Hierarchical Neural Net Architectures for Feature Extraction in ASR

František Grézl and Martin Karafiát

Brno University of Technology, Speech@FIT, Brno, Czech Republic
{grezl,karafiat}@fit.vutbr.cz

Abstract

This paper presents the use of neural net hierarchy for feature extraction in ASR. The recently proposed Bottle-Neck feature extraction is extended and used in hierarchical structures to enhance the discriminative property of the features. Although many ways of hierarchical classification/feature extraction have been proposed, we restricted ourselves to use the outputs of the first stage neural network together with its inputs. This approach is evaluated on meeting speech recognition using RT'05 and RT'07 test sets. The evaluated hierarchical feature extraction brings consistent improvement over the use of just the first level neural net.

Index Terms: Speech recognition, Feature extraction, Neural network architecture

1. Introduction

Feature extraction using Artificial Neural Network (ANN) in the field of Automatic Speech Recognition (ASR) has been studied for several years. Converting class probability estimates to feature vector – *probabilistic features* – suitable for subsequent Gaussian Mixture Model based Hidden Markov Model (GMM-HMM) recognizer was suggested and evaluated in [1]. Since then, many techniques have been introduced aiming to increase classification accuracy of the ANN and thus the quality of probabilistic features. Recently, a technique using information compressed by a bottle-neck in ANN topology was proposed [2]. These *bottle-neck features* are linear outputs of neurons in bottle-neck (BN) of an ANN and they significantly outperformed the probabilistic features.

The hierarchical classification is a common technique for increasing the classification accuracy of a system. Although all hierarchies can be implemented as one neural net with special architecture, for practical purposes it is usually better to build a classifier from several smaller ones.

Hierarchical structures can be divided into several categories. First, we can consider a hierarchy where output classes of all classifiers are the same and the input of individual stages (except for the first one) are formed by concatenation of outputs of the previous stage with a feature vector, which is also the same for all stages. In this structure, we assume that the subsequent stages are able to correct errors occurring in previous stage taking advantage from already processed input features and their "raw" form. This approach was used in [3].

Another approach to hierarchical classification can be application of subsequently more focused target classes where individual stages perform more and more difficult classification task. Thus, in the first stage, the input features are classified into broad categories, the second stage may divide each category into several groups and subsequent stages will use fine classes as outputs. The idea behind this approach is that it is advantageous to have rough but precise classification first and make it finer consecutively. This was studied in [4, 5].

Next, each stage can use different feature set which is concatenated with the outputs of the previous stage. This allows to progressively add different sources of information. Such hierarchy was used in [6] where first, the low modulation frequencies are processed and the classifiers outputs are then concatenated with high modulation frequencies.

The information about temporal evolution of the classifier's outputs can be also provided to the next level classifier. This can be done in the same way as for conventional features – i.e. by stacking several frames or by computing delta parameters.

Our goal is to examine the behavior of bottle-neck outputs/features in the hierarchical scheme. Bottle-neck features have proved to represent the underlying information better than probabilistic features and thus are able to convey more information to the following GMM-HMM classifier. The advantage of using BN outputs instead of probability estimates was also shown for a structure of several ANNs [7]. This work focuses on further improvement of the performance of BN features studied in [7] through the hierarchical structure.

2. Proposed architecture

It is clear that by combining the above approaches, one can get almost infinite number of possible combinations. To evaluate bottle-neck features in a hierarchical architecture, the first mentioned approach was chosen:

- Only two levels of hierarchy are used – each level will have 1 000 000 weights.
- Both levels have the same target classes – 135 subphonemes classes corresponding to states of phoneme GMM are used.
- Both levels share the same features – critical band energies.

To examine the effect of adding information about temporal evolution of the first level classifier outputs, the following options of the first level outputs are considered as inputs to the second level classifier:

- Only the current frame.
- Five consecutive frames (the current frame and ±2 frames).
- The current frame and its delta parameters.

To further exploit the characteristics of hierarchical approach, three classifiers of different structures are used in the first level of hierarchy. These classifiers were proposed in [7] and their structure will be further described in Sec. 3.2. The

The work was partly supported Grant Agency of Czech Republic project No. 102/08/0707, Czech Ministry of Education project No. MSM0021630528 and by BUT FIT grant No. FIT-10-S-2. F. Grézl was supported by Grant Agency of Czech Republic under project No. GP102/09/P635.

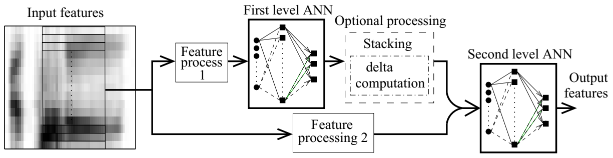

Figure 1: Block diagram of evaluated hierarchical architecture.

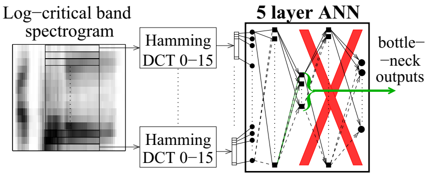

Figure 2: Block diagram of five layer ANN structure and input feature processing

general scheme of evaluated hierarchical architecture is shown in Fig. 1.

3. Specifications of hierarchy

3.1. Input features

The input parameters for both classifiers are based on Mel-scaled Critical Band Energies (CRBE). There were 23 critical bands used in our experimental setup. A part of a two dimensional time-frequency representation of length 31 frames (the current frame ± 15 frames of context) is taken. As shown in block diagram in Fig. 1, different processings are done prior to the first and second level classifiers. For the first level classifier, processing is dependent on the structure of the classifier and will be described further together with the structure itself. The processing prior to the second level classifier is the following: Temporal evolution of energy in each critical band is weighted by Hamming window and transformed by Discrete Cosine Transform (DCT). 16 values are retained including the zeroth DCT component which result in a vector of $23 \times 16 = 368$ elements.

3.2. First level classifiers

The total size of first level classifiers is about 1 M weights. All ANNs are trained to the same sub-phoneme classes and have five layers with bottle-neck in the middle layer.

Five layer ANN with a bottle-neck of size 30 in its middle layer is considered as the first and most simple first level classifier. The processing of the input block of CRBEs is the same as for the second level classifier – Hamming window is applied over each critical band followed by DCT of 0^{th} to 15^{th} bases. The block diagram is shown in Fig. 2. Output features are further denoted as *ANN BN*.

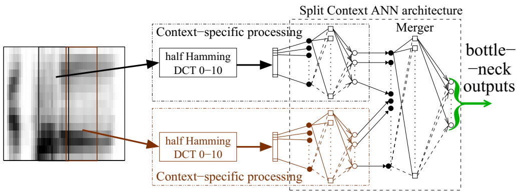
Split Context ANN architecture
Context−specific processing
half Hamming DCT 0−10
Merger
bottle− −neck outputs
half Hamming DCT 0−10
Context−specific processing

Figure 3: Block diagram of Split Context ANN structure and input feature processing

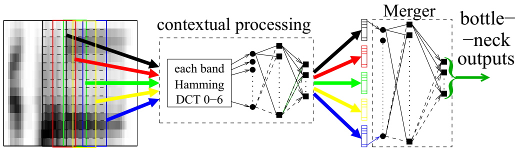

Figure 4: Block diagram of Universal Context ANN structure and input feature processing

Table 1: Frame accuracy of first level classifiers. Target classes are 135 phoneme states of 45 English phonemes.

classifier	CV accuracy [%]
5-layer ANN	53.2
Split Context	54.5
Universal Context – 3 splits	56.1
Universal Context – 5 splits	56.8

Split Context (SC) is a structure of three ANNs [3]. Here, the block of 31 frames of CRBE is split into left and right context of the current frame. Then each half of an energy trajectory in a given context is weighted by respective half of Hamming window (left or right) and transformed using 11 DCT bases (including the 0th one). The resulting parameters are processed by contextual ANN. The BN outputs of contextual ANNs are then merged together in the third ANN – *merger*. The scheme of feature processing and classifier structure is depicted in Fig. 3. The context ANN's BN has 50 neurons, the merger BN has 30 neurons. Outputs of this system are denoted as *SC BN*.

Universal Context (UC) consists of a tandem of two ANNs. The first – contextual – ANN takes a block of 11 CRBE, weights each energy trajectory by the Hamming window and projects it on 6 bases of DCT (including the 0^{th} one). The BN outputs of the contextual ANN are stacked and every fifth frame is taken to form input for the second ANN – *merger*. The process can be imagined as dividing the block of 31 CRBE into 5 overlapping sub-blocks and forwarding each through the contextual processing. The block diagram of feature processing and classifier structure is shown in Fig. 4. Another version of this system was created with just three outputs of contextual ANN (first, middle and last) at the mergers input. The contextual ANN BN size is 50 neurons, the second ANN BN has 30 neurons. The classifier outputs are denoted as *UC5 BN* and *UC3 BN* features respectively.

The size of BN layer in contextual ANNs was chosen as a trade-off between performance of the classifier and size of BN. The sizes of BN layer in 5-layer ANN and in the mergers were kept optimal for ASR system. For mode details about the above structures, refer to [7].

The accuracies of the first level classifiers can be compared on a cross-validation set, the percentages of correctly classified frames (frame accuracy) are given in Tab. 1. It can be seen that more complex classifiers gain higher classification accuracy.

3.3. Second level classifier

The second level classifier is a five layer ANN with bottle-neck in its middle layer. This ANN has about 1 000 000 weights and the number of neurons in its BN layer is 30.

The input feature vector is formed by concatenation of features described in Sec. 3.1 and outputs of the first level classifier. These outputs can be stacked to provide additional information about their time evolution. To reduce the total number of parameters, delta coefficients can be computed from stacked outputs. Note, that the stacking actually increases the span of underlying CRBE block form 31 frames to 35 frames.

The BN features from the second level classifier are the final outputs of the whole hierarchical architecture.

4. Experimental setup

Our system is based on AMI-LVCSR system used in NIST RT'07 evaluation [8] which is quite complex system running in many passes. For these experiments, the process stopped 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. Full decoding using bi-gram LM was done instead.

The task is to recognize meeting speech as defined by the NIST RT'05 and 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 with applied VTLN are appended with derivatives Δ, Δ^2 and Δ^3 and transformed by Heteroscedastic Linear Discriminant Analysis (HLDA) to 39 dimensional vector. The HLDA considers each Gaussian component as a class. The resulting parameters are mean- and variance-normalized per speaker and 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. Performance of this baseline is given in the first line of Tab. 2.

The system with HLDA-PLP features was used to generate forced alignment for ANN training. There were 135 subphoneme labels/target classes corresponding to HMM states of 45 English phonemes including silence. The ANNs were trained on 173 hours of speech.

BN features produced by different hierarchical architectures are transformed by Maximum Likelihood Linear Transform (MLLT), which considers HMM states as classes, then they are concatenated with HLDA-PLP features and the whole vector is mean- and variance-normalized. New models 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.

The two test sets were processed in a different way:

RT'05 results were obtained by rescoring lattices generated by AMI RT'05 evaluation system [9]. The LM scale factor and the word insertion penalty were tuned on this set.

RT'07 results were obtained by full decoding of test data with bi-gram language model (LM) estimated for AMI-LVCSR system used in NIST RT'07 evaluation [8]. The LM scale factor and the word insertion penalty estimated on RT'05 were used here.

5. Results and discussion

The performance of the first level classifiers is evaluated first, see Tab.2. We will compare the performance of features obtained from the hierarchy to these. We can see that adding BN features brings significant improvement over the baseline HLDA-PLP features. It can be also observed, that more complex classifiers achieve lower WER which is in agreement with cross-validation accuracies shown in Tab. 1.

Table 2: WER [%] of HLDA-PLP features and first stage classifiers BN features in concatenation with HLDA-PLP features.

features	RT'05	RT'07
HLDA-PLP	27.6	36.0
ANN BN + HLDA-PLP	23.9	31.9
SC BN + HLDA-PLP	23.5	30.7
UC3 BN + HLDA-PLP	23.5	30.6
UC5 BN + HLDA-PLP	23.1	30.3

The performance of the whole hierarchical structure is evaluated further. The results are shown in Tab. 3. Hierarchical structure tends to bring improvement over the BN features obtained by the first level classifiers in all but one case, which can be explained by non-optimal LM scale factor and word insertion penalty.

Adding information about the temporal evolution of the first level classifier outputs is beneficial for single ANN and SC and UC3 structure where it brings consistent improvement. In case of UC5 structure, the additional temporal parameters seems not to carry useful information. This behavior can be explained by the complexity of the UC5 structure – the input feature block is split into smaller, largely overlapping blocks, each being processed by the same ANN. BN outputs of this contextual ANN are then merged by a merger ANN. Shifting the input block causes shifting the divided block which for certain shift becomes the same but positioned in different time slot. The merger further smooths out the features so the delta parameters become less informative.

Over all, the hierarchy which employs the UC5 structure reaches best performance even without the use of additional temporal information.

5.1. Context classifiers in hierarchy

While experimenting with hierarchical architectures with more complex first layer classifiers, the following question arises: Is it necessary to merge the contextual information in the merger of the first level classifier or is it possible to process it in the second level together with input features?

Since the framework of class probabilities was already abandoned in bottle-neck approach and both levels of hierarchy produce general features, it is easy to include such new structure in the hierarchical architecture. The first level classifier produces features trained to discriminate the classifier targets. When this classifier is treated as one unit, different output features can be chosen. The merger in the SC or UC structure further compresses the information delivered by contextual ANNs. Thus taking directly the contextual BN outputs is nothing else than just obtaining different parameters which could contain more information (the compression by merger is skipped).

When the merger is left out from the first level classifier, its trainable parameters are assigned to the second level classifier to compensate for larger input vector and to keep the number of trainable parameters in whole hierarchical system the same.

The results of this hierarchy are shown in Tab. 4. When context BN outputs are used instead of merger ones, the WER drops in case of SC and UC3 structures and stays about the same in case of UC5 structure. Incorporating five consecutive frames of contextual BN outputs does not bring WER reduction and actually hurts us in case of SC structure. But when delta parameters are presented instead, the performance further improves. This behavior might be caused by the sizable growth of the second

Table 3: Performance of different features obtained from hierarchical architecture. WER [%].

first level classifier	5-layer ANN		Split Context		Universal Context 3		Universal Context 5	
Second layer classifier inputs	RT'05	RT'07	RT'05	RT'07	RT'05	RT'07	RT'05	RT'07
BN + 31 CRBE 16 DCT	23.7	31.0	23.2	30.8	23.4	30.7	**22.7**	**29.5**
BN stacked 5x + 31 CRBE 16 DCT	23.5	30.3	23.0	30.1	23.0	30.2	22.7	29.7
BN + delta + 31 CRBE 16 DCT	23.4	30.2	**22.7**	30.4	23.2	30.3	22.9	29.9

Table 4: Performance of features obtained from hierarchical architecture with contextual BN outputs from the first level. WER [%].

first level classifier	Split Context		Universal Context 3		Universal Context 5	
Second layer classifier inputs	RT'05	RT'07	RT'05	RT'07	RT'05	RT'07
context BN + 31 CRBE 16 DCT	22.7	30.1	23.0	30.3	22.9	29.7
context BN stacked 5x + 31 CRBE 16 DCT	23.2	30.6	23.0	30.1	22.6	29.6
context BN + delta + 31 CRBE 16 DCT	22.5	**29.4**	22.6	29.7	**22.4**	**29.4**

layer classifier input vector size when the contextual BN outputs are stacked – there are 5×100 parameters for SC structure and 5×250 parameters for UC5 structure.

6. Conclusions

In this paper, we presented the incorporation of Bottle-Neck features into hierarchical architecture of classifiers. This architecture was used for feature extraction for LVCSR of meetings and the resulting features were evaluated on NIST RT'05 and RT'07 test sets.

The evaluated architecture consists of two levels, both sharing the same input features. Three different classifiers have been used in the first level of hierarchy – a five layer ANN, Split Context structure and Universal Context structure. Bottle-Neck outputs of these classifiers are appended to input features and form input for second level classifier. In all but one cases, an improvement over just a single (first level) classifier was obtained.

The information about time evolution of the first level classifier outputs was added to the second level classifier by two ways: First, five consecutive frames were stacked and second, delta parameters were added to the current output. Improvement was obtained for a single ANN, SC and UC3 structure, but no improvement was seen in case of UC5 structure. Nevertheless, hierarchy with UC architecture achieves better results even without additional temporal information compare to hierarchies employing single ANN and SC structure.

The good performance of UC5 system can be explained in following way: the input feature block is split in several largely overlapping smaller ones, which are processed by contextual ANN. Its outputs then describe in details the dynamics in the input block, which are encoded in merger outputs. Shifting the input block by several frames then does not have such influence and contextual information (from close neighborhood) becomes redundant.

Finally, we investigated the possibility of presenting the contextual BN outputs directly to the second level classifier, i.e. omitting the merger ANN in the first level classifier. Using only direct features brings improvements over systems with merger in case of SC and UC3 structure and no improvement for UC5 structure. Additional temporal information is beneficial but only in form of delta parameters. This points out the problem of proper representation of this information and the danger of having too many similar parameters without added information.

The advantage of skipping the merger in contextual processing is twofold: first, WER reduction was achieved, and second, the hierarchical structure was simplified. In case of UC structure, we ended up with two ANNs only – contextual and second layer classifier.

The behavior of the UC structure suggests that the block of features on the input to the hierarchy should be split into more largely overlapping blocks. This blocks should be processed in the same way (by contextual processing) to obtain compressed representation of each block. Finally, these partial representation should be appended to original features and presented to the second level classifier. Additional temporal information can further improve the system performance but care has to be taken when choosing the means of its representation.

Over all we were able to improve the RT'05 results by 1% absolute, and RT'07 results by 1.5% absolute by using the hierarchical architecture rather then just one classifier (compare last line of Tab. 4 with Tab. 2).

7. 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] 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.

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

[4] S. Sivadas and H. Hermansky, "Hierarchical tandem feature extraction," in *In Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, 2002, pp. 809–812.

[5] S.-Y. Chang and L. shan Lee, "Improved clustered hierarchical tandem system with bottom-up processing," in *Proc. ICASSP 2008*, 2009, pp. 4441–4444.

[6] F. Valente, J. Vepa, C. Plahl, C. Gollan, H. Hermansky, and R. Schlüter, "Hierarchical neural networks feature extraction for LVCSR system," in *Proc. Interspeech 2007*, 2007.

[7] F. Grézl, M. Karafiát, and L. Burget, "Investigation into bottle-neck features for meeting speech recognition," in *Proc. Interspeech 2009*, Sep 2009, pp. 294–2950.

[8] 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.

[9] T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, M. Lincoln, I. McCowan, D. Moore, V. Wan, R. Ordelman, and S. Renals, "The 2005 AMI system for the transcription of speech in meetings," in *Proc. NIST MLMI Meeting Recognition Workshop*, Edinburgh, UK, 2005.