseline BN features and Split Context architecture with bottle-neck in the merger.

have $23 \times 16 = 368$ elements and form the input to a five-layer ANN with bottle-neck of size 30 in the middle layer[2]. The sizes of the first and the third hidden layer are equal[3]. The ANN of about 2 000 000 trainable parameters is trained to classify 135 targets corresponding to phoneme states on about 173 hours of speech data[4]. Finally, linear outputs of the bottle-neck layer are transformed by Maximum Likelihood Linear Transform (MLLT), which considers HMM states as classes, and mean- and variance-normalized. These features are further referred as *baseline BN* and their performance is shown in Tab. 1. The block diagram of BN feature extraction is shown in Fig. 3.

3.3. Flavors of BN features

Split Context – The left and right contexts cover 16 frames of input log-energies and they overlap by 1 frame. The energy trajectories are weighted by corresponding half of Hamming window and projected on 11 DCT bases including DC component. The number of inputs to each context-specific ANN is $23 \times 11 = 253$ elements. The total amount of trainable parameters in ANN architecture was kept the same: 2 000 000. The amount of parameters in individual ANNs was the following: $2 \times 1/6$ in context-specidic ANNs and $2/3$ in the merger[5].

First, the merger was replaced by a bottle-neck ANN. The obtained features are denoted *SC-M BN*. They achieved 1% absolute better results than the BN baseline – see Tab. 1.

The bottle-neck structure was then used also in context-specific ANNs and outputs of bottle-neck layers were used as inputs to the merger. Maximum amount of useful information from context-specific ANNs is needed to ensure proper classification ability of the merger (and thus quality of derived BN features), so experiments with different sizes of context-specific

[2]This size was chosen as optimal with respect to farther processing in previous experiments [9] and was not tuned here.

[3]This applies for all ANN with bottle-neck

[4]Some parts of data causing problems in ANN training were discarded

[5]Experiments with different proportions were done with only slight effect on the final performance

features	Context-specific ANNs BN size 50	60	70	80	90
SC BN	31.3	31.2	30.8	30.6	30.6
SC BN_D	30.4	30.0	29.8	29.6	29.5
HLDA-PLP + SC BN	30.5	30.2	29.8	29.5	29.7

Table 2: WER [%] of BN features generated by SC architecture with bottle-neck in all stages.

features	Context-independent ANN BN size 50	60	70	80	90
SC BN	31.2	30.5	31.1	30.2	30.5
UC BN_D	30.0	29.5	29.9	29.3	29.3
HLDA-PLP + UC BN	29.5	29.1	29.5	29.2	29.4

Table 3: WER [%] of BN features generated by universal context architecture.

ANN bottle-necks were done. The results can be seen in Tab. 2. Here, further improvement over 1% absolute was achieved.

Split Context – three splits – The problems with training the context-specific ANNs on the edges were encountered and the classification accuracy of the whole architecture was only slightly better than the classification ability of the central ANN alone. We hypothesize that there is not enough information in the left or right context to classify the frame outside of this context. The study in [11] was done on a small data set, so it is possible that the context ANNs learned the most frequent phoneme context of given target. The WER of obtained BN features was rather disappointing.

Universal Context After several experiments, we converged to the following configuration: The contextual ANN covers 11 frames of input mel-scaled log-energy spectrogram. The energy trajectories are weighted by Hamming window and projected on 6 DCT bases including DC component. The resulting ANN input vector has $23 \times 6 = 138$ elements. The amount of trainable parameters was about 1 000 000. The input to the merger was formed by five BN outputs of universal ANN corresponding to five blocks of log-energy spectrogram overlapping by six frames. The merger had about 1 000 000 trainable parameters, too.

The performance of this architecture is shown in Tab. 3. Small but consistent improvement over SC architecture is achieved.

3.4. Discriminative training

BN features is a feature extraction scheme based on discriminative training. Therefore, it is interesting to compare and combine our feature extraction technique with other discriminative training techniques used in speech recognition. Namely, we have examined Minimum Phone Error (MPE) training of model parameters [12] and fMPE [13]. The comparison and combination with fMPE is particularly interesting as fMPE is an alternative discriminative feature extraction technique. However, while neural net is trained to estimated phoneme state posterior probabilities for each frame in the case of BN features, in case of fMPE, the ensemble of linear feature transformations [14] is discriminatively trained to optimize the MPE criterion, which is believed to be better related to our task of speech recognition.

Table 4 presents the results for three different feature sets:

- HLDA-PLP

features	Training ML	MPE	fMPE	fMPE+MPE
HLDA-PLP	35.6	32.6	31.4	29.7
UC BN70_D	29.6	27.9	27.8	27.6
HLDA-PLP + UC BN70	29.4	27.5	26.9	26.1

Table 4: WER [%] of BN and HLDA-PLP features using different techniques.

- UC BN70_D – UC with 70 neurons in contextual ANN bottle-neck augmented with delta coefficients – one of our best performing feature sets based purely on BN processing
- HLDA-PLP+UC BN70 – feature set concatenating both the HLDA-PLP and the UC BN70 (no deltas) stream

For each feature set, the results are shown for initial ML-trained model, model re-trained using MPE, model ML-trained on the features processed using fMPE, and the last mentioned model additionally re-trained using MPE.

Comparing the two discriminative feature extraction schemes, we see that the ML results obtained with UC BN70_D features (29.9% WER) compare favorably to fMPE HLDA-PLP (31.4% WER). Applying fMPE on top of BN feature extraction and MPE training of the models brings further significant gains. Highest gains are, however, obtained with fMPE and MPE applied on HLDA-PLP+UC BN70 features consisting of both BN and HLDA-PLP feature streams. This suggests that fMPE is able to extract additional complementary discriminative information contained in the "raw" features that was already lost during the BN processing.

4. Conclusions

The improvement of Bottle-Neck features through different ANNs architecture is described in this paper. Starting with BN features generated by a single ANN, we obtained the performance of 33.3% WER – more than 2.5% absolute better than HLDA-PLP baseline. When both features are concatenated, the improvement increases to 4.3%.

When BN ANNs were introduced into Split Context architecture, the WER decreased by another 2% absolute reaching the level of 29.5%. Here, the addition of HLDA-PLP features does not bring an improvement and the same performance is achieved by BN features appended with their delta parameters. The best performance is obtained by architecture with bottle-neck of size 80 neurons in context-specific ANNs.

Increasing the number of temporal splits in SC architecture led to degradation of the system as the context-specific ANNs on the edges of the input block were not able to learn the target form its coarticulation behavior.

The developed Universal Context architecture performs about the same as SC architecture reaching the performance of 29.3% WER in the best case. This configuration is also much less sensitive to the size of the contextual ANN. In addition to better performance, the resulting system is simpler as one context-independent ANN is used for all temporal splits.

To examine the behavior of discriminative BN features together with discriminative training, two techniques were evaluated – Minimum Phone Error training of models and fMPE features. We have shown that BN features compares favorably to fMPE as an alternative discriminative feature extraction technique. The combination of BN features with fMPE and MPE training brings additional significant gains.

5. Acknowledgements

This work was partly supported by European projects AMIDA (FP6-033812) and WeKnowIt (FP7-215453), by Grant Agency of Czech Republic project No. 102/08/0707, by Czech Ministry of Education project No. MSM0021630528 and by Czech Ministry of Interior project No. VD20072010B16. František Grézl was supported by post-doctoral project of Grant Agency of Czech Republic, No. 102/09/P635.

6. References

[1] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP 2000*, Turkey, 2000.

[2] H. Hermansky and S. Sharma, "TRAPs – classifiers of temporal patterns," in *5th International Conference on Spoken Language Processing (ICSLP)*, Sydney, Nov 1998.

[3] F. Grézl and H. Hermansky, "Local averaging and differentiating of spectral plane for TRAP-based ASR," in *Proc. Eurospeech 2003*, Geneva, Switzerland, 2003.

[4] B. Chen, Q. Zhu, and N. Morgan, "Tonotopic multi-layered perceptron: A neural network for learning long-term temporal features for speech recognition," in *Proc. ICASSP 2005*, Philadelphia, PA, USA, Mar. 2005.

[5] P. Schwarz, P. Matějka, and J. Černocký, "Towards lower error rates in phoneme recognition," in *Proceedings of 7th International Conference Text, Speech and Dialogue 2004*, 2004, p. 8.

[6] A. Janin, A. Stolcke, X. Anguera, K. Boakye, O. Cetin, J. Frankel, and J. Zheng, "The ICSI-SRI Spring 2006 meeting recognition system," in *3nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, May 2006.

[7] T. Hain et al., "The AMI system for the transcription of speech meetings," in *Proc. ICASSP 2007*, Honolulu, Hawaii, USA, Apr 2007, pp. 357–360.

[8] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP 2007*, Honolulu, Hawaii, USA, Apr 2007, pp. 757–760.

[9] F. Grézl and P. Fousek, "Optimizing bottle-neck features for LVCSR," in *2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008, pp. 4729–4732.

[10] B. Chen, Q. Zhu, and N. Morgan, "Learning long-term temporal features in LVCSR using neural networks," in *Proc. ICSLP 2004*, Jeju Island, KR, Oct. 2004.

[11] P. Schwarz, P. Matějka, and J. Černocký, "Hierarchical structures of neural networks for phoneme recognition," in *ICASSP*, Toulouse, France, may 2006.

[12] D. Povey, "Discriminative training for large vocabulary speech recognitionk," Ph.D. dissertation, Cambridge University Engineering Department, Mar. 2003.

[13] D. Povey, "Improvements to fMPE for discriminative training of features," in *Proc. of Interspeech2005*, Lisbon, Portugal, Sep 2005, pp. 2977–2980.

[14] B. Zhang, S. Matsoukas, and R. Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction," in *Proc. of Interspeech2006*, Pittsburgh, PA, USA, Sep 2006, pp. 2977–2980.