INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

Connectionist Speech Recognition: Status and Prospects

Steve Renals, Nelson Morgan, Herve Bourlard*, Michael Cohen†, Horacio Franco†, Chuck Wooters and Phil Kohn

TR-91-070

December 1991

ABSTRACT

We report on recent advances in the ICSI connectionist speech recognition project. Highlights include:

- Experimental results showing that connectionist methods can improve the performance of a context independent maximum likelihood trained HMM system, resulting in a performance close to that achieved using state of the art context dependent HMM systems of much higher complexity.
- Mixing (context independent) connectionist probability estimates with maximum likelihood trained context dependent models to improve the performance of a state of the art system
- The development of a network decomposition method that allows connectionist modelling of context dependent phones efficiently and parsimoniously, with no statistical independence assumptions.

*. L&H Speechproducts, Ieper, B-8900 Belgium.
†. SRI International, Menlo Park CA 94025, USA.

Part I

INTRODUCTION

The dominant approach to automatic continuous speech recognition is statistical [5, 7]. The resulting methods, which use crude speech production models, hidden Markov models (HMMs), have been successful largely because of the existence of well-understood, consistent and provably convergent training procedures.

Since 1988, we have been investigating the use of feed-forward connectionist networks to improve HMM speech recognition systems. The basis of this approach has been a proof that feed-forward networks can be used as probability estimators [1]. We have been using such networks to estimate the output probabilities of HMMs [9, 10].

In this paper, we summarise the theory underlying our hybrid connectionist–HMM systems, and present the issues (both theoretical refinements and practical necessities) that we have had to address in order to use these methods successfully. Our recent experiments have used connectionist methods in both the baseline ICSI continuous speech recognition system and the more complex DECIPHER system developed at SRI International [3].

The connectionist systems described here estimate probabilities for context independent phone models. However, advanced continuous speech recognition systems utilise context dependent phone models [13, 7]: multiple models for each phone depending on the surrounding phonetic context. In part IV, we present a new network decomposition method that allows feed-forward networks to model phones in context efficiently and parsimoniously. We show this technique to be theoretically justified—no independence assumptions are necessary—and initial experiments have demonstrated that the decomposition method does indeed produce similar probability estimates to the much larger original networks.

Finally we outline some work in progress and future directions for this project.

Part II

BACKGROUND

HIDDEN MARKOV MODELS

Statistical speech recognition

In a statistical approach to speech recognition, the desired quantity is the posterior probability $P(\mathbf{W}_1^N|\mathbf{X}_1^T,\Theta)$ of a word sequence $\mathbf{W}_1^N = \mathbf{w}_1,...,\mathbf{w}_N$ given the acoustic evidence $\mathbf{X}_1^T = \mathbf{x}_1,...,\mathbf{x}_T$ and the parameters of the models used Θ. These models are generative models of particular units of speech (e.g. phones). This probability may be re-expressed using Bayes' rule:

$$P(\mathbf{W}_1^W|\mathbf{X}_1^T,\Theta) = \frac{P(\mathbf{X}_1^T|\mathbf{W}_1^W,\Theta)P(\mathbf{W}_1^W|\Theta)}{P(\mathbf{X}_1^T|\Theta)} = \frac{P(\mathbf{X}_1^T|\mathbf{W}_1^W,\Theta)P(\mathbf{W}_1^W|\Theta)}{\sum_{\mathbf{W}'} P(\mathbf{X}_1^T|\mathbf{W}',\Theta)P(\mathbf{W}'|\Theta)}. \quad (1)$$

$P(\mathbf{X}_1^T|\mathbf{W}_1^W,\Theta)/P(\mathbf{X}_1^T|\Theta)$ is referred to as the acoustic model. This is the likelihood of the acoustic evidence given the sequence of word models, divided by the probability of the acoustic data being generated by the models. This latter term ($P(\mathbf{X}_1^T|\Theta)$), may be regarded as a constant (over models) normalising term at recognition time. However at training time, the parameters Θ are being adapted, thus $P(\mathbf{X}_1^T|\Theta)$ is not constant. The prior, $P(\mathbf{W}_1^N|\Theta)$, is obtained from the language model.

The basic unit of speech is generally smaller than a word—here we will use phone models. Each phone is modelled by a HMM. Word models consist of concatenations of phone models (constrained by pronunciations from a lexicon), sentence models consists of concatenations of word models (constrained by a grammar). The lexicon and grammar together make up the language model, specifying prior probabilities for sentences, words and phones.

A HMM is a stochastic automaton defined by a set of states q_i, a topology specifying allowed state transitions, and a set of local output probability density functions (PDFs) $P(\mathbf{x}_t, q_i|q_j, \mathbf{X}_1^{t-1})$. Making the further assumptions that the output at time t is independent of previous outputs and depends only on the current state, we may separate the local probabilities into state transition probabilities $p(q_i|q_j)$ and output PDFs $P(\mathbf{x}_t|q_i)$. A set of initial state probabilities must also be specified.

The state of a HMM at any given time is not known, it is *hidden*. What is observed is the acoustic vector $\mathbf{x}_t$ generated at time t. Speech recognition is performed by

computing the most likely state sequence (given topological constraints) that could have generated the acoustic data. This may be achieved by a dynamic programming procedure, referred to as the Viterbi algorithm.

HMM training

The training problem for HMMs, consists of setting the parameters of the output PDFs, $P(\mathbf{x}|q_i)$, and the transition probabilities, $p(q_i|q_j)$. In practice the transition probabilities are less important (primarily encoding duration information) than the output probabilities (which are generated by a distribution in data space).

The parameters of a model are frequently estimated using a maximum likelihood training procedure known as the forward-backward algorithm. In this procedure the likelihood $P(\mathbf{x}|q_i)$ is locally maximised, with the normalising denominator $P(\mathbf{x})$ being ignored. This procedure is optimal if the several conditions are satisfied:

- The space of models being searched contains the true model;
- A "reasonable" choice of prior is made;
- The performance of the system improves as the parameters approach their optimal settings;
- There is enough data to determine the parameters well;
- An adequate training algorithm is used.

In the case of speech recognition, the first condition does not hold: we do not know the true model for speech, but we are sure that it is not a piecewise stationary HMM, with some arbitrary choice of density functions. In this case we should consider which decisions to make: or in decision theory terminology, what is the utility function to be optimised? Here, we do not wish to choose the "most accurate" model of each speech unit, chosen from the search space. Rather, we wish to make the best discrimination between classes at each time step, to ultimately choose the actual sentence that was spoken from the space of all possible utterances.

That maximising the likelihood $P(\mathbf{x}|q)$ is not necessarily the best policy may be seen by examining equation (1). The denominator is summed over all possible models. At recognition time the parameters of the models are changing: thus in addition to increasing the likelihood $P(\mathbf{x}|q_c)$ of the correct model producing the data, we must decrease the likelihoods of the data being generated by the incorrect models $P(\mathbf{x}|q_i),\ i \neq c$. So, any parameter update will have two terms, one in the direction of the maximum likelihood update and a second discriminative term.

FEED-FORWARD NETWORKS

Feed-forward networks may be used to estimate probabilities. Here, we shall regard networks that are trained to perform a '1-from-N' classification, in which there is a one-to-one correspondence between output units and classes. If we constrain the output units of such a network to be non-negative and to sum to one (e.g., by using some normalisation in the output units' transfer function), then we may regard the output units as representing a probability distribution. Note that we commonly use a sigmoid transfer function on the output units, which only constrains them to be non-negative. However Bourlard and Wellekens [1] demonstrated that at the minimum of a least squares or relative entropy objective function, the sigmoid output units of a '1-from-N' classification network will sum to 1. In practice, we find no significant difference between using a sigmoid transfer function or a normalised exponential ('softmax') transfer function on the output units of a MLP.

Furthermore it was shown in [1] that '1-from-N' networks (such as multi-layer perceptrons—MLPs) trained to minimise a relative entropy or least squares error function, in a '1-from-N' classification task will output estimates of the posterior probability $P(q_i|\mathbf{x})$ of class q_i (which corresponds to output unit i) given the input vector $\mathbf{x}$.

Note that:

$$P(q_i|\mathbf{x},\Theta) \propto \frac{P(\mathbf{x}|q_i,\Theta)}{P(\mathbf{x}|\Theta)}$$

Thus in maximising the posterior probability of the correct class, we simultaneously maximise the likelihood of the correct class and minimise the likelihoods of the incorrect classes. The proportionality constant is the prior $P(q_i|\Theta)$. In network training this is approximated by the relative frequencies of each class. However, when performing speech recognition we believe that we can set a better prior using a language model (grammar and lexicon), perhaps derived from a second, textual database.

Hence, we may use feed-forward networks as the acoustic model in speech recognition. It is important to remember that we are not getting something for nothing here. In estimating the posterior $P(q_i|\mathbf{x},\Theta)$ the networks do not estimate the input density $P(\mathbf{x}|\Theta)$ (which would be estimated by a maximum likelihood approach).

There are several reasons why this somewhat indirect method of estimating (scaled) likelihoods, is attractive for speech recognition:

1. MLPs are well matched to discriminative objective functions.
2. Although an MLP is a parametric model, a large network defines an extremely flexible set of functions. Thus only weak assumptions are made about

the input statistics. As a result of this they can combine multiple sources of evidence. For example a single MLP may be trained using input data that mixes samples drawn from several distributions, discrete or continuous.

3. Maximum likelihood estimation of HMM parameters requires the assumption of conditional independence of outputs. MLPs can model correlations across an input window of adjacent frames.
4. Since the recognition time computations are extremely regular, it is possible to have a simple, efficient implementation in parallel hardware.

Part III
CONTEXT INDEPENDENT NETWORKS

TRAINING AND RECOGNITION ISSUES

We have used the theoretical results described above to estimate observation likelihoods (output probabilities) for HMMs.

'Failure is an opportunity to learn.' In 1988 N. Morgan and H. Bourlard [unsurprisingly unpublished] first used these methods in the DECIPHER system. On the speaker independent Resource Management task, without a grammar, a word accuracy of -30% was recorded. Now, using essentially the same approach, we have improved our recognition accuracy to about 70% on the same task. What changes were necessary to make the connectionist approach effective?

1. (Scaled) likelihoods must be used in the Viterbi search, not posteriors. These may be most simply obtained by dividing each network output by the relative frequency of that class. Although the equations used in the Viterbi search hold for both posteriors and likelihoods, posteriors (which incorporate the prior probabilities estimated from the data) should not be used: when a language model and phone-structured lexicon are defined (i.e. the overall HMM topology), the priors for each class are implicitly set. Thus we must factor out the data estimates of these priors [11].
2. Cross-validation training is essential for good generalisation and preventing over-training, especially when using large networks. In our training schedule we cross-validate by withholding a certain proportion of the training data (typically 10–20%) and using this to validate the training after each epoch. When the classification performance on the validation set first fails to improve by a certain amount (typically 0.5%) the gradient descent step-size is reduced, typically by a factor of 2. This time-dependent reduction in stochastic gradient descent step-size (gain) may be understood in terms of the constraints on the gain sequence given by stochastic approximation theory [12][1]. After each succeeding epoch the step size is further reduced,

1. The conditions given by stochastic approximation theory ($\sum_n \alpha_n = \infty$ and $\sum_n \alpha_n^2 < \infty$) are not, in fact, met by our gain sequence, $\alpha_n \propto 1/2^n$, since $\sum_n 1/2^n < \infty$. A gain sequence such as $\alpha \propto 1/n$ would meet these constraints. This first constraint, that is violated by our gain sequence, may be regarded as ensuring that the gradient descent can in fact reach the minimum. Since we use a cross-validation training scheme, it may be that this condition is not necessary for us. Certainly a $1/2^n$ gain sequence results in faster training than a $1/n$ sequence.

until once again there is no improvement on the validation set. Training is then halted.

3. Input representation is important. In particular dynamic features (obtained via linear regression estimate of the temporal derivative) should be used in addition to static ones, and a multi-frame input (typically we use ± 4 frames of context), offers an improvement over single frame input [10].
4. The word transition penalty used in the Viterbi search should be increased in the case of multi-frame input. This empirical result may be explained in terms of a scaling relationship between the likelihood of a single frame and the joint likelihood of several frames, given the class.

To train large networks efficiently with a large and redundant training set, we have found a stochastic gradient descent procedure (i.e. backprop with per-pattern update) to be preferable. It appears that these stochastic methods are more efficient than batch methods, at least for our training problems.

THE DECIPHER SYSTEM

The systems into which we have previously integrated connectionist probability estimators were very simple: context independent phone models, single density models (with duration modelling) and single pronunciations of each vocabulary item. This paper continues this research by integrating such connectionist probability estimators into a large HMM continuous speech recognition system, SRI's DECIPHER [3]. DECIPHER is a much richer system than the previous baseline systems we have used. It includes multiple probabilistic word pronunciations, cross-word phonological and acoustic modelling, context dependent phone models, and models with multiple densities.

Word models are represented as probabilistic networks of phone models, specifying multiple pronunciations. These networks are generated by the application of phonological rules to baseform pronunciations for each word. In order to limit the number of parameters that must be estimated, phonological rules are chosen based on measures of coverage and overcoverage of a database of pronunciations. This results in networks which maximise the coverage of observed pronunciations while minimising network size. Probabilities of pronunciations are estimated by the forward-backward algorithm, after tying together instances of the same phonological process in different words. Phonological rules can be specified to apply across words, adding initial or final arcs which are constrained to connect only to arcs fulfilling the context of the rule [2, 3].

Context dependent phone models include word-specific phone, triphone, generalised triphone, cross-word triphone (constrained to connect to appropriate contexts), and left and right biphone (and generalised biphone). All these models are smoothed together, along with context independent models, using the deleted interpolation algorithm.

Most phone models have three states, each state having a self transition and a transition to the following state. A small number of phone models have two states, to allow for short realisations.

EXPERIMENTS

Experiments were performed on the speaker-independent DARPA Resource Management database. This database used a vocabulary of 998 words and no grammar (perplexity = 998) or a word pair grammar (perplexity = 60).

A 12th order mel cepstrum front end was used, producing 26 coefficients per frame: energy, 12 cepstral coefficients and derivatives of each static feature computed over a 4 frame window. The inputs to the MLP consisted of a frame in $\pm$ 4 frames of context, a feature vector length of 234. The MLPs that we used contained 512 hidden units (a number determined by empirical experiments, trading off representational power with computation) and 69 output units (corresponding to 69 monophone categories), giving a total of around 150,000 weights. Stochastic gradient descent training typically required about 10 passes through the training database of 1.3 million frames. This required less than 24 hours compute time, using a 5-board RAP (Ring Array Processor) [8], containing 20 TI TMS320C30 DSPs, each with 256kB of SRAM and 16MB of DRAM.

To train an MLP we require a bootstrap model to produce time-aligned phonetic labels. In this case we used the context independent DECIPHER system to perform the forced alignment between the training data and word sequence.

The baseline DECIPHER system modelled the output distributions using tied Gaussian mixtures. Training used the forward-backward algorithm to optimise a maximum likelihood criterion.

We used two sets of test sentences for evaluation. A 300 sentence development set (the June 1988 RM speaker independent test set) was used to tune the HMM recognition parameters, such as the word transition penalty. The results reported here were obtained from a 600 sentence test set (the February 1989 and October

1989 RM speaker independent test sets); no tuning of parameters was performed using this set.

Context Independent Models

We first experimented using context independent models. The baseline context independent DECIPHER system incorporated multiple pronunciations, cross-word phonological modelling, etc., but had only 69 two or three state phone models (200 distributions in all).

The baseline connectionist system had 69 single distribution phone models; the lexicon consisted of a single pronunciation for each word. Each phone model was a left-to-right model (with self-loops) with $N/2$ states, where N was the average duration of the phone. Transition probabilities were all tied to be 0.5. The connectionist probability estimator was integrated into DECIPHER in two ways:

- The usual {2,3}-state DECIPHER models were used, but each model had only a single output distribution (from the MLP). Thus the 2 or 3 states in a model shared a distribution.
- A new MLP was trained with 200 outputs, corresponding to the 200 states in the 69 context independent DECIPHER models.

The maximum likelihood transition probabilities (which basically encoded duration information) were retained.

Two heuristics were tried for combining the MLP and standard estimates of the state output probabilities. In the first weighted logs of the MLP and Gaussian mixture likelihood estimations were used:

$$\log(P(\mathbf{x}|q_j)) = \lambda_1 \log\left(\frac{P_{mlp}(q_j|\mathbf{x})}{P(q_j)}\right) + \lambda_2 \log(P_{gm}(\mathbf{x}|q_j)) \tag{2}$$

where P_{mlp} denotes the MLP estimate of a probability and P_{gm} the Gaussian mixture estimate. A single set of λs was used over all the states: they were optimised for minimum recognition error over the 300 sentence development set.

In the second heuristic, the log of a weighted average of the state output probabilities estimated by the MLP and the tied Gaussian mixtures was used:

$$\log(P(\mathbf{x}|q_j)) = \log\left(\lambda_1 \frac{P_{mlp}(q_j|\mathbf{x})P_{gm}(\mathbf{x})}{P(q_j)} + \lambda_2 P_{gm}(\mathbf{x}|q_j)\right). \tag{3}$$

In this approximation, the probability of the data $P(\mathbf{x})$ was required to ensure that the 2 likelihood estimates are scaled similarly. This cannot be obtained from the

	Parameters	% error Perplexity 998	60
Baseline MLP-69	155,717	36.1	12.8
CI-DECIPHER	125,762	44.7	14.0
MLP-69	155,717	30.1	7.8
MLP-200	222,920	34.9	11.4
MIX-69	281,548	29.4	7.5

Table 1: Results using 69 context independent phone models. The baseline MLP system uses 69 single distribution models with a single pronunciation for each word in the vocabulary. The DECIPHER system also uses 69 phone models, each with two or three states 200 independent distributions in total. The MLP-69 and MLP-200 systems use DECIPHER's multiple pronunciation and cross-word modelling. MLP-200 differs from the other MLP systems in that it has 200 outputs corresponding to DECIPHER's 200 states. The MIX-69 system is an a system interpolating the probabilities produced by the MLP and DECIPHER (rather than replacing the DECIPHER probabilities by MLP probabilities).

MLP, and was approximated by summing over the state conditional tied Gaussian likelihoods:

$$P_{gm}(\mathbf{x}) = \sum_i P_{gm}(\mathbf{x}|q_i)P(q_i). \tag{4}$$

The best results were obtained using (2), which resulted in an 8.0% error on the development set, compared with an error of 9.1% using (3). Thus (2) was used in evaluating over the 600 sentence test set.

Results for these context independent systems are shown in table 1. There are several notable aspects to these results:

- The MLP system using single pronunciations and single distribution phone models has a lower error rate than the context independent DECIPHER system, which uses multiple pronunciations and cross-word phonological modelling.
- Incorporating the MLP estimator into the context independent DECIPHER system results in still better performance, lowering the error rate substantially from 12.8% to 7.8%.
- The DECIPHER system uses multiple state, multiple distribution HMMs. Typically each phone model consists of 3 independent states. An MLP can

	Parameters	% error Perplexity 998	60
DECIPHER	5,541,844	21.9	4.9
MLP-DECIPHER	5,697,726	19.5	4.1

Table 2: Results using 3428 context dependent phone models. The hybrid system (MLP-DECIPHER) interpolates the MLP context independent probabilities with the DECIPHER context dependent probabilities.

be used to estimate these probabilities, simply by increasing the size of the output layer and using the maximum likelihood state segmentation as output targets. In maximum likelihood trained systems it is usually beneficial to move from single distribution to multi-distribution models, since acoustically different parts of a phone (e.g. onset, centre and offset of a vowel) may be modelled independently. However, such a change produced a performance degradation when using the MLP. We hypothesise that this was due to discriminative training. In many cases, different states of a phone are acoustically very similar. Thus, forcing an MLP to discriminate between such states could be counter-productive.

- Interpolation of MLP probabilities with context-independent DECIPHER maximum likelihood probabilities gives a small, but not significant (at the 0.95 level), improvement compared with MLP-69 system; the improvement compared with the CI-DECIPHER system is, of course, significant. The parameters for the interpolation were tuned on the development set. There was a more substantial improvement on the development set (around 2% error reduction), indicating that the interpolation parameters are being overfitted to the dataset used for tuning.

Context Dependent Models

Our experiments using context dependent models involved interpolating connectionist estimates of context independent output probabilities (as used above) together with the maximum likelihood tied mixture estimates of the context dependent probabilities used in DECIPHER. Equation (2) was used for these interpolations. Results for this hybrid system are shown in table 2. These results indicate that the MLP context independent probabilities may be used to afford a small improvement in recognition, which is significant at the 0.95 level for both the word-pair grammar case and the no grammar case.

Part IV
CONTEXT DEPENDENT NETWORKS

CONTEXT DEPENDENT MODELLING

In the experiments reported above, connectionist methods were used only to estimate context independent probabilities of the form $P(q_i|\mathbf{x})$. Such probabilities are considered "context independent" since the left hand side of the posterior, contains no term involving neighbouring phones[2]. However, state of the art recognisers, such as DECIPHER, are of greater complexity: in particular they use context dependent phone models. Will the consistent improvements offered by our connectionist methods be washed out in systems with more detailed models?

One difficulty with more complex models is that many more parameters must be estimated with the same limited amount of data. Brute-force application of our earlier techniques would result in an output layer with many thousands of units, and a network with many millions of connections. This network would be impractical to train, both in terms of computation and learnability, using current-sized public data bases. In each of our earlier studies, a simple context independent trained network used a single output unit for each phone. For our most recent Resource Management tests, we use 69 of these units. Were one to consider the coarticulatory effects from the right only, this number would expand out to 69^2, or over 4000. Considering both right and left context, we would require 69^3 units, or about 328,000. With a typical hidden layer of 500 units, we would have over 10^8 connections, which is far too many for a practical system.

Of course, HMM researchers have had a similar consideration in reducing the number of parameters in their VQ or tied-mixture based systems. The solution has been, in one form or another, to use a reduced number of context dependent models (typically a few thousand). However, this is still a large number. For instance, with 4000 outputs and 500 hidden units, a network would still have over 2 million connections, which makes good generalisation difficult for training sets of a few hundred thousand frames. Even if enough training data were available, networks with millions of parameters can be expected to take impractical amounts of time to train using back-propagation approaches, even with fast special-purpose machines such as our Ring Array Processor (RAP) [8].

Here, we present a method for estimating likelihoods for context dependent phone

2. Note, that these probabilities do include an acoustic context, since we typically use a 9 frame window. However this acoustic context is usually on a different timescale to phonetic context.

models, using networks that are not substantially bigger than our context independent networks, and that require only a small increase in computation. This method is a general technique for decomposing a larger classification network into several smaller networks without any simplifying assumptions, such as independence of outputs.

CDNN: A CONTEXT DEPENDENT NEURAL NETWORK

Consider a system with K phone classes. A context independent network trained to estimate class posterior probabilities would then have K outputs. For a context dependent model, we may wish to estimate the joint probability of a current HMM state with a particular neighbouring phonetic category. Using $\mathcal{C}$ to represent the set of possible contexts, we wish to estimate $p(q_k, c_j|\mathbf{x})$, where $c_j \in \mathcal{C} = \{c_1, \ldots, c_L\}$. If there are L context classes, this will require $K \times L$ output units for an MLP estimator. However, if we use the conditional probability axiom, the desired expression can be broken down as follows:

$$p(q_k, c_j|\mathbf{x}) = p(q_k|\mathbf{x}) \times p(c_j|q_k, \mathbf{x}) \tag{5}$$

Thus, the desired probability is the product of the monophone posterior probability and a new conditional. The former can be realised with the usual monophone network. Viewing an MLP as an estimator of the left side of a conditional given the right side as input, the second term can be estimated by an MLP trained to generate the correct context class given inputs of the current class and the speech input frame. The latter network only has as many outputs as there are context classes.

This procedure reduces the training of a single network with $K \times C$ outputs to the training of two smaller networks with K and C outputs respectively, and represents a generic way of splitting large MLPs used in classification mode into several smaller ones. It has the potential, however, of requiring much greater computation during the recognition phase. Indeed, if one implements this method naively, the second network must be computed K times for each frame during recognition, since the output probabilities depend on an assumption of the current class (corresponding to a monophone model in a hypothesised word sequence at that point in the dynamic programming). The next section will describe how this expense can largely be circumvented.

As a particular application of this general rule, the problem of hybrid HMM/MLP approaches for triphone modelling will be considered. In this case, the main prob-

lem lies in the estimation of probabilities like $p(x_t|q_k, c_j^\ell, c_\ell^r)$ where c_j^ℓ and c_ℓ^r respectively represent the left and right phonemic contexts of state q_k. If one wants to model triphones with neural networks, a straightforward approach could consist in having $K \times L \times L$ output units to model the $K \times L \times L$ possible contextual state probabilities. This would require an excessive number of parameters.

Expanding the joint triphone likelihood as in (5), we get:

$$p(q_k, c_j^\ell, c_\ell^r|x_t) = p(c_j^\ell|q_k, c_\ell^r, x_t).p(c_\ell^r|q_k, x_t).p(q_k|x_t) , \tag{6}$$

and

$$p(q_k, c_j^\ell, c_\ell^r) = p(c_j^\ell|q_k, c_\ell^r).p(c_\ell^r|q_k).p(q_k) . \tag{7}$$

During recognition and training, the context dependent likelihoods $p(x_t|c_j^\ell, q_k, c_\ell^r)$ can then be estimated from (6) and (7) as:

$$p(x_t|c_j^\ell, q_k, c_\ell^r) = \frac{p(c_j^\ell, q_k, c_\ell^r|x_t).p(x_t)}{p(c_j^\ell, q_k, c_\ell^r)}, \tag{8}$$

in which $p(x)$ can be ignored during dynamic time warping.

In fact, relations (5), (6) and (7) are examples of a general approach for splitting a huge MLP used in classification mode into smaller ones without requiring any simplifying assumptions. Now, exploiting the conclusions we derived from the theory of our hybrid HMM/MLP approach for phoneme models (i.e., in classification mode, the output values of the MLP are estimates of the a posteriori probabilities of the output classes conditioned on the input), it can be shown that all the right hand factors appearing in (6) and (7) can be estimated separately by different MLPs:

- $p(c_j^\ell|q_k, c_\ell^r, x_t)$ can be estimated by an MLP in which the output units are associated with the left phonemes of the triphones and in which the input field consists of the current acoustic vector x_t (possibly extended to its left and right contexts), the current state and the right phonetic contexts in the triphones (which are known during training).
- $p(c_\ell^r|q_k, x_t)$ can be estimated by a neural network in which the output units are associated with the right phonemes and in which the input field is constituted by the current acoustic vector x_t (possibly extended to its left and right contexts) and the current state (associated with x_t).
- $p(q_k|x_t)$ is estimated by the same neural network as the one used for modeling phonemes where the input field contains the current acoustic vector only and the output units are associated with the current labels.

- $p(c_j^\ell|q_k, c_\ell^r)$ can be estimated by a neural network in which the output units are associated with the left phonemes of the triphones and where the input field represents the current state and the right phonemes. This provides us with the a priori probability of observing a particular phoneme in the left part of a triphone given particular current state and right phonetic context.
- $p(c_\ell^r|q_k)$ can be estimated by a neural network in which the output units are associated with the right phonemes of the triphones and where the input field represents the current state. This provides us with the a priori probability of observing a particular phoneme on the right side of a particular state. Given the limited number of parameters in this model (i.e., $K \times L$), this probability can also be estimated by counting (i.e., this does not require a network).
- $p(q_k)$ is the a priori probability of a phoneme as also used in the standard hybrid HMM/MLP phonetic approach, and is simply estimated by counting on the training set (i.e., this also does not require a network).

By training these different MLPs and using their output activation values in (8), it is possible to estimate, without any particular assumptions, the probability $p(x_t|q_k, c_j^\ell, c_\ell^r)$ that is required for modelling triphone probabilities to be used in HMMs. This generalises to triphones the approach described above and which was restricted to phoneme models. Of course, for limited training sets, as done with standard HMMs, smoothing of context dependent and context independent probabilities still may be important even with MLPs. Also, training improvements detailed above (such as the use of cross-validation technique to improve generalisation performance) remain valid in this new approach. Additionally, if c^ℓ and c^r represent broad phonetic classes or clusters rather than phonemes, the above results apply to the estimation of "generalised triphones", such as are defined in [7]. As done previously the input field containing the acoustic data (e.g., $\mathbf{x}$) may also be supplied with contextual information. In this case, the $\mathbf{x}$ appearing in all the above (and subsequent) probabilities have to be replaced by $X_{t-c}^{t+c} = \{\mathbf{x}_{t-c}, ..., \mathbf{x}_t, ..., \mathbf{x}_{t+c}\}$, in which c represents the width of the contextual window. This leads to the estimation of triphone probabilities given acoustic contextual information, which is even more important in the case of triphone models.

IMPLEMENTATION ISSUE

While the previous section appears to have solved the problem of triphone modelling by neural network, an important implementation issue still has to be taken into account.

We have shown theoretically how to transform, without any particular assumptions or simplifications, the huge neural network which would result from the brute force application of our earlier hybrid HMM/MLP approach for phoneme modelling to triphones. Indeed, instead of having a single MLP estimating $p(q_k|x_t)$, we need to estimate

$$p(q_k, c_j^\ell, c_\ell^r|x_t) \ , \tag{9}$$

which requires an MLP with $K \times L \times L$ output units. In the previous Section, we have shown that it was possible to estimate the same probability with three smaller MLPs respectively estimating

$$p(c_j^\ell|q_k, c_\ell^r, x_t) \ , \tag{10}$$

$$p(c_\ell^r|q_k, x_t) \tag{11}$$

and

$$p(q_k|x_t) \ . \tag{12}$$

However, while this strongly reduces the memory requirement and the number of parameters, a naive implementation of these smaller networks would require much more computation.

In the case of phonetic modelling, a single MLP provided with the current acoustic vector x_t as input can estimate $p(q_k|x_t)$ for all possible classes q_k on the associated output units. This remains valid for triphone modelling if we use the huge network with x_t at its input and $K \times L \times L$ output units, each output unit being associated with a particular triphone. However, when this huge network is decomposed into smaller networks computing (10), (11) and (12), the first two networks must have input values depending on the phonetic contexts constituting the triphones. For example, the input field of the neural network estimating (10) on its output units is constituted by the concatenation of the current acoustic vector x_t and the middle and right phonetic contexts in the triphones. Since the MLP training is supervised, i.e. we know exactly which triphone is associated with a particular acoustic vector, this is not a problem during training. However, this is no longer the case during recognition where we do not know in advance which triphone is associated with x_t.

Therefore, in principle one would have to compute network activations at each frame for each possible phonetic context. This would amount to $L \times L$ times the monophone network computation, and would generally be prohibitive. Fortunately,

a simple restriction on the network topology permits the pre-calculation of contextual phonetic contributions to the output; this computation can be done at the end of the training phase, prior to the recognition of any speech. By simply partitioning the net so that no hidden unit receives input from both phonetic context units and data input units, we can pre-calculate the contribution to the output units (prior to the output nonlinearity) for all possible combinations of left and right contexts, and form a table of these contributions. During recognition, the pre-sigmoid output values resulting from data vectors can be computed by a forward pass on the net for each frame. For each hypothetical triphone model, these contributions from the data inputs can be added to the corresponding context contributions from the table. The major new computation (in comparison with the monophone case) then is simply the cost of some lookups, both for the contextual contributions, and for the final sigmoidal nonlinearity, which must now be re-computed for each hypothesised triphone (as opposed to once per frame, as in the monophone case). In practice this only doubles or triples the computation time, a reasonable cost for triphone models.

As an example, let us consider the case of $p(c_j^\ell|q_k, c_\ell^r, x_t)$. More formally, letting $Y_j(q_k, c_\ell^r)$ be the contribution to the pre-sigmoid output for state q_j for the phonetic context dependent partition of the net, and letting $Z_j(x_t)$ be the contribution to the pre-sigmoid output for state q_j for the data vector input. Then

$$p(c_j^\ell|q_k, c_\ell^r, x_t) = f(Y_j + Z_j) \ , \tag{13}$$

where f is the standard sigmoid function. A $(K \times L \times L)$-dimensional table Y is computed after network training by running the phonetic-context dependent partition of the network (which has no inputs from the data vector) $K \times L \times L$ times, i.e. for all possible output units and for all possible combinations of phonetic contexts, with no output sigmoid computation. This table loading is a negligible amount of computation compared to the training of the network. During recognition, for each acoustic vector x_t, it is then enough to run each MLP only once to get the contribution Z_j of the data inputs for each output unit q_j. For each hypothetical triphone model, this contribution Z_j just has to be added to the corresponding context contribution Y_j obtained by a simple lookup in table Y. In fact, this is equivalent to considering Y_j as an added bias (of output unit q_j) that depends on the phonetic context. Of course, the same method can be applied to the MLP computing $p(c_\ell^r|q_k, x_t)$. Also, for $p(c_j^\ell|q_k, c_\ell^r)$ and $p(c_\ell^r|q_k)$ it is sufficient to compute look-up tables at the end of the training phase for use in (7).

DISCUSSION AND RESULTS

The unrestricted split net

In equation (5), when splitting the original MLP with $K \times L$ output units into two smaller networks with K and L outputs respectively, the number of parameters is drastically reduced, which could affect the quality of the conditional distributions' estimation. However, parameter reduction is exactly the aim of the proposed approach, both to reduce computation and to improve generalisation. As it was done for $p(q_k|\mathbf{x})$ above it will be necessary to find (e.g. by using cross-validation techniques) the number of hidden units (and hence the number of parameters) leading to the best estimate of $p(c_j|q_k, \mathbf{x})$. The desired probabilities can in principle be estimated without any statistical assumptions (e.g., independence). Of course, this is only guaranteed if the training does not get stuck in a local minimum and if there are enough parameters.

The topologically restricted net

As shown above, while reducing the number of parameters, the splitting of the network into two smaller networks results in much greater computation in the contextual network. To avoid this problem it is proposed to restrict the topology of the second network so that no hidden unit shares input from both q_k and $\mathbf{x}$. Consequently, the q_k input only changes the output thresholds. However, a recent experiment with frame classification for continuous speech (trained using 160,000 patterns from 500 sentences uttered by a speaker in the Resource Management continuous speech recognition corpus) suggested that this did not affect the correct estimation of $p(c_j|q_k, \mathbf{x})$. In this example, the network with a split hidden layer predicted (for a test set of 32,000 patterns from 100 sentences) the correct right context 63.6% of the time, while a network with a unified hidden layer predicted the context 63.5% of the time, an equivalent figure.

Preliminary results and conclusion

Prior to experimenting with the CDNN for continuous speech recognition using biphone and triphone models (to be reported at a later date), we wanted to check experimentally that the split MLP was equivalent to the original one. We compared biphone probabilities generated by the original and split MLP for the speaker independent Resource Management database. The number of hidden units in each MLP was chosen such that the number of parameters was approximately the same in both cases. After having trained both cases on 4,000 sentences, biphone probabilities were computed on a test set of 100 sentences pronounced by 4 different

speakers, yielding a total of 17,012,088 probabilities. To compare both sets of probabilities we computed the correlation coefficient to be 0.65, and the mean absolute difference that was equal to 0.0017. Thus, the two sets of probabilities are significantly correlated. This suggests that CDNN may be a good way to compute context dependent probabilities with nets that have a limited number of parameters and that require an acceptably small increase in computation over the context independent case.

Part V

PROSPECTS AND CONCLUSIONS

WORK IN PROGRESS

We are currently working on several areas of speech recognition, including improved front-end signal processing, acoustic modelling and language modelling. A particular focus is to integrate our various methods into working, near-real-time systems.

The RASTA-PLP (Relative Spectral–Peceptual Linear Prediction) method [4], developed by Hermansky et al. at ICSI, is an analysis technique designed to be robust to steady-state or slowly varying factors in speech, which are assumed to carry little linguistic information. The essential idea, is that part of an analysis should include a bandpass filtering in log spectral domain. Application of the RASTA-PLP method in online recognition experiments at ICSI have shown it to be robust to changes in microphone, compared with other analysis methods, such as mel-cepstrum and PLP. This, and other experiments, seem to indicate that RASTA is an extremely effective approach to adopt for speech recognition in "real world" situations.

Much of our current effort is in the area of improved acoustic modelling. Konig et al. [6] have added gender models to the acoustic modelling component. A gender-dependent network was trained to estimate the probability of a specker being male or female. This network was then used with two context independent networks (one trained on male data, the other on female) to estimate two sets of likelihoods $P(\mathbf{x}|q_i, male)$ and $P(\mathbf{x}|q_i, female)$, with the maximum probability Viterbi path (male or female) being chosen for a new speaker. This gave a statistically significant improvement from 17.2% error to 15.5% error on our baseline system.

Renals et al. [11] showed how a tied-mixture density estimator may be discriminatively trained using connectionist methods. Work is in progress to determine if this method can improve our current acoustic models, and if it can further improve the performance of an already trained tied mixture density estimator.

In October 1991, we constructed an online demonstration system, using the Resource Management task. This system uses the baseline ICSI system (single pronunciation words, etc.) with a RASTA-PLP front end. It has proven surprisingly robust to microphone variation. Work is in progress on a new demonstration system, Y_0 ("Y-naught") which is based on a Berkeley restaurant guide. This task

has a more substantial language modelling component: in addition to correctly transcribing a string of words, a correct database query must also be generated; a dialog must be maintained with the user; and out of vocabulary words must be handled appropriately. Additional features will include dealing with disfluencies and recognising non-native English speakers.

Our current recogniser is entirely implemented on a SPARC-2: data collection is performed on the RAP (and, of course, the network training). This system runs at about 35 $\times$ real-time. We aim to achieve real-time performance by running the network recognition computations on the RAP, using pruning in the Viterbi search, and possibly exploiting the parallelism of the RAP at recognition time by running the dynamic programming in parallel (although this latter step may prove unnecessary with the next generation of faster workstations).

CONCLUSIONS

Two substantial advances in connectionist methods for continuous speech recognition have been reported:

- We have demonstrated that connectionist probability estimation may be used to improve the performance of a state of the art continuous speech recognition system. In particular:
  1. Comparing like with like, a discriminatively trained connectionist context independent system performs considerably better than the corresponding maximum likelihood tied mixture system.
  2. The context independent MLP-DECIPHER system has an error of 7.8% compared with a 4.9% error produced by context dependent DECIPHER. However the latter system has 50 times the number of models and 35 times the number of parameters compared with the MLP system.
  3. Interpolating MLP context independent probabilities into the context dependent DECIPHER system produces a significant increase in word accuracy.
- We have presented a general method for factoring a multi-layered classification network with a large output layer into a number of smaller networks, with no statistical independence assumptions. We have demonstrated that this technique may be used estimate likelihoods for context dependent phone models.

ACKNOWLEDGEMENTS

This work was partially funded by DARPA contract MDA904-90-C-5253 via a subcontract from SRI International.

REFERENCES

[1] H. Bourlard and C. J. Wellekens. Links between Markov models and multilayer perceptrons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12:1167–1178, 1990.

[2] M. Cohen. *Phonological Structures for Speech Recognition.* PhD thesis, University of California at Berkeley, 1989.

[3] M. Cohen, H. Murveit, J. Bernstein, P. Price, and M. Weintraub. The DECIPHER speech recognition system. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 77–80, Albuquerque, 1990.

[4] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. RASTA-PLP speech analysis. Technical Report TR-91-069, International Computer Science Institute, Berkeley CA, 1991.

[5] F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64:532–556, 1976.

[6] Y. Konig, N. Morgan, and C. Chandra. GDNN: A gender dependent neural network for continuous speech recognition. Technical Report TR-91-071, International Computer Science Institute, Berkeley CA, 1991.

[7] K.-F. Lee. *Large Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System.* PhD thesis, School of Computer Science, Carnegie Mellon University, 1988.

[8] N. Morgan, J. Beck, P. Kohn, J. Bilmes, E. Allman, and J. Beer. The Ring Array Processor (RAP): A multiprocessing peripheral for connectionist applications. *Journal of Parallel and Distributed Computing*, page In Press, 1992.

[9] N. Morgan and H. Bourlard. Continuous speech recognition using multi-layer perceptrons with hidden markov models. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 413–416, Albuquerque, 1990.

[10] N. Morgan, H. Hermansky, H. Bourlard, C. Wooters, and P. Kohn. Continuous speech recognition using PLP analysis with multi-layer perceptrons. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 49–52, Toronto, 1991.

[11] S. Renals, N. Morgan, and H. Bourlard. Probability estimation by feed-forward networks in continuous speech recognition. Technical Report TR-91-030, International Computer Science Institute, Berkeley CA, USA, 1991.

[12] H. Robbins and S. Munro. A stochastic approximation method. *Annals of Mathematical Statisitics*, 29:400–407, 1951.

[13] R. Schwartz, Y. Chow, O. Kimball, S. Roucous, M. Krasner, and J. Makhoul. Context-dependent modelling for acoustic-phonetic recognition of continuous speech. In *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1205–1208, Tampa FL, 1985.