Proceedings of the Institute of Acoustics

A COMPARISON OF HIDDEN CONTROL NEURAL NETWORKS AND HIDDEN MARKOV MODELS.

G D Tattersall (1), G E Lee (1) & S G Smyth (2).

(1) School of Information Systems, University of East Anglia, Norwich.
(2) Speech Applications Division, BT Laboratories, Ipswich.

1. INTRODUCTION

Early neural network speech recognition systems were notable for their failure to perform as well as classical recognisers based on techniques such as Dynamic Time Warping (DTW) and the Hidden Markov Models (HMM). The failure may be due to the difficulty of explicitly dealing with the time variability of speech when using neural networks, and various architectures such as the Time Delay Neural Network (TDNN) and recurrent Multi Layer Perceptron (MLP) have been proposed to solve this problem.

Another method of using a neural network in a way which takes explicit account of time variability has been proposed by Levin and is called the Hidden Control Neural Network (HCNN) [1]. Like the HMM, the HCNN attempts to model an utterance by a Markov state sequence. However, instead of associating a specific observed vector emission probability distribution with each state, the HCNN assumes that the vectors generated during a particular state are characterised by a specific prediction function which enables the next observed vector to be predicted from previous vectors.

The HCNN classifier consists of a number of word models. Each word model contains an MLP which has been trained to predict the next frame in the observed frame sequence, for words of a particular class, from a few of the previous frames. Recognition is performed by applying the observation sequence of an unknown utterance to every word model and the utterance is assigned the class of the word model whose MLP produces the least prediction error energy aggregated over the entire utterance.

The experimental work described in this paper was designed to evaluate the HCNN for the recognition of isolated words and to compare the results with those obtainable using continuous density HMMs on the same data. It will be shown that the performance of HCNNs is significantly worse than HMMs and an investigation into the reasons is described. In particular, it will be shown that frame to frame prediction functions are not a good characterisation of particular speech states and that in reality, the HCNN probably does not learn such prediction functions.

As a result of the disappointing performance of HCNNs, a new technique called the Hidden Control Density Mapper (HCDM) is proposed. The HCDM has a similar architecture to the HCNN but does not attempt to characterise the speech by predictive functions. Instead, the MLP in each word model is trained to produce as high an output as possible on utterances belonging to its class and as low output as possible on utterances of other classes. Thus the recogniser is trained in a *class discriminative* fashion, unlike the HMM and HCNN. Recognition is performed by applying the unknown utterance's frame sequence to all models and aggregating each word model MLP's output over the entire utterance. The utterance is assigned the class of the word model producing the highest score.

The HCDM will be shown to perform almost as well as an HMM but without the need for the estimation of state transition probabilities, and in spite of its currently inferior performance it is thought that further development may yield a system which works as well as the HMM.

2. PRINCIPLE OF THE HCNN.

2.1 Architecture of the HCNN.

The architecture of a single HCNN word model is shown in figure 1. As already described, the word model consists of an MLP which attempts to predict the next frame, C_n, in an utterance from previous frames. In practice, an extremely complex MLP would be required in each word model if it were required to synthesise the complete range of state specific prediction functions in a particular class of utterance. A more robust approach is to select a separate MLP to synthesise the prediction function associated with each state. Alternatively, a single MLP can be used in conjunction with an extra input to which is applied a state control code, S_n. The effect of the state control input is to "steer" the MLP to produce the required state specific prediction function. It is the last method which was proposed by Levin and from which the technique derives its name of Hidden Control Neural Network.

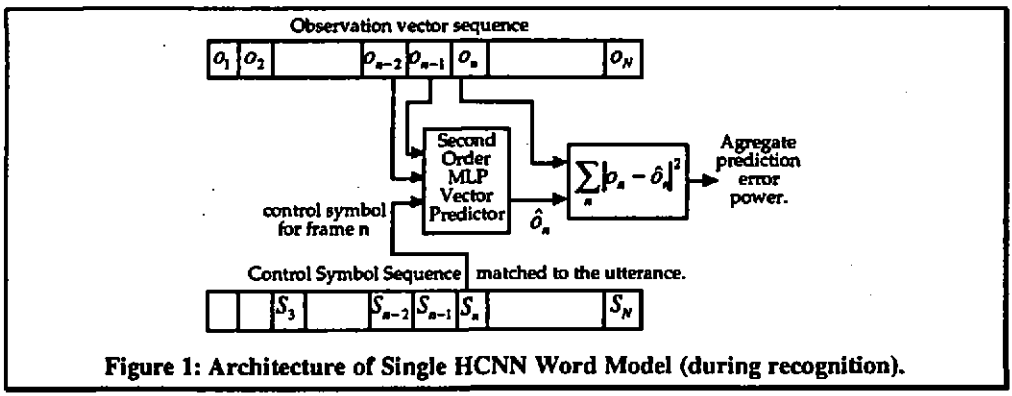

Figure 1: Architecture of Single HCNN Word Model (during recognition).

Typically, "one-out-of-N" codes are used to represent each state and these are applied to the input of the MLP in concatenation with the chosen number of past frames. For example, if it is assumed that the utterance can be modelled by 8 states, then the control code consists of an 8 element vector, with just 1 non-zero element. Typically, the prediction is based upon one previous composite Mel Frequency Cepstral Coefficient (MFCC) vector of 17 dimensions, and so the total number of inputs to the MLP will be 25.

The optimal sequence of control codes is found using the Viterbi algorithm [2] such that the aggregated prediction error energy is minimised. The sequence of control codes effectively defines the sequence of underlying states in the speech during which the required speech frame predictor has a state specific transfer function. In common with HMMs, the sequence of control codes is assumed to be governed by a Markov process [3].

2.2 Training the HCNN.

The MLP predictor for each class is trained separately by alternating between two distinct phases which are referred to as *re-estimation* and *segmentation*. It is of note that the HCNN does not

perform discriminative learning like most other neural net classifiers because each MLP predictor is trained separately. This is probably a significant deficiency in the existing HCNN concept.

Training is started by choosing an initial segmentation of the utterance. Typically, the utterance is uniformly segmented so that equal numbers of speech frames are associated with each of the control codes. MLP weight re-estimation is then performed using Backward Error Propagation (BEP) [4] using a randomly picked observation vector, o_n, from a chosen utterance example. The appropriate control code S_n can be referenced for the chosen observation and it is appended to o_{n-1} and o_{n-2} to produce the overall MLP input vector E_i. The target T_i will correspond to o_n. A series of such examples are randomly selected and used to train the MLP whose weights are updated with each new example.

Following many weight re-estimations, the segmentation training phase is entered. In this phase, the MLP is used to predict the next frame in each of the utterance frame sequences and the sequence of control codes is optimized using the Viterbi algorithm to minimise the mean prediction error over all utterance examples of the particular class. During segmentation, the MLP weights are fixed at whatever values they had after the previous re-estimation phase.

2.3 Classification Using The HCNN.

Once the optimal MLP weights sets have been found for each class of word, the system can be used for recognition of unlabelled words. The unknown observation sequence is applied to each HCNN word model and the Viterbi algorithm used to find the optimal sequence of control codes which minimises the prediction error energy for each model. The utterance is assigned the class of the model producing the least error energy. Note that none of the code sequences derived during training are used during recognition. The segmentation process is performed afresh for each class predictor and is unique for each unlabelled utterance to be classified.

3. WHOLE WORD RECOGNITION USING HCNNs AND HMMs

All experiments presented in this paper were done using the BT speaker independent 'S1' corpus which consists of a training set of 3 examples by 52 speakers of each of the letters of the alphabet. The test set consists of 3 examples of each of the letters spoken by a different set of 52 speakers. The utterances were encoded as 17 dimensional vectors containing a differential log frame energy, MFCC coefficients C_1 to C_8 and their time differentials ΔC_1 to ΔC_8. These delta-MFCC frames (or observations) were produced every 16ms.

The HCNN's MLP predictors were provided with an input consisting of a sequence of pairs of delta-MFCC frames concatenated with a "one-out-of-four" control code. Each MLP used 38 input units, 5 sigmoidal hidden units and had 17 linear output units. A separate model was trained for each of the 26 word classes. The HMM recogniser consisted of 4 state, no skip, models using a Mixture of 7 Gaussians to model the emission probabilities in each state (this was the same number of states used in the HCNN tests). The HMM was trained using Viterbi alignment and clustering, followed by Baum Welch re-estimation operating on 17 dimensional delta MFCC vectors as described in section 3.1.

The HCNN predictor achieved a performance of $34.7 \pm 1.5\%$ on the S1 test set and $38.2 \pm 1.5\%$ on its training set suggesting a reasonable level of generalisation was being achieved. The HMM provided an accuracy of $94.0 \pm 0.8\%$ and $84.5 \pm 1.1\%$ on the training and test sets respectively. Confidence limits are set at twice the estimated standard deviation [5].

4. CHARACTERISING SPEECH BY INTER FRAME RELATIONSHIPS

4.1 Frame to frame prediction function as a speech characteristic.
The essential idea underlying the HCNN recogniser is that each speech state within an utterance should be characterised by a particular frame to frame prediction function. However, the rather poor performance of the HCNN compared to an HMM suggests that frame to frame prediction functions are not in reality good characterisers of different speech states. Instead, the MLP may act as a "look up table", producing an output value which is determined primarily by the state control code at its input and which is unrelated to its previous frame inputs. The optimal output in the latter case would be a vector having the mean value of the observed vectors for that state and the system would exhibit similar behaviour to an HMM which had a single entry codebook for its state dependent emission probabilities.

An experiment was devised to test this hypothesis. It involved a simplified form of HCNN in which the MLP in each word model was replaced by a set of linear predictors. Each predictor is selected in turn by the sequence of state control codes. Thus, the non-linear mapping of the MLP is replaced by a set of linear mappings. The system is called the Hidden Control Linear Predictor (HCLP) and it is impossible for this system to produce an output just in response to the state control code. If the HCLP enabled good recognition, it would indicate that prediction functions really are characteristic of particular speech states. It might be argued that speech states could be characterised by non-linear prediction but not by linear prediction functions. However, previous work on the prediction of line spectral pair descriptions of speech has shown that non-linear prediction has negligible advantage over linear prediction and it is concluded that MFCC observations are in general linearly related.

4.2 Architecture and operation of the Hidden Control Linear Predictor.
The architecture of HCLP is shown in figure 2 which illustrates the set of linear predictors which are selected by the sequence of state control codes. The state control sequence is identical to the sequence which would have been an input to the MLP in an HCNN.

The HCLP is trained in a very similar manner to the HCNN. Each training utterance is initially divided into equal length state segments which define the initial state control sequence. The coefficients of each linear predictor are then adapted using gradient descent minimisation of its prediction error during the period when it is selected by the state control sequence. The coefficients are adapted at the frame rate of the speech. The error power given by a linear predictor is a quadratic function of its coefficients and so convergence to a global minimum is assured.

After each of the training examples have been used several times, the Viterbi algorithm is used to re-estimate the optimal state segmentation of each utterance, and back propagation of the prediction error is re-commenced to adapt the predictor coefficients. The 2 training phases of state segmentation and coefficient adaptation are alternately repeated many times.

4.3 Recognition using the HCLP.
The HCLP was trained and tested on the S1 corpus using the standard front end processing scheme described in 3.1. 26 HCLP models were trained corresponding to each of the classes. Each HCLP used 4 control states, a second order linear predictor being associated with each. The system achieved $14.3 \pm 1.1\%$ accuracy on the test set and $14.7 \pm 1.1\%$ on its training set. Tests were also conducted using fourth order linear predictors but these offered only $9.4 \pm 0.9\%$ accuracy on the test set.

Although the performance achieved by the HCLP is considerably poorer than any other technique tested on this database, it is still performing approximately 4 times better than a totally random

classifier. None the less, the results do suggest that frame to frame prediction is a very poor way of characterising speech states.

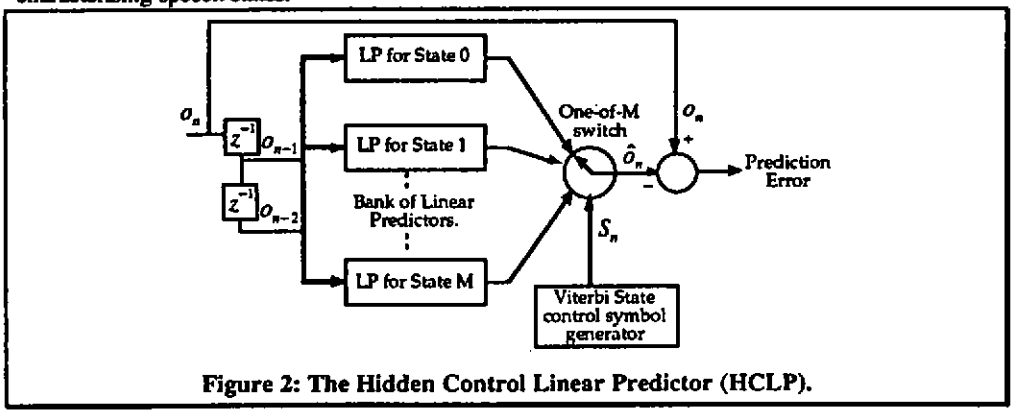

Figure 2: The Hidden Control Linear Predictor (HCLP).

4.4 The effect of the MFCC representation on predictive characterisation.
The previous experiments have shown that the prediction functions relating successive frame values are poor characterisers of specific speech states. However, it is possible that this is due to the particular speech representation which has been used, the MFCC.

MFCCs represent the log energy in a section of speech by the value of just one coefficient, C_0. The remaining coefficients, C_1 to C_8, encode the log power spectral shape by resolving it onto a set of cosine basis functions. If we consider low energy sounds such as fricatives or background noise, the value of C_0 will be low but the values of the other MFCC coefficients will be just as large as when the speech is a high energy event such as a vowel sound. This is because scaling down spectral power corresponds to a vertical offset in the log spectral envelope; the envelope shape and hence the Discrete Cosine Transform (DCT) coefficients that describe it remain unaffected. This means that the estimate of spectral shape encoded by coefficients C_1 to C_8 will be highly variable during low energy parts of an utterance such as fricatives or stops. This will have a profound effect on the operation of an HCNN because state-specific prediction of successive frames will be impossible in these circumstances.

The previous argument suggests that the HCNN may work better with a simple spectral representation, such as unprocessed filter bank coefficients. Alternatively, it may be possible to use MFCCs but to weight the HCNN prediction error by the current frame energy. This would allow the compactness of the MFCC representation to be exploited whilst reducing the significance of noise-like sounds in the prediction error.

5. THE HIDDEN CONTROL DENSITY MAPPER

5.1 Principle of the Class Discriminative Hidden Control Density Mapper.
The Hidden Control Density Mapper (HCDM) is an attempt to allow a state based neural-network classifier to model the distributions associated with training observations, rather than their temporal correlation, as in the HCNN. In this respect it is similar to the HMM. However, rather than being trained to directly model the probability density function of the observations associated with each state, the HCDM is trained to maximise the average discrimination between the output of the correct

classifier and those of all incorrect classifiers. In other words, the system undergoes Class Discriminative Learning (CDL). Figure 3 shows the HCDM architecture.

If a specific HCDM classifier operates on a training observation sequence associated with class k,

$$O_k = \{o_{1k} \quad o_{2k} \quad o_{3k} \quad \cdot \quad o_{Tk}\} \tag{1}$$

it will produce a response which is the sum of the density mapping values associated with each of the observation vectors. If the HCDM corresponding to the j^{th} class is used we can denote its density mapping function F_j and its response to a pattern associated with class k as r_{jk} (equation 2).

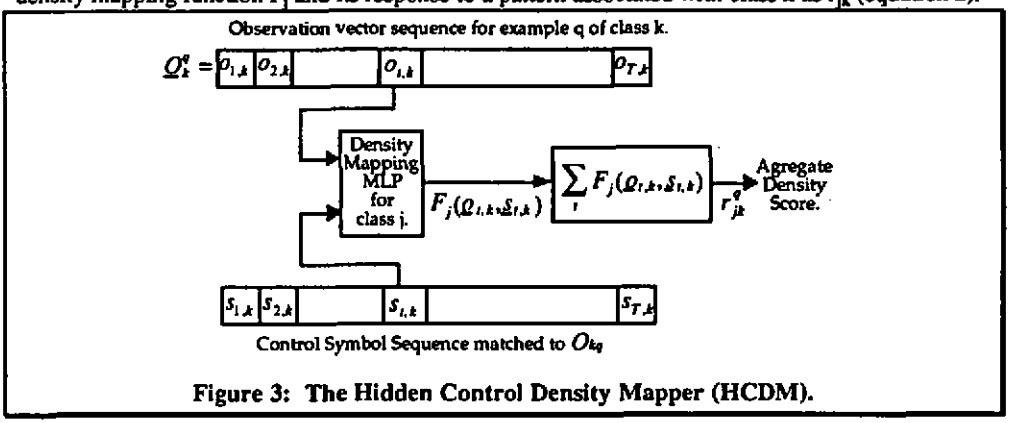

Figure 3: The Hidden Control Density Mapper (HCDM).

$$r_{jk}^q = \sum_{t=1}^{T_q} F_j(o_{tk}, s_{tk}) \tag{2}$$

Here r_{jk}^q denotes the response of the j^{th} HCDM to the q^{th} example within the training set labelled as class k. T_q is the number of vectors constituting the q^{th} example and F_j denotes the density mapping performed (by an MLP) within the HCDM associated with class j. Finally s_{tk} is the control symbol encoding the correct state for the t^{th} frame of the O_k which would be produced by the Viterbi algorithm. If N_k denotes the number of training set examples labelled as class k, we can generate a mean response by considering all of the examples of class k within the training set,

$$R_{jk} = \frac{1}{N_k} \sum_{q=1}^{N_k} \frac{r_{jk}^q}{T_q} \tag{3}$$

5.2 Class Discriminative Learning.

The aim of the CDL algorithm is to maximise the discrimination between the correct HCDM response and all the incorrect HCDM responses. The discrimination for O_k can be defined as the ratio of the response of the correct HCDM to the total response of the HCDM associated with all of the classes,

$$p_k = \frac{R_{kk}}{\sum_{j=1}^{M} R_{jk}} \tag{4}$$

COMPARISON BETWEEN HCNNs AND HMMs

To recap, R_{jk} is the mean response of the HCDM corresponding to class j to observation sequences associated with class k and M is the number of classes. We can go further by defining a gross discriminative metric accumulated across all classes,

$$P_G = \sum_{k=1}^{M} P_k \tag{5}$$

To obtain maximal discrimination across all classes we need to maximise P_G which we can do using gradient ascent. If we denote a parameter within the F_p (the MLP within the p^{th} HCDM) as ϕ_p we can iteratively increase P_G using a steepest ascent equation,

$$\phi_p^{n+1} = \phi_p^n + k\frac{\partial P_G}{\partial \phi_p} \tag{6}$$

where k is an adaptation step size and the superscript of ϕ denotes the update iteration. We can derive an expression for the partial differential of P_G from equations (5) and (4) in terms of the differentials of the density mapping results,

$$\frac{\partial P_G}{\partial \phi_p} = \sum_{k=1}^{M} \frac{\partial}{\partial \phi_p}\left\{\frac{R_{ik}}{\sum_{j=1}^{M} R_{jk}}\right\} \tag{7}$$

$$= \sum_{k=1}^{M}\left(\frac{1}{\sum_{j=1}^{M} R_{jk}}\frac{\partial R_{ik}}{\partial \phi_p} - \frac{R_{ik}}{\left(\sum_{j=1}^{M} R_{jk}\right)^2}\frac{\partial}{\partial \phi_p}\left\{\sum_{j=1}^{M} R_{jk}\right\}\right) \tag{8}$$

Since ϕ_p is defined as a parameter in the p^{th} model it will only influence the response of that model resulting in non-zero partial derivitives. Its derivatives with respect to all other models will be zero, These zero terms considerably simplify equation (8) allowing the partial derivitive of P_G to be expressed as,

$$\frac{\partial P_G}{\partial \phi_p} = \frac{1}{\sum_{j=1}^{M} R_{jp}}\frac{\partial R_{pp}}{\partial \phi_p} - \sum_{k=1}^{M}\frac{R_{ik}}{\left(\sum_{j=1}^{M} R_{jp}\right)^2}\frac{\partial R_{pk}}{\partial \phi_p} \tag{9}$$

Equation (9) provides a way of representing the partial differential of P_G in terms of the partial differentials of each of the mean MLP outputs. Using equations (3) and (2) we can express the differentials of the mean outputs in terms of the differentials of the individual density outputs within the training set as,

$$\frac{\partial R_{pk}}{\partial \phi_p} = \frac{1}{N_k}\sum_{q=1}^{N_k}\left(\frac{1}{T_q}\sum_{t=1}^{T_q}\frac{\partial}{\partial \phi_p}\left\{F\left(\underline{o}_{qt}, \underline{s}_{qt}, \Phi_p\right)\right\}\right) \tag{10}$$

The notation for F_p has been amended to reflect the fact that the MLP mapping depend jointly on its input vectors $\underline{o}$ and $\underline{s}$ and on its controlling parameter set, denoted Φ_p. A specific weight ϕ_p is a member of this weight set. The combination of equations (9) and (10) therefore allow us to represent the differential of the gross discriminative metric in terms of the differentials of the individual MLP mappings, both with respect to a specific weight within the MLP. Minor modifications to the standard BEP algorithm allow us to derive the differential of a specific MLP output in terms of a one of its controlling parameters. The HCDM is trained by alternating between parameter re-estimation and re-segmentation in an identical fashion to the HCNN and HCLP.

5.3 Recognition using the HCDM.
Training is complete when the weight parameters controlling the MLP density mappings associated with each class have been estimated. The alignment sequences used during training are not saved since these are unique to the utterances in the training set. During recognition a separate response value r_{jk} is calculated to an unlabelled utterance for each of the possible classes, j, using the Viterbi algorithm. The unknown utterance is labelled as the class which has the greatest response k, since the CDL training aims to ensure that $r_{kk} \geq r_{jk} \quad \forall j$. The HCDM was tested on a subset on the S1 corpus and compared with identical HCNN and HMM tests. Classes 'A', 'B' and 'C' only were used in the comparison. The HCNN provided the poorest test set accuracy of 63.8 ± 4.5%, the HCDM was significantly better at 85.5 ± 3.3% whilst the HMM performed best with 93.3 ± 2.3%. The training set results of 66.2%, 87.6% and 94.4% respectively, suggest all three classifiers were providing a reasonable level of generalisation.

6.0 CONCLUSIONS

The HCNN was proposed by Levin as a way of explicitly modelling time variability in speech whilst using a neural net classifier. The system relies on the idea that each speech state should be characterised by a unique frame to frame prediction function which is modelled by the HCNN's MLP predictor.

This paper has sought to compare the performance of HCNN and HMM classifers for recognition of speech in the S1 corpus and it has been shown that the HCNN is very inferior to the HMM. Insight into the reasons was gained by conducting recognition tests using a set of switched linear predictors in place of the MLPs in the HCNN. This system is called the Hidden Control Linear Predictor (HCLP), and the very poor recognition performance it provides suggest that speech states cannot be adequately characterised by frame to frame prediction functions.

In view of these experimental results, it is concluded that predictive classifiers, such as the HCNN and HCLP, are unsuitable for the recognition of speech and an alternative neural net classifier called the Hidden Control Density Mapper (HCDM) has been proposed and tested. The HCDM has a similar structure to the HCNN but uses its MLPs to model the observation probabilities instead of acting as a predictors. This sytem provides recognition performance approaching that of an HMM and it is believed that further development may improve its performance still more.

7.0 REFERENCES

[1] E LEVIN, 'Word Recognition using Hidden Control Neural Architecture', Proc. IEEE ICASSP-90, Paper S8.6, pp 433-436.
[2] G D FORNEY JR, 'The Viterbi algorithm', Proc. of the IEEE, pp 268-278, Vol. 61, No. 3, March 1973.
[3] A W DRAKE, 'Discrete-state Markov processes.', Chapter five of *Fundamentals of Applied Probability Theory*. McGraw-Hill, New York, 1967.
[4] D E RUMELHART, G E HINTON, R J WILLIAMS, 'Learning representations by back-propagating errors.', Nature, Vol. 323, pp 533-536, October 1986.
[5] L GILLICK, S J COX, 'Some statistical issues in the comparison of speech recognition systems', Proc., IEEE Conf. on Acoustics, Speech and Signal Processing, Glasgow, May 1989.