# A HIERARCHICAL, CONTEXT-DEPENDENT NEURAL NETWORK ARCHITECTURE FOR IMPROVED PHONE RECOGNITION

*László Tóth*

Research Group on Artificial Intelligence, Hungarian Academy of Sciences and University of Szeged

## ABSTRACT

In this paper we combine three simple refinements proposed recently to improve HMM/ANN hybrid models. The first refinement is to apply a hierarchy of two nets, where the second net models the contextual relations of the state posteriors produced by the first network. The second idea is to train the network on context-dependent units (HMM states) instead of context-independent phones or phone states. As the latter refinement results in a lot of output neurons, combining the two methods directly would be problematic. Hence the third trick is to shrink the output layer of the first net using the bottleneck technique before applying the second net on top of it. The phone recognition results obtained on the TIMIT database demonstrate that both the context-dependent and the 2-stage modeling methods can bring about marked improvements. Using them in combination, however, results in a further significant gain in accuracy. With the bottleneck technique a further improvement can be obtained, especially when the number of context-dependent units is large.

***Index Terms—*** Phone recognition, MLP, HMM/ANN, bottleneck, TIMIT

## 1. INTRODUCTION

The most successful approaches of applying artificial neural nets (ANNs) to automatic speech recognition (ASR) use multi-layer perceptrons (MLP) to estimate local phone or state posteriors. With a slight modification these can replace the Gaussian mixture likelihood estimates in hidden Markov models (HMMs), resulting in the HMM/ANN hybrid model [1]. Alternatively, the MLP can be interpreted as a non-linear transformation, and its output used (after some postprocessing) as input features for a conventional HMM. This tandem construct [2] allows one to combine the advantages of MLPs with the capabilities of sophisticated HMM development tools without the need for modifying the latter.

Since the introduction of the HMM/ANN hybrid, several modifications of the MLP have been proposed in order to make its posterior estimates more accurate. A simple idea is to replace the standard 3-layer "big dumb neural network" with a hierarchic architecture, which then provides room for many modifications, like training the subnets on different features and/or targets. The tandem system also consists of one learner stacked on another, the lower being an MLP, while the upper is the Gaussian mixture model of the HMM. The same idea can be implemented in HMM/ANN hybrids as well, with two MLPs being trained on top of each other. Although this idea appeared a few years ago [3], it was not thoroughly studied until quite recently [4, 5]. 2-stage modelling will be one of the refinements we apply here in order to obtain enhanced posterior estimates.

With conventional HMMs the use of context-dependent (CD) phone models is standard practice. The commonly applied decision tree-based clustering decomposes the CD models into thousands of building units ('tied states' or 'physical states' [6]). As in the hybrid model one MLP output is dedicated to each unit, adapting this methodology to the hybrid directly would require a neural net with a huge output layer. Most authors consider this infeasible, and hence the difficulty of CD phone modelling is still one of the main claims against hybrid recognizers, and only sporadic efforts have been made to solve it [7]. However, the simple idea of constructing CD ANNs by using decision tree-clustered states as training targets re-appeared quite recently, and good results were reported [8, 9]. This method will be the second refinement that we apply in this study.

In a 2-stage hierarchic learner the lower stage is not required to have the same '1-of-N' output representation as the upper stage. In our case this means that there is no need for an output layer of thousands of neurons in the lower net of the hierarchy. A popular technique for reducing the size of the output layer of an MLP while forcing it to represent the same information is the 'bottleneck' method proposed in the framework of tandem systems [10]. This is the third refinement that we are going to apply here.

The structure of the paper is as follows. First we create baseline results in Section 2 by training standard HMM and HMM/ANN hybrid systems on TIMIT. We then introduce and test the first two refinements; that is, 2-stage modelling and CD training in Sections 3 and 4. The combined model is refined further by applying the bottleneck technology in Section 5. Lastly, some extended tests and a comparison with results taken from the literature are given in Section 6.

This research was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

## 2. BASELINE RESULTS

To train and test the various models the TIMIT corpus was used. The training set consisted of the standard 3696 'si' and 'sx' sentences. As training the MLPs requires a cross-validation (CV) set, a random 10% of the training set was separated for this purpose. During the experimentation phase, recognition scores on the core test set (192 sentences) will be reported. All these experiments use a bigram phone model estimated from the training set. In the last section we will also evaluate the best performing model on the complete test set (1344 sentences) and without the bigram model to aid a comparison with results taken from the literature.

As is usual, the 61 phone labels of TIMIT were mapped to a set of 39 labels *before* performing the experiments. However, some researchers carry out the fusion of labels only *after decoding*, just for the evaluation step [11]. We will report results with this strategy as well in Section 6.

For the training and evaluation of the HMM model the HTK Toolkit was applied [6]. The MLP of the hybrid was trained using our own implementation, while the decoding from the frame-level probabilities produced by the MLP was performed by a modified version of HTK's HVite tool. The HMM/ANN hybrid worked directly with the posteriors, that is, without division by the priors, as in earlier phone recognition experiments we obtained better scores with this configuration. As acoustic features the standard 13 MFCCs were used (including the 0th one) along with their $\Delta$ and $\Delta\Delta$ coefficients extracted from 25 ms frames at 10 ms frame skips.

A special feature of HTK's language modelling utilities is that the decoding process must start and end on dedicated start and end symbols, and these must have an acoustic counterpart. If we map these to the leading-ending silences (labelled as h#) of the TIMIT sentences, we have two options during evaluation: if we include the h# labels in the recognition score, we then bias the results because these are guaranteed to be hit by the language model. We could also ignore these labels, but this biases the results the other way because these segments would be recognized with better-than-average accuracy. Following [11], we chose this latter scenario for the experiment phase, but in Section 6 for comparison purposes we also report scores with the h# symbols included.

To obtain baseline scores, first we trained a monophone (context-independent, CI) HMM system with 3-state phone models. The number of Gaussians was gradually increased, using the 'mixture splitting' feature of HTK. The phone recognition error peaked around 30-40 components with scores of about 33.3-33.1%. The next step was to obtain a baseline triphone (context-dependent, CD) system. We configured the decision tree-based state clustering tool of HTK so that it resulted in relatively few, 613 tied states. The corresponding CD-HMM yielded accuracy scores of 29.5-29.0% with 20 to 30 Gaussian components per state. The motivation for working with few physical states was that this many units seemed manageable by MLPs, but we should mention that we could not obtain significantly better results with more states.

| Model | PhER |
|---|---|
| CI-HMM | 33.04% |
| CD-HMM | 29.01% |
| HMM/ANN | 26.77% |

**Table 1**. Phone error rates of the baseline systems.

The baseline HMM/ANN hybrid was trained as follows. The CI-HMM system was used to create force-aligned state labels for the training set. Separate ANN outputs were dedicated to the 3 states of each phone, so altogether the MLP had 117 output neurons. As input 9 neighboring frames were used, so the size of the input layer was 351. The net contained one hidden layer with 5000 neurons, and was trained using backpropagation in semi-batch mode on batches of 256 frames. Training was stopped when the error rate on the CV set stopped improving. The resulting HMM/ANN hybrid produced a phone recognition error rate of 26.77%. The three baseline scores are listed in Table 1.

## 3. TWO-STAGE ESTIMATION OF POSTERIORS

In theory, an MLP with one hidden layer is sufficiently flexible for any practical machine learning task. In practice, however, we can guarantee only locally optimal training. Hence it makes sense to construct a hierarchy of learners, especially if we have a priori knowledge about how the information is structured. Many efforts have been made to create hierarchic phone classifiers, a recent example being described in [12].

In tandem systems the original motivation for combining the ANN and GMM models was simply to maintain compatibility with classic HMMs. However, the hierarchic stacking of classifiers may also enhance the probability estimates they produce. A tandem-like hierarchic combination of MLPs was tested in [3] with good results. The effect of training a second MLP on a longer context of posterior estimates yielded by a first-stage MLP was studied quite recently in [4] and [5]. Both sets of authors found that the hierarchical system attained higher recognition accuracy scores, as the temporal context of posteriors helps the second MLP correct the phonetic confusions of the first MLP, and also supports the modelling of sub-lexical transitions. In the following we will refer to this architecture of two MLPs as the '2-stage model'.

In our 2-stage experiments the MLP of the HMM/ANN hybrid described in Section 2 served as the 1st-stage MLP. The 117 posterior estimates it produced were then logarithmized and normalized (the effect of normalization is discussed in [5]). An MLP with the same structure as above was trained on this data; that is, 9 neighboring frames served as input (now corresponding to $117 * 9 = 1053$ input neurons), while the number of hidden and output neurons was 5000 and

| MLP architecture | PhER |
|---|---|
| 1-stage, CI | 26.77% |
| 2-stage, CI | 26.10% |
| 1-stage, CD | 25.70% |
| 2-stage, CD | 23.76% |

**Table 2**. Phone error rates of the HMM/ANN hybrid for various MLP architectures and training targets.

| | 2nd-stage training targets | |
|---|---|---|
| 1st-stage training method | 613 states | 858 states |
| direct, on CI targets | 23.76% | 23.15% |
| bottleneck, full rand. init. | 24.65% | — |
| bottleneck, 1 rand. layer | 23.53% | — |
| bottleneck, 2 rand. layers | 23.44% | 22.42% |

**Table 3**. Phone error rates for the various training strategies of the 2-stage architecture.

117, respectively (the target labelling was also the same as for the first net). No attempt was made to optimize the size of the input context, though a thorough analysis would be worthwhile ([4] and [5] use a longer temporal context). The phone error rate obtained with these 2-stage posterior estimates was 26.10% (see Table 2). In accord with Ketabdar [4], we found that the 2-stage model is much less sensitive to the phone insertion penalty and language model weight parameters than the conventional systems. Thus in all the 2-stage experiments no particular effort was made to fine-tune these parameters: in all bigram experiments the language model weight was set to 1.0 with no insertion penalty, while in the tests with no language model the insertion penalty was set to -2.

## 4. CONTEXT-DEPENDENT POSTERIORS

In this set of experiments the physical states of the baseline CD-HMM served as training targets for the MLP. As we mentioned in Section 2, this resulted in 613 different target labels, which were then synchronized with the recordings using forced alignment. Because of the increased size of the output layer we reduced the size of the hidden layer of the MLP from 5000 to 2000, in order to decrease the training time and the possibility of overfitting. The input layer and input data was the same as with the conventional HMM/ANN hybrid; that is, 9 frames of MFCC vectors. The phone recognition error rate obtained with this model was 25.70% (see Table 2).

The idea of combining CD training with the 2-stage approach appears natural. Unfortunately, stacking a second MLP on the outputs in this case seems infeasible: with the 613 output neurons, 9 neighboring frames would require an input layer of 5517 units for the second net. To avoid the training of such a huge net, *only the 2nd-stage* was adjusted to the CD targets. That is, the lower MLP was the same CI net that served as the lower stage in the 2-stage CI model. This way the 2nd-stage net had layer sizes of 1053x2000x613. Decoding from the posteriors estimated by this MLP yielded a phone recognition error rate of 23.76% (see Table 2).

## 5. ENHANCEMENT VIA BOTTLENECK TRAINING

Although the above result is very encouraging, there seems to be room for even more improvement. As the first stage was trained to discriminate CI units, the resulting posteriors are not necessarily optimal as features for discriminating the CD units in the second stage. Hence, in the following we seek to modify the training of the 1st stage *while preserving its size*. The effect of modified training will be evaluated by re-training the second net and checking the phone error rate.

The bottleneck training technique [10] was introduced within the framework of tandem models with the aim of making the output size of the MLP independent of the number of training classes (within reasonable limits). What makes this possible is that the MLP of the tandem (or in our case, the 1st-stage MLP) is not required to have a dedicated output for each class. Rather, it should provide a compact representation of the same information. The basic idea of bottleneck training is to construct an MLP (usually with 4-5 layers) with the middle layer having few neurons (this is the bottleneck layer). The net is trained the usual way, but after training the layers above the bottleneck layer are simply thrown away.

We wanted our bottleneck MLP to have the same structure as the lower net in our earlier experiments (to aid comparison), but to be trained with the 613 CD training targets instead of the 117 CI states. Thus we extended the 351x5000x117 network structure with an additional layer of 613 units. This net was then trained on the MFCC features with the 613 context-dependent targets, and after training the topmost layer of 613 units was thrown away. Then the second net of size 1053x2000x613 was re-trained, and the resulting recognizer evaluated. Unfortunately, as Table 3 shows, the score obtained was significantly worse than the previous best result.

Frankel et al. found that bottleneck MLPs are difficult to train, as the narrow bottleneck layer increases the chance of getting stuck in a local optimum [13]. They proposed a 'network growing' technique to alleviate this problem, which could be trivially adapted to our system. We performed network growing as follows: instead of randomly initializing all the weights of the bottleneck MLP, the weights for the first-to-second and second-to-third layer connections were copied from the CI 1st-stage net, and only the weights between the third and fourth layers were initialized randomly. We also ran a variant of this experiment where only the first layer of connections was taken from the CI net, and the two upper layers were initialized randomly. Both initalization schemes brought slight improvements compared to the CI-trained net (see Table 3).

| Language model | core test set | complete test |
|---|---|---|
| bigram (h# not counted) | 22.42% | 22.17% |
| bigram (h# counted) | 21.24% | 21.02% |
| no LM (h# counted) | 21.59% | 21.04% |

**Table 4**. Phone error rates of the best model for both test sets.

## 6. EXTENDED TESTS WITH 61 PHONE LABELS AND THE COMPLETE TEST SET

Some authors train on the 61 phones of TIMIT and fuse the labels just for the evaluation [11]. Motivated by this, we re-trained our two best-performing models with the original 61 labels. For the 1st-stage MLP we kept the size of 351x5000x117; that is, it was not adjusted to the $61 * 3 = 183$ CI states of the 61 labels. For the 2nd-stage MLP we allowed slightly more outputs: re-running the CD-HMM training with the 61 labels we got 858 physical states. Two tests were then performed with this new label set. In the first the 1st-stage MLP was the same CI MLP as that applied in Sections 3 and 4, while in the second test it was adapted to the 858 targets by applying the network growing technique with the two upper layers initialized randomly. The two results are shown in the third column of Table 3. In this case the system with the bottleneck 1st-stage MLP performed significantly better. Also, the scores are much better than those obtained with 613 tied state targets.

The final step of evaluation was to compare our results with those given in the literature. Unfortunately, some authors report results on the core test set, while others report those on the complete test set. Some apply a bigram language model and some do not. Moreover, as was explained in Section 2, it is not clear whether the h# symbols should be ignored or not when decoding with bigrams in HTK (other decoder implementations might behave differently in this respect). Hence, all combinations were evaluated and the results are listed in Table 4. Hifny and Hinton et al. collected a lot of phone recognition results on TIMIT [14, 11], and they report only one better result than ours [15].

## 7. CONCLUSIONS

In this paper we combined three simple training strategies in a novel way to obtain enhanced posterior estimates for HMM/ANN hybrids. We showed that it is possible to replace the usual CI state training targets with CD tied states. The 2-stage modelling strategy can also improve the performance, especially with CD targets. Lastly, the bottleneck training technique introduced for tandem systems both suitable and beneficial for the 2-stage MLP architecture as well. However, more studies are required to see whether the method is scalable to much larger databases, where the number of tied states used is usually much larger as well.

## 8. REFERENCES


[1] H. Bourlard and N. Morgan, *Connectionist Speech Recognition - A Hybrid Approach*, Kluwer, 1994.

[2] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ICASSP*, 2000, pp. 1635–1638.

[3] P. Schwarz, P. Matějka, and J. Černocký, "Hierarchical structures of neural networks for phoneme recognition," in *ICASSP*, 2006, pp. 325–328.

[4] H. Ketabdar and H. Bourlard, "Enhanced phone posteriors for improving speech recognition systems," *IEEE Trans. ASLP*, vol. 18, no. 6, pp. 1094–1106, 2010.

[5] J. Pinto et al., "Analysis of MLP based hierarchical phoneme posterior probability estimator," *IEEE Trans. ASLP*, vol. in press, 2010.

[6] S. Young et al., *The HTK Book*, Cambridge University Engineering Department, 2005.

[7] H. Bourlard, N. Morgan, C. Wooters, and S. Renals, "CDNN: A context dependent neural network for continuous speech recognition," in *ICASSP*, 1992, vol. II, pp. 349–353.

[8] T. Pavelka and P. Král, "Neural network acoustic model with decision tree clustered triphones," in *MLSP*, 2008, pp. 216–220.

[9] A. Abad, T. Pellegrini, I. Trancoso, and J. Neto, "Context dependent modelling approaches for hybrid speech recognizers," in *Interspeech*, 2010.

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

[11] A.-R. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *NIPS 22 workshop on deep learning for speech recognition*, 2009.

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

[13] J. Frankel, D. Wang, and S. King, "Growing bottleneck features for tandem ASR," in *Interspeech*, 2008, p. 1549.

[14] Y. Hifny and S. Renals, "Speech recognition using conditional random fields," *IEEE Trans. ASLP*, vol. 17, no. 2, pp. 354–365, 2009.

[15] S.-M. Siniscalchi, P. Schwarz, and C.-H. Lee, "High-accuracy phone recognition by combining high performance lattice generation and knowledge based rescoring," in *ICASSP*, 2007, pp. 869–872.