ADVANCED TRAINING METHODS AND NEW NETWORK TOPOLOGIES FOR HYBRID MMI-CONNECTIONIST/HMM SPEECH RECOGNITION SYSTEMS

Christoph Neukirchen *Gerhard Rigoll*

Department of Computer Science
Faculty of Electrical Engineering
Gerhard-Mercator-University Duisburg, Germany
e-mail: {chn,rigoll}@fb9-ti.uni-duisburg.de

ABSTRACT

This paper deals with the construction and optimization of a hybrid speech recognition system that consists of a combination of a neural vector quantizer (VQ) and discrete HMMs. In our investigations an integration of VQ based classification in the continuous classifier framework is given and some constraints are derived that must hold for the pdfs in the discrete pattern classifier context. Furthermore it is shown that for ML training of the whole system the VQ parameters must be estimated according to the MMI criterion. A novel training method based on gradient search for Neural Networks that serve as optimal VQ is derived. This allows faster training of arbitrary network topologies compared to the traditional MMI-NN training. An integration of multilayer MMI-NNs as VQ in the hybrid discrete HMM based speech recognizer leads to a large improvement compared to other supervised and unsupervised single layer VQ systems. For the speaker independent Resource Management database the constructed hybrid MMI-connectionist/HMM system achieves recognition rates that are comparable to traditional sophisticated continuous pdf HMM systems.

1. INTRODUCTION

In former work (see [5]) it has been shown, that hybrid speech recognition systems which use Neural Networks (NN) as labelers combined with discrete output Hidden Markov Models (HMM) compare well to other complex systems. These hybrid systems use a special purpose kind of Neural Network that is trained on the maximum mutual information (MMI) objective function and serves as a vector quantizer (VQ) for the HMMs. So the network output labels (called $\hat{M}$) provide as much information about the phonetic classes (called W) as possible. In particular the MMI-NN/HMM hybrid has shown nearly similar recognition rates as other State-of-the-Art multi-mixture continuous density systems for the Resource Management (RM) speaker independent continuous speech recognition task [5].

However, there were several drawbacks in hybrid system training: Although it is well known that the usage of hidden units can improve the performance of NNs in general (see e.g. [2]), the system described in [5] did not have any hidden nodes. This is mainly due to the facts that is has turned out: i) it is nearly impossible to train a multilayer MMI-Network in reasonable time on standard hardware, ii) the current training technique just allows a very specific network topology that was used in [3]. In contrast to usual NN training, that is commonly based on gradient search, the traditional MMI-NN training scheme (see e.g [5]) is a blind weight change by trail-and-error. The training algorithm steps through the network weights and changes each of them by a fixed offset. If this change leads to an increase of the objective function (i.e. $I(\hat{M}, W)$), the change is accepted, otherwise it is discarded. This causes a lot of unnecessary and time consuming weight changes and entropy calculations that grow the larger the networks are. Furthermore the type of network output layer was restricted to some kind of LVQ-like activation function that looks for the minimum Euclidean distance between the network weights and the input vector.

Hence a novel training algorithm had to be developed based on gradient descent, that improves training time and allows arbitrary kinds of NN topologies as shown in this paper. The main problem with gradient methods here is, that it is difficult to calculate finite but non-zero derivatives of the maximum/minimum search needed at the output nodes of a Winner-Takes-All Network (as used in VQs and the MMI-NN). So a soft and differentiable decision function is used at the output nodes of the new proposed NNs. This leads to a novel NN training algorithm that does not need target values for the output nodes (unlike other supervised training methods of NNs), while all well known acceleration methods (like Quickprop or second order methods) for weight update of classification networks can be applied. In addition any kind of network topology (with hidden nodes) can be trained now and even if the feature extraction module is added as a first layer to the NN, the parameters of the feature extraction can be optimized simultaneously.

In the following, at first a theoretical integration of discrete models in the statistical pattern recognition (using continuous models) framework is given, and it is shown how VQ parameters have to be estimated for different classifier system training paradigms. Then we show how a gradient based optimal NN training algorithm can be derived using these foundations. And finally some experimental results for the RM database obtained with multilayer MMI-NN systems trained by the novel algorithm are given.

2. OPTIMAL PARAMETER ESTIMATION FOR DISCRETE PATTERN CLASSIFIERS

In statistical pattern recognition, likelihood based modelling of class dependent probability density functions (pdf) $p(\mathbf{x}|w)$ is very common for classifier design. Here $\mathbf{x}$ denotes a (continuous) feature vector and w denotes a pattern class (e.g. a phoneme or an HMM state in speech recognition). A bayes classifier also needs a-priori information $P(w) = P_w$ about these classes that does not depend on the observation $\mathbf{x}$ (in the following the P_w are assumed fixed). A very general choice for $p(\mathbf{x}|w)$ is a mixture of J_w different parametric

continuous basic pdfs $p(\mathbf{x}|m_j, w)$ (e.g. Gaussians):

$$p_\theta(\mathbf{x}|w) = \sum_{j=1}^{J_w} p_\theta(\mathbf{x}|m_j, w) \cdot P_\theta(m_j|w) \quad (1)$$

In speech recognition this is called a *continuous model* with $P_\theta(m_j|w)$ as the mixture weights. θ denotes the set of system parameters that are estimated during training. Sharing of the basic densities among all different pattern classes leads to the *semi-continuous (tied-mixture) models*:

$$p_\theta(\mathbf{x}|w) = \sum_{j=1}^{J} p_\theta(\mathbf{x}|m_j) \cdot P_\theta(m_j|w) \quad (2)$$

It must be noted, that in both cases above the class-independent pdf $p_\theta(\mathbf{x})$ is automatically also determined when model assumptions about the $p_\theta(\mathbf{x}|m_j, ..)$ are made (since $p_\theta(\mathbf{x}) = \sum_w p_\theta(\mathbf{x}|w) \cdot P_w$). Hence if $p_\theta(\mathbf{x}|m_j, ..)$ is a Gaussian, $p_\theta(\mathbf{x})$ is a mixture of Gaussians.

In *discrete modelling* the continuous feature space is subdivided by a VQ into J different regions (partitions) associated with the discrete labels m_j $(1 \le j \le J)$; the VQ parameters (e.g. centroids, NN weights) are contained in the parameter set θ. The label of the actual VQ region the current feature vector $\mathbf{x}$ is in, is called $\hat{m}_\theta(\mathbf{x}) \in \{m_1, .., m_J\}$. The likelihood pdf of the discrete model can be derived from eqn. (2) by using $p_\theta(\mathbf{x}|m_j) = \frac{P_\theta(m_j|\mathbf{x}) \cdot p_\theta(\mathbf{x})}{P_\theta(m_j)}$. Here $P_\theta(m_j|\mathbf{x})$ is the probability of $\mathbf{x}$ being in the j-th VQ partition, that is given by $\delta_{m_j, \hat{m}_\theta(\mathbf{x})}$ in the non-fuzzy-VQ case, and thus the continuous pdf associated with the j-th VQ partition is given by:

$$p_\theta(\mathbf{x}|m_j) = \begin{cases} \frac{p_\theta(\mathbf{x})}{P_\theta(m_j)} & \text{if } m_j = \hat{m}_\theta(\mathbf{x}) \\ 0 & \text{else} \end{cases} \quad (3)$$

In this case any class independent pdf $p_\theta(\mathbf{x})$ that holds the condition:

$$P_\theta(m_j) = \int_{m_j = \hat{m}_\theta(\mathbf{x})} p_\theta(\mathbf{x})\, d\mathbf{x} \quad \forall j \in \{1, .., J\} \quad (4)$$

(to make eqn. (3) become a real pdf) can be chosen. Hence in opposite to the (semi-)continuous models, the assumptions made about $P_\theta(m_j|\mathbf{x})$ in the VQ model case do not directly determine the form of the class independent pdf. Using eqn. (3) and eqn. (2), the class dependent continuous pdf for the discrete model case is given by:

$$p_\theta(\mathbf{x}|w) = \frac{p_\theta(\mathbf{x})}{P_\theta(\hat{m}_\theta(\mathbf{x}))} \cdot P_\theta(\hat{m}_\theta(\mathbf{x})|w) \quad (5)$$

From eqn. (5) follows that in the VQ case the modelled class dependent pdf $p_\theta(\mathbf{x}|w)$ is piecewise proportional to the class independent pdf $p_\theta(\mathbf{x})$ for all different classes w with $\frac{P_\theta(\hat{m}_\theta(\mathbf{x})|w)}{P_\theta(\hat{m}_\theta(\mathbf{x}))}$ as weighting factor. This is a constraint that may limit the modelling power of discrete models. On the other hand, since $\frac{p_\theta(\mathbf{x})}{P_\theta(\hat{m}(\mathbf{x}))}$ in eqn. (5) does not depend on w (i.e. it is equal for all different classes), during classification (when the $P_w \cdot p_\theta(\mathbf{x}|w)$ are compared) it can be omitted in the calculation of the $p_\theta(\mathbf{x}|w)$. Thus the recognition system actually does not need information about the assumed class independent pdf $p_\theta(\mathbf{x})$.

For classifier training the pattern samples may be used to learn the parameters of the whole discrete system, i.e. as well the discrete bayes classifier as the VQ. It is assumed that there are N feature vector samples $\mathbf{x}(n)$ $(1 \le n \le N)$ for training; the VQ partition, the n-th sample is in, is called $\hat{m}(n) = \hat{m}(\mathbf{x}(n))$. There are K different pattern classes w_k $(1 \le k \le K)$; the class of the n-th pattern sample is denoted $w(n)$. A theoretically optimal training criterion (that is discriminative by nature) is the maximization of the a-posteriori class probabilities for all samples; in the discrete model case this yields (using eqn. (5)):

$$\theta_{discrim} = \operatorname*{argmax}_\theta \prod_{n=1}^{N} \frac{p_\theta(\mathbf{x}(n)|w(n)) \cdot P_{w(n)}}{\sum_{k=1}^{K} p_\theta(\mathbf{x}(n)|w_k) \cdot P_{w_k}} \quad (6)$$

$$= \operatorname*{argmax}_\theta \prod_{n=1}^{N} \frac{P_\theta(\hat{m}_\theta(n)|w(n))}{\sum_{k=1}^{K} P_\theta(\hat{m}_\theta(n)|w_k) \cdot P_{w_k}} \quad (7)$$

As shown in [2] the widely used maximum likelihood (ML) estimation can be derived from eqn. (6) under the assumption that the class independent pdf in the denominator of eqn. (6) is not affected by training (i.e. $p_\theta(\mathbf{x}) = p(\mathbf{x})$). In general this assumption is not valid and for the discrete model case condition (4) may be violated. For the discrete system the ML estimation using eqn. (5) is given by:

$$\theta_{ML} = \operatorname*{argmax}_\theta \prod_{n=1}^{N} p_\theta(\mathbf{x}(n)|w(n)) \cdot P_{w(n)} \quad (8)$$

$$= \operatorname*{argmax}_\theta \prod_{n=1}^{N} \frac{P_\theta(\hat{m}_\theta(n)|w(n))}{P_\theta(\hat{m}_\theta(n))} \quad (9)$$

If the fixed a-priori probability P_{w_k} matches $P(w_k)$ of the training samples, both estimates θ_{ML} and $\theta_{discrim}$ are equal for the discrete classifier.

By eqn. (9) and eqn. (7) two rules for designing an optimal VQ in different training frameworks are given. To allow a simple interpretation, eqn. (9) can by transformed into: $\operatorname{argmax}_\theta(H(\hat{M}_\theta) - H(\hat{M}_\theta|W))$. This is the maximization of the mutual information $I(W; \hat{M}_\theta)$ between the stream of pattern classes W and the stream of labels $\hat{M}_\theta$ produced by the VQ.

3. TRAINING OF MMI NEURAL NETWORKS

As shown above, the parameters θ of a VQ, used in a classification system trained by ML, have to be estimated by maximizing $I(W; \hat{M}_\theta)$. This conclusion was also drawn by the authors of [4] and [5]. Since the mutual information can be rewritten as $H(W) - H(W|\hat{M}_\theta)$, VQ parameter optimization can be done by finding: $\operatorname{argmax}_\theta \left(-H(W|\hat{M}_\theta)\right) =$

$$\operatorname*{argmax}_\theta \left(\sum_{k=1}^{K} \sum_{j=1}^{J} P_\theta(w_k, \hat{m}_j) \cdot \log \frac{P_\theta(w_k, \hat{m}_j)}{\sum_{r=1}^{K} P_\theta(w_r, \hat{m}_j)} \right) \quad (10)$$

with the derivative $\frac{\partial(-H(W|\hat{M}_\theta))}{\partial P_\theta(w_l, \hat{m}_i)} = \log P_\theta(w_l|\hat{m}_i)$.

In the following, for vector quantization a Winner-Takes-All NN is utilized; the NN uses the feature vector $\mathbf{x}$ as input

and has J different output nodes with output activations called $f_j(\mathbf{x})$ $(1 \leq j \leq J)$. The parameters θ are associated with the network weights, that will be estimated by a gradient descent approach according to the MMI objective function. This NN training principle will be called the MMI-NN paradigm. During recognition the network determines the VQ partition label $\hat{m}(\mathbf{x}) = m_{\hat{j}}$ by $\hat{j} = \text{argmax}_j f_j(\mathbf{x})$. During the network training, the maximum operation is replaced by a soft approximation, due to the need of finite derivatives. So we use the Softmax function (with a quite small choice for the softness parameter T to be not too smooth) defined in [6] by:

$$O_j(\mathbf{x}) = \frac{e^{\frac{f_j(\mathbf{x})}{T}}}{\sum_{i=1}^{J} e^{\frac{f_i(\mathbf{x})}{T}}} \tag{11}$$

With the derivative $\frac{\partial O_j(\mathbf{x})}{\partial f_l(\mathbf{x})} = \frac{1}{T} \cdot O_l(\mathbf{x}) \cdot (\delta_{l,j} - O_j(\mathbf{x}))$. Since in our (sharp) case the Softmax output is nearly 1 for the maximum network output and zero for the other ones, it can be used to approximate the probabilities needed in eqn. (10) by averaging over the training samples. That yields:

$$P_\theta(w_k, \hat{m}_j) \simeq \frac{1}{N} \cdot \sum_{n=1}^{N} \delta_{w_k, w(n)} \cdot O_j(\mathbf{x}(n)) \tag{12}$$

So the gradient of eqn. (10) with respect to the NN weights θ can be written using the chain rule and the derivatives given above by:

$$\frac{\partial\big(-H(W|\hat{M})\big)}{\partial\theta} = \sum_{k=1}^{K}\sum_{j=1}^{J} \frac{\partial\big(-H(W|\hat{M})\big)}{\partial P(w_k, \hat{m}_j)} \cdot \sum_{n=1}^{N} \frac{\partial P(w_k, \hat{m}_j)}{\partial O_j(\mathbf{x}(n))} \cdot \cdot \sum_{l=1}^{J} \frac{\partial O_j(\mathbf{x}(n))}{\partial f_l(\mathbf{x}(n))} \cdot \frac{\partial f_l(\mathbf{x}(n))}{\partial\theta} \tag{13}$$

$$= \sum_{n=1}^{N}\sum_{l=1}^{J} \frac{\partial f_l(\mathbf{x}(n))}{\partial\theta} \cdot A_l(\mathbf{x}(n)) \tag{14}$$

With:

$$A_l(\mathbf{x}(n)) \sim O_l(\mathbf{x}(n)) \cdot \sum_{j=1}^{J} \log P(w(n)|\hat{m}_j) \cdot (\delta_{l,j} - O_j(\mathbf{x}(n))) \tag{15}$$

In eqn. (14) $\frac{\partial f_l(\mathbf{x}(n))}{\partial\theta}$ just depends on the structure of the NN. In principle any kind of NN topology can be used as MMI-Net (e.g. MLP, RBF, Kohonen Maps, etc.).

This novel MMI-Net paradigm differs from usual neural classification and probability estimating networks in several points: in the MMI-Net the number of output nodes J can be chosen arbitrarily and may be larger and (theoretically) even smaller than the number of pattern classes K. Furthermore there are no output target values presented to the network during training. Instead of this the network finds the optimal outputs in a self-organizing way by considering all training patterns simultaneously.

On the other hand there are some interesting relations to classical NN training algorithms: Considering a classification NN with $K = J$ output nodes and the Softmax output nonlinearity that is trained on some objective function called E (e.g. squared error criterion, cross entropy, etc.). It turns out that its gradient is also calculated by eqn. (14) and

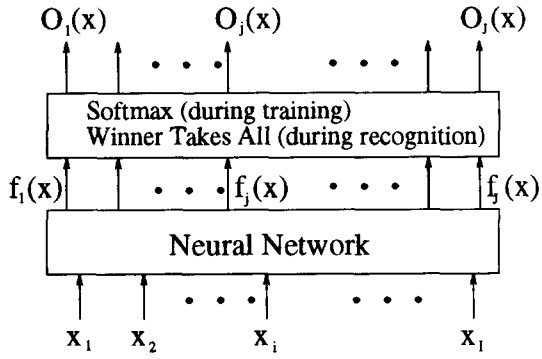

Figure 1. General structure of MMI-NN trained as optimal VQ

(15) but $\frac{\partial E}{\partial O_j(\mathbf{x}(n))}$ is used instead of $\log P(w(n)|\hat{m}_j)$. In the squared error minimization case this is $O_j(\mathbf{x}(n)) - \hat{O}_j(n)$, with the target value at the j-output node for the n-th pattern denoted $\hat{O}_j(n)$. Hence both training algorithms are quite similar and the same acceleration methods (i.e. hardware as well as software) can be applied.

4. EXPERIMENTS AND RESULTS

To investigate the behavior of the novel training method, several kinds of MMI-Nets are trained, and used as VQ in a discrete HMM speech recognition system. NN weight update is done by a method similar to the off-line RProp training (see [7]). For training we use the 3990 speaker independent sentences of the RM database. In the first experiments we extract 12 MFCC features every 10 ms, resulting in ca. 1.3 million training patterns. No power and delta-features are used in these experiments. To determine the pattern classes $w(n)$, corresponding to the feature vector frames, the database is Viterbi-aligned using a method as described in [5]. Two kinds of pattern classes are obtained by the alignment: i) monophone HMMs, ii) states of monophone HMMs.

At first a Euclidean distance NN, as used in [5], is trained using the new algorithm. It turns out that after 50 RProp iterations the objective function reaches similar values as obtained in [5] with the former training method. In terms of training time, that is a more than two times acceleration compared to the traditional MMI-NN training algorithm. In practice it turns out that the softness parameter T in the Softmax output must be chosen very small to achieve a fast training convergence. The recognition results of both systems were quite the same.

To compare the recognition performance of the NNs in the discrete HMM framework we use the official Feb'89, Oct'89, Feb'91 and Sep'92 DARPA RM SI test sets. The used models are discrete monophone HMMs for the MFCCs. Recognition is done via a beam search guided Viterbi decoder using the DARPA word pair grammar (perplexity: ca. 60). In tab. 1 the obtained results are given as the average recognition rate for the four RM test sets in the form: correct (accuracy). Tab. 1 also shows the number of input nodes, number of hidden nodes and the kind of pattern class w_k that is used in MMI-NN training. The output layer size is fixed to 200 and the input layer size varies with the number of adjacent MFCC frames that are used as NN input to capture larger context. The first two rows in tab. 1 are baseline results obtained with k-means and the former training algorithm using the Euclidean distance NN type as used in [5]. The other results demonstrate the capabilities of the

VQ type	# inp.	# hid.	w_k	Corr. (Acc.)
K-means	12	0	-	71,2% (70,3%)
eucl. NN	12	0	HMM	76,0% (74,5%)
SLP	12	0	HMM	74,7% (73,5%)
MLP	12	12	HMM	76,9% (75,7%)
MLP	36	36	HMM	80,2% (78,9%)
MLP	36	36	states	80,9% (79,7%)
MLP	60	60	states	83,3% (82,0%)

Table 1. Comparison of RM recognition rates of monophone HMMs with MFCC features using different MMI-NN topologies

novel training method with single-layer perceptrons (SLP) NN and two-layer perceptrons (MLP). It can be seen that the gradient based MMI-Net paradigm in conjunction with NNs that use hidden nodes leads to superior recognition rates compared to the former methods.

Finally the new kind of multi layer MMI-Net is integrated in the 'large' speech recognition system that was used in [5]. This system extracts 39 features consisting of 12 MFCCs, LogEnergy, plus the first and second derivatives, every 10 ms. All in all four different MMI-NNs are used as multi codebook VQ: the MLP with 3 adjacent input frames for the MFCCs as described above; 3 Euclidean distance NNs as described in [5] for the first and second MFCC-derivatives and the power with its derivatives. The output layer size of each NN is 200. The system uses 2309 context dependent HMMs to model word internal triphones and the 33 most frequent function words. All MMI-NNs are trained on the whole speaker independent part of the RM database as well as all HMMs using the forward-backward-algorithm (i.e. ML training). To overcome some problems due to insufficient training data the discrete pdfs are smoothed and the HMM states are tied. Tying is performed via a phonetically based decision tree that assigns the triphone states to several equivalence classes according to similar phonetic context and similar discrete pdfs. This enables us to generate unseen triphone states that may be used in future systems with larger vocabulary and/or cross word context HMMs.

The recognition results for this hybrid MMI-NN/HMM system are given in tab. 2 for all four used RM test sets. As a comparison the recognition results of a classical discrete k-means system are also given that uses the same kind of detailed triphone HMMs. So these two systems are quite similar, the only difference is the usage of MMI-NNs as VQ in the hybrid system case in contrast to the k-means codebooks in the other case. As tab. 2 clearly shows, on the average the hybrid systems outperforms the classical one by more than 2% (absolute). This result also compares very well to other complex systems that were tested on the RM task. In [8] another hybrid NN/HMM is described that uses simple monophone HMMs and a recurrent NN to estimate local phone posterior probabilities. This system achieves an average recognition result of 94,3% (93,4%) for the RM test sets using context independent HMMs. Results for a multi Gaussian mixture continuous pdf system are reported e.g. in [9]. The system uses word internal state-clustered triphone HMMs (similar to the MMI-NN hybrid system described here) and achieves average RM recognition results of 95,4% (94,7%) what is equal to the recognition rate given in tab. 2.

5. CONCLUSIONS

The paper gives an integration of VQ based discrete pattern classifiers into a continuous pdf modelling framework

Test set	MMI-NN/HMM	K-means system
Feb'89	**96,3% (95,6%)**	94,3% (93,6%)
Oct'89	**95,4% (94,5%)**	93,5% (92,0%)
Feb'91	**96,7% (95,9%)**	94,4% (93,5%)
Sep'92	**93,9% (92,5%)**	90,7% (88,9%)
average	**95,6% (94,6%)**	93,2% (92,0%)

Table 2. Comparison of RM SI recognition rates (Corr. (Acc.)) between context dependent hybrid MMI-NN/HMM and baseline k-means system

and it is shown that for optimal ML training of discrete classifiers, the VQ parameters have to be estimated according to the MMI objective function. A novel gradient based MMI training method is proposed for optimal neural network learning as a VQ in a discrete HMM framework. This training method is faster than a traditional MMI-NN weight optimization procedure and allows training of arbitrary hidden NN layers. Using these MMI-NNs leads to a hybrid speech recognition system that performs equal or even better than other high sophisticated continuous pdf and hybrid HMM systems. Future improvements may be obtained by training of more complex NN types (e.g. recurrent nets) and application of techniques to improve generalization as well as the integration of context dependent HMMs that allow cross-word modelling of speech.

6. ACKNOWLEDGMENTS

This work was partly supported by the DFG (German Research Foundation) under contract Ri 658/3-1. Responsibility for the content of this paper is with the authors.

REFERENCES

[1] G. Rigoll, Ch. Neukirchen, J. Rottland, "A new hybrid system based on MMI-Neural Networks for the RM speech recognition task", *Proc. IEEE-ICASSP,* 1996, pp. 865–868.

[2] N. Morgan, H. Bourlard, "Neural Networks for Statistical Recognition of Continuous Speech," *Proc. IEEE,* Vol. 83, No. 5, May 1995, pp. 742–770.

[3] G. Rigoll, "Speech recognition experiments with a new multilayer LVQ network (MLVQ)", *Proc. Eurospeech,* 1995, pp. 2167–2170.

[4] M. Osterndorf, J.R. Rohlicek, "Joint quantizer design and parameter estimation for discrete Hidden Markov Models", *Proc. IEEE-ICASSP,* 1990, pp. 705–708.

[5] G. Rigoll, Ch. Neukirchen, "A new approach to hybrid HMM/ANN speech recognition using mutual information neural networks", *Advances in Neural Information Processing Systems 9,* NIPS*96, Denver, 1996.

[6] J.S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition", *Neurocomputing: Algorithms, Architectures and Applications,* NATO ASI Series, Springer, Berlin, 1990, pp. 227–236.

[7] M. Riedmiller, H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," *Proc. IEEE-ICNN,* 1993.

[8] A.J. Robinson, "An Application of Recurrent Nets to Phone Probability Estimation", *IEEE-Trans. Neural Networks,* Vol. 5, No. 2, Mar. 1994, pp. 298–305.

[9] P.C. Woodland, S.J. Young, "The HTK tied-state continuous speech recognizer", *Proc. Eurospeech,* 1993, pp. 2207–2210.