Integrating recent MLP feature extraction techniques into TRAP architecture

František Grézl, Martin Karafiát

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

Abstract

This paper is focused on the incorporation of recent techniques for multi-layer perceptron (MLP) based feature extraction in Temporal Pattern (TRAP) and Hidden Activation TRAP (HATS) feature extraction scheme. The TRAP scheme has been origin of various MLP-based features some of which are now indivisible part of state-of-the-art LVCSR systems. The modifications which brought most improvement – sub-phoneme targets and Bottle-Neck technique – are introduced into original TRAP scheme. Introduction of sub-phoneme targets uncovered the hidden danger of having too many classes in TRAP/HATS scheme. On the other hand, Bottle-Neck technique improved the TRAP/HATS scheme so its competitive with other approaches.

Index Terms: TRAP processing, Bottle-Neck technique, subphoneme classes, LVCSR features

1. Introduction

Contrary to classical speech recognition schemes where standard features (such as MFCC or PLP) are fed into one (usually GMM-HMM) classifier, the TANDEM approach proposed in [2] treats the outputs of one classifier as features for the second classifier. The first one is a Neural Network (NN) (or a structure of several NNs) trained to produce estimates of posterior probabilities of phonetically motivated classes. The second one is standard GMM-HMM system. As probabilities do not have the desired Gaussian distribution, they were usually processed by logarithm and decorrelated by Principal Component Analysis. The resulting features are called probabilistic features.

In the early days of TANDEM, TempoRAl Patterns (TRAP) processing [1] was often used to generate phoneme posteriors. TRAP probabilistic feature extraction consists of two stages of NNs. The inputs to first stage are derived from long temporal context (up to 1s) of primary/raw features, mostly outputs of Mel-filter bank - critical band energies (CRBE). The temporal evolution of one coefficient (energy in one critical band) forms *TRAP vector*. This *TRAP vector* is converted into phoneme probability estimates by its own NN (band NN). This is done for all coefficients/bands. Outputs from all band NNs are concatenated into one vector which, after logarithm nonlinearity, forms input to Merger NN. This NN combines all band-conditioned estimates into one final set of probability estimates.

Though TRAP probabilistic features neither reached the performance of standard cepstral features nor the performance of the probabilistic features derived by a single neural network with cepstral (PLP) features as its inputs, they are complementary to both of them and were used in the state-of-the-art LVCSR systems [3].

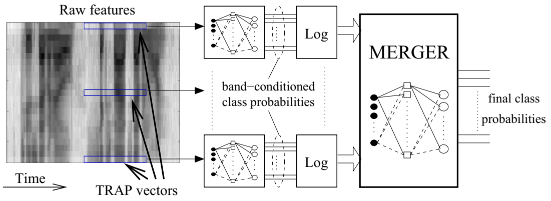
Raw features
Log
band−conditioned class probabilities
MERGER
final class probabilities
Log
Time
TRAP vectors

Figure 1: Scheme of basic TRAP architecture.

Several modifications and enhancement were suggested throughout the years, from which the most promising were dimensionality reduction of the TRAP vector and processing of TRAP vectors from several adjacent bands together [4, 5]. The NN architecture evolved through Hidden Activation TRAPS (HATS) and the Tonotopic NN reaching 4-layer (2 hidden layers) NN with similar performance [6]. HATS architecture was inspiration for a neural network with a narrow-layer in it and development of Bottle-Neck (BN) feature extraction technique [7]. This technique delivers inner product of NN as final features. Also, significant improvement obtained along the way origins from training NN towards phoneme-state targets instead of phoneme ones [8].

All these improvements and simplifications resulted in abandoning TRAP/HATS architecture which required training of multiple NNs in the first stage and a merger in the second one. The handling of whole system was impractical and simpler solutions were preferred.

The enhancements were evaluated on different tasks and direct comparison between some of them is not possible. We evaluated them on the same task – meeting speech recognition as defined by NIST RT'05 and RT'07 evaluations.

Our goal is to put the latest enhancement of MLP-based feature extraction back into TRAP structure and evaluate if the approach can be viable once again. The obvious choice of enhancements is usage of sub-phoneme NN targets and employing the Bottle-Neck structure in NNs. Promising TRAP techniques were identified for: the original TRAP structure is accompanied with TRAP vector dimensionality reduction and one of the three-band processings.

2. TRAP techniques

We will distinguish between *TRAP processing* – the processing done on the TRAP vector(s), and *TRAP architecture*, which will refer to specific configuration of NNs.

2.1. TRAP architecture

The architecture was described in Sec. 1 and in [1]. It consists of two stages with three layer NNs. In the first stage, there are as many NNs as (processed) TRAP vectors. The scheme can be seen in Fig. 1.

This work was partly supported by Technology Agency of the Czech Republic grant No. TA01011328, Czech Ministry of Education project No. MSM0021630528, Grant Agency of Czech Republic projects Nos. GP102/09/P635 and 102/08/0707, and by BUT FIT grant No. FIT-11-S-2

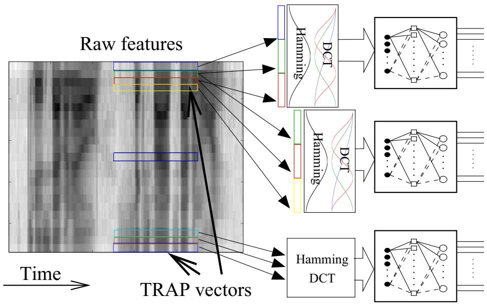

Figure 2: Scheme of three band processing.

2.2. Basic TRAP processing

The TRAP vectors are obtained in simple way in this case. Context of ±25 frames of raw features is stacked and then the temporal evolution of each coefficient forms a TRAP vector. There is no processing between creation of the TRAP vector and NN input.

2.3. TRAP vector dimensionality reduction

TRAP vector dimensionality reduction is an efficient processing which improves the performance of resulting probabilistic features. The 51 points of *Basic TRAP* vector are weighted by Hamming window and projected on 26 DCT bases including the 0^{th} one. We will call this modification *TRAP DCT*.

2.4. Three-band processing

Three-band processing can be seen as taking three coefficients of raw features instead of just one. There are quite some possibilities how to process this block of raw features. Many different approaches were studied in [9] and the most promising ones were evaluated on our experimental setup. The best performing processing selected for further experiments was the following: Three *Basic TRAP* vectors are concatenated and the resulting vector is weighted by Hamming window and projected on 78 DCT bases including the 0^{th} one. This is done with overlap of two *Basic TRAP* vectors. The scheme of three-band processing is shown in Fig. 2. If the number of coefficients in the raw features is N, there is $N-2$ NNs in the first stage. This processing will be denoted as *TRAP3b DCT*.

3. System description

The task is meeting speech recognition as defined by the NIST RT'05 and RT'07 STT evaluations. The independent head set microphone (IHM) condition with reference segmentation was used in our experiments.

The **raw features** are Critical Band Energies (CRBE) computed from 25ms of speech every 10ms. The speech signal is sampled at 16 KHz and there are 23 filters in the filter-bank analysis. CRBE raw features are subject to mean and variance normalization on speaker basis. When creating the context at the beginning and end of speech segment, the samples behind the segment boundary are taken.

The **Neural Networks** in TRAP processing have the same number of weights over all experiments. The total number of weights is 2 000 000. We have experimentally tuned the number of weights associated to the first and second stage of the *Basic TRAP* processing. The best ratio was 0.2:1.8, which means that all NNs in the first stage have together 200 000 weights. The number of inputs to the first stage NNs is either 51, 26 or 78 depending, whether the *basic TRAP*, *TRAP DCT* or *TRAP3b DCT* processing is used. The number of outputs is 45 for phoneme classes or 135 for sub-phoneme classes corresponding to phoneme states. The Merger input size is equal to sum of all first stage NNs outputs and its output size again depends on used target classes.

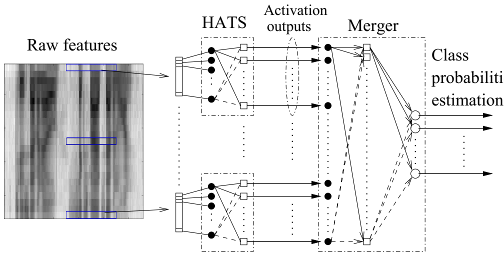

Figure 3: Scheme of HATS architecture.

The transcription for NN training were obtained by forced alignment of training data using enhanced PLP features [10].

Post-processing of Mergers output consists of logarithm and Heteroscedastic Linear Discriminant Analysis (HLDA) decorrelation and dimensionality reduction to 30 dimensions. The HLDA treats every state of corresponding HMM model as class.

The **Recognition system** is based on AMI-LVCSR used in NIST RT'07 evaluation [10] 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 LM scale factor and the word insertion penalty estimated on RT'05 were used here.

The **training set** consists of the complete NIST, ISL, AMI and ICSI meeting data – about 180 hours. The NN were trained on subset of 173 hours.

The **features** used in recognition system are the post-processed outputs from Merger only. Although delta parameters or concatenation with cepstral features improves the performance, for the purpose of comparison of individual techniques it is better to use only outputs from the system under evaluation.

4. Experimental results and discussion

First, the performance of the TRAP architecture with all described processing is evaluated. The NN targets are 45 phonemes and this experiments are our baseline. The results are shown on first line in Tab. 1. We can see that with more elaborate TRAP processing the WER decreases which confirms our previous statements.

4.1. HATS architecture

The Hidden Activation TRAPS architecture (Fig. 3) further improved the performance of resulting probabilistic features [11]. As the name suggests, the outputs of NNs hidden neurons (after sigmoid nonlinearity) are taken to create inputs for merger. The logarithm between first stage outputs and second stage inputs is omitted.

The size of Merger input is now given by the size of hidden layer of NNs in the first stage (with structure 51-90-45 neurons for *Basic TRAP* processing), which significantly increases the Merger input size. We found that changing the ratio of weights in the first and second stage from 0.2:1.8 to 0.1:1.9 and thus effectively decreasing the size of hidden layer of the first stage

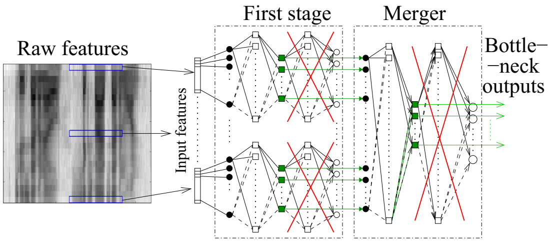

Figure 4: Scheme of Bottle-Neck TRAP architecture.

NN improves the system performance. We hyphotetize that this is caused by the fact, that the hidden layer encodes the information about classes as it would be delivered by the final layer. The increased dimensionality does not bring additional information about the classes needed by Merger and it then suffers from increase of input parameters which do not provide useful information.

The results for HATS probabilistic features are given on the second line in Tab. 1. It can be seen that for all kinds of TRAP processing, the HATS architecture improves by at least 1% absolute over the TRAP architecture.

4.2. Sub-phoneme classes

Training NN for sub-phoneme classes – phoneme states labeled by three state HMM model – instead of phoneme ones improved the phoneme recognition accuracy [8]. This can be accommodated by both architectures, TRAP and HATS.

The phoneme state targets were used in the merger at the beginning. Both architectures were evaluated and results are given in the third an fourth lines of Tab. 1. Comparing TRAP and HATS architectures with phoneme states as Merger targets, we see that HATS still performs better than TRAP but the difference is decreased. When we compare the systems with different Merger targets, we mostly see degradation of system performance, namely for HATS architecture. This might be caused by the mismatch between the first and second stage targets – as the sub-phoneme information is suppressed by the first stage classifiers, the Merger might not be able to recover it.

In the following experiments, the phoneme states targets were introduced also into the first stage classifiers. As the increase of target classes decreases the number of hidden units, the former ratio of 0.2:1.8 between the numbers of first and second stage weights was used in HATS architecture.

The results are given in the next two lines of Tab. 1. We can see that the TRAP architecture degrades drastically. The degradation can be assigned to the fact that the number of inputs to Merger increased three times, but the first stage NNs are not able to provide enough information for sufficient training. On the other hand, the HATS architecture improves over the system with phoneme state targets only in Merger. However, it cannot be said whether it performs better than the original approach with only phoneme targets as for *TRAP DCT* processing the results are better and worse in case of *TRAP3b DCT* processing.

To understand this behavior, more detailed analysis would be needed. It would be necessary to take into account the size of hidden layer in the first stage and to run optimization experiments because the requirements put on the first stage NNs are contradictory: On one hand, we want a good performance of these NNs, which requires bigger size of NN and therefore bigger hidden layer. On the other hand, we want to deliver compact information on Mergers input which requires smaller hidden layer.

Table 2: Frame accuracies and number of hidden units of 6^{th} first stage classifier and three layer Merger in BN-TRAP architecture. Phoneme outputs are used in both stages.

first stage		Merger	
hidden units	6th NN accuracy	merger HU	accu
46,**45**,46	31.2	1666	66.7
49,**40**,49	31.2	1865	67.1
55,**30**,55	31.0	2449	67.7
63,**20**,63	31.2	3564	68.2
74,**10**,74	30.0	6545	67.0
TRAP 90	28.9	1666	65.6
HATS 45	27.7	1759	67.0

4.3. Bottle-Neck approach

The Bottle-Neck (BN) approach [7] was inspired by the HATS architecture and the idea was following: If the outputs of hidden layer can provide better input to Merger than output probabilities, can they also provide better features for GMM-HMM system then probabilistic features are? And the answer was: Yes.

The BN approach is introduced in the Merger NN first, thus it has five layers with a narrow middle layer of size 30 (found optimal in [7]). The other two hidden layers have the same size. As in case of HATS, the logarithm is omitted, because the BN outputs have Gaussian mixture-like distribution. Since the Bottle-Neck outputs have the desired dimensionality, the following HLDA does not perform dimensionality reduction and only rotates the feature space to obtain Bottle-Neck features.

The results from the best performing systems are presented in Tab. 1, Section II. – TRAP and HATS architecture with phoneme targets in both stages and HATS architecture with phoneme states targets in both stages. Comparing the corresponding lines, significant improvement can be seen as it was seen elsewhere when switching from probabilistic to Bottle-Neck features.

4.4. Bottle-Neck TRAP

At this point we propose new Bottle-Neck TRAP (BN-TRAP) architecture which introduces Bottle-Neck NN structure also into the first stage. Thus the problem of contradictory requirements set on the HATS first stage discussed in paragraph of Sec. 4.2 is solved. The scheme of BN-TRAP architecture is shown in Fig. 4.

The optimal size of bottle-neck in the first stage NN should be found first. As mentioned above, these NNs are rather small and we did not know how they would perform. The *Basic TRAP* processing and phoneme targets were chosen for this optimization experiment. A problem was encountered right at the beginning of our effort – the five layer NN training failed to converge (remember that the NN is trained on single TRAP vector and generally does not reach high accuracy). This problem was overcome by training a three layer NN first. Once it was trained, the structure was split and two randomly initialized layers (1^{st} hidden to bottle-neck; bottle-neck to 2^{nd} hidden) were inserted. Then, the whole structure was retrained. Tab. 2 shows frame accuracies and number of hidden units of 6^{th} first stage classifier and three layer Merger in BN-TRAP architecture. The same is given for TRAP and HATS architectures described above on the last two lines in Tab. 2. The proposed approach outperformed both former ones. For further experiments, 45 (for comparison with TRAP/HATS) and 20 bottle-neck hidden units will be used in the first stage classifiers.

Table 1: Performance of different feature extraction techniques. WER [%].

TRAP processing		output	Basic TRAP		TRAP DCT		TRAP3d DCT	
architecture	NN targets	features	RT'05	RT'07	RT'05	RT'07	RT'05	RT'07
Section I: Probabilistic features for TRAP/HATS architectures								
TRAP	45 / 45	probabilistic	28.6	38.7	28.1	38.3	27.3	36.8
HATS	45 / 45	probabilistic	27.2	36.9	27.1	37.0	25.8	35.4
TRAP	45 / 135	probabilistic	28.9	38.5	28.6	38.9	27.0	35.9
HATS	45 / 135	probabilistic	28.2	37.4	28.0	37.1	26.7	35.4
TRAP	135 / 135	probabilistic	29.6	39.3	29.0	39.3	27.9	38.0
HATS	135 / 135	probabilistic	27.5	36.4	27.6	36.9	27.9	38.0
Section II: Bottle-Neck features for TRAP/HATS architectures								
TRAP	45 / 45	Bottle-Neck	27.1	36.2	26.8	36.2	25.9	34.9
HATS	45 / 45	Bottle-Neck	26.4	35.3	26.3	35.3	25.2	33.8
HATS	135 / 135	Bottle-Neck	24.7	33.0	25.3	33.6	24.3	32.0
Section III: Bottle-Neck features for BN-TRAP architecture								
BN-TRAP 45	135 / 135	Bottle-Neck	26.0	34.5	26.4	35.3	25.7	34.1
BN-TRAP 20	135 / 135	Bottle-Neck	24.3	32.9	24.5	32.8	**23.7**	**31.7**

The results obtained with Bottle-Neck features derived from BN-TRAP architecture are shown in the last two lines in Tab. 1. Comparing this two lines, it can be seen that BN-TRAP with smaller bottle-neck layer perform significantly better. Comparing over different TRAP processings shows, that *TRAP DCT* processing performs similar to *Basic TRAP* one. It suggests that the TRAP processing might not be necessary in BN-TRAP architecture. Additional improvement is achieved over HATS BN features for all kinds of TRAP processings.

5. Conclusions

The TRAP and HATS architectures with three different kinds of TRAP processing were evaluated on RT'05 and RT'07 tasks. First, the phoneme targets were used for all NNs as it was originally proposed. The best performing feature extraction scheme was HATS architecture with *TRAP3b DCT* processing at this point.

Further, phoneme-state targets were introduced into Merger NN first, which mostly led into degradation of the system. This can be caused by the fact that first stage classifiers suppress information about phoneme states and it is impossible for the Merger to recover it back. Then, the sub-phoneme classes were used also in the first stage classifiers. This caused failure of TRAP architecture because of enormous increase of Merger inputs which did not bring discriminative information. The HATS architecture improved over the previous case but compared to phoneme targets the results are ambiguous.

The Bottle-Neck technique was applied in Merger NN in the next step. The best performing feature extraction schemes from previous part were evaluated. The results showed significant improvement in all cases.

Finally, the Bottle-Neck technique was applied also in first stage NNs and we named the resulting architecture BN-TRAP. The Bottle-Neck technique separates the number of provided outputs from the number of training targets which is very useful here. Thus the Merger can be provided with compact information regardless the number of the first stage NNs targets. The results show that having smaller bottle-neck layer is beneficial. Note, that the NNs are small and remaining hidden layers are not much bigger, which might negatively influence the resulting performance. Another interesting observation is, that *Basic TRAP* processing reached the performance of the *TRAP DCT* one, thus suggesting that DCT compression removed information which can be utilized now. In all cases, the BN-TRAP improves over previous approaches.

We conclude that TRAP architecture should still be considered interesting approach for feature extraction. Compared to originally proposed BN features (obtained with NN with 2M weights trained on current training data) with WER of 24.8% on RT'05 and 33.3 on RT'07 obtained with the same HMM system, the proposed architecture is superior. During our experiments, many decisions were made with respect to the original setup (3-layer NNs, phoneme targets) which might not be optimal for the system we end up with.

6. References

[1] S. R. Sharma, "Multi-stream approach to robust speech recognition," Ph.D. dissertation, Oregon Graduate Institute of Science and Technology, Oct. 1999.

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

[3] Q. Zhu, A. Stolcke, B. Chen, and N. Morgan, "Using MLP features in SRI's conversational speech recognition system," in *Proc. INTERSPEECH 2005*, Lisbon, Portugal, Sep. 2005.

[4] P. Jain and H. Hermansky, "Beyond a single critical-band in TRAP based ASR," in *Proc. Eurospeech 2003*, Geneva, Switzerland, 2003, pp. 437–440.

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

[6] Q. Zhu, B. Chen, F. Grézl, and N. Morgan, "Improved MLP structures for data-driven feature extraction for ASR," in *Proc. INTERSPEECH 2005*, Lisbon, Portugal, Sep. 2005.

[7] 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.

[8] 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.

[9] F. Grézl, "TRAP-based probabilistic features for automatic speech recognition," Ph.D. dissertation, Brno University of Technology, Czech Republic, 2007, http://www.fit.vutbr.cz/ grezl/publi/dis.pdf.

[10] 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.

[11] 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.