Selective Adaptation of Speaker Characteristics within a Subcluster Neural Network

S.J.Haskey and S.Datta

Electronic and Electrical Engineering Department
Loughborough University, Loughborough, LE11 3TU, U.K.
S.Datta@lboro.ac.uk

ABSTRACT

This paper aims to exploit inter/intra-speaker phoneme sub-class variations as criteria for adaptation in a phoneme recognition system based on a novel neural network architecture.

Using a subcluster neural network design based on the One-Class-in-One-Network (OCON) feed forward subnets, similar to those proposed by Kung [2]and Jou[1], joined by a common front-end layer, the idea is to adapt only the neurons within the common front-end layer of the network. Consequently resulting in an adaptation which can be concentrated primarily on the speakers vocal characteristics. Since the adaptation occurs in an area common to all classes, convergence on a single class will improve the recognition of the remaining classes in the network.

Results show that adaptation towards a phoneme, in the vowel sub-class, for speakers *MDAB0* and *MWBT0* improve the recognition of remaining vowel sub-class phonemes from the same speaker.

INTRODUCTION

Inter/intra speaker variations can cause significant problems with speaker independent recognition systems. Variations such as vocal tract length and dialect differences from speaker to speaker or the intonation, rhythm or stress variations from the same speaker. To over come this problem it is necessary to have a recognition system, that has been trained with utterances from a representative subset of speakers, to dynamically adapt after an initial correct utterance, latching onto the new speakers vocal characteristics.

Adaptation of conventional connectionist architectures generally involves network-wide weight changes. This is undesirable for the purposes of phoneme recognition due in part to computational inefficiency, but mainly due to the fact that the network will be susceptible to cross-class interference.

The main objectives of the new neural network architecture were to avoid cross-class interference during adaptation towards a phoneme class and to separate the phoneme information from the speaker information within the network. Separation of speaker and phoneme information would allow adaptation to be concentrated purely on speaker variations, reducing the need for network-wide adaptation. An assumption however is made that for similar dialects, inter speaker phoneme sub-class variations are roughly constant, i.e. vowel sound differences from speaker to speaker are consistent for all vowel sounds.

NETWORK ARCHITECTURE

Unlike conventional subnet structures (OCONs)[1][2] , fig 1, this new neural network architecture consists of OCONs, one for each phoneme class, joined by a common front-end adaptation layer, fig 2. Each OCON structure consists of a fully connected two layered network with a single output neuron. The adaptation layer fully connects to each of the OCON structures and in turn fully connects to the input layer. All the neurons within the network use the sigmoidal activation function and the weights of each connection are trained using the back-propagation algorithm [5].

After the network is initially trained with speech data it is assumed that all class specific information unique to that phoneme is stored in the relevant OCON subnet and that information common to all the classes, such as speaker information, is stored within the weights of the common front-end adaptation layer. When the network is introduced with speech data from a new speaker the score at the output is computed in much the same way as a conventional network. All the OCON outputs connect to a MAXNET[1] which finds the highest score, as long as it exceeds a minimum threshold level, which is assumed to be the correct utterance. Using back-propagation, the error is then fed back through the OCON structure to the front-end adaptation layer where the weights are adapted to minimise the error. As each new speaker uses the system the updated adaptation weights are reset to their initial values ready for adaptation towards the next speaker.

By concentrating the adaptation only on this front layer it is expected that only information unique to the speaker will change, resulting in a more efficiently controlled application-driven (speech recognition) connectionist regime. (Since the adaptation occurs in an area common to all classes, it is envisaged that convergence on a single class will improve the recognition of the remaining classes in the network, for

the same speaker, by eliminating the need to update each class for full adaptation to take place.)

ADAPTATION PROCEDURE

Forward Pass

When confronted with an utterance from a new speaker the output score is computed in much the same way as any conventional neural network.

Define:
I : Network Input.
A_j: Output of the j-th adaptation neuron.
ω_{ij}: the weights from the i-th input neuron to the j-th adaptation neuron.
$\bar{\omega}_{jk}$: the weights from the j-th adaptation neuron to the k-th hidden neuron in the subnet m.
$\hat{\omega}_k^{[m]}$: the weights from the k-th hidden neuron to the output neuron in the subnet m.
θ_j: the bias of the j-th adaptation neuron.
$\bar{\theta}_k^{[m]}$: the bias of the k-th hidden neuron in the subnet m.
$\hat{\theta}^{[m]}$: the bias of the output neuron in the subnet m.
$H_k^{[m]}$: Output of the k-th hidden neuron in the subnet m.
$O^{[m]}$: Output from the subnet m.
η : Learning rate of the adaptation layer.

Each neuron uses the sigmoidal activation function. Therefore the output of the j-th adaptation neuron is :

$$A_j = f\left(\sum_i \omega_{ij} \cdot I_i\right)$$
$$= 1/\left[1+\exp-\left(\sum_i \omega_{ij} \cdot I_i + \theta_j\right)\right]$$

Using the values of A_j the output of the k-th hidden neuron of the m-th subnet is calculated according to :

$$H_k^{[m]} = f\left(\sum_j \bar{\omega}_{jk}^{[m]} \cdot A_j\right)$$
$$= 1/\left[1+\exp-\left(\sum_j \bar{\omega}_{jk}^{[m]} \cdot A_j + \bar{\theta}_k^{[m]}\right)\right]$$

Finally, using the output of the hidden layer, $H_k^{[m]}$, from each corresponding subnet the output of the m-th subnet is :

$$O^{[m]}(I) = f\left(\sum_k \hat{\omega}_k^{[m]} \cdot H_k^{[m]}\right)$$
$$= 1/\left[1+\exp-\left(\sum_k \hat{\omega}_k^{[m]} \cdot H_k^{[m]} + \hat{\theta}^{[m]}\right)\right]$$

The outputs from each OCON subnet are fed through a MAXNET to find the winner, assuming the highest score achieves a minimum threshold score.

Backward Pass

Now we begin the back pass of the back-propagation algorithm to adapt the weights and bias values of the adaptation layer. Firstly we need the error E of each of the m subnets to feed-back. The error is :

$$E^{[m]}(I) = \left(T - O^{[m]}(I)\right)$$

where T is the target values.

If the input pattern I belongs to the m-th subnet then the target T is 1. Otherwise T is 0.

For the sigmoidal activation function, the error signal ,$\hat{\delta}^{[m]}$, for the output of the hidden layer is given by :

$$\hat{\delta}^{[m]} = E^{[m]}(I) \cdot O^{[m]}(I) \cdot \left(1 - O^{[m]}(I)\right)$$

Feed-back this error through now to the hidden neuron:

$$\bar{\delta}_k^{[m]} = H_k^{[m]}\left(1 - H_k^{[m]}\right) \cdot \sum_k \hat{\delta}^{[m]} \hat{\omega}_k^{[m]}$$

Feed-back this error through to the adaptation layer, adding the errors from all the OCON subnets.

$$\delta_j = \sum_m \left\{ A_j\left(1 - A_j\right) \sum_j \bar{\delta}_k^{[m]} \bar{\omega}_{jk}^{[m]} \right\}$$

Now we have fed back the errors through the whole network we can modify the adaptation weights and bias values using the following:

$$\Delta\omega_{ij} = \eta \cdot \delta_j \cdot I$$

$$\Delta\theta_{ij} = \eta \cdot \delta_j$$

As the adaptation weights and bias values are modified, the old weight and bias values are stored so that the adaptation layer can be reset for each new user.

RESULTS

The main objective was to monitor the improved recognition rates of every phoneme class within the neural network after adaptation towards a single phoneme class from the same speaker.

Since we made the assumption concerning inter speaker phoneme sub-class variations remaining roughly constant, all the training data was from one phoneme sub-class, the vowel sub-class, of the DARPA TIMIT database. From this sub-class, 8 phonemes */ix, iy, eh, ah, ax, ih, ey, aa/* from 24 male speakers from dialect region one were used to train the network. The back-propagation algorithm was used for training, with all the weights within the network initially randomised, along with the order of the speech training data, to maximise convergence

The network consisted of 8 OCON subnet structures, one for each of the phoneme classes, all having a single output and containing a 15 neuron fully connected hidden layer. The 8 hidden layers from each of the OCONs were fully connected to the 15 neuron adaptation layer which in turn was fully connected to the 75 neuron input layer. The input data comprised of the sampled phonemes being split into 15 overlapping hamming windows, each of which was represented by 5 linear predictive coefficients [6].

The test set for the experiment contained utterances of the 8 selected vowel sub-class phonemes spoken by 2 male speakers (*DAB0, WBT0*) from the same dialect region as the training set. Initial recognition rates were noted for all the 8 phonemes from both speakers before adaptation began.

The adaptation procedure involved adapting the network towards a phoneme by feeding back any errors through the network and using these to modify the weights and bias values within the adaptation layer. After adaptation, recognition rates of the phonemes uttered by the same speaker were recorded and any variation calculated. After adaptation towards a phoneme the average recognition rate of that phoneme increased by 16.5% for speaker *DAB0* and by an average 20.2% for speaker *WBT0*. As for the average recognition rates of the remaining phonemes, for speaker *DAB0* the rate increased by an average 8.3% and for speaker *WBT0* by an average 9.8%. See Table 1 and 2 for speakers *DAB0* and *WBT0* respectively.

After each test, the adaptation weights and bias values were reset.

CONCLUSION

It can be seen from Table 1 and Table 2 that adaptation towards a phoneme, in the vowel sub-class, for speakers *MDAB0* and *MWBT0* can indeed improve the recognition of the remaining phonemes from the same speaker. After adaptation towards a phoneme the average recognition rate of that phoneme increases by 18.35% and the recognition rate of the remaining phonemes increases by 8.9%. This highlights the idea of speaker information being stored in the common front end adaptation layer, resulting in a more efficient adaptation system. Further tests need to be applied to other phoneme sub-classes such as stops and fricatives and at present, adaptation itself is still slow. This is because only simple back propagation is being used, although faster existing adaptation techniques can be applied to the same architecture

REFERENCES

(1) I. C. Jou, Y. J. Tsay, S. C. Tsay, Q. Z. Wu, and S.S. Yu. **Parallel distributed processing with multiple one-output back-propagation neural networks.** Proceedings, International Symposium on Circuits and Systems, Singapore, pp 1408-11, 1991.

(2) S. Y. Kung, J. S. Taur **Decision-Based Neural Networks with Signal/Image Classification Applications.** IEEE Transactions on Neural Networks, Vol 6, No 1, pp170-81, 1995.

(3) R. P. Lippmann **An Introduction to Computing with Neural Nets.** IEEE ASSP Magazine, April 1987, pp 4-22.

(4) R.P.Lippmann **Review of Neural Networks for Speech Recognition.** Neural Computation 1, pp 1-38, 1989.

(5) D.Rumelhart, J.McClelland **Parallel Distributed Processing.** Cambridge, MIT Press, 1986.

(6) J.Makhoul **Linear Prediction: a tutorial review**, Proc IEEE, Vol 63, No 4, pp 561-580, April, 1975.

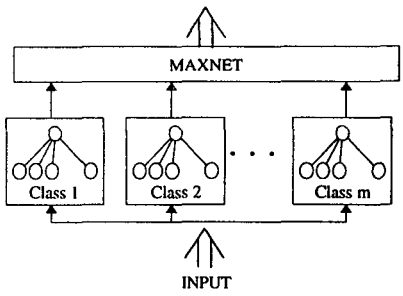

Fig. 1: Conventional OCON Architecture

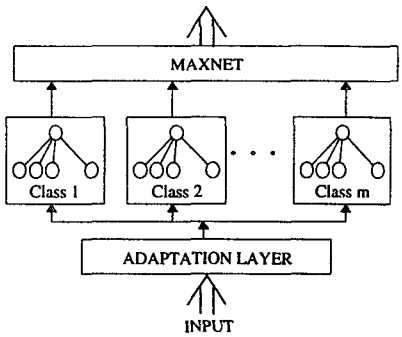

Fig 2: OCON Architecture with Common Front-End Adaptation Layer.

Adapted Phonemes

Tested Phonemes	ix	iy	eh	ah	ax	ih	ey	aa
ix	+13.3%	+13.3%	+13.3%	0	0	+13.3%	+6.6%	0
iy	+7.1%	+21.4%	+7.1%	+7.1%	0	+21.4%	+7.1%	+7.1%
eh	0	+11.1%	+11.1%	0	0	+11.1%	+11.1%	0
ah	+14.3%	+14.3%	0	+28.6%	+14.3%	+28.3%	+14.3%	0
ax	+10%	+10%	0	+10%	+20%	+20%	+10%	+10%
ih	+12.5%	+12.5%	0	0	+12.5%	+12.5%	+12.5%	0
ey	+25%	0	+25%	+25%	+25%	+25%	+25%	0
aa	0	0	0	0	0	0	0	0

Table 1: Recognition Results Using Speaker *DAB0*

Adapted Phonemes

Tested Phonemes	ix	iy	eh	ah	ax	ih	ey	aa
ix	+18.2%	+18.2%	+18.2%	+9.1%	0	+18.2%	+9.1%	0
iy	+12.5%	+37.5%	+12.5%	+12.5%	0	+37.5%	+12.5%	+12.5%
eh	0	+10%	+10%	0	0	+10%	+10%	0
ah	+9.1%	+9.1%	0	+18.2%	+9.1%	+18.2%	+9.1%	0
ax	+12.5%	+12.5%	0	+12.5%	+25%	+25%	+12.5%	+12.5%
ih	+11.1%	+11.1%	0	0	+11.1%	+11.1%	+11.1%	0
ey	+16.7%	0	+16.7%	+16.7%	+16.7%	+16.7%	+16.7%	0
aa	+25%	+25%	+25%	0	0	0	0	+25%

Table 2: Recognition Results Using Speaker *WBT0*