

Investigations into Tandem Acoustic Modeling for the Aurora Task

Daniel P.W. Ellis and Manuel J. Reyes Gomez

Dept. of Electrical Engineering, Columbia University, New York 10027

dpwe@ee.columbia.edu, mjr59@columbia.edu

Abstract

In tandem acoustic modeling, signal features are first processed by a discriminantly-trained neural network, then the outputs of this network are treated as the feature inputs to a conventional distribution-modeling Gaussian-mixture model (GMM) speech recognizer. This arrangement achieves relative error rate reductions of 30% or more on the Aurora task, as well as supporting feature stream combination at the posterior level, which can eliminate more than 50% of the errors compared to the HTK baseline. In this paper, we explore a number of variations on the tandem structure to improve our understanding of its effectiveness. We experiment with changing the subword units used in each model (neural net and GMM), varying the data subsets used to train each model, substituting the posterior calculations in the neural net with a second GMM, and a variety of feature condition such as deltas, normalization and PCA rank reduction in the ‘tandem domain’ i.e. between the two models. All results are reported on the Aurora-2000 noisy digits task.

1. Introduction

The 1999 ETSI Aurora evaluation of proposed features for distributed speech recognition stipulated the use of a specific Hidden Markov Model (HMM)-based recognizer within the HTK software package, using GMM acoustic models [1]. We were interested in using posterior-level feature combination which had worked well in other tasks [2], but this approach requires posterior phone probabilities. Such posteriors are typically generated by the neural networks that replace the GMMs as the acoustic models in hybrid connectionist-HMM speech recognizers [3]. This led us to experiment with taking the output of the neural net classifier, suitably conditioned, as input features for the HTK-based GMM-HMM recognizer. To our surprise, this approach of using two acoustic models in tandem—neural network and GMM—significantly outperformed other configurations, affording a 35% relative error rate reduction compared to the HTK baseline system, averaged over the 28 test conditions, when using the same MFCC features as input [4]. Posterior combination of multiple feature streams, as made possible by this approach, achieved further improvements to reduce the HTK error rate by over 60% relative [5].

The origins of the improvements are far from clear. In the original configuration, there were several possible factors identified:

- The use of different subword units in each acoustic model—context-independent phones for the neural net, and whole-word substates for the GMM-HMM.
- The very different natures of the two models and their training schemes, with the neural net being discriminantly trained to Viterbi targets, and the GMM making independent distribution models for each state, based on EM re-estimation.

- Peculiarities of the way that the posteriors were modified to make them more suitable as features for the GMM. PCA orthogonalization of the net activations before the final nonlinearities gave more than 20% relative improvement compared to raw posterior probabilities.

This paper reports our further investigations, based on the revised Aurora-2000 noisy digits task, to establish the relative roles of these factors, and also to see if the tandem result can be improved even further.

The next section briefly reviews the baseline tandem system. In section 3 we look at varying the training data used for each model, then at the influence of the subword units used in each model, then at the effect of simulating posterior calculation via GMMs. Section 4 presents our investigation into tandem-domain processing including normalization, deltas and PCA rank reduction. We conclude with a brief discussion of our current understanding of tandem acoustic modeling in light of these results.

2. Baseline Tandem System

The baseline tandem system is illustrated in figure 1. In this configuration, a single feature stream (13 PLP cepstra, along with their deltas and double-deltas) is fed to a multi-layer-perceptron (MLP) neural network, which has an input layer of the 39 feature dimensions sampled for 9 adjacent frames providing temporal context. The single hidden layer has 480 units, and the output layer 24 nodes, one for each TIMIT-style phone class used in the digits task. The net has been Viterbi trained by back propagation using a minimum-cross-entropy criterion to estimate the posterior distribution across the phone classes for the acoustic feature vector inputs. Conventionally, these posteriors would be fed directly into an HMM decoder to find the best-matching wordstring hypotheses [3].

To generate tandem features for the HTK system, the net’s final ‘softmax’ exponentiation and normalization is omitted, resulting in the more Gaussian-distributed linear output activation. These vectors are rotated by a square (full rank) PCA matrix (derived from the training set) in order to remove correlation between the dimensions. The unmodified GMM-HMM model is then trained on these inputs. In particular, the GMM-HMM is unaware of the special interpretation of the pre-PCA posterior features as having one particular element per phone.

The results for the tandem baseline system applied to the Aurora-2000 data are shown in table 1. The format of this table is repeated for all the results quoted in this paper, and reproduces numbers from the standard Aurora spreadsheet. Only multicondition training is used in the current work (i.e. training with noise-corrupted data). Test A is a ‘matched noise’ test; Test B’s examples have been corrupted with noises different from those in the training set, and Test C adds channel mismatching. Improvement figures indicate performance rela-

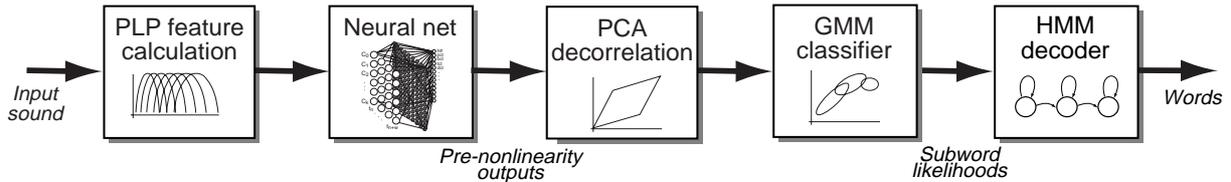


Figure 1: Block diagram of the baseline tandem recognition setup. The neural net is trained to estimate the posterior probability of every possible phone, then the pre-nonlinearity activation for these estimates is decorrelated via PCA, and used as training data for a conventional, EM-trained GMM-HMM system implemented in HTK.

	Avg. WAc 20-0			Imp. rel. MFCC			Avg. imp.
	A	B	C	A	B	C	
Baseline	92.3	88.9	90.0	36.8	19.0	38.5	30.0

Table 1: Performance of the Tandem baseline system (PLP features, 480 hidden unit network, standard HTK back-end). The first three columns are the average word accuracy percentages over SNRs 20 to 0 dB for each of the three test cases; the next three columns are the corresponding percentages of improvement (or deterioration if negative) relative to the MFCC HTK standard; the final column is the weighted average percent improvement, as reported on the standard Aurora spreadsheet.

tive to the standard MFCC-based HTK system; an improvement of +30.0 indicates the test system committed 30% fewer errors than the standard.

Note at this stage how the tandem system shows much less improvement relative to the GMM HTK baseline in test B than in test A: The neural net is apparently learning specific characteristics of the noise, and this limits the benefit of tandem at higher noise levels; for test B, the average improvement across the 10 dB SNR cases is 33.4%, but at 0 dB SNR (which dominates the overall average figures) it is only 10.1%.

3. Experiments

This section presents three experiments conducted to test specific hypotheses about the tandem system. First we looked at the effect of using the same or different data to train each model, then we tried changing the subword units used in both the neural net and the GMM system. Finally we substituted a distribution-based GMM for the posterior-estimating neural net.

3.1. Training Data

Since the tandem arrangement involves separate training of two acoustic models (net and GMM), there is a question of what data to use to train each model. Ideally, each stage should be trained on different data: if we train the neural network, then pass the same training data through the network to generate features to train the GMM, the GMM will learn the behavior of the net on its *training* data, not on the kind of unseen data that will be encountered in testing. However, in the situation of finite training data, performance of each individual stage will be hurt if we use less than the entire training set.

To investigate this issue, we divided the training set utterances into two randomly-selected halves, T1 and T2. The effect of using different training data for each model can be seen by comparing the performance of a system with both the net and the GMM trained on T1 against a system using T1 to train the

	Avg. WAc 20-0			Imp. rel. MFCC			Avg. imp.
	A	B	C	A	B	C	
T1:T1	91.4	88.3	88.9	29.7	14.7	31.8	24.2
T1:T2	91.5	88.7	89.2	29.8	17.9	33.6	25.9

Table 2: Effect of dividing the training data in two halves, and using same or different halves to train each model (net and GMM). T1:T1 uses the same half for both trainings; T1:T2 uses different halves.

net then T2 to train the GMM. Comparing these split systems to the tandem baseline reveals the impact of reducing the amount of training data for each stage. The results are shown in table 2.

We see a slight advantage to using separate training data for each model (T1:T2 condition), but this does not appear to be significant. It is certainly much smaller than the negative impact of using only half the training data, as shown by the overall reduction in average improvement from 30% to 25.9%. We surmise that the separate model trainings are able to extract different information from the same training data, and that the difference in net performance on its training data and on unseen data is not particularly important from the perspective of the GMM classifier.

3.2. Subword Units

One notable difference between the two models in the original tandem format is that the net is trained to context-independent phone targets (imposing a prior and partially shared structure on the digits lexicon), whereas the GMM-HMM system used fixed-length whole-word models. This might have helped overall performance, by focusing the two models on different aspects of the data, or it might have hurt by making the net outputs only indirectly relevant to the GMM's classes.

To test this, we tried to equate the units used in both models. There are two ways to do this: modify the HTK back-end to use phone states, or modify the neural network to discriminate between the subword units being used by the HTK back-end. We tried both: For the all-phone system, we had only to change the HTK setup, providing it with the same pronunciations that had been used to train the net.

For the all-whole-word model, we had to train a new network to targets matching the HTK subword units. To do this, we made forced alignments for the entire training set to the 181 subword states employed by the HTK back-end (16 states for each of the 11 word vocabulary, plus 5 states for the silence and pause models), and trained a new net with 181 outputs. We then orthogonalized these outputs through PCA and passed them to the standard whole-word-modeling HTK back-end. However, the large increase in dimensionality (181 dimensions compared

	Avg. WAc 20-0			Imp. rel. MFCC			Avg. imp.
	A	B	C	A	B	C	
Ph 24:24	92.2	88.3	89.5	35.6	14.4	35.3	27.0
181:181	91.5	90.1	90.2	30.2	27.7	39.3	31.3
181:40	91.9	90.2	90.4	33.8	28.3	40.8	33.2

Table 3: Matching the subword states used in each model. “Ph 24-24” uses 24 phone states for both neural net and HTK back-end. “181:181” uses a 181-output neural net, trained to the whole-word state alignments from the tandem baseline HTK system, and uses all 181 components from the PCA as GMM input features. “181:40” uses only the top 40 PCA components, to reduce GMM complexity.

to 24 for the phone-net) resulted in a back-end with very many parameters relative to the training set size. Thus we also tried restricting the HTK input to reduced-rank outputs from the PCA orthogonalization; we report the case of taking the top 40 principal components, which was by a small margin the best of the cases tried. The results are shown in table 3.

Constraining the HTK back-end to use phones caused a small decrease in performance, indicating that this arbitrary restriction on how words were modeled caused more harm than the possible benefits of harmony between the two models. The net trained to whole-word substates gave rise to systems performing a little better than our baseline, particularly when the feature vector size was limited to avoid an overparameterized GMM. Note that this improvement is mainly due to improvements on Test B; it appears that the whole-word state labels are somehow improving the ability of the net to generalize across different noise backgrounds.

We note that for a digits vocabulary, there are in fact relatively few shared phonemes, and we can expect that the phone-based net is in fact learning what amounts to word-specific subword units (or perhaps the union of a couple of such units). Thus it is not surprising that the differences in unit definition are not so important in practice.

3.3. GMM-derived Posteriors

Our suspicion is that the complementarity of discriminant nets and distribution-modeling GMMs is responsible for the improvements seen with tandem modeling. However, an alternative explanation could give credit to the process of using two stages of model training, or perhaps some fortunate aspect of modeling the distribution of transformed posteriors. Using Bayes’ rule, we can derive posterior probabilities for each phone class $p(q|X)$ from the likelihoods $p(X|q)$ by multiplying by the class prior and normalizing by the sum over all classes. Thus, we can take the output of a first-stage GMM distribution model, trained to model phone states, and convert it into posteriors that are an approximation to the neural net outputs. The main difference of such a system from the tandem baseline is that the discriminant training of the net in the latter should deploy the ‘modeling power’ differently in order to focus on the most critical boundary regions in feature space, whereas independent modeling of each state’s distribution in the former is effectively unaware of where these boundaries lie. Comparing our tandem baseline to a tandem system based on posteriors derived from a GMM should reveal just how much benefit is derived from discriminant training in tandem systems.

We trained a phone-based GMM-HMM system directly on

	Avg. WAc 20-0			Imp. rel. MFCC			Avg. imp.
	A	B	C	A	B	C	
PLP GM	79.4	67.1	75.7	-69	-140	-50	-93
GM:GM	74.8	74.5	75.9	-107	-86	-49	-85

Table 4: Tandem modeling on the output of a GMM first-stage model. The top line, “PLP GM”, shows the performance of our initial phone-based HTK system using PLP features, which is making almost twice as many errors as the HTK baseline (93% worse). Posteriors derived from the corresponding likelihoods were used as the basis for tandem-style features for a standard Aurora HTK back-end, resulting in the second line, “GM:GM”. Neither system involves any neural net.

the PLP features, including deltas. We then calculated posterior probabilities across the 24 phones based on the likelihood models from that training. The log of these posteriors was PCA-orthogonalized, then used as feature inputs to a new GMM-HMM training, using the standard Aurora HTK back-end. The result is a tandem system in which both acoustic models are GMMs. The results are shown in table 4.

The performance of the PLP feature, phone-modeling GMM system is inexplicably poor, and we are continuing to investigate this result. However, the tandem system based upon these models performs only slightly better, in contrast to the large improvements seen when using a GMM to model processed outputs from a neural net. Thus, we take these results to support our expectation that it is specifically the use of a neural net as the first acoustic model, followed by a GMM, that furnishes the success of tandem modeling.

4. Tandem-Domain Processing

In our original development of tandem systems, we found that conditioning of the outputs of the first model into a form better suited to the second model—i.e. by removing the net’s final nonlinearity and by decorrelating with PCA—made a significant contribution to the overall system performance. We did not, however, exhaust all possible modifications in this domain. On the basis that there may be further improvements available from this conditioning stage, we have experimented with some additional processing.

Figure 2 illustrates the basic setup of these experiments. In between the central stages of figure 1 are inserted several possible additional modifications. Firstly, delta calculation can be introduced, either before or after the PCA orthogonalization. Secondly, feature normalization can be applied, again before or after PCA. (In our experiments we have used non-causal per-utterance normalization, where every feature dimension is normalized to zero mean and unit variance within every utterance, but online normalization could be used instead.) Thirdly, some of the higher-order elements from the PCA orthogonalization may be dropped to achieve rank reduction i.e. a smaller feature vector, as with the 181-output network in section 3.2. The motivation for each of these modifications is that they are frequently beneficial when used with conventional features, so they are worth trying here.

We tried a wide range of possible configurations; only a few representative results are given in table 5. Firstly we see that even a small reduction in rank after the PCA transformation reduces performance. Delta calculation helps significantly, as does normalization, and their effects are somewhat cumu-

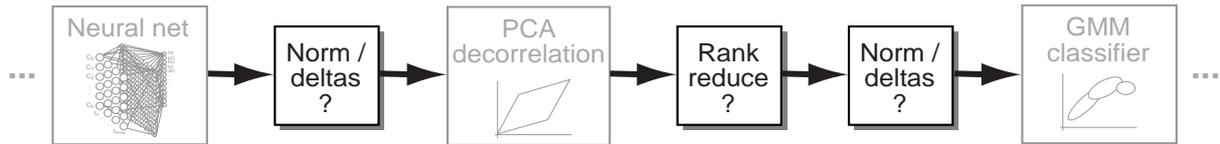


Figure 2: Options for processing in the ‘tandem domain’ i.e. between the two acoustic models.

	Avg. WAc 20-0			Imp. rel. MFCC			Avg. imp.
	A	B	C	A	B	C	
P21	92.2	88.8	89.8	35.6	18.5	37.0	29.0
Pd	92.6	90.4	91.1	39.4	30.1	45.0	37.0
Pn	93.0	91.0	92.4	42.4	34.4	53.1	41.7
dPn	93.2	91.7	92.8	44.0	39.4	55.5	44.9

Table 5: Results of various kinds of tandem-domain processing. ‘‘P21’’ uses only the top 21 PCA components (of 24). ‘‘Pd’’ inserts delta calculation after the PCA (marginally better than putting it before). ‘‘Pn’’ inserts normalization after PCA (much better than putting it first). ‘‘dPn’’ applies delta calculation, then takes PCA on the extended feature vector, then normalizes the full-rank result—the best-performing arrangement in our experiments.

relative when applied in the order shown, giving an overall average improvement that is one-and-a-half times that of the tandem baseline (or a relative reduction of about 20% in absolute word error rate). Improvements occur in all test cases, but are particularly marked in test B; it is encouraging that fairly simple normalization is able to improve the generalization of the tandem approach to unseen noise types.

We also tried taking double-deltas which performed no better than plain deltas. However, delta calculation increases the dimension of the feature vectors being modeled by the GMM. We have yet to try combining delta calculation with rank reduction, although the results with the 181 output net indicate this might be beneficial.

4.1. Feature Combination

As explained in the introduction, one motivation for tandem modeling was to find a way to apply multistream posterior combination to the Aurora task. We have tried some simple versions of this on the Aurora-2 task, just to indicate the kind of benefit that can be achieved. To the PLP features and network we added modulation-filtered spectrogram features (MSG [6]) along with their own, separately-trained neural network. The pre-nonlinearity activations of the corresponding output layer nodes of the two nets were added together, which is the most successful form of posterior combination for tandem systems. Table 6 reports on two variants: baseline tandem processing applied to the combined net outputs, and the same system with normalization after the PCA orthogonalization. Delta calculation offered no benefits for the combined feature system, perhaps because of the slower temporal characteristics of MSG features working through to the posteriors. Using multiple feature streams achieves further significant improvements in both reported cases.

	Avg. WAc 20-0			Imp. rel. MFCC			Avg. imp.
	A	B	C	A	B	C	
Cmb	93.2	91.4	91.9	44.4	37.1	50.0	42.8
CmbN	93.8	92.1	93.7	48.8	42.3	61.3	49.1

Table 6: PLP+MSG feature combinations. ‘‘Cmb’’ is baseline combination, and ‘‘CmbN’’ adds normalization after PCA.

5. Conclusions

The tandem connection of neural net and Gaussian mixture models continues to offer significant improvements over standard GMM systems, even in the mismatched test conditions of Aurora-2000. We have investigated several facets of this approach, finding that it is better to use the same data to train both models than to reduce training set size, that matching subword units in the two models may help a little without being critical, but that including a neural net apparently is critical. We have also tried some variations on the conditioning applied between the two models, which, added to a simple feature combination scheme, can eliminate half the errors of the HTK standard.

6. Acknowledgments

This work was supported by the European Commission under LTR project RESPITE (28149). We would like to thank Rita Singh for helpful discussions on this topic.

7. References

- [1] H.G. Hirsch and D. Pearce, ‘‘The AURORA Experimental Framework for the Performance Evaluations of Speech Recognition Systems under Noisy Conditions,’’ *ISCA ITRW ASR2000*, Paris, September 2000.
- [2] A. Janin, D. Ellis and N. Morgan, ‘‘Multi-stream speech recognition: Ready for prime time?’’ *Proc. Eurospeech*, II-591-594, Budapest, September 1999.
- [3] N. Morgan and H. Bourlard, ‘‘Continuous Speech Recognition: An Introduction to the Hybrid HMM/Connectionist Approach,’’ *Signal Processing Magazine*, 25-42, May 1995.
- [4] H. Hermansky, D. Ellis and S. Sharma, ‘‘Tandem connectionist feature extraction for conventional HMM systems,’’ *Proc. ICASSP*, Istanbul, June 2000.
- [5] S. Sharma, D. Ellis, S. Kajarekar, P. Jain and H. Hermansky, ‘‘Feature extraction using non-linear transformation for robust speech recognition on the Aurora database,’’ *Proc. ICASSP*, Istanbul, June 2000.
- [6] B. Kingsbury, *Perceptually-inspired Signal Processing Strategies for Robust Speech Recognition in Reverberant Environments*, Ph.D. dissertation, Dept. of EECS, University of California, Berkeley, 1998.