COMPARISON OF NEURAL NETWORK TECHNIQUES FOR SPEAKER VERIFICATION

S. Hussain†‡, F. R. McInnes†and M. A. Jack†

†Centre for Communication Interface Research, University of Edinburgh, UK
‡Department of Electronic Engineering, Universiti Teknologi Malaysia, Malaysia

ABSTRACT - In this paper a comparison is made between two alternative approaches for speaker verification(SV) using neural network. Firstly, a vector quantization preprocessing stage was used as the front end. The preprocessing stage measures the local spectral similarities by using a vector quantizer to select the index. The indices of the winner units are fed to a second stage neural network in which the system can be trained and evaluated. Two experiments were performed. The first experiment used a neural network model(NNM) with frame labelling performed from a client codebook known as NNM-C. Better performance was obtained from this model when compared with SCHMM(Semi Continuous Hidden Markov Model). The second set of experiments used the NNM with frame labelling from the client and the impostors codebook known as NNM-CI. The results were not as good when compared with the NNM-C and SCHMM.

INTRODUCTION

Speaker verification(SV) is a process of accepting or rejecting a person's claimed identity based on a sample of speech from that person. Speaker verification vector quantization(VQ) based methods were proposed by [Soong *et al.*, 1985][Rosenberg & Soong, 1987]. The technique of hidden Markov models(HMM's) is one of the popular methods for speech recognition and has been successfully applied to speaker recognition[de Veth & Bourlard, 1994][Che & Lin, 1995]. In recent years there is growing interest in using neural networks for speaker recognition. In one study, Hattori used a predictive neural network. It non-linearly predicts the next frame from several preceding frames. This predictive neural network is based on MLP(Multi Layer Perceptron). In a text-dependent SV an equal error rate(EER) of 1.5% was achieved for 12 male speakers[Hattori, 1994]. Oglesby and Mason have proposed using MLP and Radial Basis Function(RBF). The performance of the system is greatly affected by the training tokens as well as the network architecture. One of the main problems faced using these methods is that it requires large number of hidden units and the training can be time consuming[Oglesby & Mason, 1990][Oglesby & Mason, 1991].

For any given database there will be a variation in performance among the speakers. Some speakers have voices that are distinctive and have no difficulty with the SV systems. The false acceptance rate in this case will be low. Other speakers have difficulty using the SV systems as they might have common voice characteristics. In theory neural networks should be able to produce the desired outputs from any input representation that encodes the relevant information. In any practical cases, an optimal input representation and preprocessing is normally required for an efficient network. For each client speaker the codebooks are produced from its own training tokens. The codebook design in this case may well represent the distribution of speech features for each speaker but not discriminate well the different characteristics among different speakers. In this paper the work will examine and compare the use of :

[1.] Neural network SV that uses separate client codebooks for different digits.

[2.] Separate impostor codebooks for different digits. It is assumed that including these impostor codebooks in the training data of each network will increase the efficiency of this data to enable direct modeling of the differences between the client speaker and the impostors. For example, the client tokens that have minimum or similar values using the impostor codebooks are highlighted and vice versa.

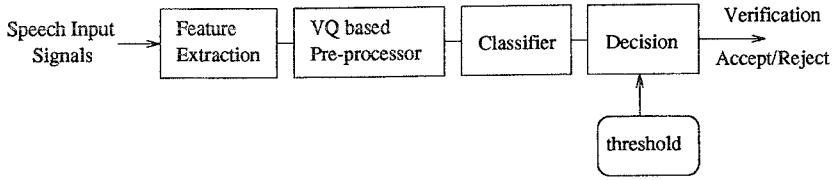

Figure 1 Basic structure of NNSV system

Since each word has its own codebook, this simplifies the vocabulary reconstruction. New words can be simply added to the system. The number of codevectors can be modified at will while other codebooks remain unchanged. Each small network can be trained separately, this gives the possibility to expand the classifier without an increase in computational effort. The preprocessing stage allows a smaller network configuration. This can eliminate the difficulties in the training phase and facilitates training on limited data. It is important to note that data preparation can make the difference between a network that performs well and a network that performs excellently. If the improvement in performance is significant then this addition of impostor codebooks would be acceptable. The preprocessing stage that makes use of codebooks from the client and the impostors is a subject of further experimentation. In practice, it is often necessary to experiment with a variety of input representations.

This paper is organised as follows. The next section describes the SV system. The remaining sections describe the database, report the results of the experiments and give a conclusion.

SPEAKER VERIFICATION

The SV system is made up of two phases : the preprocessing vector quantization(self-organization network) and the MLP classifier. A self-organization network consists of the input layer and the output competitive layer. In this network, the codewords W_i associated with the units are initialized to small random values. The training algorithm iterates by selecting the winning unit N_i and adjusting the weight W_i after each presentation of the input vector $\mathbf{X}$. The classifier system is based on a three layer perceptron. This network is trainable with the back-propagation algorithm. The verification system is shown in Figure 1. The initial stage used the commonly used feature set, cepstral coefficients for the speaker modelling stage as the speech signals. In NNM-C, the output of the preprocessor for each time frame would contain the index j of the codevector with the minimal distortion and the corresponding distortion value d. In NNM-CI, there are two pairs(j,d) per frame, one for the client codebook and one for the impostor codebook. The input pattern is linear time normalized(LTN) either by linear compression or expansion so that the total number of frames becomes a constant regardless of the word duration. Through this preprocessing, the highly redundant speech data are reduced so that only the useful information regarding codevector and the distance measure is retained in the feature vector to feed the MLP. For example, if the number of frames after LTN=40, two coefficients per frame will fit the 80 input units. By using the client codebook and the impostor codebook each hidden unit is fed with 160 input units resulting in an architecture of 160-N-1 where N is the number of hidden units. The output unit is trained to respond with a high value(0.9) output for the desired speaker and a low value(0.1) for the other speaker data. Separate nets were trained for each of the 12 digits for each of the 11 speakers.

SPEECH DATA

The database consists of the isolated digits from a large number of speakers. Twelve isolated digits(digits 'one' to 'nine' plus 'zero','nought' and 'oh') were used in the experiments. A group of 11 speakers are modelled by the system and an independent set of 83 impostor speakers is used for testing. The data are all end-point detected to remove excess silence and minimize storage requirements. The framesize was 20ms with 15ms overlap. The training templates consisted of 5 tokens from the client speaker and 19 from impostors who were different from the impostors used in testing. The templates from the target group and the impostor group were alternated in the training set. The implemented verification system used another set of data (not used during training) for further evaluation of its performance. It was tested on 20 true speaker tokens and 83 impostor tokens for each digit for each speaker.

RESULTS

The global success rate of the verification system is evaluated as the mean value of the success rates obtained with each speaker. Given the test token, it can be accepted or rejected if the output score is respectively greater or lower than the predetermined threshold. Thus, two kind of errors are evident: falsely reject(FR), when the speaker is a client but wrongly rejected and falsely accept(FA), when an impostor is mistaken for the client. In the evaluation of the verification system the use of equal error rate(EER) thresholds means that all thresholds are determined a posteriori. This approach sets the proportion of FA equal to the proportion of FR resulting in the said EER. In addition to the EER results based on single digits, results are also quoted for error rates that combined the results for all the twelve digits. Combining the scores from several digits can further improve the SV performance. The combination of these scores is commonly referred to as using digit sequence.

Figure 2 and Figure 3 show the average EER for both SV models for the 5 training tokens. EER for the single digits ranged from 7% to 16% for the NNM-C and 13% to 24% for the NNM-CI. The EERs for the 12 digit sequences shown in Figure 4 were 1.04% (NNM-C) and 3% (NNM-CI). The NNM-C performs better than the NNM-CI in all the digits. The performance of the NN models is linked with the hidden layer of units with its ability to generalize. The knowledge which is stored in the hidden layer is abstracted from the information contained in the input patterns. Each hidden unit will respond to the different input patterns presented to it. The NNM-CI design works but is not optimal for two reasons. First, in this model the decision surface may be more complex due to the complexity of the input to the classifier. On the other hand, NNM-C has a simpler feature vector representation which increases the chance of convergence to a working set of connection weights. Second, the input layer has the largest number of connections and the first hidden layer is often the largest computing layer in the network. This may require more training data before settling to a set of weights that would provide a good generalization. Both these methods were trained with equivalent amount of training data.

COMPARISON WITH THE ESTABLISHED TECHNIQUE HMM

The performance of neural networks and HMM(Hidden Markov Models) varies strongly with the amount of training tokens available. The number of training tokens needed varies according to the network structure and the input data. It seems that the larger and more complex the input space of a particular pattern the more training tokens will be required [Maren *et al.*, 1990]. As noted before in speaker verification applications for telephone speech it is important to be able to construct models with the least amount of data. Five training tokens is common among researchers [Rosenberg *et al.*, 1990]. This is based on the minimum amount of data required to train a reliable HMM SV system. When comparing the work carried out in this paper with other published results it is important to consider that different systems are also trained on a limited number of training tokens and the performances of the systems are evaluated on the same data base. Comparisons of different systems which use different amount of training data and different data base are not very meaningful. In view of this, a comparison is made with the established technique of HMM[Forsyth *et al.*, 1993] applied to the same data base. For DHMM(Discrete Hidden Markov Models), models with 3 and with 6 states were constructed for each digit. SCHMM(Semi Continuous Hidden Markov Model) with 6 states was constructed for each digit. Both of these models are trained with 5 and 10 training tokens and tested with 10 true client tokens and 95 or 100 impostor tokens for each digit for each speaker. NNM-CI and NNM-C SV on the other hand was trained with 5 tokens and tested on 20 true client tokens and 83 impostor tokens. EER for individual digits for DHMM ranged from 12%-28% for the 5 token models and 8%-17% for the 10 token models. Average EER of 14%(DHMM) and 12%(SCHMM) were achieved for single isolated digits and 4%(DHMM) and 2%(SCHMM) for a sequence of 12 isolated digits for the 10 token models. NNM-C SV experiments, with 40LTN produce an EER of 1.04% compared to 4% for DHMM trained with 10 tokens. NNM-C using 5 training tokens produced an average EER of 13% compared to 18.7%(DHMM) and 14.3%(SCHMM) for single isolated digits. One difference in Forsyth's system from the NNM is that a standard codebook is used for all speakers and for all digits instead of different codebooks for all speakers and digits. Another important difference is that HMM just models the client data whereas NNM is trained to discriminate between client and impostor data. Some indication of the effect of different systems for the verification tasks can be gained from the results. As mentioned earlier the NNM-C for the 12 digit sequence using 5 training tokens per digit produced an EER of 1.04%. This compares favourably with the 2% EER of Forsyth's system.

Despite these results the full benefits of the neural network approach have not yet been utilized for verification. Preliminary results from client A and B have shown that selecting the best LTN from

different digits resulted in better performance on sequence of the 12 digits. The lengths of the inputs after LTN are 30, 40, 50 and 60. For speaker A there is small variation of EER performance with different sizes of LTN, however selecting the best LTN from different digits results in better performance of the sequence of the 12 digits. Results from speaker B show significant variation of performance from different values of LTN. Further improvement can be seen after combining the different values of LTN. The usefulness of this approach will be investigated in future and the performance of the neural network SV approach can be expected to improve.

CONCLUSIONS

Two methods for creating vector sequences were investigated: these methods make use of a VQ based preprocessor which reduced the amount of input to be fed to the MLP classifier. This can speed the learning process and facilitates training on limited data. Initially it was assumed with the addition of the impostor codebook this would increase the efficiency of the training data as well as its performance. However, it is reasonable to conclude from the results that NNM-C model should be used in preference to NNM-CI model when training with limited data. Using the NNM-C also means the amount of input to be fed to MLP is reduced by 50%.

References

[Che & Lin, 1995] Che, C., & Lin, Q. 1995 (September). Speaker Recognition Using HMM With Experiments On the YOHO Database. *Pages 625–628 of: Procedings of the European Conference on Speech Technology.*

[de Veth & Bourlard, 1994] de Veth, J., & Bourlard, H. 1994 (April). Comparison of Hidden Markov Model Techniques for Automatic Speaker Verification. *Pages 11–14 of: ESCA Workshop On Automatic Speaker Recognition Identification And Verification.*

[Forsyth *et al.*, 1993] Forsyth, M., Sutherland, A., Elliot, J., & Jack, M. 1993. HMM speaker verification with sparse training data on telephone quality speech. *Pages 411–416 of: Speech Communication*, vol. 13.

[Hattori, 1994] Hattori, Hiroaki. 1994 (April). Text-Independent Speaker Verification Using Neural Networks. *Pages 103–106 of: ESCA Workshop On Automatic Speaker Recognition Identification And Verification.*

[Maren *et al.*, 1990] Maren, A.J, Harston, Craig, & Pap, Robert M. 1990. *Handbook of Neural Computing Application.* Academic Press.

[Oglesby & Mason, 1990] Oglesby, J., & Mason, J. S. 1990 (April). Optimization of Neural Models For Speaker Identification. *Pages 261–264 of: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 1.

[Oglesby & Mason, 1991] Oglesby, J., & Mason, J. S. 1991 (May). Radial Basis Function for Speaker Recognition. *Pages 393–396 of: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 1.

[Rosenberg & Soong, 1987] Rosenberg, A., & Soong, F. 1987 (September). Evaluation of a Vector Quantization Talker Recognition System in Text Independent and Text Dependent Modes. *Pages 143–157 of: Computer Speech and Language*, vol. 22.

[Rosenberg *et al.*, 1990] Rosenberg, A., Lee, C., & Soong, F. 1990 (April). Sub-word Unit Talker Verification Using Hidden Markov Models. *Pages 269–272 of: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 1.

[Soong *et al.*, 1985] Soong, F., Rosenberg, A., & Juang, L. Rabiner B. 1985 (March). A Vector Quantization Approach To Speaker Recognition. *Pages 387–390 of: Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 1.

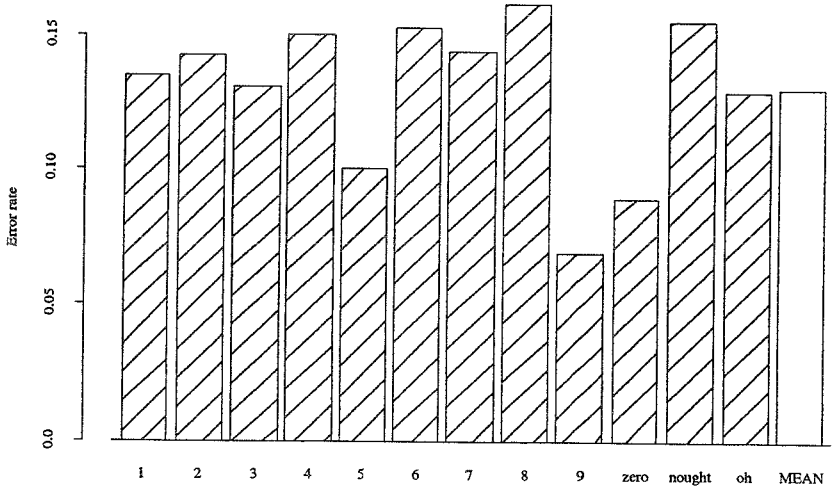

Figure 2: Digit EER (NNM-C, 5 training tokens)

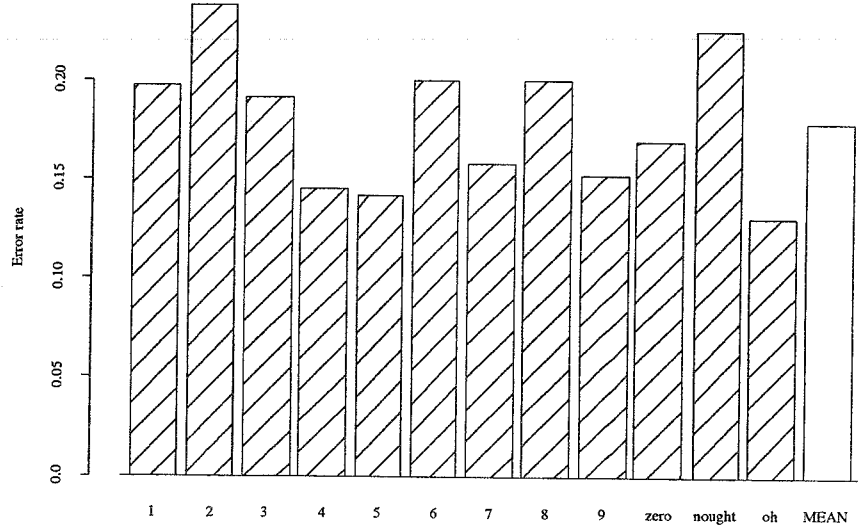

Figure 3: Digit EER (NNM-CI, 5 training tokens)

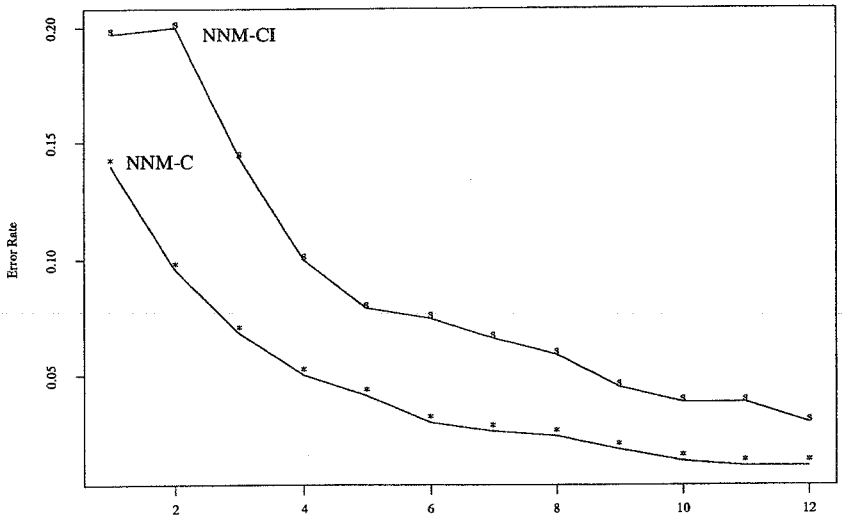

Figure 4: Digit Sequence EER

(The Improvement of NNM-C over NNM-CI For Various Digit Sequence Lengths)