4th WSEAS International Conference on ELECTRONICS, CONTROL and SIGNAL PROCESSING, Miami, Florida, USA, 17-19 November, 2005 (pp.121-125)

EMBEDDED CLASSIFICATION KERNEL USING SOM CLUSTERING AND MIXTURE OF EXPERTS

S.MEENAKSHISUNDARAM, S.S.DLAY AND W.L.WOO
School of Electrical Electronic and Computer Engineering,
University of Newcastle upon tyne,
Newcastle upon tyne NE1 7RU
UNITED KINGDOM

Abstract --- In this paper, we introduce a new classification kernel by embedding self organized map (SOM) clustering with mixture of radial basis function (RBF) networks. The model's efficacy is demonstrated in solving a multi-class TIMIT speech recognition problem where the kernel is used to learn the multidimensional cepstral feature vectors to estimate their posterior class probabilities. The tests results have shown that this model provides a better alternative to the state of the art models achieving a significant improvement in error performance, reduction in complexity and gain in training time.

Key words---Supervised mixture models, Mixture of Experts, and Self organizing map

1 Introduction

In machine learning, Mixture of Experts (ME), a class of supervised mixture models, learn a problem by using expert components and a gating component that soft partitions the input space. For classification problems as in automatic speech recognition (ASR), the goal of the ME is to establish a model that can predict the target label given the input pattern where the target labels may represent the labels of multiple classes. The state of art ME models used for such applications were built by using expert neural networks such as multilayered perceptron (MLP) or Recurrent Neural Network (RNN) with a gating network which can be a unsupervised model (GMM) or Linear model (GLM) or a neural network such as MLP [1, 3, 7]. In [1] authors have used ME to estimate the posterior class probabilities for speech corpus data. In this direction, we have introduced a new kernel by embedding a two staged SOM clustering [4] [10] to choose hidden unit parameters for ME framework of RBF experts. Ghosh in [6] has reported speed up of training times by embedding SOM within RBF and demonstrated it for a single dimensional problem [6]. In other direction Bin Tang [4] introduced SOM as a pre-processor to MoE to identify the MoE initial components and used them with existing MoE. However with this method experts are constrained to clusters with a one to one correspondence. In our method we utilize SOM to maximum potential by replacing hidden layer stage with SOM and the optimization layer restricted to only output layer of RBF experts instead of a two stage optimization as in MLP counterparts. This helps our kernel achieve great reduction in complexity as well as improved accuracy.

2 Classification Using Mixture of Experts

2.1. Posterior Probability Estimation through Mixture of Experts- Motivation:

Bayes theorem suggests that a classifier can be optimal if we could perfectly estimate priors $P(C_k)$ and class conditional densities $p(\mathbf{x}\,|\,C_k)$. Accordingly if we consider a 1-of-C_k classification case to model the input data $\mathbf{x}$ we can write the posterior probabilities as

$$P(C_k\,|\,x) = p(x\,|\,C_k)P(C_k) \Big/ \sum_{k'} p(x\,|\,C_{k'})P(C_{k'}) \qquad (1)$$

Using supervised mixture models to estimate the posterior probabilities stems from the theory that if the labels $\mathbf{t}$ represent a class

label C_k, the classification problem becomes modelling the conditional density $p(\mathbf{t}|\mathbf{x})$ directly. By definition, Mixture of Experts (MoE), a class of supervised mixture models can estimate $p(\mathbf{t}|\mathbf{x})$ directly using a gating function and K experts with an interpretation

$$p(\mathbf{t}|\mathbf{x}) = \sum_{k=1}^{K} g_k(\mathbf{x}, \mathbf{w}_{ji}) \phi(\mathbf{t}|\mathbf{x}, \mathbf{w}_{rk}) \quad (2)$$

where $\phi_k(\mathbf{t}|\mathbf{x})$ represents class conditional densities derived by experts, $g_k(\mathbf{x})$ denotes the probability that input $\mathbf{x}$ is attributed to expert k and $\mathbf{w}_{ji}$ and $\mathbf{w}_{rk}$ represent the gate and expert parameters respectively. The gating network $g_k(\mathbf{x})$ is normally softmax or network using softmax activation function $\exp(a_j)/\sum_{j'}\exp(a_{j'})$ where a_j is the intermediate variable or hidden layer output for single or multilayered gate respectively and experts are neural networks. For modelling multidimensional feature vector space, the choice of expert with response $\phi_k(\mathbf{t}|\mathbf{x})$ is crucial to the complexity and the MoE performance. Using complex networks such as MLP or mixture of MLPs increase the complexity as the training involves hidden and output layer parameters. Existing MoE architectures for speech recognition combine MLP networks or RNN networks to perform the posterior probability estimation [1] [2]. Critically these networks achieve the accuracy at the expense of complex architecture. It is important to note that the aim of the MoE is to not to estimate the posterior probability accurately but instead assist in determining the class to which a set of input vectors belong to. In brief the aim is to build a simplified learning kernel to approximate the input space to 1 of K target classes with faster convergence rate and lesser training time.

2.2. Description of proposed framework:

The analysis reveals that the learning models efficacy can be increased if the modules involve in MoE training stage has a sequel effect towards faster convergence. In literature methods the information derived from the input space as in pre-processor stage model is passed onto MoE. However if a SOM model, known for topological preserving within the Mixture of RBF, is embedded in the hidden layer stages of Mixture of RBF experts, the speech data described the vectors of dimension d from input space, $\mathbf{X}$ can be mapped to a codebook C that consists of N codewords. This is explained as follows: Consider an input $\mathbf{X}$ represented with multidimensional feature vectors $\mathbf{x}(l,d)$ where l and d denote length and dimension of feature vector respectively. If SOM is used to map $\mathbf{x}(l,d)$ to rectangular nodes of matrix size $m \times m$ then the multidimensional vectors are represented in the form of nodes using Euclidean distance minimization

$$\mathbf{som}\left((m \times m), d\right) = \arg\min_m \left\{ \left\| \mathbf{x}(l,d) - \mathbf{w}_m \right\| \right\} \quad (3)$$

These multidimensional node vectors n_d are clustered using k-means algorithm to obtain K clusters with mean vectors $\boldsymbol{\mu}_k(\mathbf{x})$. If we denote the number of nodes within each cluster as N_d then using sum of squares minimization as shown in (4) we can obtain the cluster parameters which are indices of input data distribution to form RBF experts.

$$\boldsymbol{\mu}_k(\mathbf{x}) = \sum_{k=1}^{K} \sum_{n_d=1}^{N_d} \left| \left(\arg\min_m \left\{ \left\| \mathbf{x}(l,d) - \mathbf{w}_m \right\| \right\} \right) - \boldsymbol{\mu}_k \right|^2 \quad (4)$$

The multidimensional mean vectors $\boldsymbol{\mu}_k(\mathbf{x})$ for each cluster c_k are used to construct K RBF experts with unit variance and hidden unit activations using $\varphi_k = \exp\left(-\left\| \mathbf{t} - \boldsymbol{\mu}_k \right\|^2 / 2\right)$. For the gating network, we use a Generalized Lineal Model (GLM) with activation $a_j = \sum_{j=1}^{d} w_{ji} x_i + b_j$. By substituting the GLM and expert activation functions in (2) we obtain the MoE output

$$p(\mathbf{t}|\mathbf{x}) = \sum_{i=1}^{K} \left(\left(\exp(a_j) / \sum_{j'} \exp(a_{j'}) \right) \left(\sum_{k=1}^{h_d} w_{rk} \varphi_k + w_{r0} \right) \right)$$

where w_{rk}, w_{r0} and h_d representing RBF weights, bias, and the number of hidden units respectively.

2.3. Learning procedure using Mixture of Experts:

Different learning algorithms are approached in the state of the art models and the common ME training approaches being gradient descent and expectation optimization (EM) algorithm. We have adopted batch gradient descent approach [9] for our evaluation as EM algorithm, despite decoupling the parameter estimation, is computationally very intensive and training large data can be a very cumbersome task. With batch gradient descent approach the proposed model can be trained on likelihood approach [6] resulting in an error function

$$E=-\sum_{n}\ln p(\mathbf{t}|\mathbf{x})-\sum_{n}\ln\sum_{k=1}^{K}g_k(\mathbf{x}^n,w_{ji})\phi_k(\mathbf{t}^n|\mathbf{x}^n,w_{rk}) \quad (5)$$

If we concatenate expert and gating weights w_{ji} and w_{rk} into a parameter $\mathbf{w}_k$ and assuming unobserved variables $\mathbf{z}$ then minimizing this error function with respect to the parameters yield

$$\frac{\partial E}{\partial \mathbf{w}_k}=-\sum_{n,}p(\mathbf{z}=k|\mathbf{x}^n,\mathbf{t}^n,\mathbf{w}_k)\frac{\partial}{\partial \mathbf{w}_k}\ln p(\mathbf{t}^n,\mathbf{z}|\mathbf{x}^n,\mathbf{w}_k) \quad (6)$$

Substituting for $p(\mathbf{t}|\mathbf{x})$ from (1) into (6), we have

$$\frac{\partial E}{\partial \mathbf{w}_k}=-\sum_{n}\sum_{k=1}^{K}p(\mathbf{z}=k|\mathbf{x}^n,\mathbf{t}^n,\mathbf{w}_k)\left[\frac{\partial}{\partial \mathbf{w}_k}\ln g_k(\mathbf{x}^n,w_{ji})+\frac{\partial}{\partial \mathbf{w}_k}\ln\phi_k(\mathbf{t}^n|\mathbf{x}^n,w_{rk})\right] \quad (7)$$

The term $p(\mathbf{z}=k|\mathbf{x}^n,\mathbf{t}^n,\mathbf{w}_k)$ is the posterior probabilities of unobserved variables $\mathbf{z}$ defined by Bayes rule:

$$p(\mathbf{z}=k|\mathbf{x}^n,\mathbf{t}^n,\mathbf{w}_k)=\frac{P(\mathbf{t}=k|\mathbf{x}^n,\mathbf{w}_k)p(\mathbf{t}^n|\mathbf{z}=k,\mathbf{x}^n,\mathbf{w}_k)}{p(\mathbf{t}^n|\mathbf{x}^n,\mathbf{w}_k)} \quad (8)$$

Substituting (2) in (8) we get

$$p(\mathbf{z}=k|\mathbf{x}^n,\mathbf{t}^n,\mathbf{w}_k)=\frac{g_k(\mathbf{x}^n,w_{ji})\phi_k(\mathbf{t}^n|\mathbf{x}^n,w_{rk})}{\sum_{k'}g_{k'}(\mathbf{x}^n,w_{ji})\phi_{k'}(\mathbf{t}^n|\mathbf{x}^n,w_{rk'})} \quad (9)$$

Using (7), (8) and (9) we train the MoE using gradient descent algorithm to obtain the derivative with respect to gate and expert outputs as

$$\frac{\partial E}{\partial y_k^n}=-\sum_{k=1}^{K}\pi_k\frac{\partial \ln g_k(\mathbf{x}^n,w_{ji})}{\partial y_k^n}=g_k(\mathbf{x}^n,w_{ji})-\pi_k \quad (10)$$

$$\frac{\partial E}{\partial y_{jk}^n}=-\sum_{k=1}^{K}\pi_k\frac{\partial \ln\phi_k(\mathbf{t}^n|\mathbf{x}^n,w_{rk})}{\partial y_{jk}^n}=\pi_k\left\{y_{jk}^n(\mathbf{x}^n,w_{rk})-t_{jk}^n\right\} \quad (11)$$

where π_k denotes $p(\mathbf{z}=k|\mathbf{x}^n,\mathbf{t}^n,\mathbf{w}_k)$. Concatenating these derivates into a gradient term Δ_{moe} we obtain the following gradient descent weight updates for w_k optimization.

$$\mathbf{w}_k(t+1)=\mathbf{w}_k(t)-\eta_{moe}\Delta_{moe}+\mu_{moe}\mathbf{w}_k(t-1) \quad (12)$$

where η_{moe} and μ_{moe} are the learning rate and momentum factor, respectively.

3 Experiments and Results

For experiments a database of 6300 sentences spoken by 63 speakers from 8 dialect regions (TIMIT speech corpus) are subjected to classification experiments. The classification kernel programmed in MATLAB using parallel Intel® Pentium® 4 CPUs running at 2.8GHz. We have applied 4620 sentences for training and 1680 sentences for testing purpose.

The speech data was parameterized in spectral domain using MFCC feature extraction process. Every sentence is divided into units of 20 ms duration with 10 ms overlap and for each unit is represented with 39 Mel feature vectors. These feature vectors are subjected to a two layer feed forward MLP network with 351 inputs, 200 hidden and 39 output nodes analyzed for modelling with back propagation and separately to a mixture of MLP experts' model for comparison. For our method the feature vectors are at first subjected to two-stage SOM clustering process. A rectangular grid 10×10 SOM is used to map the MFCC vectors each represented by 351 scalar values into a codebook of size 100×351. The SOM is batch trained for 250 epochs to map these vectors to codebook vector based partitions. The SOM parameters are trained to reach the optimal conditions and the optimal learning rate of 0.001 is adopted [5].The codebook vectors are grouped into a number of clusters using k-means algorithm to find the basis function parameters. K-Means algorithm is used to select an optimal set of points which are placed at the centroids of clusters of training data. Given K radial units, it adjusts the positions of the centers so that each training point "belongs to" a cluster center, and is nearer to this center than to any other center and each cluster center is the centroid of the training points which belong to it.

MoE is then constructed with a softmax expert mixture of RBF experts and a single feed-forward neural network or a simple linear model (GLM) is used as gate. Each expert assigned the cluster means to its basis

functions, unit variance. The parameters of the proposed system are estimated using gradient descent approach with early stopping criterion of Δmse <0.0001. The learning rate and momentum factor are determined through a set of Monte Carlo experiments and the best rates are kept constant at 0.001 and 0.002 respectively. The trained network is then analyzed for test performance. In the results section, First set compares the proposed ME models performance with a single MLP classifier (Table 1, Fig 2). Second set of experiments analyze the MoE model with MLP mixture model counterparts (Table 2).The test results show that a single MLP invariably suffers longer training time of 482 hours of training while the mixture of MLP provides a good alternative to single MLP at the expense of more complex network. For larger datasets lesser training times and complexity are crucial and proposed MoE models alleviates both issues and achieves better error performance with less complex core and lesser training times. Further the results under different number of experts, gating networks and Davies Bouldin Index [5] to determine clustering validity are given in Table 2 and Table 3, Fig. 3, Fig. 4 and Fig. 5 respectively. The results clearly demonstrates that the proposed MoE model efficacy for the speech recognition applications.

4 Conclusion

In this letter we have demonstrated the benefits of embedding SOM clustering with mixture of experts' framework. We have shown that for classification problems better convergence can be achieved with this kernel than a single MLP classifier or state of the art models such as mixtures of MLP. The test results reported for TIMIT speech corpus show that accuracy improvement, lesser training times can be achieved with this reduced complex kernel and it provides a better alternative to achieve better speech recognition solutions.

References:

[1] Waterhouse.S., Classification and Regression using Mixtures of Experts, PhD Thesis, Department of Engineering, Cambridge University, 1997.

[2] P. Moerland, Mixture of Experts Estimate A-Posteriori Probabilities, International Conference on Artificial Neural Networks (ICANN'97), pp. 499 -- 505, 1997

[3] Wenxin Jiang, Martin A. Tanner: On the asymptotic normality of hierarchical mixtures-of-experts for generalized linear models. IEEE Transactions on Information Theory 46(3): 1005-1013 (2000)

[4] Bin Tang, Malcolm I. Heywood, and Michael Shepherd. Input partitioning to mixture of experts. In 2002 International Joint Conference on Neural Networks, pp.227-232, Hawaii, May 2002.

[5] Vesanto, J. & Alhoniemi, E. Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks, 2000. Vol. 11, no 3, pp. 586-600.

[6] J. Ghosh and S.V. Chakravarthy, "The Rapid Kernel Classifier: A Link Between the Self-Organizing Feature Map and the Radial Basis Function Network", Jl. of Intelligent Material Systems and Structures, (spl. issue on Neural Networks), pp.211-219, Vol. 5(2), March 1994.

[7] Ramamurti, V.; Ghosh, J., "Regularization and error bars for the mixture of experts network", Neural Networks, 1997. International Conference on Volume 1, 9-12 June 1997.Page(s):221 – 225.

[8] Li Jun and Tom Duckett, "Learning Robot Behaviours with Self-Organizing Maps and Radial Basis Function Networks", In Proc. SWAR'02, Second Swedish Workshop on Autonomous Robotics, Stockholm, Sweden, October 10-11, 2002.

FIGURE CAPTIONS:

Fig. 1 Mixture of Experts Layout

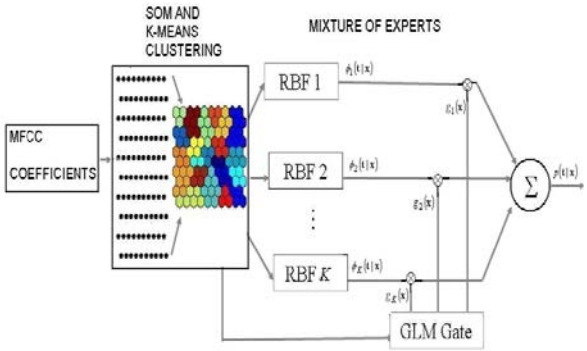

Fig. 2: Training and Test MSE with MLP and MoE

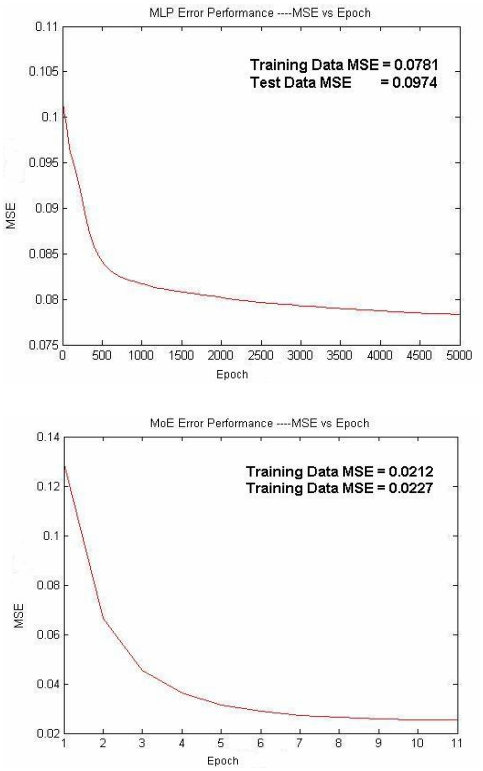

Fig. 3: Test MSE with Different Number of Experts

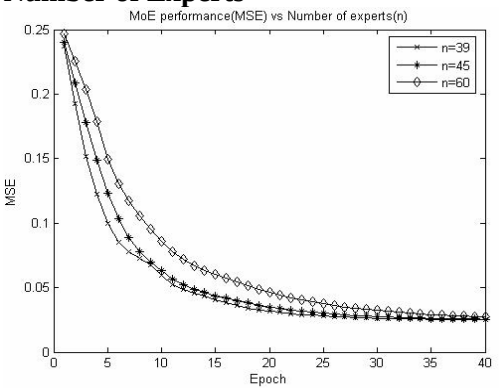

Fig. 4: Test MSE with Different Gates

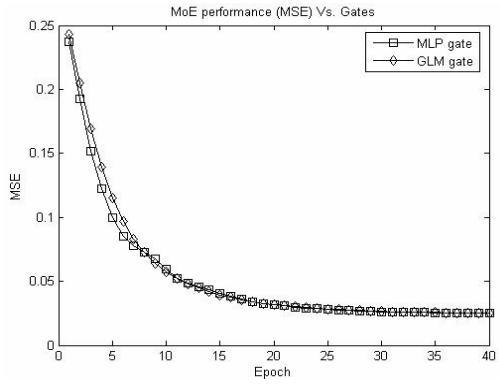

Fig 5: Davies-Bouldin Index for Clustering for Data set.

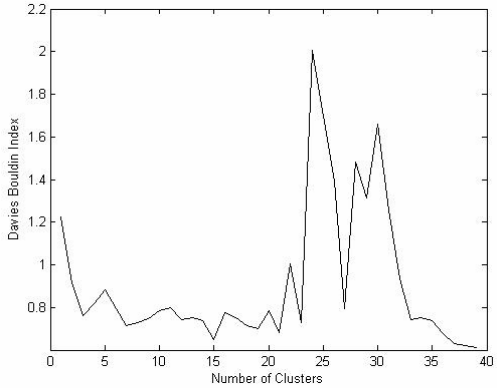

Table 1: Training and Test data Error Performance comparison with Single MLP

TIMIT MSE	Single MLP	MoE	Accuracy Gain (n times)
Training Data Set	0.0781	0.0212	2.683
Validation	0.0797	0.0214	2.724
Complete	0.0974	0.0227	3.291

Table 2: Comparison of MoE with MLP mixture

Architecture	Mix MLP with MLP gate	MoE with MLP gate	MoE with GLM gate
Training Parameters	611250	76128	75928
Epochs to train	82	30	12
Training MSE	0.0241	0.0228	0.0212
Test MSE	0.0254	0.0234	0.0227
Accuracy Improvement	-------	8.5470%	11.894%

Table 3: Mixture of Experts performance at different Number of experts with GLM gate

Number of experts	Parameters	Test Set MSE
39 experts	75928	0.0212
45 experts	85683	0.0255
60 experts	109728	0.0274