# MODULAR NEURAL NETWORKS EXPLOIT LARGE ACOUSTIC CONTEXT THROUGH BROAD-CLASS POSTERIORS FOR CONTINUOUS SPEECH RECOGNITION

*Christos Antoniou*

University of Essex
Department of Computer Science
Colchester, CO4 3SQ, UK
cantona@essex.ac.uk

**ABSTRACT**

Traditionally, neural networks such as multi-layer perceptrons handle acoustic context by increasing the dimensionality of the observation vector, in order to include information of the neighbouring acoustic vectors, on either side of the current frame. As a result the monolithic network is trained on a high multi-dimensional space. The trend is to use the same fixed-size observation vector across the one network that estimates the posterior probabilities for all phones, simultaneously. We propose a decomposition of the network into modular components, where each component estimates a phone posterior. The size of the observation vector we use, is not fixed across the modularised networks, but rather accounts for the phone that each network is trained to classify. For each observation vector, we estimate very large acoustic context through broad-class posteriors. The use of the broad-class posteriors along with the phone posteriors greatly enhance acoustic modelling. We report significant improvements in phone classification and word recognition on the TIMIT corpus. Our results are also better than the best context-dependent system in the literature.

## 1. INTRODUCTION

### 1.1. Problem Statement

Neural networks such as multi-layer perceptrons (MLPs) offer great potential for acoustic modelling when coupled with one state per phone HMM. Traditionally, neural networks handle acoustic context by increasing the dimensionality of the observation vector, in order to include information of the neighbouring acoustic vectors, on either side of the current frame; thus making the minimum of assumptions about the speech generation process. As a result the monolithic network is trained on a high multi-dimensional space. When longer sequences of input features are employed, the dimensions of the input space increase and the data sparsity problem worsens.

Typically, a single monolithic MLP with thousands of hidden nodes is used to simultaneously model all phones. Minimising output error by several criteria will lead to the MLP outputs yielding phone probabilities, posterior on the input space, under the assumptions that the MLP contains sufficient hidden nodes, unlimited training data is available and training does not get stuck in a local minimum. In practice, unlimited speech training data is not available, so training is stopped early using a minimum error criterion on a separate data set. The reasoning is that training must be prevented from fitting too closely to a finite training data set. Applying this procedure to a network with multiple outputs is a necessary awkward compromise. The best stopping point for all outputs combined can be computed, but it is not necessarily ideal for any particular phone. Recent work we published in [1] supports this view.

The above observations have motivated work on decomposing the acoustic modelling task into several sub-tasks. Hierarchical Mixtures of Experts [7] and the ACID/HNN architecture [6] have been used to decompose acoustic modelling in a data-driven fashion. However, the decomposition algorithms for both these architectures depend on a uniform observation feature vector size for all acoustic units. We decompose the network into modular components, where each component estimates a phone posterior. We use a variable observation vector size that depends on the phone that each modularised network is trained to classify. Moreover, for each observation vector, we estimate very large acoustic context through broad-class posteriors.

## 2. A MODULAR DECOMPOSITION METHOD

### 2.1. Acoustic Decomposition by Phone

A monolithic MLP with multiple outputs suggests the possibility of decomposing the mappings it represents, to mappings represented by individual modules and learn these partial mappings separately. The simplest principle of such decomposition is to assign a “part” of the network to a sub-task. We chose to relate these “parts” to each of the 39 phone classes in the classification task and further assign a **detector MLP** with a single output node to each phone. The term ”detector” is adopted as each MLP is trained with a target of 1 for its phone and with a target of 0 for the remaining phones. For any given phone detector, training attempts to maximise the response for the target phone while minimising the response for the remaining competing phones.

Using this decomposition method, we can explicitly match network resources and training to the needs of each phone, depending on their complexity, i.e. hidden nodes, learning regime, input feature dimension, input feature space and early stopping of training. This is demonstrated in previous work we published in [1] [2] [3]. As a result of this decomposition, network optimisation is performed both in training times (e.g. explore parallel training naturally as well as reduction in parameters employed) and in performance on both acoustic and word recognition level.

A possible argument against modular decomposition by phone, is that it repeats the modelling of boundaries between phones. This

is true since modular networks contain in total, a relatively large number of parameters. However, these parameters are not extra degrees of freedom that might be under-determined by the training data. They represent merely the repetition of the same modelling. The extra time taken to train redundant parameters is more than offset by the fact that training modular networks can be distributed across a large number of machines. We literally use whole labs full of PC's to do our training, and this proves an effective experimental approach.

### 2.2. Broad-class Posteriors

It is well known that the expression of a phone is modified by the context in which it is expressed. Co-articulation may be responsible for most of this variability. It follows that it should be possible to estimate a phone posterior conditioned not just on the actual expression of the phone, but on its observed acoustic context as well. Unfortunately, extending the field of view presented to a phone detector as we demonstrate in section 3.2 by simply showing it more and more input-feature frames creates practical problems.

We propose an alternative approach that provides the classification task with a more concise indication of left and right context, such as **broad-class posterior** probabilities. The 39 standard phones were divided into 7 broad-classes of plosives (PL), fricatives (FR), nasals (NA), semi-vowels (SV), vowels (VO), diphthongs (DP) and closures (CL).

In particular, there are two ways of calculating these broad-class posteriors: by creating a separate model for each broad-class, or by simply summing the posterior phone probabilities of the existing phone detectors belonging to each broad-class. Using a feature window of 9 frames and 125 hidden nodes, we calculated posterior probabilities for these methods on the cross-validation set to select that with the highest classification. Simply summing the posteriors of the phone detectors delivers the highest classification. Table 1 shows the phone classification confusion matrix for this method, which results for the 7204 phone segments in the evaluation set. Vertical lines represent the response of each broad-class and the diagonal line the response of the broad-classes on their own phones. Despite some expected confusions (e.g. plosives with fricatives, vowels with semi-vowels and diphthongs), the broad-class posteriors provide a very good estimation of confidence for class discrimination since they deliver 98.1% segments labelled correctly.

| | PL | FR | NA | SV | V | DP | CL |
|---|---|---|---|---|---|---|---|
| PL | **872** | 1 | 0 | 0 | 3 | 1 | 1 |
| FR | 4 | **1030** | 4 | 0 | 2 | 0 | 5 |
| NA | 1 | 3 | **625** | 4 | 4 | 1 | 2 |
| SV | 0 | 5 | 2 | **951** | 13 | 11 | 3 |
| V | 5 | 5 | 7 | 26 | **1716** | 7 | 0 |
| DP | 0 | 0 | 1 | 2 | 5 | **331** | 1 |
| CL | 0 | 3 | 2 | 3 | 2 | 1 | **1540** |

**Table 1**. Broad-class phone classification confusion matrix.

### 2.3. The Modular/Posterior Broad-class Architecture

Fig. 1 presents the modular/posterior broad-class architecture. The first layer uses the phone detectors - each estimating probabilities for their own class -, and the broad-class information described in section 2.2. The second layer, namely the **posterior MLP** layer, serves for a co-operative ensemble combination of posteriors. Each posterior MLP is trained in the same fashion as that of phone detectors. Its job is to combine the outputs of the first layer and deliver an estimate of posterior probability for its phone. In principle it will be posterior on the original observation of the raw speech. In reality, it will be posterior in some complex way on the union of all feature vectors employed by the phone detectors and broad-classes. Training of the posterior MLP is done with the primary detectors held fixed. As most of the work is done by the primary detectors, posterior MLP's are simple and quick to train.

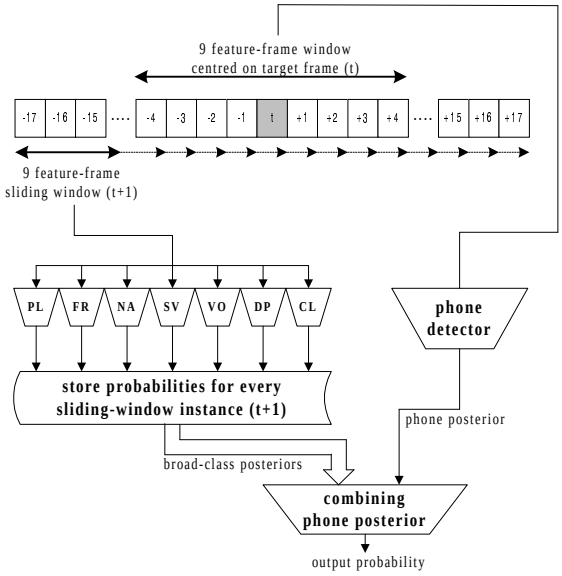


**Fig. 1**. Modular network architecture using broad-class posteriors.

For the classification of a given frame, two sources of information are provided to the combining phone posterior MLP: (1) the output probability of the phone detector, posterior on a 9 frame window centred on the current frame, and (2) the output probabilities of the broad-class detectors, posterior both on the current frame and on the adjacent contexts on either side of the current frame (computed on a 9 frame window). This way, we can determine the broad-class context in which the current phone is expressed, thus literally taking the role of context-dependent modelling in a much simplified fashion.

## 3. EXPERIMENTAL EVALUATION

We selected the continuous speaker-independent TIMIT corpus to evaluate our architecture. It allows us to compare phone classification and recognition results with many other results in the literature. Unfortunately, there is no direct way comparing word error rates with others. We performed two sets of experiments:

1. Employ a number of uniform and steadily increasing input-feature frames to the phone detectors. Furthermore, investigate how these different input dimensions can be customised to benefit different phones belonging to particular group sounds.
2. Employ the broad-class posterior to the system and show

the effectiveness of such information towards the improvements obtained in acoustic and word recognition level.

### 3.1. System Setup

We employ the recommended training set (3696 sentences) to train our models, while the test (1152 sentences) and the core (192 sentences) sets are used for cross-validation and evaluation purposes respectively. Our front-end extracts log-energy along with 15 Mel-Frequency Cepstral Coefficients from a 20ms window at a 10ms frame rate. Additionally, we add delta information computed over a window of 5 frames, to give a feature vector containing 32 coefficients. For comparison purposes with other researchers, we collapsed the TIMIT 61 phone labels into the remaining 39 labels, according to a CMU/MIT mapping taken from [8]. Training of the complete modular architecture (see Fig. 1) took place on 39 Pentium II-450MHz machines and was completed in 7 hours.

We evaluate the performance of our methods using phone classification and word recognition rates. Although not reported very often, phone classification can provide a good measure of the quality of acoustic modelling, since it does not require a grammar or any other syntactical rules. Phone classification is performed by allowing phone models to compete to label each phone segment. To obtain word error rates, the best word string for each sentence is obtained from a time-synchronous Viterbi beam search decoder. A bi-gram model was estimated by scanning through a very large amount of sentences (roughly 90 million words) from a text corpus, namely the British National Corpus (BNC), with a floor value to protect infrequent transitions from becoming zero. A pronunciation model for the TIMIT lexicon containing a total of 6224 words was constructed by the simple expedient of allowing all plausible alternative pronunciations to occur.

### 3.2. Exploring Detectors' Input Layer Size

It is common practice for traditional monolithic MLP's to include a fixed-size input representation. Typically, a constant number of 9 adjacent frames centred on the current moment in time is used to allow modelling of the time development of the phones. However, as the number of input frames get larger (i.e. the space we attempt to model grows), the task for training a monolithic network becomes extremely hard. On the other hand, it can be possible that acoustic modelling improves as more adjacent input frames are employed since we begin to model the target phone in its wider acoustic context. We identify two sources towards this improvement:

1. A wider view of the target phone can be obtained, even when the input frames to the network are mis-aligned with the boundaries of the observed phone;
2. the network is able to "see" additional information about the adjacent phones on either side of the target phone, by taking their expression into account when estimating the probability for the current phone.

For each phone, we trained a set of 7 detectors, using 9, 11, 13, 15, 17, 19 and 21 input frames. All phone detectors carry a fixed number of 125 hidden nodes. Earlier work we published [1] shows how increasing the hidden nodes to a detector MLP adds very little to the recognition task. Table 2 summarises the results, along with the network parameters consumed by each configuration.

The best result is obtained by all phone detectors using the maximum 21 input frames. It is also evident that there is an initial dramatic improvement in performance as more input frames are employed, while performance gradually converges at about 15 input frames. This is mainly due to the large complexity of the input space and the excessive amount of network parameters present. Since we use only roughly 1.5M frames to do our training, the detectors do not have enough freedom to benefit from a 21 input frame window (i.e. 672 dimensions), while such task is computationally very demanding. It is also possible that not all phone detectors require the same input dimension.

| input-layer | phone classif. | word recog. | # parameters |
|---|---|---|---|
| 9 frames | 75.9% | 71.1% | 1,408,875 |
| 11 frames | 76.3% | 71.5% | 1,720,875 |
| 13 frames | 76.8% | 71.8% | 2,032,875 |
| 15 frames | 77.4% | 72.2% | 2,344,875 |
| 17 frames | 77.6% | 72.3% | 2,656,875 |
| 19 frames | 77.7% | 72.3% | 2,968,875 |
| 21 frames | 77.8% | 72.4% | 3,280,875 |

**Table 2**. Phone detectors with variable input-layer size.

| phn | fr | phn | fr | phn | fr | phn | fr | phn | fr |
|---|---|---|---|---|---|---|---|---|---|
| b | 9 | z | 9 | r | 9 | uh | 9 | sh | 9 |
| d | 9 | f | 11 | er | 13 | uw | 11 | l | 13 |
| g | 9 | th | 11 | w | 17 | ey | 21 | ah | 11 |
| p | 9 | v | 13 | y | 15 | aw | 19 | s | 9 |
| t | 9 | dh | 11 | iy | 13 | ay | 17 | ng | 15 |
| k | 11 | h | 15 | ih | 17 | oy | 21 | aa | 9 |
| jh | 13 | m | 13 | eh | 13 | ow | 21 | dx | 11 |
| ch | 11 | n | 15 | ae | 11 | h# | 9 | | |

**Table 3**. Input-frame selection for phone detectors using MSE.

To investigate this hypothesis, we selected for each phone its detector whose given input-layer size delivers the lowest MSE on the cross-validation set (Table 3). In fact, for cases where two or more detectors of the same phone have very similar MSE's, we select that of less computational effort, i.e. less parameters. The performance of the resulting system is evaluated in Table 4. Interestingly, the result is very close to that using 21 frames across all phone detectors, while training times are dramatically reduced, i.e. 40.5% of less parameters. This also verifies the fact that some groups of phones can still achieve high class discrimination with a much reduced input dimension, without of course, any decrease in performance. For example, plosives only require 9 frames, in contrast to diphthongs that require 21 frames. A lot of this can be explained in the frame duration of these sounds.

| input-layer | phone classif. | word recog. | # parameters |
|---|---|---|---|
| as in Table 3 | 77.6% | 72.3% | 1,952,875 |

**Table 4**. Customised input-layer size for phone detectors.

### 3.3. Incorporating Broad-class Posteriors

In section 3.2 we have shown, how extending the field of view presented to a phone detector, by simply showing it more and more input-feature frames, creates practical problems. To overcome this limitation, we introduce the broad-class posteriors. To determine

the appropriate broad-class context, we experimented with a number of window sizes, as shown in Table 5. The left column specifies the broad-class context visible to the combining posterior MLP's. As more broad-class context is provided, performance improves for both the acoustic model and word recognition while network parameters are slightly increased. The best result is obtained with the broad-class posteriors scanning through a context of 35 frames, although simply providing 5 frames gives an equally attractive performance. A comparison performance for the best system that uses the broad-class information and the system without that information, shows an improvement of 10.4% in phone classification and 8.3% in word recognition. Furthermore, network parameters are only increased by 13.3%.

| broad context | phone classif. | word recog. | #parameters |
|---|---|---|---|
| None | 75.9% | 71.1% | 1,408,875 |
| 5 frames | 82.8% | 74.6% | 1,444,950 |
| 9 frames | 83.1% | 75.0% | 1,497,600 |
| 15 frames | 83.5% | 75.7% | 1,596,660 |
| 19 frames | 83.5% | 76.2% | 1,803,750 |
| 23 frames | 83.7% | 76.5% | 1,949,220 |
| 27 frames | 83.9% | 76.7% | 2,153,775 |
| 31 frames | 84.0% | 76.9% | 2,348,385 |
| 35 frames | 84.1% | 77.0% | 2,564,835 |

**Table 5**. Broad-class posteriors providing contextual information.

## 4. COMPARISON WITH PUBLISHED RESULTS

We compare our best result with the best results in the literature. Most methods use different techniques to capture relevant acoustic contextual information. Robinson [11] uses a fully connected NN with recurrent loops to keep information about past inputs for an amount of time that is not fixed *a* priori, but rather depends on its weights and on the input data. They achieve this at the expense of being computationally very expensive to train. Chengalvarayan explains in [5] how an HMM/Gaussian mixtures system can be trained using minimum classification error (MCE) criterion for both context-independent (CI) and context-dependent (CD) modelling. Our result compares favourably with the CD modelling. This is a good result because HMM/NN hybrid methods have historically compared well with standard CI systems, but struggled for parity with CD systems.

| Author | Method | Acc. | Classif. |
|---|---|---|---|
| This Paper | Modular NN | 75.8% | 84.1% |
| Chengal. [5] | CD MCE HMM | — | 83.5% |
| Chengal. [5] | CI MCE HMM | — | 72.4% |
| Robinson [11] | Recurrent NN | 73.8% | — |
| Ming [9] | Bayesian Triphone | 74.4% | — |

**Table 6**. Comparison results with others in the literature.

## 5. CONCLUSION

We have provided some illustration of the effectiveness of a modular decomposition to acoustic modelling, by phone. In particular, we have shown how different input observation vector dimensions for different phones result in a dramatic reduction of network parameters, without any decrease in performance. Furthermore, we have shown how co-articulatory effects can be effective modelled using broad-class posteriors. It seems that our notion of the broad-class posteriors is also a useful one for capturing contextual information for the phones. We achieved an improvement of 8.3% in word recognition, without the need for excessive training of parameters. In fact, our result (84.1%) compares favourably with the best published method (83.5%) that uses a context-dependent Gaussian mixture system with thousands of models to represent each phone in a different left-right context [5].

## 6. REFERENCES


[1] Antoniou, C.A. and Reynolds, T.J. (1999) Using Modular/Ensemble Neural Networks ofr the Acoustic Modelling of Speech. *In IEEE on Automatic Speech Recognition and Understanding, Acoustic Modelling and Adaptation* 1, pp. 115-118.

[2] Antoniou, C.A. and Reynolds T.J. (2000) Modular Neural Networks Exploit Multiple Front-ends to Improve Speech Recognition Systems. *In International Conference in Knowledge Based Intelligent Engineering Systems and Allied Technologies, Neural Networks* 1, pp. 205-208.

[3] Antoniou, C.A. and Reynolds T.J. (2000) Acoustic Modelling using Modular/Ensemble Combinations of Heterogeneous Neural Networks. *In International Conference on Speech and Language Processing, Acoustic Modelling* 1.

[4] Auda, G. and Kamel, M. (1998) Modular Neural Network Classifiers: A Comparative Study. *In International Conference on Acoustics, Speech and Signal Processing*.

[5] Chengalvarayan, R. (1998) Speech Trajectory Discrimination Using the Minimum Classification Error Learning. *In IEEE Transactions on Speech and Audio Processing* 6(6), pp. 505-512.

[6] Fritsch, J. (1998) ACID/HNN - Clustering Hierarchies of Neural Networks for Context-Dependent Connectionist Acoustic Modelling. *In International Conference on Acoustics, Speech and Signal Processing*.

[7] Jordan, M.I. (1994) Hierarchical Mixtures of Experts and the EM Algorithm. *In Advances in Neural Computation* 6, pp. 181-214.

[8] Lee, K.F. and Hon, H.W. (1989) Speaker-independent Phoneme Recognition using Hidden Markov Models. *In IEEE on Acoustic, Speech and Signal Processing* 37(12), pp. 1641-1648.

[9] Ming, J. and Smith F.J. (1998) Improved Phone Recognition Using Bayesian Triphone Models. *In International Conference on Acoustics, Speech and Signal Processing*, pp. 409-412.

[10] Reynolds, T.J., Pizzolato, E.B. and Antoniou, C. (1998) Multinet: A New Connectionist Architecture for Speech Recognition. *In International Conference on Artificial Neural Networks*, pp. 257-262.

[11] Robinson, A.J. and Fallside, F. (1991) A Recurrent Error Propagation Network Speech Recognition System. *In Computer Speech and Language* 5(3), pp. 259-274.