HIERARCHICAL STRUCTURES OF NEURAL NETWORKS FOR PHONEME RECOGNITION

Petr Schwarz , Pavel Matějka and Jan Černocký

Speech@FIT group, Brno University of Technology, Czech Republic
{schwarzp,matejkap,cernocky}@fit.vutbr.cz

ABSTRACT

This paper deals with phoneme recognition based on neural networks (NN). First, several approaches to improve the phoneme error rate are suggested and discussed. In the experimental part, we concentrate on TempoRAl Patterns (TRAPs) and novel split temporal context (STC) phoneme recognizers. We also investigate into tandem NN architectures. The results of the final system reported on standard TIMIT database compare favorably to the best published results.

1. INTRODUCTION

Phoneme recognition plays very important role in speech processing. Phoneme strings are basic representation for automatic language recognition and it is proved that language recognition results are highly correlated with phoneme recognition results [1]. Phoneme posteriors are useful representation for acoustic keyword search, they contain enough information to distinguish among all words and they are small enough to store compared for example to the size of posteriors from context dependent Gaussian Mixture Models (GMM) [2]. Another usable representation for acoustic keyword search are phoneme lattices [3] that can be also generated by phoneme recognizer. Phoneme recognition can also improve speaker recognition [4].

Nowadays, many recognizers are based on Hidden Markov Model (HMM) with probability distributions modeled by GMM. The second big group of phoneme recognizers are HMM / Artificial Neural Network hybrids. In this paper, the second group of recognizers is investigated. Many different structures of neural networks were proposed for this task: for example multilayer perceptrons [5] and recurrent neural networks [7]. We are particularly interested in TempoRAl Patterns (TRAPs) [10] – a hierarchical structure of multilayer perceptrons with separate classification of input patterns in frequency bands, and split temporal context (STC) system [13] – a hierarchical structure of multilayer perceptrons, where a block of spectral vectors is split into several blocks precessed separately. Other architectures are treated for example in [14].

The aim of this article is to compare multiple hierarchical structures of neural networks for phoneme recognition and to emphasize approaches to reach lower recognition error rates, which may be not obvious while working with these structures.

This work was partially supported by EC project Augmented Multi-party Interaction (AMI), No. 506811 and by Grant Agency of Czech Republic under project No. 102/05/0278

2. APPROACHES FOR IMPROVEMENT OF ERROR RATES

The goal is to perform recognition from a long temporal context of speech. The ideal situation would be: 1) infinitely long block of features (for example mel-bank energies), 2) infinitely big training data, 3) infinitely big net, 4) perfect training algorithm. The network would then produce always correct phoneme posteriors and no research would be needed. Unfortunately, none of these conditions is ever met. The following paragraphs make some suggestions for real life:

2.1. Provide additional information

Additional information can be inserted to the NN in several forms:

windowing – if there is a prior knowledge about importance of features in the *input* block, the features can be weighed according to this importance. For example, Hamming window can be used to emphasize features in the center of temporal context that are more important. Weights of neural net are initialized randomly in certain dynamic range of values. After weighting, the center of input block has greater dynamic range, so the training focus is first at these central values. The marginal values are taken into account later.

output representation – additional information added at the *output* of network can reduce phoneme error rate too. The neural network can classify *phoneme states* instead of phonemes (each state is a part of phoneme). In case of whole phonemes, the neural network has hard time to learn all patterns because the time center of actual block of parameters can represent arbitrary part of phoneme. In case of states, there are less patterns per class and these patterns are better time localized. Some improvements have been also seen when a net was trained for multiple tasks in the same time (see [8] for experiments on simultaneous phoneme recognition and gender classification).

2.2. Lower the number of input features

Sometimes, the neural network has too many parameters and there is not enough training data to train them. If the parameters are spread between input and hidden layer, the size of input pattern can be reduced while still keeping the input meaningful. There is usually no degradation in phoneme error rate if, for example, a temporal trajectory of mel-bank energies. Usually, the reduction can be done for example by discrete cosine transform (DCT) projection with subsequent shortening of the output vector.

2.3. Incorporate task specific knowledge to the network

If some knowledge about the task is available, it can be incorporated directly into the network structure. For example, there are many psycho-acoustic experiments showing that speech is processed in critical bands separately [9]. If this assumption is true, the individual bands can be treated separately in early stages of processing and the band-specific knowledge can be merged later.
One representative of such systems is the **TRAPs system** [10], which uses separate neural network for each critical band. These "front-end nets" are trained to classify input patterns to phoneme posteriors. Another neural network (a "merger" or "back-end net") is trained to merge the posteriors from all bands. The outputs are again phoneme posteriors. Separate input patterns for all the front-end nets are simpler than the whole input pattern: they are more easily learned by networks and the input patterns can be longer than if the whole pattern was processed by one net. Phoneme posteriors are not the best representation for input to the merger [14] but they are simple to obtain and in many applications they work pretty well.
Split temporal context system [13] introduces the assumption that two parts of phoneme can be processed independently. The trajectory representing a phoneme in feature space can be split into two parts. The system uses two blocks of features – for left and right contexts (the blocks have one frame overlap). Before splitting, the Hamming window is applied on the whole block so that the original central frame is emphasized. Dimensions of vectors are then reduced by DCT and results are sent to two neural networks. These front-end neural networks are trained to produce phoneme posteriors. The posteriors from both contexts are merged by back-end neural network. In [13], we tested both systems (without and with context splitting), evaluated available amounts of training data for both of them, and showed clear advantage of the split context system.

2.4. Tandem of two nets

Another possibility to improve phoneme error rate is to use a tandem of two neural networks. The front-end network is trained in classical way (for example to classify multiple frames of MFCCs to phoneme or state posteriors). The posteriors from front-end network are sent to back-end net together with the *original input features* (those seen also by the front-end net). In our interpretation, the front-end net prepares phoneme or state space for the back-end network: It could by said that the front-end net it is able to roughly localize phonemes or states and the back-end one performs precise classification. A great benefit of this scheme is that the back-end net is able to process longer temporal context than the front-end net and classify phonemes or states using new information.

2.5. Relations to recurrent neural networks

Recurrent neural networks (RNNs) are reported to reach low phoneme error rates [7]. At least two links between RNNs and the approaches described above can be found:

1. RNN could be decomposed into two networks – front-end which generates state vector and back-end using this state vector for finer classification. RNN actually works similarly as described in section 2.4; one frame delay used in RNN does not really matter in comparison to lengths of contexts (around 30 frames).
2. RNNs create the state vector implicitly, the size is usually 3 to 4 times number of phonemes [7]. This information is actually used during training similarly as it is described in section 2.1.

The advantage of our approaches described above over RNNs is that they are based purely on standard forward neural networks, common training algorithms and existing tools.

3. EXPERIMENTAL SETUP

Databases: The TIMIT database was chosen for phoneme recognition experiments. All SA* records were removed as we felt that the phonetically identical sentences over all speakers in the database could bias the results. The database was divided into three parts – training (412 speakers), cross-validation (CV – 50 speakers) and test (168 speakers). The training and CV subsets are included in the original TIMIT training part.

Phoneme set: The phoneme set consists of 39 phonemes. It is very similar to the CMU/MIT phoneme set [6], but closures were merged with burst instead of with silence (bcl b $\rightarrow$ b). We believe that this is more appropriate for features which use a longer temporal context.

Evaluation criteria: NNs were trained on the training part of the database. The increment in classification error on the cross-validation part during training was used as stopping criterion to avoid over-training. There is one ad hoc parameter in the system, the word (phoneme) insertion penalty, which has to be set. This constant was tuned to minimal phoneme error rate on the cross-validation part of the database. The number of neurons in hidden layer of neural networks was increased until the saturation of phoneme error rate (PER) was observed. The obtained number of hidden layer neurons was approximately 500. All experiments reported in this paper use this number of hidden layer neurons unless stated otherwise.

Training of neural networks: All neural networks were trained using classical back-propagation algorithm with cross-entropy error function. Several iterations of training of the whole system followed by realignment of labels were done. For multi-state systems, the algorithm started with uniform segmentation of phonemes into states. Then, the networks were trained, state posteriors were generated and these posteriors were used in classical Baum-Welch algorithm to produce new labels. The algorithm creates hard labels – one label per frame. The label corresponds to a state with the highest state occupation probability. These new labels are used in the following iteration of NN training.

Software: Quicknet tool[1] employing three layer perceptron with the softmax non-linearity at the output, was used in all experiments. Our own implementation of Viterbi decoder[2] was applied to post-process neural network outputs to produce phoneme strings.

4. EXPERIMENTS

4.1. One network systems

This section compares four different parameterization in ANN/HMM hybrid system with one neural network

1. one vector of MFCC features: 13 cepstral features including C_0, Δ, $\Delta\Delta$ (23 mel-banks were used during calculation), totally 39 features.
2. four vectors of MFCC features. This number of vectors was found to be optimal.

[1] part of SPRACHcore package, developed at ICSI, http://www.icsi.berkeley.edu/~dpwe/projects/sprach/

[2] part of STK toolkit developed at Brno University of Technology, http://www.fit.vutbr.cz/speech/sw/stk.html

system	1 state	3 states
MFCC, 9 frames	39.87	35.56
MFCC, Δ,$\Delta\Delta$	37.67	32.77
MFCC, Δ,$\Delta\Delta$, 4 frames	34.10	29.88
Block of MBE, Hamming, DCT	**29.89**	**28.72**

Table 1. *One network systems - Phoneme Error Rates (%).*

posteriors	PER (%)
3 state	**29.88**
converted to 1 state	31.08
converted to 1 state, min. duration	31.09
1 state	34.10

Table 2. *Three state posteriors converted to one state posteriors in MFCC system with four frames*

3. nine vectors of MFCC features without Δ and $\Delta\Delta$.
4. Block of 31 vectors of mel-bank energies (MBE) $= 310\ ms$. Temporal trajectories in bands were weighted by Hamming window and down-sampled by DCT to 11 coefficients.

Table 1 shows the superiority of long Mel-bank energies but also great improvement coming from three state models. Not the whole improvement is however caused by finer representations of neural network outputs. A part of this improvement comes from the decoding process. To evaluate this, an experiments was done: posteriors from the three state system (with four vectors of MFCC features) were converted to one state posteriors by summing posteriors for each phoneme. This representation was sent to the decoder. Then a minimum duration of phonemes (3 frames) was fixed and the decoder was run again. The results are in Table 2.

The improvement between one and three states is 4.22%. We see that finer representation of neural network output removes 3.02% from PER. The limitation of minimum phoneme duration has no effect and 1.20% comes from the three state strucure in decoder. The improvement in decoder is not surprising: if three-state posteriors are summed within one phoneme, de facto a three state model with arbitrary order of states is created. We know however, that the order of parts of phonemes matters for the recognition.

4.2. Time and/or frequency split architectures

The main question in multi-net architectures is: "Which assumptions are correct – Independent processing of speech in critical bands? Independent processing of different parts of phonemes? Both?" Three architectures were tested: 1) the TRAPs system (Fig. 1a) – separate networks for processing of speech in frequency bands were trained. 2) the split temporal context system (Fig. 1b) – separate networks for processing of blocks of spectral vectors. 3) combination of both (Fig. 1c) – split in both frequency and time.

The question to ask for these architectures is "How many frequency bands to join together or how many blocks of spectral vectors are optimal?" Table 3 shows that it is the best to join 5 bands together in the TRAPs systems (remember that TIMIT is broad-band speech; another experiments showed that this number decreases to 3 bands in case of 8 kHz speech filtered by 15 bands). Two border bands are always shared by two neighboring nets (we have seen that greater overlap improved results). In comparison to simple 1-band TRAPs, the PER decreases dramatically by 3.4% if 5 adjacent bands

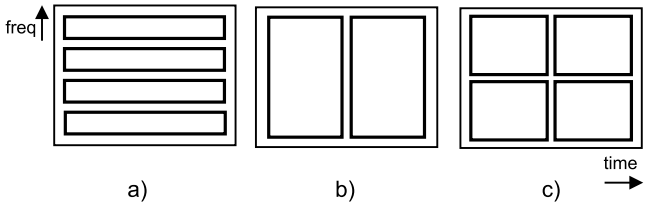

Fig. 1. *Time and/or frequency split architectures.*

# bands per net	1	3	5	7	13
PER (%)	28.24	25.78	**24.84**	24.93	25.62

Table 3. *Optimal number of bands for one neural networks in the 3 state TRAPs system*

are used. This means that the assumption of independent early-stage processing of speech in bands was not verified.

Table 4 shows that it is good to split the temporal context into five blocks. In case of two blocks, half of Hamming window was applied in all bands followed by DCT reduction to 11 coefficients. For 3 or 5 blocks, full Hamming window was applied on each block followed by reduction to 8 or 5 coefficients.

Table 5 compares three different hierarchical structures: TRAPs, split temporal context system (STC) and combination of both called "2x2 system" – two temporal blocks and two frequency bands. This system contains 5 neural networks (2x2=4 and 1 merger). The preprocessing of features for the front-end nets is similar to the preprocessing for the two block STC system. As can be seen, the five block STC works better by 1.4% absolute than the best TRAPs system. The 2x2 system works better than the TRAPs system too and it is better than two- and three-block STC, but it does not reach the performances of the five block STC. Results for one and three states are presented for comparison.

4.3. Tandem of neural networks

In Table 6, properties of concatenation of two neural networks, where the second net sees both posteriors from the front-end net and the original features, are presented. The net replaced by the tandem was the left block of the STC system with two blocks – Table .

Second network (left panel of Fig. 2) is able to add one percent. Third network (right panel) adds another 0.3%. If the context for the third network is extended from 160 ms to 230 ms, the third network is able to add 1%. Tandems of two networks at place of front-end networks in the STC system with two blocks can improve its PER by 1.02% (from 24.41% to 23.39%).

4.4. Tuning the best performance

The STC with 5 blocks was taken and tuned to the best performance mainly by improved NN training: The scheduler for neural network

# blocks	1	2	3	5
PER (%)	26.81	24.41	24.20	**23.44**

Table 4. *Optimal number of blocks in 3 state split temporal context system*

system	1 state	3 states
3 band TRAPs	29.24	25.78
5 band TRAPs	-	24.84
STC - 2 blocks	28.47	24.41
STC - 5 blocks	-	**23.44**
2 x 2	-	24.06

Table 5. *PER for different time and/or frequency split architectures.*

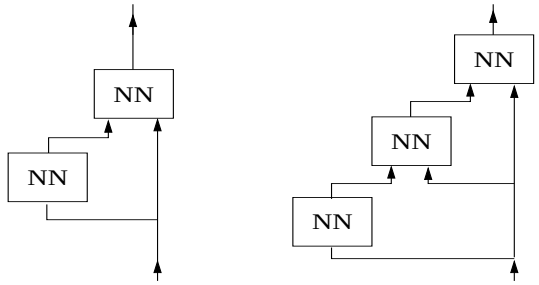

Fig. 2. *Tandem architectures.*

learning rate was changed to use *training set* and set to halve the learning rate learning if the decrease in the frame error-rate (FER) is less than 0.5%. The number of training epochs was fixed at 20. Then, the numbers of hidden layer neurons in nets were increased from 500 to 800. We have seen that it was almost impossible to overtrain neural networks with 800 neurons in 20 epochs, therefore the CV set was added to the training one. At the end, bigram language network trained on phonetic transcriptions of the training part was included. All described steps are summarized in table 7.

5. CONCLUSION

This paper presented several thoughts and experiments concerning architectures of neural nets used for phoneme recognition. The main message should be "adding the most of knowledge about what we want to recognize in all levels (features, output, architecture) is necessary to obtain good results". We have compared TRAPs and split temporal context (STC) systems and concluded the later offer better results. We have also experimented with tandem-NN architectures. Preliminary results show that using one net to "focus" another net on features is advantageous, though this approach needs more experiments.

At the end, we have tuned the five-block STC system by increasing the sizes of neural nets and modifying the training algorithm. The resulting PER of 21.48% is very competitive. Many researches publish their results also on phoneme classification task, therefore we run our system with fixed phoneme boundaries. The classification error rate was 17.19%.

# nets	1	2	3	3 ext.
PER (%)	31.64	30.63	30.33	**29.66**

Table 6. *Tandem of neural networks*

system	PER (%)
baseline	23.44
20 epochs in training	22.71
20 epochs in training + 800n	22.11
+ CV part (18 minutes)	21.82
+ bigram LM	**21.48**

Table 7. *Improvements to the 5-block STC system*

6. REFERENCES

[1] P. Matějka, P. Schwarz , J. Černocký, P. Chytil, "Phonotactic Language Identification using High Quality Phoneme Recognition" in Proc. Eurospeech2005, Sept. 2005.

[2] I. S ozke, P. Schwarz, L. Burget, M. Fapšo, M. Karafiát, J. Črnocký and P. Matěka: "Comparison of Keyword Spotting Approaches for Informal Continuous Speech", in Proc. Eurospeech2005, Sept. 2005.

[3] O. Siohan and M. Bacchiani: "Fast Vocabulary-Independent Audio Search Using Path-Based Graph Indexing", in Proc. Eurospeech2005, Sept. 2005.

[4] Q. Jin, T. Schultz, A Waibel, "Speaker identification using multilingual phone strings", in Proc. ICASSP2002, Orlando, USA, May 2002.

[5] H. Bourlard and N. Morgan. "Connectionist speech recognition: A hybrid approach." Kluwer Academic Publishers, Boston, USA, 1994.

[6] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models", IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(11):1641-1648, November 1989.

[7] A. Robinson," An application of recurrent nets to phone probability estimation", IEEE Transactions on Neural Networks, vol. 5, No. 3, 1994

[8] J. Stadermann, W. Koska and G. Rigoll, "Multi-task Learning Strategies for a Recurrent Neural Net in a Hybrid Tied-Posteriors Acoustic Model", in Proc. Eurospeech2005, Sept. 2005.

[9] J. B. Allen, "How do humans process and recognize speech?", IEEE Trans. on Speech and Audio Processing, 2(4):567-577, 1994

[10] H. Hermansky and S. Sharma, "Temporal Patterns (TRAPS) in ASR of Noisy Speech", in Proc. ICASSP'99, Phoenix, Arizona, USA, Mar, 1999

[11] S. Sharma, D. Ellis, S. Karajekar, P. Jain and H. Hermansky, "Feature extraction using non-linear transformation for robust speech recognition on the Aurora database", in Proc. ICASSP2000, Turkey, 2000.

[12] P. Jain and H. Hermansky, "Beyond a single critical-band in TRAP based ASR", in Proc. Eurospeech2003, Geneva, Switzerland, Sept. 2003

[13] P. Schwarz, P. Matějka, J. Černocky, "Towards Lower Error Rates in Phoneme Recognition", in Proc. TSD2004, Brno, Czech Republic, 2004.

[14] B. Y. Chen, "Lerning Discriminant Narrow-Band Temporal Patterns for Automatic Recognition of Conversational Telephone Speech", doctoral thesis, ICSI, Berkeley, Spring, 2005.