Pergamon

Neural Networks, Vol. 10, No. 2, pp. 231–241, 1997

Printed in Great Britain
0893–6080/97 $17.00+.00

PII: S0893-6080(96)00050-0

CONTRIBUTED ARTICLE

A Bayesian Approach to Model Selection in Hierarchical Mixtures-of-Experts Architectures

ROBERT A. JACOBS,[1] FENGCHUN PENG[2] AND MARTIN A. TANNER[3]

[1]University of Rochester, [2]University of Nebraska, Lincoln and [3]Northwestern University

(*Received* 29 *June* 1995; *revised and accepted* 8 *April* 1996)

Abstract—*There does not exist a statistical model that shows good performance on all tasks. Consequently, the model selection problem is unavoidable; investigators must decide which model is best at summarizing the data for each task of interest. This article presents an approach to the model selection problem in hierarchical mixtures-of-experts architectures. These architectures combine aspects of generalized linear models with those of finite mixture models in order to perform tasks via a recursive "divide-and-conquer" strategy. Markov chain Monte Carlo methodology is used to estimate the distribution of the architectures' parameters. One part of our approach to model selection attempts to estimate the worth of each component of an architecture so that relatively unused components can be pruned from the architecture's structure. A second part of this approach uses a Bayesian hypothesis testing procedure in order to differentiate inputs that carry useful information from nuisance inputs. Simulation results suggest that the approach presented here adheres to the dictum of Occam's razor; simple architectures that are adequate for summarizing the data are favored over more complex structures.* © *1997 Elsevier Science Ltd. All Rights Reserved.*

Keywords—Modular architecture, Hierarchical architecture, Model selection, Bayesian analysis, Gibbs sampling.

1. INTRODUCTION

Statisticians have known for many years that there does not exist a statistical model that shows good performance on all tasks. Whereas some models perform well on some tasks, other models perform well on other tasks. The *model selection* problem is, therefore, unavoidable; investigators must decide which model is best at summarizing the data for each task of interest. A common approach is to compare the expected squared error for different models. A model's expected squared error is equal to the sum of its variance (across different data sets) and the square of its bias (the difference between the expected value of the estimate produced by the model and the expected value of the random variable of interest). Classical statistics has focused on unbiased models, and has viewed unbiased models with the smallest variance to be best. More recently, statisticians have considered the comparison of biased models.

The problem of model selection is often divided into at least two subproblems. The first subproblem is to discover an organization of a model's parameters that is well-matched to the task that the model is to perform. In the context of neural networks, this means that an investigator attempts to discover the network topology (e.g., number of layers, number of hidden units per layer, connectivity pattern among units) that results in the best generalization performance. A common result in the neural network literature is that "simple is good." Networks with too many free parameters tend to overfit the training data and, thus, show poor generalization performance. In contrast, networks with as few free parameters as possible that are still adequate for summarizing the data tend to generalize comparatively well (Denker et al., 1987).

At least three approaches to the model selection problem have been reported in the neural network literature. Some researchers have attempted to discover optimal network structures by searching the space of topologies using optimization techniques

Acknowledgements: R. Jacobs was supported by NIH grant R29-MH54770. F. Peng was supported by NIH grant T32-CA09667. M. Tanner was supported by NIH grant RO1-CA35464, NSF grant DMS-9505799, and by a grant from Sun Microsystems.

Requests for reprints should be sent to Professor Robert Jacobs, University of Rochester, Brain and Cognitive Sciences Department, Meliora Hall, River Campus, Rochester, NY 14627, USA.

such as genetic algorithms. Belew (1993), for example, presented a genetic algorithm system in which the "genomes" correspond to an ordered sequence of productions from a grammar that defines the topological structure of neural networks. A network's generalization error is used to determine its genome's fitness. Other researchers have attempted to "grow" good network structures by starting with a simple topology, and then adding units and connections to the network during the course of training. For example, the cascade-correlation learning architecture proposed by Fahlman and Lebiere (1990) adds new hidden units to a network whenever the network's error reaches an asymptote. Still other researchers have started with complex topologies, and "pruned" units and connections from the network during training. Weigend et al. (1991), for example, trained a network using an objective function that is minimized when the network approximates the target function with as few non-zero weights as possible. As a second example, Le Cun et al. (1990) evaluated the second derivatives of an error function with respect to the weights in order to determine which connections could be deleted from a network with minimal effect on the network's performance. It is this last approach to the model selection problem that has, in our view, been the most mathematically rigorous and the most empirically successful. As discussed below, it is also the approach that is most closely related to the methods presented here.

The problem of model selection has a second subproblem that is referred to as the covariate selection problem. A model attempting to estimate the value of a random variable may have potential access to a wide range of measurements regarding the state of the environment. Some of these quantities may provide the model with useful information regarding the random variable, whereas others may not. The latter are commonly referred to as "nuisance" variables. The covariate selection problem is to discover the quantities that carry useful information. In the context of neural networks, only the useful quantities should be used as inputs to a network. A network that receives both useful inputs and "nuisance" inputs will contain too many free parameters and, thus, be prone to overfitting the training data leading to poor generalization. The covariate selection problem is, therefore, closely related to the first subproblem of model selection discussed above. Although the covariate selection problem has received only sporadic attention in the neural network literature, it is widely viewed as an important problem in the statistics literature (e.g., Breiman et al., 1984).

This article studies the problem of model selection in a class of models known as hierarchical mixtures-of-experts (HME) architectures. HME architectures contain multiple neural networks organized into tree-structures. Networks referred to as *expert networks* form the leaves of the tree; *gating networks* are located at the branch-points. Tasks are approached using a recursive "divide-and-conquer" strategy; complex tasks are decomposed into subtasks which, in turn, are themselves decomposed into subsubtasks. It is assumed that the data can be summarized by a collection of relatively simple mappings, each of which is defined over a restricted region of the input space. During the course of training, HME architectures learn to allocate different expert networks to summarize the data located in different regions. HME architectures are of interest on both theoretical and empirical grounds. From a theoretical viewpoint, they combine aspects of finite mixture models and generalized linear models, two well-studied statistical frameworks (McLachlan & Basford, 1988; McCullagh & Nelder, 1989). From an empirical viewpoint, they have been shown to be capable of comparatively fast learning and good generalization on a wide variety of regression and classification tasks (Jacobs et al., 1991; Nowlan & Hinton, 1991; Jordan & Jacobs, 1994; Waterhouse & Robinson, 1994).

Most previous applications of HME architectures have used fixed tree-structures whose particular organizations were selected by researchers in a heuristic or intuitive manner (Jordan & Jacobs, 1992, 1994; Waterhouse & Robinson, 1994; Peng et al., 1996). In contrast, this article presents a Bayesian approach to the model selection problem through the use of Markov chain Monte Carlo methodology. Initially complex HME architectures are gradually pruned on the basis of quantities computed via Gibbs sampling until a tree-structure is discovered that is well-matched to the task of interest. Gibbs sampling is also used to differentiate useful input variables from nuisance inputs. The article is organized as follows. Section 2 presents an overview of HME architectures and the Bayesian approach to estimating these architectures' parameters. Section 3 presents a technique for pruning HME tree-structures. Results are reported from applications of the technique to a breast cancer classification task and a speech recognition task. Section 4 presents a technique for differentiating useful inputs from nuisance inputs. The merits of this technique are evaluated using the breast cancer classification task and the speech recognition task.

2. HIERARCHICAL MIXTURES-OF-EXPERTS ARCHITECTURES

This section briefly presents the class of hierarchical mixtures-of-experts architectures and the Bayesian

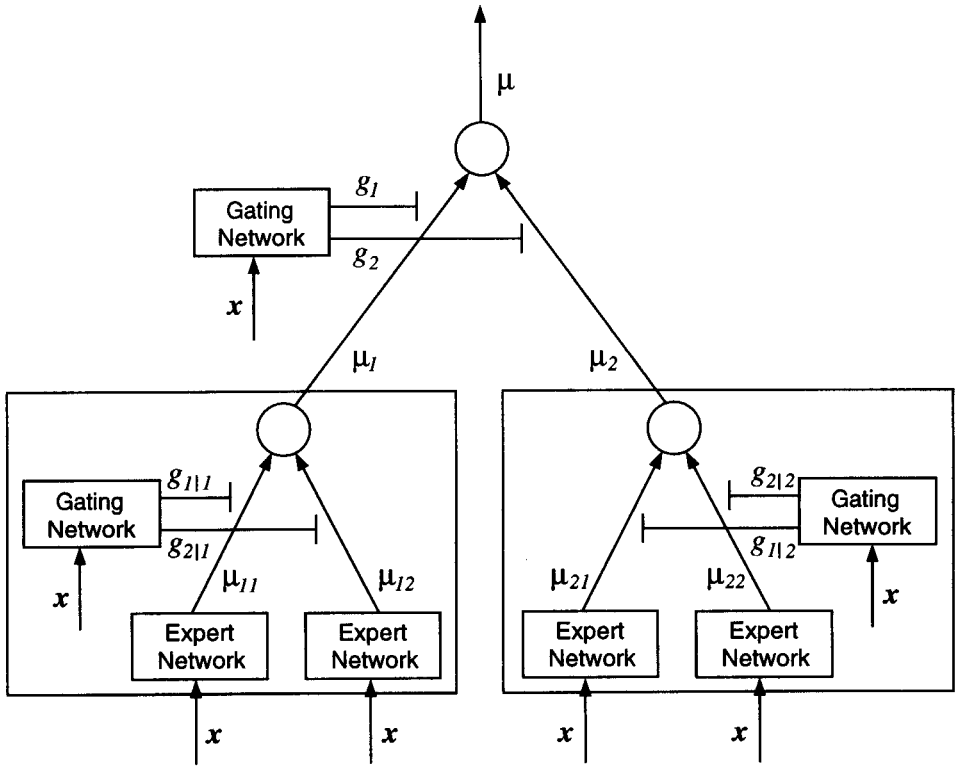

FIGURE 1. A hierarchical mixtures-of-experts architecture.

approach to estimating these architectures' parameters. A fuller discussion of this class of architecture can be found in Jordan and Jacobs (1994); a more detailed presentation of the Bayesian approach to training these architectures can be found in Peng et al. (1996).

HME architectures can be characterized as fitting a piecewise function to the training data. These data are assumed to form a countable set of paired variables $\chi = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}_{t=1}^{T}$, where $\mathbf{x}$ is a vector of input variables and $\mathbf{y}$ is a vector of target outputs. An example of an HME architecture is illustrated in Figure 1. For explanatory reasons, we limit our presentation to a two-level tree; the extension to trees of arbitrary depth and width is straightforward. A number of *clusters* are located at the bottom-level of the tree, each cluster consisting of a number of expert networks and a gating network (the tree in the figure contains two clusters, each enclosed in a box). An additional gating network is located at the top-level of the tree. As a matter of notation, the letter i is used to index branches at the top-level, and the letter j to index branches at the bottom-level. A cluster's output is formed by linearly combining the outputs of the experts that comprise the cluster. The linear coefficients are computed by the cluster's gating network. In particular, each expert network (i, j) maps the input vector $\mathbf{x}$ to the output vector $\boldsymbol{\mu}_{ij}$. The outputs of the gating network are a set of scalar coefficients $g_{j|i}$, where the subscript denotes that the coefficient corresponds to the jth expert within the ith cluster. These coefficients are computed on the basis of the input $\mathbf{x}$, and are constrained to be nonnegative and to sum to one. The output of the ith cluster, denoted $\boldsymbol{\mu}_i$, is given by

$$\boldsymbol{\mu}_i = \sum_j g_{j|i}\, \boldsymbol{\mu}_{ij}. \tag{1}$$

The output of the architecture as a whole is formed in an analogous manner; that is, the architecture's output, denoted $\boldsymbol{\mu}$, is a linear combination of the cluster outputs for each input $\mathbf{x}$:

$$\boldsymbol{\mu} = \sum_i g_i \boldsymbol{\mu}_i. \tag{2}$$

The set of scalar coefficients g_i that weight the contributions of the clusters are the outputs of the gating network at the top-level of the tree-structure. These coefficients are computed on the basis of $\mathbf{x}$, and are constrained to be nonnegative and to sum to one.

To understand the mixtures-of-experts framework, it is necessary to provide a probabilistic interpretation of the architecture. Assume that the process generating the data is decomposable into a set of subprocesses defined on distinct regions of the input space. For each data item, a subprocess is selected, based on the input $\mathbf{x}^{(t)}$, and the selected subprocess

maps $\mathbf{x}^{(t)}$ to the output $\mathbf{y}^{(t)}$. Furthermore, assume that the selection of a subprocess is based on a nested sequence of decisions that each depend on the input $\mathbf{x}^{(t)}$. For explanatory purposes, we limit our presentation to a two-stage sequence (as discussed below, this two-stage sequence matches the two-level tree-structure). Specifically, data are generated as follows. For each input $\mathbf{x}^{(t)}$:

- a label i is chosen from a multinomial distribution with probability $P(i|\mathbf{x}^{(t)}, V)$, where $V = [\mathbf{v}_1, \ldots, \mathbf{v}_I]$ is the matrix of parameters underlying the multinomial distribution;
- a label j is chosen from another multinomial distribution with probability $P(j|\mathbf{x}^{(t)}, V_i)$, where the matrix of parameters $V_i = [\mathbf{v}_{i1}, \ldots, \mathbf{v}_{iJ}]$ underlying this distribution is dependent on the value of label i;
- an output $\mathbf{y}^{(t)}$ is generated by subprocess (i, j) with probability $P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})$, where U_{ij} is a parameter matrix and Φ_{ij} represents other parameters.

The total probability of generating $\mathbf{y}^{(t)}$ from $\mathbf{x}^{(t)}$ is given by the hierarchical mixture density

$$P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, \Theta) = \sum_i P(i|\mathbf{x}^{(t)}, V) \sum_j P(j|\mathbf{x}^{(t)}, V_i) P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij}) \quad (3)$$

where $\Theta = [\mathbf{v}_1, \ldots, \mathbf{v}_I, \mathbf{v}_{11}, \ldots, \mathbf{v}_{IJ}, U_{11}, \ldots, U_{IJ}, \Phi_{11}, \ldots, \Phi_{IJ}]^T$ is the matrix of all parameters. The total probability of the data set χ, assuming independently distributed data, is the product of T such densities, with the likelihood given by:

$$L(\Theta|\chi) = \prod_t \sum_i P(i|\mathbf{x}^{(t)}, V) \sum_j P(j|\mathbf{x}^{(t)}, V_i) P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij}). \quad (4)$$

From the perspective of statistical mixture modeling, we identify the gating network at the top-level of the HME architecture with the selection of the label i, and the bottom-level gating networks with the selection of the label j. That is, the gating outputs g_i are interpreted as the input-dependent, multinomial probabilities of selecting a value for i; the gating outputs $g_{j|i}$ are the input-dependent, multinomial probabilities of selecting a value for j given a value for i. The values for i and j index a particular subprocess. Different expert networks are identified with different subprocesses; each expert models the input-dependent distributions associated with its corresponding subprocess.

This article is concerned with multiway classification problems, meaning that the expert networks' outputs, like those of the gating networks, are interpreted as input-dependent multinomial distributions. Both types of networks produce their outputs in two stages; first they compute a linear sum of the components of the input vector $\mathbf{x}$, and then they normalize this sum using the "softmax" activation function (Bridle, 1989). The top-level gating network outputs g_i and the bottom-level gating network outputs $g_{j|i}$ are given by:

$$g_i = \frac{e^{\mathbf{v}_i^T \mathbf{x}}}{\sum_{l=1}^{I} e^{\mathbf{v}_l^T \mathbf{x}}}, \qquad g_{j|i} = \frac{e^{\mathbf{v}_{ij}^T \mathbf{x}}}{\sum_{l=1}^{J} e^{\mathbf{v}_{il}^T \mathbf{x}}} \quad (5)$$

where $\mathbf{v}_i$ and $\mathbf{v}_{ij}$ are the parameter vectors for the top-level and bottom-level gating networks, respectively. The kth component of expert (i, j)'s output vector, interpreted as this expert's conditional probability of classifying the input as a member of class k, is denoted μ_{ijk} and is given by:

$$\mu_{ijk} = \frac{e^{\mathbf{u}_{ijk}^T \mathbf{x}}}{\sum_{l=1}^{n} e^{\mathbf{u}_{ijl}^T \mathbf{x}}} \quad (6)$$

where $U_{ij} = [\mathbf{u}_{ij1}, \ldots, \mathbf{u}_{ijn}]$ is the parameter matrix of expert (i, j), and n is the number of possible classes. Under these conditions, the likelihood function given in eqn (4) can be rewritten as

$$L(\Theta|\chi) = \prod_t \sum_i g_i^{(t)} \sum_j g_{j|i}^{(t)} \prod_k (\mu_{ijk}^{(t)})^{y_k^{(t)}} \quad (7)$$

which is a hierarchical mixture of multinomial densities.

In this article, Bayesian inference regarding the gating and expert networks' parameters is performed using Markov chain Monte Carlo methods. An important advantage of Bayesian estimation procedures as compared to maximum likelihood estimation procedures is that they provide an estimate of the entire distribution of each parameter value instead of a simple point estimate. A disadvantage is that they can be computationally expensive. This article uses a Bayesian technique known as Gibbs sampling. In short, a Gibbs sampler obtains samples from the joint distribution of all the parameter values by iteratively sampling from the conditional distribution of each parameter given values for the remaining parameters (Neal, 1991; Tanner, 1993).

In the context of the hierarchical mixtures-of-experts architecture, the parameter estimation problem is greatly simplified by augmenting the observed data with the unobserved indicator variables indicating the expert in the model. Let $z_i^{(t)} = 1$ with probability

$$h_i^{(t)} = \frac{P(i|\mathbf{x}^{(t)}, V)\sum_j P(j|\mathbf{x}^{(t)}, V_i)P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})}{\sum_i P(i|\mathbf{x}^{(t)}, V)\sum_j P(j|\mathbf{x}^{(t)}, V_i)P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})}, \tag{8}$$

and $z_{j|i}^{(t)} = 1$ with probability

$$h_{j|i}^{(t)} = \frac{P(j|\mathbf{x}^{(t)}, V_i)P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})}{\sum_j P(j|\mathbf{x}^{(t)}, V_i)P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})} \tag{9}$$

Then $z_{ij}^{(t)} = z_i^{(t)} \times z_{j|i}^{(t)}$ takes value 1 with probability

$$h_{ij}^{(t)} = \frac{P(i|\mathbf{x}^{(t)}, V)P(j|\mathbf{x}^{(t)}, V_i)P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})}{\sum_i P(i|\mathbf{x}^{(t)}, V)\sum_j P(j|\mathbf{x}^{(t)}, V_i)P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})}. \tag{10}$$

Let $\chi' = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}, \mathbf{z}^{(t)})\}_{t=1}^T$ where $\mathbf{z}^{(t)}$ is the vector of indicator variables. Then the augmented likelihood for the HME architecture is

$$L(\Theta|\chi') = \prod_t \prod_i \prod_j \{P(i|\mathbf{x}^{(t)}, V)P(j|\mathbf{x}^{(t)}, V_i) \times P(\mathbf{y}^{(t)}|\mathbf{x}^{(t)}, U_{ij}, \Phi_{ij})\}^{z_{ij}^{(t)}} \tag{11}$$

$$= \prod_t \prod_i \prod_j \prod_k \{g_i^{(t)} g_{j|i}^{(t)} \mu_{ijk}^{(t)y_k^{(t)}}\}^{z_{ij}^{(t)}} \tag{12}$$

$$= \prod_t \prod_i \prod_j \prod_k \left\{ \frac{e^{v_i^T \mathbf{x}^{(t)}}}{\sum_{l=1}^I e^{v_l^T \mathbf{x}^{(t)}}} \frac{e^{v_{ij}^T \mathbf{x}^{(t)}}}{\sum_{l=1}^J e^{v_{il}^T \mathbf{x}^{(t)}}} \left(\frac{e^{u_{ijk}^T \mathbf{x}^{(t)}}}{\sum_{l=1}^n e^{u_{ijl}^T \mathbf{x}^{(t)}}} \right)^{y_k^{(t)}} \right\}^{z_{ij}^{(t)}}. \tag{13}$$

This equation with independent normal priors (mean= 0; variance= σ_0^2) on Θ yields a proper posterior distribution, though the full conditionals are not standard densities. Consequently, we apply the "Metropolis-within-Gibbs" approach of Müller (1991) to draw the posterior sample for the top-level gating network parameter matrix V, the bottom-level gating network parameter matrices V_i, and the expert network parameter matrices U_{ij}. In particular, to draw a deviate from a full conditional we use the Metropolis algorithm. Consider, for example, the top-level gating network parameters. A candidate value for the next point in the Metropolis chain ($V^{(k+1)}$) is drawn from the multivariate normal distribution with the current sample values as its mean and a diagonal variance-covariance matrix to allow for variation around the current sample values, i.e., $V^{(k+1)} \leftarrow N(V^{(k)}, \gamma^2 \mathbf{I})$. This candidate value is accepted or rejected according to the standard Metropolis scheme (Tanner, 1993). This Metropolis algorithm is iterated and the final value in this chain is treated as a deviate from the full conditional distribution.

3. PRUNING HME TREE-STRUCTURES

The structures of HME architectures have typically been designed in a heuristic or intuitive manner. In general, how can one assess whether an architecture is overspecified or under-specified for a given dataset? A basic approach to be investigated consists of examining the indicator variables $z_{ij}^{(t)}$. These quantities are Bernoulli distributed deviates which indicate whether the jth expert in branch i generated the $\{\mathbf{x}^{(t)}, \mathbf{y}^{(t)}\}$ data item. By pooling the indicator variables $z_{ij}^{(t)}$ over the data items, one can estimate the value of an expert over the observed sample space. Define the *worth index* for expert (i, j) to be the sum of the $z_{ij}^{(t)}$ over the observed sample points divided by the total number of sample points:

$$\text{worth index for expert } (i, j) = \frac{1}{T}\sum_{t=1}^T z_{ij}^{(t)}. \tag{14}$$

The worth index is an average of random variables. Throughout the remainder of this article, we focus on realizations of averages of this type based on the simulated values of the indicator variables. For simplicity, we will not distinguish the worth index from the simulated value of the index. If the worth indices for the experts are all of similar magnitudes, then this suggests that the current tree-structure may be too small, and that a structure with additional experts should be considered. Alternatively, if the worth index for an expert is small relative to that of other experts then there is evidence to suggest that this expert can be pruned from the architecture. Note, moreover, that this approach can be extended to entire branches of a tree-structure. If the worth index for branch i, given by

$$\text{worth index for branch } i = \frac{1}{T}\sum_{t=1}^T z_i^{(t)}, \tag{15}$$

is small relative to that of other branches there is evidence to suggest that this branch may not be needed.

Two tasks were used to evaluate the tree-structure pruning technique. The first task is a breast cancer classification task. The database was provided by Dr. William H. Wolberg of the University of Wisconsin Hospitals, Madison (Wolberg et al., 1987; Mangasarian & Wolberg, 1990; Wolberg & Mangasarian, 1990). (This database is available on the Internet from the UC Irvine Repository of Machine Learning Databases and Domain Theories.) For each data item, the nine input variables give values for nine

attributes of a breast tumor (clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cells size, bare nuclei, bland chromatin, normal nucleoli, and mitoses). The target output indicates whether the tumor was benign or malignant. Out of the 683 data items, 100 were randomly selected and used for training; the remaining 583 were used to test generalization performance.

The second task was a speech recognition task. The values of the two input variables are the first and second formant values from ten classes of spoken vowels. The values of the ten target outputs are all zero except for the target corresponding to the correct vowel class which is assigned a value of one. The data items were segmented from utterances of words that began with an "h", contained a vowel in the middle, and ended with a "d" (e.g., heed, hid, head, had). In all, 75 speakers (32 males, 28 females, and 15 children) uttered each word twice, though the utterances of one word for three of the speakers is missing from the database. The graph in Figure 2 presents the data. The horizontal and vertical axes give the first and second format values respectively. Each data point is labeled with a digit (0–9) that indicates the vowel class to which the data point belongs. From the collection of 1494 data items, 149 were randomly selected and assigned to a training set; the remaining 1345 were assigned to a generalization set. The data was originally collected by Peterson and Barney (1952) and is a benchmark database in the speech recognition literature.

Both the Gibbs sampler algorithm described above and the expectation-maximization (EM) algorithm were used to train the HME architectures [see Jordan and Jacobs (1994) for the EM algorithm]. The parameter values obtained via EM were used as the starting values for the Gibbs sampler. A total of five chains were created, each of length 7500 iterations. All performance statistics reported in this article are based on the final 500 iterations of each chain. Because posterior distributions based on mixture models can be highly multimodal, it is important to permit mode jumping so as to avoid oversampling in the neighborhoods of local modes. To facilitate this, after every 100 iterations of the Gibbs sampler a candidate value for the next point in the Markov chain was selected at random based on modes defined by 50 independent and randomly initialized runs of the EM algorithm. [This approach is similar to the use by Gelman and Rubin (1992) of an overdispersed approximation to the posterior distribution.] This candidate value was compared to the current point in the Gibbs sampler chain using a Metropolis test based on the observed posterior. The Metropolis algorithm used to sample from the full conditional distributions was run for 40 iterations, with the variance on the normal distribution equal to 1.0. The variance of each of the normal priors on the components of the expert and gating networks parameters was equal to 50 000. Convergence of the five chains was evaluated using the technique of Gelman and Rubin (1992). Intuitively, this technique for assessing convergence compares the between-

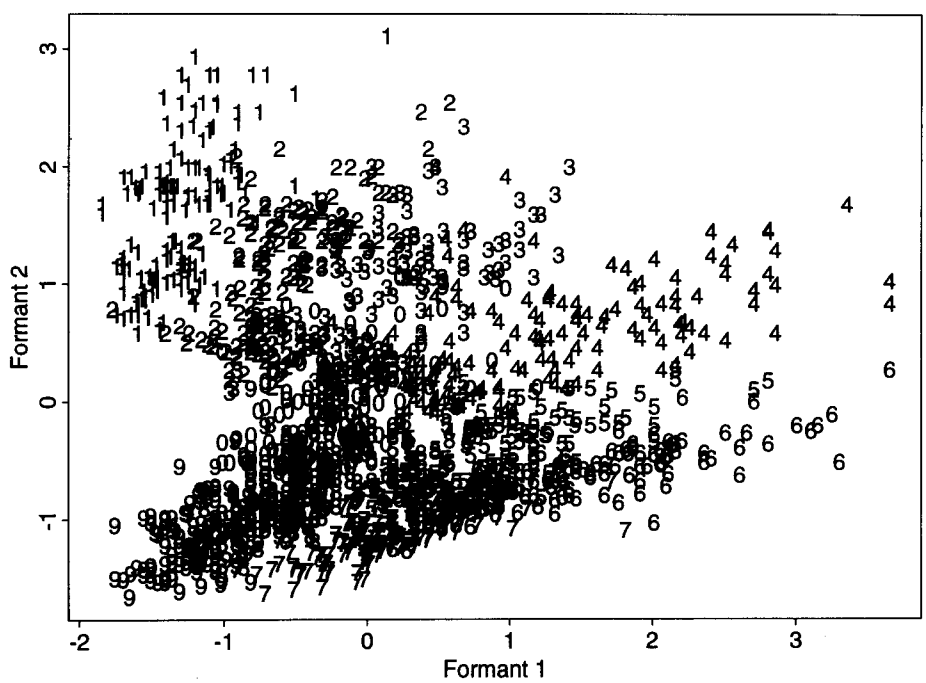

FIGURE 2. The horizontal and vertical axes give the values of the first and second formants, respectively. Each data point is labeled with a digit (0–9) that indicates the vowel class to which the data point belongs.

other expert networks can be pruned from an architecture. For example, suppose that the worth indices for the four experts that comprise an architecture have the values 0.4, 0.1, 0.3, and 0.2. According to our criterion, this suggests that three experts are required by the architecture $(0.4 + 0.3 + 0.2 = 0.9)$, and one expert can be pruned. Care, however, has to be taken; if an expert is used for a specific type of data which occurs relatively infrequently in the dataset then dropping the expert may not be optimal. Although the 0.8 cutoff criterion is arbitrary, based on our empirical work we have found this value to work well in

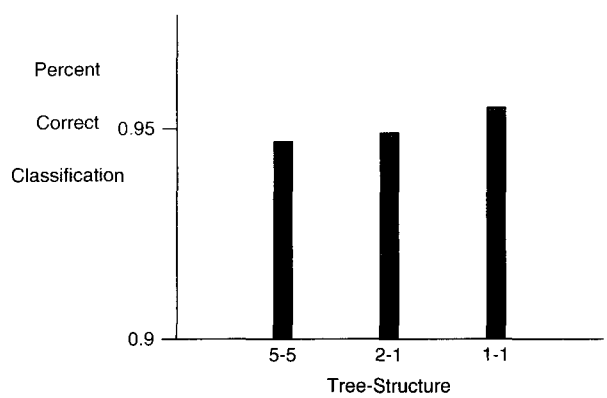

FIGURE 4. The generalization performance of several architectures on the breast cancer classification task.

architecture to the breast cancer classification task that contained two expert networks in one cluster and one expert in another cluster. The results for this "2-1" architecture are shown in the middle graph of Figure 3. Of 25 mode sets 17 suggested that only two expert networks were useful for the task, and that one expert could be pruned. Therefore, the next architecture that we tried had a 1-1 structure; it was a one-level architecture with two expert networks and one gating network. The rightmost graph of Figure 3 shows the results for this structure. A total of 19 mode sets suggested that both of the architecture's expert networks were needed to perform the task, meaning that no further pruning of the structure should be conducted.

The generalization performance of the 5-5, 2-1, and 1-1 architectures are shown in Figure 4. The horizontal axis gives the architecture; the vertical axis gives the percentage of data items in the generalization dataset that were correctly classified (averaged over all mode sets). Small performance improvements can be seen for the smaller architectures. Based upon the entire set of results, we conclude that the 1-1 structure is an appropriate architecture for the breast cancer classification task.

Figures 5 and 6 give the results of applying the pruning algorithm to the speech recognition task. We started our simulations using a 5-5 architecture. The assessment of the number of needed experts is shown in the top-left graph of Figure 5. The mode of this data is at 4, suggesting that four experts are useful for performance of this task, and that six experts can be pruned from the architecture. Therefore, we next used a 2-2 architecture; it was a two-level structure with two clusters and two expert networks per cluster. The top-right graph of Figure 5 shows the results for this architecture. Nearly equal numbers of modes sets suggested that either two or three experts were useful, and that either one or two experts could be pruned. The next structure that we tried was a 2-1 architecture. Its results are shown in the bottom-left graph. The maximum number of mode sets suggested that two experts were useful for performing this task, and so we next tried a 1-1 structure. The results for this architecture are shown in the bottom-right graph. All 25 mode sets suggested that both expert networks

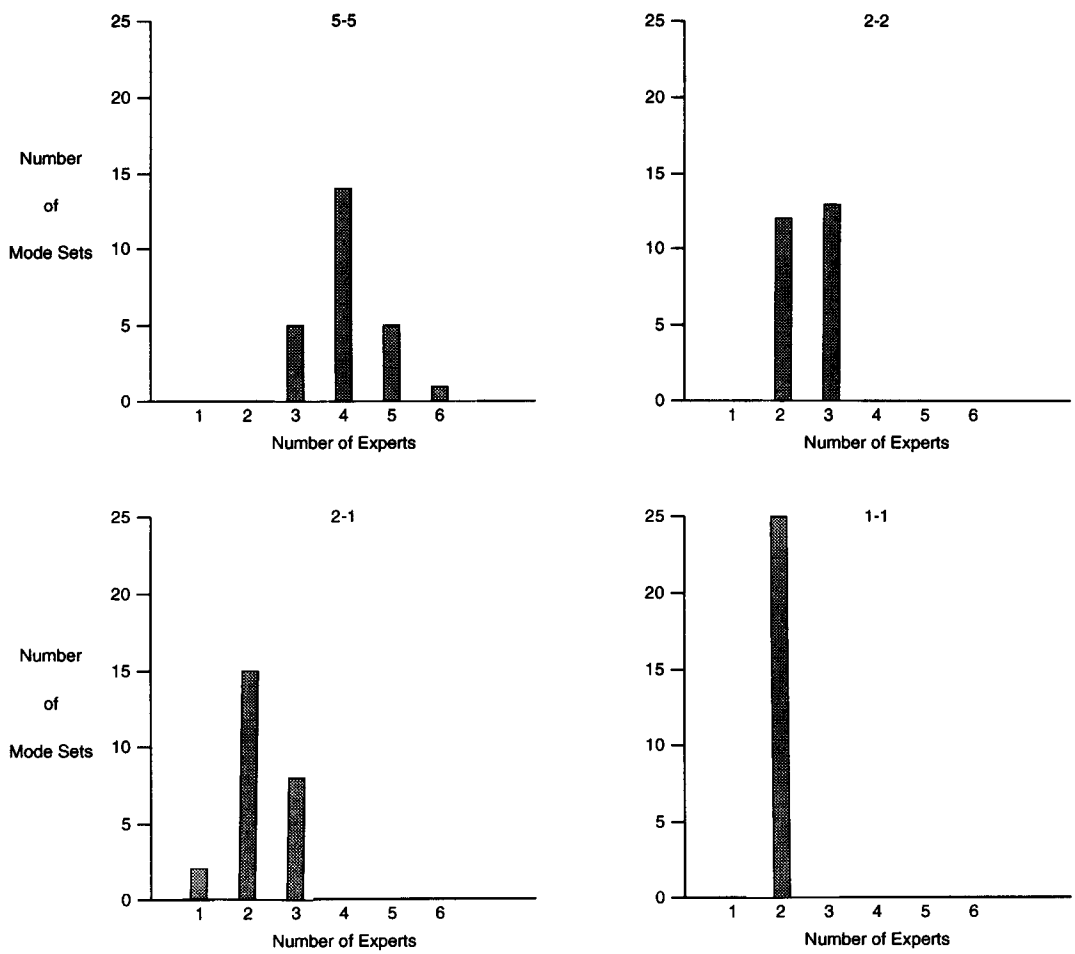

FIGURE 5. The four graphs give the results of the pruning algorithm for four different architectures applied to the speech recognition task. The horizontal axis of each graph gives the number of experts that should be retained according to the algorithm. The vertical axis gives the number of mode sets that suggested that a particular number of experts should be retained.

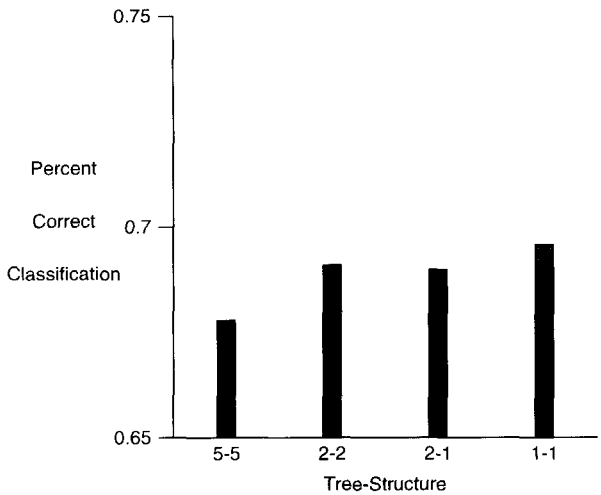

FIGURE 6. The generalization performance of several architectures on the speech recognition task.

were needed to perform the task, and that no additional pruning of the structure should be conducted.

Figure 6 shows the generalization performance of the various architectures as measured by the percentage of data items in the generalization dataset that were correctly classified. The 1-1 architecture generalized best. Based upon the entire set of results, it appears that the 1-1 structure is an appropriate architecture for the speech recognition task.

4. DETECTING NUISANCE INPUTS

A model attempting to estimate the value of a random variable may have potential access to a wide range of measurements regarding the state of the environment or the current stimulus situation. Some of these quantities may provide the model with useful information regarding the random variable, whereas others may not. The former are referred to as useful input variables; the latter are nuisance inputs. The covariate selection problem is to discover the quantities that carry useful information. This section presents an approach to this problem that is based upon Bayesian hypothesis testing procedures.

As pointed out in Box and Tiao (1973), to summarize the information in a posterior distribution $p(\Theta|\chi) \propto L(\Theta|\chi)p(\Theta)$ it is sometimes of value to identify a region of the parameter space which contains most of the mass under this distribution. One approach for such a delineation is the region of highest posterior density (HPD). Box and Tiao (1973, p. 122) presented the following definition:

A region R in the parameter space of Θ is called a highest posterior density region of content $1-\alpha$ if

(a) $p(\Theta \in R|\chi) = 1-\alpha$, *and*

(b) *for* Θ_1 *in* R *and* Θ_2 *not in* R, $p(\Theta_1|\chi) \geq p(\Theta_2|\chi)$.

Box and Tiao (1973) pointed out that for a given probability content $1-\alpha$, the region of highest posteror density has smallest volume in parameter space. In addition, these authors noted that a point Θ_0 is covered by the highest posterior density region of content $1-\alpha$ if and only if the following inequality holds:

$$P_\Theta[p(\Theta|\chi) \geq p(\Theta_0|\chi)|\chi] \leq 1-\alpha \tag{16}$$

where the density $p(\Theta|\chi)$ is treated as a random variable.

The HPD region can be viewed as a Bayesian analogue of the frequentist confidence interval. As such, to test the null hypothesis that $\Theta = \Theta_0$, we can investigate whether Θ_0 is located, for example, in the HPD region of content 0.95. If so, there is little evidence against the null hypothesis. If not, we can reject the null hypothesis at the 0.05 level. The Bayesian p-value is equal to one minus the content of the smallest HPD region which contains the null value Θ_0. See Lee (1991, p. 130) for a further discussion of this approach to Bayesian hypothesis testing.

Wei and Tanner (1990) showed how to compute the content of the smallest highest posterior density region which contains the point Θ_0 using the output from a Gibbs sampler analysis. Given a set of samples from the density $p(\Theta|\chi)$, the probability

$$P_\Theta[p(\Theta|\chi) \geq p(\Theta_0|\chi)|\chi] \tag{17}$$

can be estimated as the proportion of samples for which $p(\Theta|\chi) \geq p(\Theta_0|\chi)$, where $p(\Theta|\chi)$ is evaluated over the sample values of Θ. In our context, however, this approach is difficult to implement because of difficulties in computing marginal distributions from the posterior distribution of the parameter vector.

We have, therefore, proceeded as follows. In order to determine whether or not an input variable carries useful information, we compared an architecture that received this input variable with an architecture that did not. The former architecture is referred to as a full architecture; the latter is called a reduced architecture. The reduced architecture may be thought of as having fewer parameters than the full architecture because it does not contain expert and gating network parameters corresponding to the input variable of interest. Alternatively, the two architectures may be considered to have the same number of parameters, though the parameters in the reduced architecture corresponding to the input being studied are set to a value of zero. The parameter vector for the full architecture is denoted by $\Theta = (\Theta_1, \Theta_2)$, where Θ_2 is the vector of expert and gating network parameters corresponding to the variable of interest

and Θ_1 is the vector of remaining parameters. We estimated the following probability using the simulated Θ_2 values from the Gibbs sampler to obtain an approximation to the Bayesian p-value:

$$P_{\Theta_2}[p(\hat{\Theta}^1, \Theta_2|\chi) \leq p(\tilde{\Theta}_1, \Theta_2 = 0|\chi)|\chi] \quad (18)$$

where $(\hat{\Theta}_1, \hat{\Theta}_2)$ is the mode of the posterior $p(\Theta|\chi)$ and $\tilde{\Theta}_1$ is the mode of the posterior distribution $p(\Theta_1, \Theta_2 = 0|\chi)$.

To test the procedure for discriminating useful from nuisance inputs, we modified the breast cancer classification task by adding an extra input variable to the original set of nine inputs. The values of this new input were uncorrelated noise; they were independently sampled from a normal distribution with a mean of zero and a variance of one. The HME architecture consisted of two expert networks and one gating network, as this structure was found to be appropriate for this task (see Section 3). The Bayesian p-value (averaged over five chains) for the new input variable was 0.93 suggesting that this noisy input has no predictive value. In contrast, the Bayesian p-value for the input concerning the breast tumor clump thickness was 0.06 suggesting that this input variable has some prognostic value in determining whether the breast mass is benign or malignant.

For the speech recognition task, we investigated the importance of the two inputs giving the first two formant values for the speech utterances. The HME architecture had a 1-1 structure. The Bayesian p-values for the first and second formants were 0.01 and 0.04, respectively. Thus there is evidence to suggest that both inputs are important for classifying the vowel utterances.

5. SUMMARY

In summary, because no statistical model shows good performance on all tasks, the model selection problem is unavoidable; investigators must decide which model is best at summarizing the data for a particular task. This article has presented an approach to the model selection problem in hierarchical mixtures-of-experts architectures. One part of this approach examines the indicator variables that index the expert network which generated each data item. These variables are used to estimate the worth of each expert so that relatively unused experts can be pruned from the tree-structure. A second part of this approach uses a Bayesian hypothesis testing procedure in order to differentiate inputs that carry useful information from nuisance inputs. Simulation results suggest that the approach presented here adheres to the dictum of Occam's razor; simple architectures that are adequate for summarizing the data are favored over more complex structures.

REFERENCES

Belew, R. K. (1993). Interposing an ontogenic model between genetic algorithms and neural networks. In S. J. Hanson, J. D. Cowan, & C. L. Giles (Eds.), *Advances in neural information processing systems 5*. San Mateo, CA: Morgan Kaufmann.

Box, G. E. P.. & Tiao, G. (1973). *Bayesian inference in statistical analysis*. Reading, MA: Addison-Wesley.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth International Group.

Bridle, J. (1989). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In F. Fogelman-Soulie, & J. Hérault (Eds.), *Neuro-computing: algorithms, architectures, and applications*. New York: Springer-Verlag.

Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., and Hopfield, J. (1987). Large automatic learning, rule extraction, and generalization. *Complex Systems*, **1**, 877–922.

Fahlman, S. E., & Lebiere, C. (1990). The cascade-correlation learning architecture. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 2*. San Mateo, CA: Morgan Kaufmann.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, **7**, 457–472.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, **3**, 79–87.

Jordan, M. L., & Jacobs, R. A. (1992). Hierarchies of adaptive experts. In J. E. Moody, S. J. Hanson, & R. P. Lippmann (Eds.), *Advances in neural information processing systems 4*. San Mateo, CA: Morgan Kaufmann.

Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, **6**, 181–214.

Le Cun, Y., Denker, J. S., & Solla, S. A. (1990). Optimal brain damage. In D. S. Touretzky (Ed.), *Advances in neural information processing systems 2*. San Mateo, CA: Morgan Kaufmann.

Lee, P. M. (1991). *Bayesian statistics: an introduction*. New York: Oxford University Press.

Mackay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation*, **4**, 448–472.

Mangasarian, O. L., & Wolberg, W. H. (1990). Cancer diagnosis via linear programming. *SIAM News*, **23**, 1–18.

McCullagh, P., & Nelder, J. A. (1989). *Generalized linear models*. London: Chapman and Hall.

McLachlan, G. J., & Basford, K. E. (1988). *Mixture models: inference and applications to clustering*. New York: Marcel Dekker.

Müller, P. (1991). *A genetic approach to posterior integration and Gibbs sampling*. (Technical Report 1991–09), Department of Statistics, Purdue University.

Neal, R. M. (1991). *Bayesian mixture modeling by Monte Carlo simulation*. (Technical Report CRG-TR-91-2), Department of Computer Science, University of Toronto.

Nowlan, S. J., & Hinton, G. E. (1991). Evaluation of adaptive mixtures of competing experts. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in neural information processing systems 3*. San Mateo, CA: Morgan Kaufmann.

Peng, F., Jacobs, R. A., & Tanner, M. A. (1996). Bayesian

inference in mixtures-of-experts and hierarchical mixtures-of-experts architectures with an application to speech recognition. *Journal of the American Statistical Association*, in press.

Peterson, G. E., & Barney, H. L. (1952). Control methods used in a study of vowels. *Journal of the Acoustical Society of America*, **24**, 175–184.

Taner, M. A. (1993). *Tools for statistical inference: methods for the exploration of posterior distributions and likelihood functions*. New York: Springer-Verlag.

Waterhouse, S. R., & Robinson, A. J. (1994). Classification using hierarchical mixtures of experts. In J. Vloutzos, J.-N. Hwang, & E. Wilson (Eds.), *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*. New York: IEEE Press.

Wei, G. C. G., & Tanner, M. A. (1990). Calculating the content and boundary of the HPD region via data augmentation. *Biometrika*, **77**, 649–652.

Weigend, A. S., Rumelhart, D. E., & Huberman, B. A. (1991). Generalization by weight-elimination with application to forecasting. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in neural information processing systems 3*. San Mateo, CA: Morgan Kaufmann.

Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, **87**, 9193–9196.

Wolberg, W. H., Tanner, M. A., Loh, W. Y., & Vanichsetakul, N. (1987). Statistical approach to fine needle aspiration diagnosis of breast masses. *Acta Cytologica*, **31**, 737–741.