7

THE USE OF RECURRENT NEURAL NETWORKS IN CONTINUOUS SPEECH RECOGNITION

Tony Robinson, Mike Hochberg[1]
and Steve Renals[2]

Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, U.K.

ABSTRACT

This chapter describes a use of recurrent neural networks (i.e., feedback is incorporated in the computation) as an acoustic model for continuous speech recognition. The form of the recurrent neural network is described along with an appropriate parameter estimation procedure. For each frame of acoustic data, the recurrent network generates an estimate of the posterior probability of of the possible phones given the observed acoustic signal. The posteriors are then converted into scaled likelihoods and used as the observation probabilities within a conventional decoding paradigm (e.g., Viterbi decoding). The advantages of using recurrent networks are that they require a small number of parameters and provide a fast decoding capability (relative to conventional, large-vocabulary, HMM systems)[3].

1 INTRODUCTION

Most – if not all – automatic speech recognition systems explicitly or implicitly compute a *score* (equivalently, *distance, probability,* etc.) indicating how well an input acoustic signal matches a speech model of the hypothesised utterance. A fundamental problem in speech recognition is how this score may be computed,

[1]Mike Hochberg is now at Nuance Communications, 333 Ravenswood Avenue, Building 110, Menlo Park, CA 94025, USA.
[2]Steve Renals is now at the Department of Computer Science, University of Sheffield, Sheffield S1 4DP, UK.
[3]This chapter was written in 1994. Further advances have been made such as: context-dependent phone modelling; forward-backward training and adaptation using linear input transformations.

given that speech is a non-stationary stochastic process. In the interest of reducing the computational complexity, the standard approach used in the most prevalent systems (e.g., dynamic time warping (DTW) [1] and hidden Markov models (HMMs) [2]) factors the hypothesis score into a local acoustic score and a local transition score. In the HMM framework, the observation term models the local (in time) acoustic signal as a stationary process, while the transition probabilities are used to account for the time-varying nature of speech.

This chapter presents an extension to the standard HMM framework which addresses the issue of the observation probability computation. Specifically, an artificial recurrent neural network (RNN) is used to compute the observation probabilities within the HMM framework. This provides two enhancements to standard HMMs: (1) the observation model is no longer local, and (2) the RNN architecture provides a nonparametric model of the acoustic signal. The result is a speech recognition system able to model long-term acoustic context without strong assumptions on the distribution of the observations. One such system has been successfully applied to a 20,000 word, speaker-independent, continuous speech recognition task and is described in this chapter.

2 THE HYBRID RNN/HMM APPROACH

2.1 The HMM Framework

The HMM framework has been well documented in the speech recognition literature (e.g., [2]). The framework is revisited here in the interest of making this chapter relatively self-contained and to introduce some notation. The standard statistical recognition criterion is given by

$$\mathcal{W}^* = \underset{\mathcal{W}}{\operatorname{argmax}} \Pr(\mathcal{W}|\mathcal{U}) = \underset{\mathcal{W}}{\operatorname{argmax}}\, p(\mathcal{U}|\mathcal{W}) \Pr(\mathcal{W}) \tag{7.1}$$

where $\mathcal{W}^*$ is the recognised word string, $\mathcal{W}$ is a valid word sequence, and $\mathcal{U}$ is the observed acoustic signal (typically a sequence of feature vectors $\mathbf{u}$)[4]. For typical HMM systems, there exists a mapping between a state sequence $Q = \{q_1, q_2, \ldots, q_T\}$ on a discrete, first-order Markov chain and the word sequence $\mathcal{W}$. This allows expressing the recognition criterion (7.1) as finding the

[4]Throughout this chapter, the terms Pr and p indicate the probability mass and the probability density function, respectively.

maximum a posteriori (MAP) state sequence of length T, i.e.,

$$Q^* = \underset{Q}{\text{argmax}} \prod_{t=1}^{T} \Pr(q_t|q_{t-1}) p(\mathbf{u}_t|q_t). \tag{7.2}$$

Note that the HMM framework has reduced the primary modelling requirement to stationary, local (in time) components; namely the observation terms $p(\mathbf{u}_t|q_t)$ and transition terms $\Pr(q_t|q_{t-1})$. There are a number of well known methods for modelling the observation terms. Continuous density HMMs typically use Gaussian mixture distributions of the form

$$p(\mathbf{u}|q) = \sum_m c_{qm} \mathcal{N}(u; \mu_{qm}, \Sigma_{qm}). \tag{7.3}$$

Recently, there has been work in the area of hybrid connectionist/HMM systems. In this approach, nonparametric distributions represented with neural networks have been used as models for the observation terms [3, 4].

2.2 Context Modelling

Context is very important in speech recognition at multiple levels. On a short time scale such as the average length of a phone, limitations on the rate of change of the vocal tract cause a blurring of acoustic features which is known as co-articulation. Achieving the highest possible levels of speech recognition performance means making efficient use of all the contextual information.

Current HMM technology primarily approaches the problem from a top-down perspective by modelling phonetic context. The short-term contextual influence of co-articulation is handled by creating a model for all sufficiently distinct phonetic contexts. This entails a trade off between creating enough models for adequate coverage and maintaining enough training examples per context so that the parameters for each model may be well estimated. Clustering and smoothing techniques can enable a reasonable compromise to be made at the expense of model accuracy and storage requirements (e.g., [5, 6]).

Acoustic context in HMMs is typically handled by increasing the dimensionality of the observation vector to include some parameterisation of the neighbouring acoustic vectors. The simplest way to accomplish this is to replace the single frame of parameterised speech by a vector containing several adjacent frames along with the original central frame. Alternatively, each frame can be augmented with estimates of the temporal derivatives of the parameters [7].

However, this dimensionality expansion quickly results in difficulty in obtaining good models of the data. Multi-layer perceptrons (MLPs) have been suggested as an approach to model high-order correlations of such high-dimensional acoustic vectors. When trained as classifiers, MLPs approximate the posterior probability of class occupancy [8, 9, 10, 11, 12]. For a full discussion of this result to speech recognition, see [13, 4].

2.3 Recurrent Networks for Phone Probability Estimation

Including feedback into the MLP structure gives a method of efficiently incorporating context in much the same way as an infinite impulse response filter can be more efficient than a finite impulse response filter in terms of storage and computational requirements. Duplication of resources is avoided by processing one frame of speech at a time in the context of an internal state as opposed to applying nearly the same operation to each frame in a larger window. Feedback also gives a longer context window, so it is possible that uncertain evidence can be accumulated over many time frames in order to build up an accurate representation of the long term contextual variables.

There are a number of possible methods for incorporating feedback into a speech recognition system. One approach is to consider the forward equations of a standard HMM as recurrent network-like computation. The HMM can then be trained using the maximum likelihood criterion [14] or other discriminative training criteria [15, 16, 17]. Another approach is to use a recurrent network only for estimation of the emission probabilities in an HMM framework. This is similar to the hybrid connectionist-HMM approach described in [3] and is the approach used in the system described in this chapter.

The form of the recurrent network used here was first described in [18]. The paper took the basic equations for a linear dynamical system and replaced the linear matrix operators with non-linear feedforward networks. After merging computations, the resulting structure is illustrated in figure 1. The current input, $\mathbf{u}(t)$, is presented to the network along with the current state, $\mathbf{x}(t)$. These two vectors are passed through a standard feed-forward network to give the output vector, $\mathbf{y}(t)$ and the next state vector, $\mathbf{x}(t+1)$. Defining the combined input vector as $\mathbf{z}(t)$ and the weight matrices to the outputs and the next state

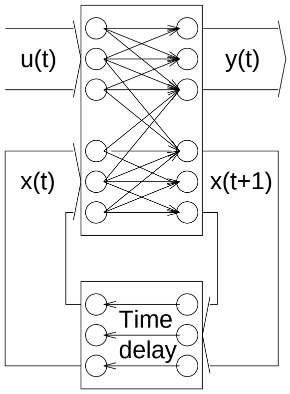

Figure 1 The recurrent network used for phone probability estimation.

as $\mathbf{W}$ and $\mathbf{V}$, respectively:

$$\mathbf{z}(t) = \begin{bmatrix} 1 \\ \mathbf{u}(t) \\ \mathbf{x}(t) \end{bmatrix} \tag{7.4}$$

$$y_i(t) = \frac{\exp(\mathbf{W}_i \mathbf{z}(t))}{\sum_j \exp(\mathbf{W}_j \mathbf{z}(t))} \tag{7.5}$$

$$x_i(t+1) = \frac{1}{1+\exp(-\mathbf{V}_i \mathbf{z}(t))} \tag{7.6}$$

The inclusion of "1" in $\mathbf{z}(t)$ provides the mechanism to apply a bias to the non-linearities. As is easily seen in (7.4)–(7.6), the complete system is no more than a large matrix multiplication followed by a non-linear function.

A very important point to note about this structure is that if the parameters are estimated using certain training criteria (see section 4), the network outputs are consistent estimators of class posterior probabilities. Specifically, the outputs $y_i(t)$ are interpreted as

$$y_i(t) = \Pr(q_t = i | u_1, \ldots, u_t, \mathbf{x}(0)). \tag{7.7}$$

The softmax non-linear function of (7.5) is an appropriate non-linearity for estimating posterior probabilities as it ensures that the values are non-negative and sum to one. Work on generalised linear models [19] also provides theoretical justification for interpreting $y_i(t)$ as probabilities. Similarly, the sigmoidal non-linearity of (7.6) is the softmax non-linearity for the two class case and is appropriate if all state units are taken as probability estimators of hidden independent events.

In the hybrid approach, $y_i(t)$ is used as the observation probability within the HMM framework. It is easily seen from (7.7) that the observation probability is extended over a much greater context then is indicated by local models as shown in (7.3). The recurrent network uses the internal state vector to build a representation of past acoustic context. In this fashion, the states of the recurrent network also model dynamic information. Various techniques used in non-linear dynamics may be used to describe and analyse the dynamical behaviour of the recurrent net. For example, different realisations of the network show a variety of behaviours (e.g., limit cycles, stable equilibriums, chaos) for zero input operation of the network (i.e., $\mathbf{u}(t) = \mathbf{0}$). For example, limit cycle dynamics for a recurrent network are shown in figure 2. The figure shows the projection onto two states of the network state vector over seven periods.

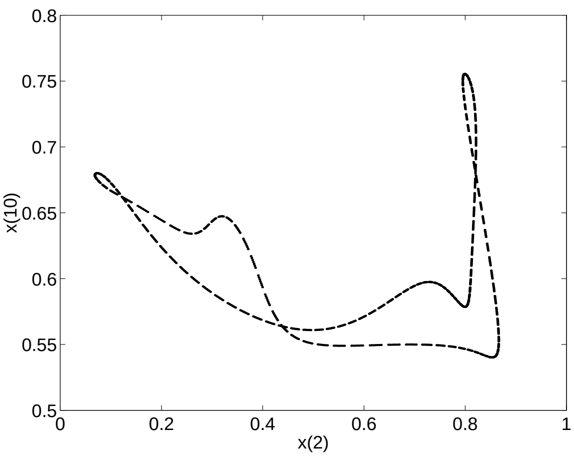

Figure 2 Projection of recurrent network state space trajectory onto two states.

3 SYSTEM DESCRIPTION

The basic hybrid RNN/HMM system is shown in figure 3. Common to most

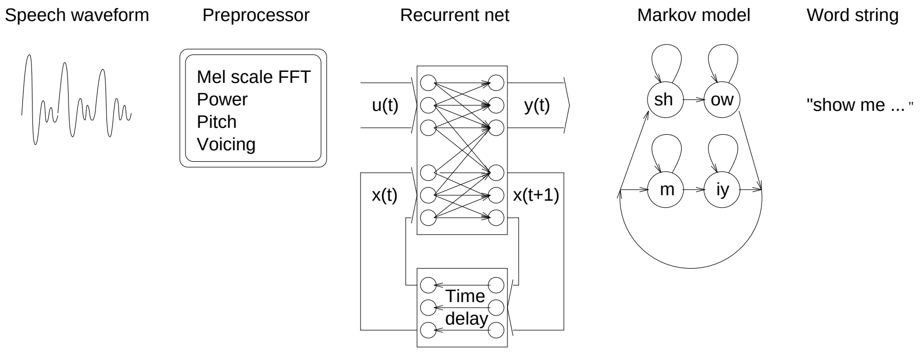

Figure 3 Overview of the hybrid RNN/HMM system.

recognition systems, speech is represented at the waveform, acoustic feature, phone probability and word string levels. A preprocessor extracts acoustic vectors from the waveform which are then passed to a recurrent network which estimates which phones are likely to be present. This sequence of phone observations is then parsed by a conventional hidden Markov model to give the most probable word string that was spoken. The rest of this section will discuss these components in more detail.

3.1 The Acoustic Vector Level

Mapping the waveform to an acoustic vector is necessary in speech recognition systems to reduce the dimensionality of the speech and so make the modelling task tractable. The choice of acoustic vector representation is guided by the form of the acoustic model which will be required to fit this data. For example, the common use of diagonal covariance Gaussian models in HMM systems requires an acoustic vector that has independent elements. However, the connectionist system presented here does not require that the inputs be orthogonal, and hence a wider choice is available. The system has two standard acoustic vector representations, both of which give approximately the same performance: `MEL+`, a twenty channel power normalised mel-scaled filterbank representation augmented with power, pitch and degree of voicing, and PLP, twelfth order perceptual linear prediction cepstral coefficients plus energy.

Another feature used for describing the acoustic processing is the ordering of the feature vectors. In systems which use non-recurrent observation modelling, this property is ignored. With a recurrent network, the vector ordering – or equivalently, the direction of time – makes a difference in the probability estimation process. In the experiments described later in this chapter, results are reported for systems using both forward and backward (in-time) trained recurrent networks.

3.2 The Phone Probability Level

Figure 4 shows the input and output representation of the recurrent network for a sentence from the TIMIT database. The top part of the diagram shows the MEL+ acoustic features. The top twenty channels represent the power at mel-scale frequencies up to 8 kHz. The bottom three channels represent the power, pitch and degree of voicing. Some features, like the high frequency fricative energy in /s/ and /sh/ and the formant transitions are clearly visible. The lower part of the diagram shows the output from the recurrent network. Each phone has one horizontal line with the width representing the posterior probability of the phone given the model and acoustic evidence. The vowels are placed at the bottom of the diagram and the fricatives at the top. As the TIMIT database is hand aligned, the dotted vertical lines show the boundaries of the known symbols. The identity of these hand aligned transcriptions is given on the top and bottom line of the diagram. Further information concerning this representation can be obtained from [20].

The complete sentence is "She had your dark suit in greasy wash water all year". Some of the phone labels may be read from the diagram directly; for example, the thick line in the bottom left is the initial silence and is then followed by a /sh/ phone half way up the diagram. Indeed, by making "reasonable" guesses and ignoring some of the noise, the first few phones can be read off directly as /sh iy hv ae dcl d/ which is correct for the first two words. Thus, the problem of connected word recognition can be rephrased as that of finding the maximum likelihood path through such a diagram while taking into account lexical and grammatical constraints.

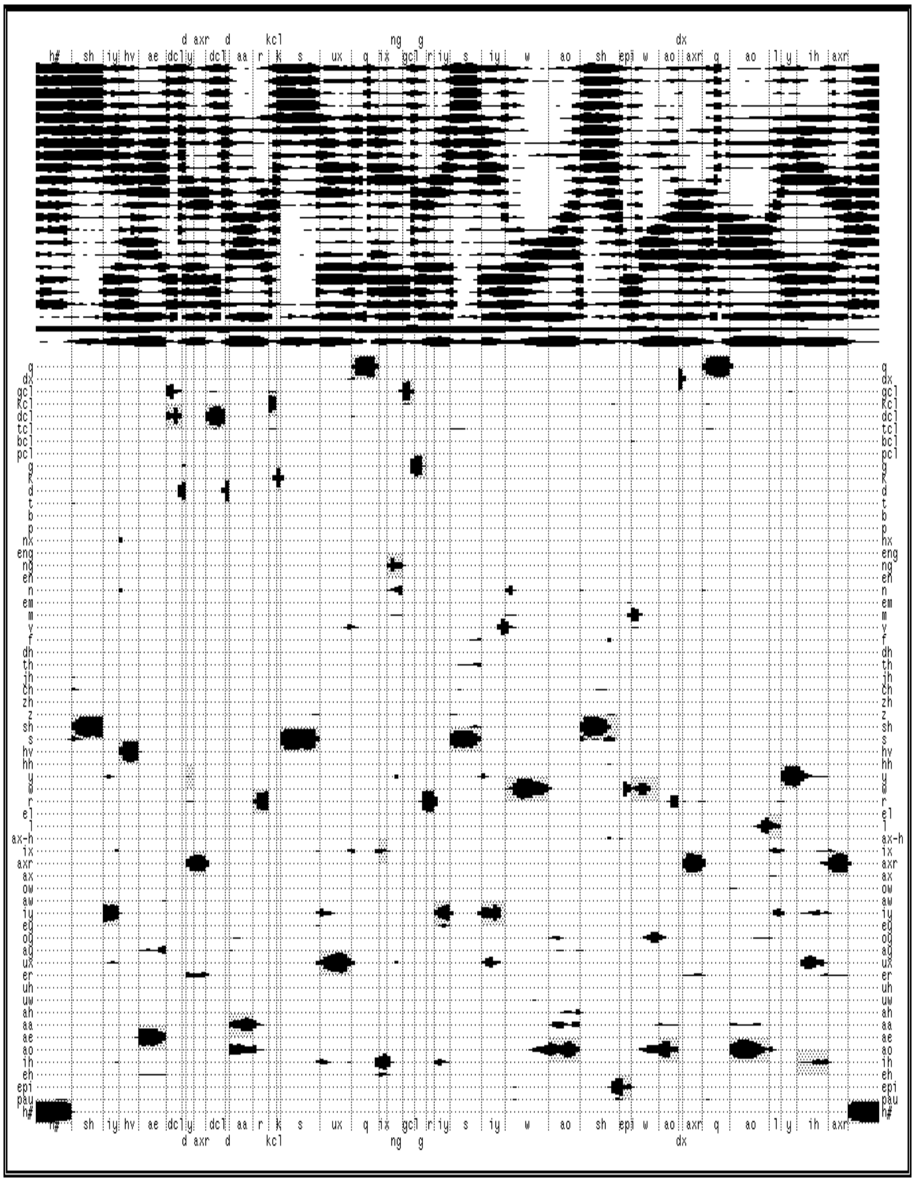

Figure 4 Input and output of the recurrent network for a TIMIT sentence "she had your dark suit in greasy wash water all year".

3.3 Posterior Probabilities to Scaled Likelihoods

The decoding criterion specified in (7.1) and (7.2) requires the computation of the likelihood of the data given a phone (state) sequence. Using the notation $\mathbf{u}_1^t = \mathbf{u}_1, \ldots, \mathbf{u}_t$, the likelihood is given by

$$p(\mathbf{u}_1^t|q_1^t) = \prod_{s=1}^{t} p(\mathbf{u}_s|q_1^t, \mathbf{u}_1^{s-1}). \tag{7.8}$$

In the interest of computational tractability and ease of training, standard HMMs make the assumptions of observation independence and that the Markov process is first order, i.e., $p(\mathbf{u}_s|q_1^t, \mathbf{u}_1^{s-1}) = p(\mathbf{u}_s|q_s)$. The recurrent hybrid approach, however, makes the less severe assumption that $p(\mathbf{u}_s|q_1^t, \mathbf{u}_1^{s-1}) = p(\mathbf{u}_s|q_s, \mathbf{u}_1^{s-1})$ which maintains the acoustic context in the local observation model. Manipulation of this results in an expression for the observation likelihood given by

$$p(\mathbf{u}_1^t|q_1^t) = \prod_{s=1}^{t} p(\mathbf{u}_s|\mathbf{u}_1^{s-1}) \frac{\Pr(q_s|\mathbf{u}_1^s)}{\Pr(q_s|\mathbf{u}_1^{s-1})}. \tag{7.9}$$

The computation of (7.9) is straightforward. The recurrent network is used to estimate $\Pr(q_s|\mathbf{u}_1^s)$. Because $p(\mathbf{u}_s|\mathbf{u}_1^{s-1})$ is independent of the phone sequence, it has no effect on the decoding process and is ignored. The one remaining issue in computing the scaled local likelihood is computation of $\Pr(q_s|\mathbf{u}_1^{s-1})$. The simplest solution is to assume $\Pr(q_s|\mathbf{u}_1^{s-1}) = \Pr(q_s)$ where $\Pr(q_s)$ is determined from the relative frequency of the phone q_s in the training data[5]. Although this works well in practice, it is obviously a wrong assumption and this area deserves further investigation.

3.4 Decoding Scaled Likelihoods

Equation (7.2) specified the standard HMM recognition criterion, i.e., finding the MAP state sequence. The scaled likelihoods described in the previous section are used in exactly the same way as the observation likelihoods for a standard HMM system. Rewriting (7.9) in terms of the network outputs and

[5] This computation is consistent with the MLP hybrid approach to computing scaled likelihoods [13].

making the assumptions stated above gives

$$Q^* = \operatorname*{argmax}_Q \prod_{t=1}^{T} \Pr(q_t|q_{t-1}) \frac{y_{q_t}(t)}{\Pr(q_t)}. \tag{7.10}$$

The non-observation constraints (e.g., phone duration, lexicon, language model, etc.) are incorporated via the Markov transition probabilities. By combining these constraints with the scaled likelihoods, we may use a decoding algorithm (such as time-synchronous Viterbi decoding or stack decoding) to compute the utterance model that is most likely to have generated the observed speech signal.

4 SYSTEM TRAINING

Training of the hybrid RNN/HMM system entails estimating the parameters of both the underlying Markov chain and the weights of the recurrent network. Unlike HMMs which use exponential-family distributions to model the acoustic signal, there is not (yet) a unified approach (e.g., EM algorithm [21]) to simultaneously estimate both sets of parameters. A variant of Viterbi training is used for estimating the system parameters and is described below.

The parameters of the system are adapted using Viterbi training to maximise the log likelihood of the most probable state sequence through the training data. First, a Viterbi pass is made to compute an alignment of states to frames. The parameters of the system are then adjusted to increase the likelihood of the frame sequence. This maximisation comes in two parts; (1) maximisation of the emission probabilities and (2) maximisation of the transition probabilities. Emission probabilities are maximised using gradient descent and transition probabilities through the re-estimation of duration models and the prior probabilities on multiple pronunciations. Thus, the training cycle takes the following steps:

1. Assign a phone label to each frame of the training data. This initial label assignment is traditionally done by using hand-labelled speech (e.g., the TIMIT database).
2. Based on the phone/frame alignment, construct the phone duration models and compute the phone priors needed for converting the RNN output to scaled likelihoods.

3. Train the recurrent network based on the phone/frame alignment. This process is described in more detail in section 4.1.

4. Using the parameters from 2. and the recurrent network from 3., apply Viterbi alignment techniques to update the training data phone labels and go to 2.

We generally find that four iterations of this Viterbi training are sufficient.

4.1 Training the RNN

Training the recurrent network is the most computationally difficult process in the development of the hybrid system. Once each frame of the training data has been assigned a phone label, the RNN training is effectively decoupled from the system training. An objective function which insures that the network input-output mapping satisfies the desired probabilistic interpretation is specified. Training of the recurrent network is performed using gradient methods. Implementation of the gradient parameter search leads to two integral aspects of the RNN training described below; (1) computation of the gradient and (2) application of the gradient to update the parameters.

RNN Objective Function

As discussed in earlier sections, the recurrent network is used to estimate the posterior probability of a phone given the input acoustic data. For this to be valid, it is necessary to use an appropriate objective function for estimating the network weights. An appropriate criterion for the softmax output of (7.5) is the cross-entropy objective function. For the case of Viterbi training, this objective function is equivalent to the log posterior probability of the aligned phone sequence and is given by

$$E = \sum_t \log y_{q_t}(t). \tag{7.11}$$

It has been shown in [9] that maximisation of (7.11) with respect to the weights is achieved when $y_i(t) = \Pr(q_t = i | u_1^t)$.

Gradient Computation

Given the objective function, the training problem is to estimate the weights to maximise (7.11). Of the known algorithms for training recurrent nets, back-propagation through time (BPTT) was chosen as being the most efficient in space and computation [22, 23]. The basic idea behind BPTT is illustrated in figure 5. The figure shows how the recurrent network can be expanded (in time) to represent an MLP where the number of hidden layers in the MLP is equal to the number of frames in the sequence. Training of the expanded recurrent network can be carried out in the same fashion as for an MLP (i.e., using standard error back-propagation [22]) with the constraint that the weights at each layer are tied. In this approach, the gradient of the objective function with respect to the weights (i.e., $\partial E / \partial w_{ij}$ and $\partial E / \partial v_{ij}$) is computed using the chain-rule for differentiation.

An overview of the gradient computation process for a sequence of N frames can be described as follows[6]:

1. Initialise the initial state $\mathbf{x}(0)$.
2. For $t = 0, \ldots, N-1$, compute $\mathbf{y}(t)$ and $\mathbf{x}(t+1)$ by forward propagating $u(t)$ and $\mathbf{x}(t)$ as specified in (7.4)–(7.6).
3. Set the error on the final state vector to zero as the objective function does not depend on this last state vector. Set the error on the output nodes to be the target value given by the Viterbi alignment less the actual output, $y_i(N-1)$, as in normal back-propagation training.
4. For $t = N-1, \ldots, 0$, back-propagate the error vector back through network. The error corresponding to the outputs is specified by the Viterbi alignment, while the error corresponding to the state are computed in the same way as backpropagation of the error to hidden units in a MLP.
5. Compute the gradient of the objective function with respect to the weights by accumulating over all frames.

Note that the state units have no specific target vector. They are trained in the same way as hidden units in a feedforward network and so there is no obvious "meaning" that can be assigned to their values. It should be pointed out that the proposed method is subject to boundary effects in that the frames at the

[6] The reader is directed to [23] for the details on the error back-propagation computations.

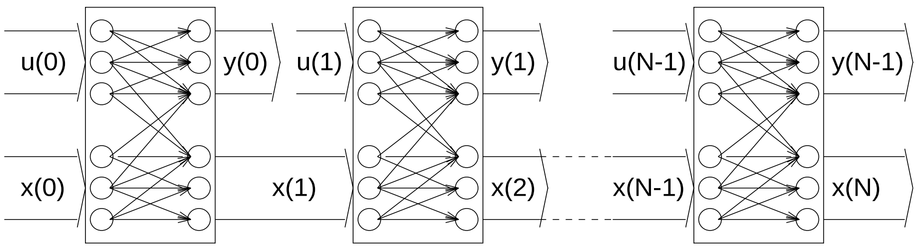

Figure 5 The expanded recurrent network.

end of a buffer do not receive an error signal from beyond the buffer. Although methods exist to eliminate these effects (e.g., [23]), in practice it is found that the length of the expansion (typically 256 frames) is such that the effects are inconsequential.

Weight Update

There are a number of ways in which the gradient signal can be employed to optimise the network. The approach described here has been found to be the most effective in estimating the large[7] number of parameters of the recurrent network. On each update, a local gradient, $\partial E^{(n)}/\partial W_{ij}^{(n)}$, is computed from the training frames in the nth subset of the training data. A positive step size, $\Delta W_{ij}^{(n)}$, is maintained for every weight and each weight is adjusted by this amount in the direction of the smoothed local gradient, i.e.,

$$W_{ij}^{(n+1)} = \begin{cases} W_{ij}^{(n)} + \Delta W_{ij}^{(n)} & \text{if } \frac{\partial E^{(n)}}{\partial W_{ij}^{(n)}} > 0 \\ W_{ij}^{(n)} - \Delta W_{ij}^{(n)} & \text{otherwise} \end{cases} . \tag{7.12}$$

The local gradient is smoothed using a "momentum" term by

$$\frac{\partial \tilde{E}^{(n)}}{\partial W_{ij}^{(n)}} = \alpha^{(n)} \frac{\partial \tilde{E}^{(n-1)}}{\partial W_{ij}^{(n-1)}} + (1 - \alpha^{(n)}) \frac{\partial E^{(n)}}{\partial W_{ij}^{(n)}}. \tag{7.13}$$

The smoothing parameter, $\alpha^{(n)}$, is automatically increased from an initial value of $\alpha^{(0)} = 1/2$ to $\alpha^{(\infty)} = 1 - 1/N$ by

$$\alpha^{(n)} = \alpha^{(\infty)} - (\alpha^{(\infty)} - \alpha^{(0)}) e^{-n/2N} \tag{7.14}$$

[7] The term *large* is relative to neural networks, not standard HMM acoustic modelling techniques.

where N is the number of weight updates per pass through the training data. The step size is geometrically increased by a factor ϕ if the sign of the local gradient is in agreement with the averaged gradient, otherwise it is geometrically decreased by a factor $1/\phi$, i.e.,

$$\Delta W_{ij}^{(n+1)} = \begin{cases} \phi \Delta W_{ij}^{(n)} & \text{if } \frac{\partial \tilde{E}^{(n-1)}}{\partial W_{ij}^{(n-1)}} \frac{\partial E^{(n)}}{\partial W_{ij}^{(n)}} > 0 \\ \frac{1}{\phi} \Delta W_{ij}^{(n)} & \text{otherwise} \end{cases} . \tag{7.15}$$

In this way, random gradients produce little overall change.

This approach is similar to the method proposed by Jacobs [24] except that a stochastic gradient signal is used and both the increase and decrease in the scaling factor is geometric (as opposed to an arithmetic increase and geometric decrease). Considerable effort was expended in developing this training procedure and the result was found to give better performance than the other methods that can be found in the literature. Other surveys of "speed-up" techniques reached a similar conclusion [25, 26].

5 SPECIAL FEATURES

The recurrent network structure applied within the HMM framework provides a powerful model of the acoustic signal. Besides the obvious advantages of increased temporal context modelling capability and minimal assumptions on the observation distributions, there are a number of less apparent advantages to this approach. Four such advantages are described in this section.

5.1 Connectionist Model Combination

Connectionist model combination refers to the process of merging the outputs of two or more networks. The original motivation for model merging with the hybrid system came from analysis of the recurrent network. Unlike a standard HMM, the recurrent network structure is time asymmetric. Training a network to recognise forward in time will result in different dynamics than training to recognise backwards in time. As different information is available to both processes, it seems reasonable that better modelling can be achieved by combining both information sources.

Significant improvements have been observed by simply averaging the network outputs [27], i.e., setting

$$y_i(t) = \frac{1}{K}\sum_{k=1}^{K} y_i^{(k)}(t) \tag{7.16}$$

where $y_i^{(k)}(t)$ is the estimate of the kth model. Although this merging has been successful, the approach is somewhat ad-hoc. A more principled approach to model merging is based on using the Kullback-Leibler information as a distance-like measure on multinomial distributions. Consider the following criterion

$$E(p) = \sum_{k=1}^{K} D(p||y^{(k)}) \tag{7.17}$$

where

$$D(p||q) \equiv \sum_i p_i \log \frac{p_i}{q_i} \tag{7.18}$$

is the Kullback-Leibler information. Minimisation of E with respect to the distribution p can be interpreted as choosing the distribution which minimises the average (across models) Kullback-Leibler information. Solving the minimisation in (7.17) results in the log- domain merge of the network outputs, i.e.,

$$\log y_i(t) = \frac{1}{K}\sum_{k=1}^{K} \log y_i^{(k)}(t) - B \tag{7.19}$$

where B is a normalisation constant such that $\mathbf{y}$ is a probability distribution. This technique has been applied to merging four networks for large vocabulary speech recognition [28]. The four networks represented forward and backward MEL+ and PLP acoustic preprocessing described in section 3.1. Recognition results are reported in table 1 for three different test sets.

Whilst the exact gains are task specific, it is generally found that linear merging of four networks provide about 17% fewer errors. The log domain merging performs better with approximately 24% fewer errors when four networks are combined.

5.2 Duration Modelling

The recurrent network is used to estimate the local observation probabilities within the HMM framework. Although the dynamics of the network encode

	Word Error Rate %		
Merge Type	spoke 5	spoke 6	H2
FORWARD MEL+	17.3	15.0	16.2
FORWARD PLP	17.1	15.1	16.5
BACKWARD MEL+	17.8	15.5	16.1
BACKWARD PLP	16.9	14.4	15.2
AVERAGE	17.3	15.0	16.0
UNIFORM MERGE	15.2	11.4	13.4
LOG-DOMAIN	13.4	11.0	12.6

Table 1 Merging results for the ARPA 1993 spoke 5 development test, 1993 spoke 6 development test, and the 1993 hub 2 evaluation test. All tests utilised a 5,000 word vocabulary and a bigram language model and were trained using the SI-84 training set.

some segmental information, explicit modelling of phone duration improves the hybrid system's performance on word recognition tasks[8].

Phone duration within the hybrid system is modelled with a hidden Markov process. In this approach, a Markov chain is used to represent phone duration. The duration model is integrated into the hybrid system by expanding the phone model from a single state to multiple states with tied observation distributions, i.e.,

$$p(\mathbf{u}|q_t = i) = p(\mathbf{u}|q_t = j) \tag{7.20}$$

for i and j states of the same phone model.

Choice of Markov chain topology is dependent on the decoding approach. Decoding using a maximum likelihood word sequence criterion is well suited to complex duration models as found in [29]. Viterbi decoding, however, results in a Markov chain on duration where the parameters are not hidden (given the duration). Because of this, a simple duration model as shown in figure 6 is employed. The free parameters in this model are (1) the minimum duration of the model, N, (2) the value of the first $N-1$ state transitions, a, (3) the self-transition of the last state x, and (4) the exit transition value, b. The duration score generated by this model is given as

$$p_\tau = \begin{cases} 0 & \text{if } \tau < N, \\ a^{N-1} b x^{\tau - N} & \text{if } \tau \geq N \end{cases} \tag{7.21}$$

[8]This is not necessarily the case for phone recognition where the network training criterion and the actual task are more closely linked.

and p_τ is not necessarily a proper distribution.

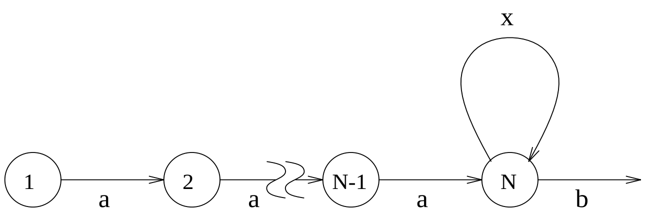

Figure 6 Phone-deletion penalty duration model.

The parameters are determined in the following manor. First, the minimum duration N is set equal to half the average duration of the phone. The average duration of the phone is computed from Viterbi alignment of the training data. The parameters a and x are arbitrarily set to 0.5. The parameter b represents a phone-deletion penalty and is empirically set to maximise performance on a cross-validation set.

5.3 Efficient Models

One of the great benefits of this approach is the efficient use of parameters. In a comparable HMM system, acoustic context is modelled via context-dependent phone models. For a large vocabulary, speaker independent task (e.g., the Wall Street Journal), this typically results in thousands of phone models. In addition, these phone models are comprised of some number of states which model the dynamics of the signal within the phone. In contrast, the RNN models context with the hidden state units and only context-independent outputs are required. Because the RNN is a dynamic model, it is only necessary to model the individual phones, not sub-phone units. This results in an HMM system with a single state per context-independent phone.

A typical RNN will have 20 to 50 inputs, 256 state units, and 50 to 80 outputs for approximately 100,000 parameters specifying the complete observation model. This is one to two orders of magnitude fewer parameters than an HMM with similar performance. The efficiency of the model is also a consequence of the training paradigm. The error-corrective training allocates parameters based in reducing errors, rather than on modelling distributions. The efficient representation of the acoustic model results in a number of desirable properties, e.g., fast decoding.

5.4 Decoding

The task of decoding is to find the maximum likelihood word sequence given the models and acoustic evidence. The time synchronous Viterbi algorithm provides an efficient means of performing this task for small vocabularies (e.g., less than 1000 words) and short span language models (e.g., bigrams). However, with larger vocabularies and longer span language models a simple exhaustive search is not possible and the issue of efficient decoding becomes critical to the performance of the system.

Search Algorithm

A search procedure based on stack decoding [30, 31] has been adopted. This search procedure may be regarded as a reordered time-synchronous Viterbi decoding and has the advantage that the language model is decoupled from the search procedure. Unlike time-synchronous Viterbi decoding, the Markov assumption is not integral to the search algorithm. Thus, this decoder architecture offers a flexible platform for single-pass decoding using arbitrary language models. The operation of the algorithm is described in some detail in [32, 33]. Discussed below are some new approaches to pruning that have been developed to take advantage of hybrid system properties.

Pruning

Two basic pruning criteria are used to reduce the computation required in decoding. *Likelihood* based pruning is similar to the various types of pruning used in most decoders and is based on the acoustic model likelihoods. *Posterior* based pruning is specific to systems which employ a local posterior phone probability estimator.

Likelihood based methods are used to compute the envelope and also to set a maximum stack size. These rely on the computation of an estimate of the least upper bound of the log likelihood at time t, `lub(t)`. This is an updated estimate and is equal to the log likelihood of the most probable partial hypothesis at time t. The size of the envelope is set heuristically and is dependent on the accuracy of the estimate of `lub(t)`. The second pruning parameter is used to control the maximum number of hypotheses in a stack. This parameter may be regarded as adaptively tightening the envelope, while ensuring that M hypotheses are still extended at each time (subject to the overall envelope).

A second pruning method has been developed to take advantage of the connectionist probability estimator used in the hybrid system. The phone posteriors may be regarded as a local estimate of the presence of a phone at a particular time frame. If the posterior probability estimate of a phone given a frame of acoustic data is below a threshold, then all words containing that phone at that time frame may be pruned. This may be efficiently achieved using a tree organisation of the pronunciation dictionary. This process is referred to as *phone deactivation pruning*. The posterior probability threshold used to make the pruning decision may be empirically determined in advance using a development set and is constant for all phones.

This posterior-based approach is similar to the likelihood-based channel-bank approach of Gopalakrishnan et al. [34], which used phone-dependent thresholds. However, that system incurred a 5–10% relative search error to obtain a factor of two speedup on large vocabulary task. This new approach is extremely effective. On a 20K trigram Wall Street Journal task, phone deactivation pruning can result in close to an order of magnitude faster decoding, with less than 2% relative search error (see table 2).

20K Trigram, Trained on SI-84					
Pruning Parameters		si_dt_s5		Nov '92	
Envelope	Threshold	Time	%Error	Time	%Error
10	0.000075	16.1	12.1	15.7	12.6
10	0.0005	4.3	12.2	3.9	12.9
10	0.003	1.4	14.3	1.3	14.9
8	0.0	46.8	12.5	50.4	12.6
8	0.000075	5.4	12.2	4.9	12.8
8	0.0005	1.7	12.6	1.5	13.6
8	0.003	0.6	15.0	0.6	15.8

Table 2 Decoding performance on the Wall Street Journal task using a 20,000 word vocabulary and a trigram language model. Accuracy and CPU time (in multiples of realtime on an HP735) are given with respect to varying the likelihood envelope and the posterior-based phone deactivation pruning threshold. The maximum stack size was set to be 31.

6 SUMMARY OF VARIATIONS

This section provides a concise description of the differences between the hybrid RNN/HMM and standard HMM approaches. It should be pointed out that many of the capabilities attributed to the recurrent network can also be represented by standard HMMs. However, the incorporation of these capabilities into standard HMMs is not necessarily straightforward.

6.1 Training Criterion

The parameters of the recurrent network are estimated with a discriminative training criterion. This leads to a mechanism for estimation of the posterior probability for the phones given the data. Standard HMM training utilises a maximum likelihood criterion for estimation of the phone model parameters. The recurrent network requires substantially fewer parameters because discriminative training focuses the model resources on decision boundaries instead of modeling the complete class likelihoods.

6.2 Distribution Assumptions

One of the main benefits of the recurrent network is that it relaxes the conditional independence assumption for the local observation probabilities. This results in a model which can represent the acoustic context without explicitly modeling phonetic context. This has positive ramifications in terms of the number of required parameters and the complexity of the search procedure.

The second main assumption of standard HMMs is that the observation distributions are from the exponential family (e.g., multinomial, Gaussian, etc.) or mixtures of exponential family distributions. The recurrent network, however, makes much fewer assumptions about the form of the acoustic vector distribution. In fact, it is quite straightforward to use real-valued and/or categorical data for the acoustic input. In theory, a Gaussian mixture distribution and a recurrent network can both be considered nonparametric estimators by allowing the size (e.g., number of mixtures or state units, respectively) to increase with additional training data. However, because standard HMMs employ maximum likelihood estimation there is the practical problem of sufficient data to estimate all the parameters. Because the recurrent network shares the state units for all phones, this data requirement is less severe.

6.3 Practical Issues

There are a number of practical advantages to the use of a recurrent network instead of an exponential family distribution. The first, mentioned in section 6.1, is that the number of required parameters is much fewer than standard systems. In addition, section 5.4 shows that the posterior probabilities generated by the network can be used efficiently in the decoding – both for computing likelihoods and pruning state paths (similar to fast-match approaches which are add-ons to standard systems). Of course, a major practical attraction of the approach is that it is very straightforward to map the recurrent network to standard DSP architectures.

7 A LARGE VOCABULARY SYSTEM

A hybrid RNN/HMM system has been applied to an open vocabulary task; namely the 1993 ARPA evaluation of continuous speech recognition systems. The hybrid system employed context-independent phone models for a 20,000 word vocabulary with a backed-off trigram language model. Forward and backward in time `MEL+` and `PLP` recurrent networks were merged to generate the observation probabilities. The performance of this simple system (17% word error rate using less than a half million parameters for acoustic modelling) was similar to that of much larger, state-of-the-art HMM systems. This system has recently been extended to a 65,533 word vocabulary and the simplicity of the hybrid approach resulted in decoding with minimal search errors in only 2.5 minutes per sentence.

8 CONCLUSION

Recurrent networks are able to model speech as well as standard techniques such as hidden Markov models based on Gaussian mixture densities. Recurrent networks differ from the HMM approach in making fewer assumptions on the distributions of the acoustic vectors, having a means of incorporating long term context, using discriminative training, and providing a compact set of phoneme probabilities which may be efficiently searched for large vocabulary recognition. There are also practical differences such as the fact that the training of the systems is slower than the HMM counterpart, but this is made up for by faster execution at recognition time. The RNN system also has relatively few parameters, and these are used in a simple multiply and accumulate loop so hardware implementation is more plausible. In summary, recurrent networks are an attractive, alternative statistical model for use in the core of a large vocabulary recognition system.

Acknowledgements

Two of the authors, T.R. and S.R., held U. K. Engineering and Physical Sciences Research Council Fellowships. This work was supported in part by ESPRIT project 6487, WERNICKE. For the reported experiments, the pronunciation dictionaries were provided by Dragon Systems and the language models were provided by MIT Lincoln laboratories.

REFERENCES

[1] H. F. Silverman and D. P. Morgan, "The application of dynamic programming to connected speech recognition," *IEEE ASSP Magazine*, vol. 7, pp. 6–25, July 1990.

[2] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, February 1989.

[3] N. Morgan and H. Bourlard, "Continuous speech recognition using multilayer perceptrons with hidden Markov models," in *Proc. ICASSP*, pp. 413–416, 1990.

[4] S. Renals, N. Morgan, H. Bourlard, M. Cohen, and H. Franco, "Connectionist probability estimators in HMM speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 2, Jan. 1994.

[5] F. Jelinek and R. Mercer, "Interpolated estimation of Markov source parameters from sparse data," *Pattern Recognition in Practice*, pp. 381–397, 1980.

[6] K.-F. Lee, *Automatic Speech Recognition: The Development of the SPHINX System.* Boston: Kluwer Academic Publishers, 1989.

[7] S. Furui, "Speaker-independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, pp. 52–59, Feb. 1986.

[8] E. B. Baum and F. Wilczek, "Supervised learning of probability distributions by neural networks," in *Neural Information Processing Systems* (D. Z. Anderson, ed.), American Institute of Physics, 1988.

[9] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neuro-computing: Algorithms, Architectures and Applications* (F. Fougelman-Soulie and J. Hérault, eds.), pp. 227–236, Springer-Verlag, 1989.

[10] H. Bourlard and C. J. Wellekens, "Links between Markov models and multilayer perceptrons," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1167–1178, Dec. 1990.

[11] H. Gish, "A probabilistic approach to the understanding and training of neural network classifiers," in *Proc. ICASSP*, pp. 1361–1364, 1990.

[12] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian a posteriori probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.

[13] H. Bourlard and N. Morgan, *Continuous Speech Recognition: A Hybrid Approach.* Kluwer Academic Publishers, 1994.

[14] J. S. Bridle, "ALPHA-NETS: A recurrent 'neural' network architecture with a hidden Markov model interpretation," *Speech Communication*, vol. 9, pp. 83–92, Feb. 1990.

[15] J. S. Bridle and L. Dodd, "An Alphanet approach to optimising input transformations for continuous speech recognition," in *Proc. ICASSP*, pp. 277–280, 1991.

[16] L. T. Niles and H. F. Silverman, "Combining hidden Markov models and neural network classifiers," in *Proc. ICASSP*, pp. 417–420, 1990.

[17] S. J. Young, "Competitive training in hidden Markov models," in *Proc. ICASSP*, pp. 681–684, 1990. Expanded in the technical report CUED/F-INFENG/TR.41, Cambridge University Engineering Department.

[18] A. J. Robinson and F. Fallside, "Static and dynamic error propagation networks with application to speech coding," in *Neural Information Processing Systems* (D. Z. Anderson, ed.), American Institute of Physics, 1988.

[19] P. McCullagh and J. A. Nelder, *Generalised Linear Models.* London: Chapman and Hall, 1983.

[20] T. Robinson, "The state space and "ideal input" representations of recurrent networks," in *Visual Representations of Speech Signals*, pp. 327–334, John Wiley and Sons, 1993.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. Roy. Statist. Soc.*, vol. B39, pp. 1–38, 1977.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. I: Foundations.* (D. E. Rumelhart and J. L. McClelland, eds.), ch. 8, Cambridge, MA: Bradford Books/MIT Press, 1986.

[23] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, pp. 1550–1560, Oct. 1990.

[24] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Networks*, vol. 1, pp. 295–307, 1988.

[25] W. Schiffmann, M. Joost, and R. Werner, "Optimization of the backpropagation algorithm for training multilayer perceptrons," tech. rep., University of Koblenz, 1992.

[26] T. T. Jervis and W. J. Fitzgerald, "Optimization schemes for neural networks," Tech. Rep. CUED/F-INFENG/TR144, Cambridge University Engineering Department, Aug. 1993.

[27] M. M. Hochberg, S. J. Renals, A. J. Robinson, and D. J. Kershaw, "Large vocabulary continuous speech recognition using a hybrid connectionist-HMM system," in *Proc. of ICSLP-94*, pp. 1499–1502, 1994.

[28] M. M. Hochberg, G. D. Cook, S. J. Renals, and A. J. Robinson, "Connectionist model combination for large vocabulary speech recognition," in *Neural Networks for Signal Processing IV* (J. Vlontzos, J.-N. Hwang, and E. Wilson, eds.), pp. 269–278, IEEE, 1994.

[29] T. H. Crystal and A. S. House, "Segmental durations in connected-speech signals: Current results," *J. Acoust. Soc. Am.*, vol. 83, pp. 1553–1573, Apr. 1988.

[30] L. R. Bahl and F. Jelinek, "Apparatus and method for determining a likely word sequence from labels generated by an acoustic processor." US Patent 4,748,670, May 1988.

[31] D. B. Paul, "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model," in *Proc. ICASSP*, vol. 1, (San Francisco), pp. 25–28, 1992.

[32] S. J. Renals and M. M. Hochberg, "Decoder technology for connectionist large vocabulary speech recognition," Tech. Rep. CUED/F-INFENG/TR.186, Cambridge University Engineering Department, 1994.

[33] S. Renals and M. Hochberg, "Efficient search using posterior phone probability estimates," in *Proc. ICASSP*, pp. 596–599, 1995.

[34] P. S. Gopalakrishnan, D. Nahamoo, M. Padmanabhan, and M. A. Picheny, "A channel-bank-based phone detection strategy," in *Proc. ICASSP*, vol. 2, (Adelaide), pp. 161–164, 1994.