

A Framework for Nonparametric Regression Using Neural Networks

Herbert K. H. Lee
ISDS, Duke University

Abstract

Neural networks are a useful statistical tool for nonparametric regression. In this paper, I develop a methodology for nonparametric regression within the Bayesian paradigm. I address the problem of model selection and model averaging, particularly the problem of searching the model space in terms of both the optimal number of hidden nodes in the network as well as the best subset of explanatory variables. I implement this with a method I call Bayesian Random Searching (BARS). I also demonstrate how to use a noninformative prior for a neural network, which is useful because of the difficulty in interpreting the parameters. Finally, I present results on the asymptotic consistency of the posterior for neural network regression.

Keywords: Bayesian statistics; Model selection; Model averaging; Noninformative prior; Bayesian random searching; Asymptotic consistency

1 Introduction

Nonparametric regression addresses the problem of trying to fit a model for a variable Y on a set of possible explanatory variables X_1, \dots, X_p , where the relationship between X and Y may be more complicated than a simple linear relationship. Neural networks are a useful tool for nonparametric regression (see for example, Stern 1996). However, several important questions arise when doing neural network regression: What is the optimal number of hidden nodes and subset of explanatory variables to use? How much uncertainty is there in my choice of model? Is this method asymptotically consistent? This paper is meant to address these questions in a Bayesian framework.

The methods of this paper differ from competing Bayesian neural network regression methods (MacKay 1992; Neal 1996; Müller and Rios Insua 1998b) in two main ways. First, I propose a simple noninformative prior, rather than the complex hierarchical priors typically used. Second, I use the full posterior to address the question of model selection for both the size of the network and the subset of explanatory variables. As part of the implementation of model selection, I use Bayesian Random Searching, which builds upon model space searching work by Raftery, Madigan, and Hoeting (1997) and is similar to the approach of Chipman, George, and McCulloch (1998) in their implementation of Bayesian CART. Finally I present results on the asymptotic properties of the posterior for neural network models.

An example which illustrates the ideas of this paper is the ozone data of Breiman and Friedman (1985). There are 330 observations of the response variable, groundlevel ozone (as a pollutant) in Los Angeles, and nine explanatory variables: VH, the altitude at which the pressure is 500 millibars; WIND, the wind speed (mph); HUM, the humidity (%); TEMP, the temperature ($^{\circ}$ F); IBH, the temperature inversion base height (feet); DPG, the pressure gradient (mm Hg); IBT, the inversion base temperature (degrees F); VIS, the visibility (miles); and DAY, the day of the year. The goal is to model groundlevel ozone concentration as a function of the day of the year and the eight meteorological variables. Figure 1 shows the matrix of all two-way scatterplots of the variables. Note that the response variable, ozone, is in the first row and column. Ozone appears to be a nonlinear function of each of the explanatory variables, and so a nonparametric regression procedure is called for. In this case, I use a neural network.

Figure 2 shows a sample fitted curve for ozone versus the day of the year, and it is clearly nonlinear. The details of fitting a neural network model are in the following sections. For now, I just want to point out two main issues that will be addressed in this paper. First, there is an issue of how much smoothing is necessary in the fit. The number of hidden nodes controls the amount of smoothing, so one needs to be able to find

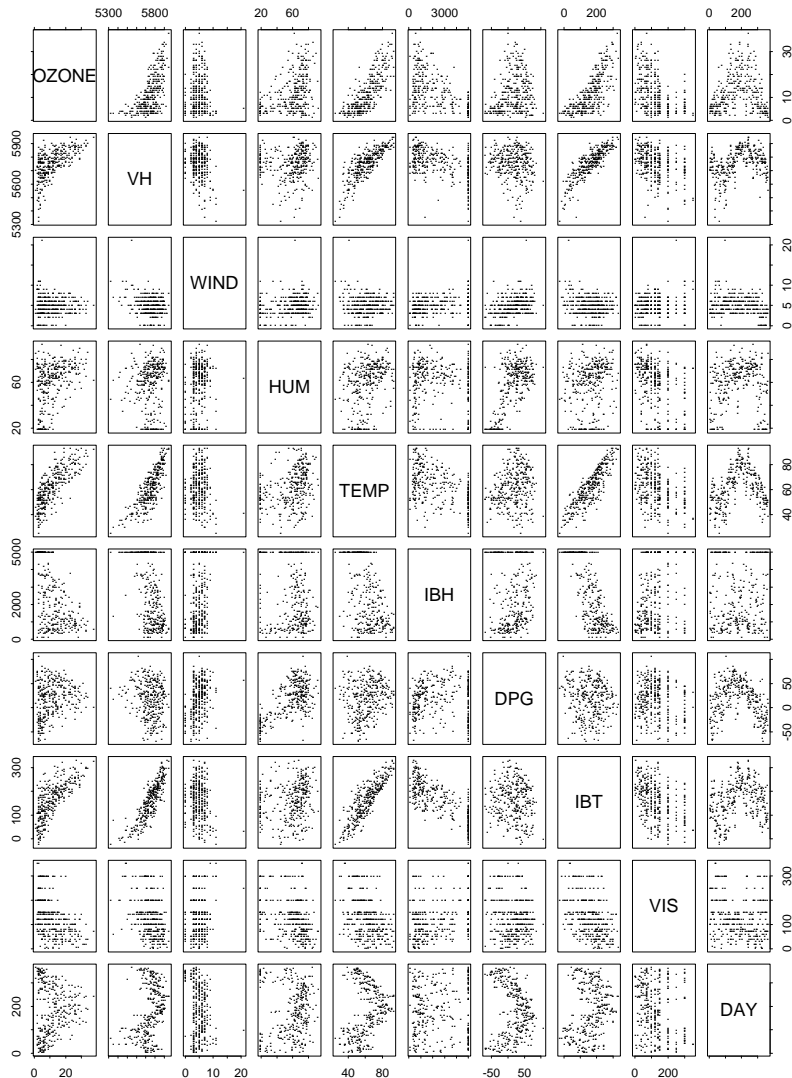


Figure 1: Ozone Data

the optimal number of hidden nodes. Second, the explanatory variables are highly correlated, for example, there is very strong linear correlation between TEMP and IBT. Because of these dependencies between the explanatory variables, predictive performance will likely be increased by using only a subset of the variables. This paper will discuss a method for finding those subsets with highest posterior probability.

2 Neural Networks

Neural network regression is a special case of nonparametric regression. The idea of nonparametric regression is to use models of the form

$$Y_i = f(\mathbf{X}_i) + \varepsilon_i,$$

where $f \in \mathcal{F}$, some class of regression functions, and ε is *iid* additive error with mean zero and constant variance. Sometimes normality of ε is assumed. The main distinction between the competing nonparametric methods is the class of functions, \mathcal{F} , to which f is assumed to belong. In all cases, \mathcal{F} is taken to be some class rich enough to be able to sufficiently approximate a very large set of possible regression functions,

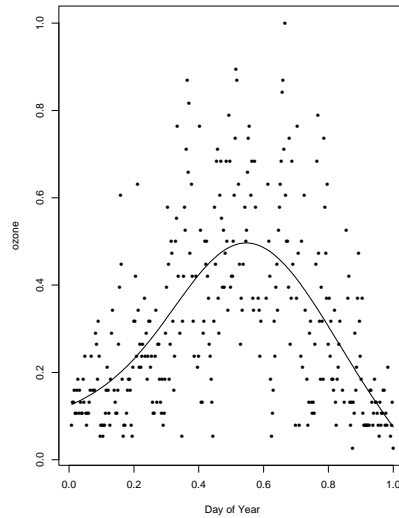


Figure 2: Fitted Function for Ozone Concentration on Day of Year

such as all continuous functions. The advantage of nonparametric regression over simpler parametric models (such as linear regression) is the increased flexibility and reduction in assumptions of the model. One can get an improved fit for the data when the relationship between the explanatory and response variables is complicated.

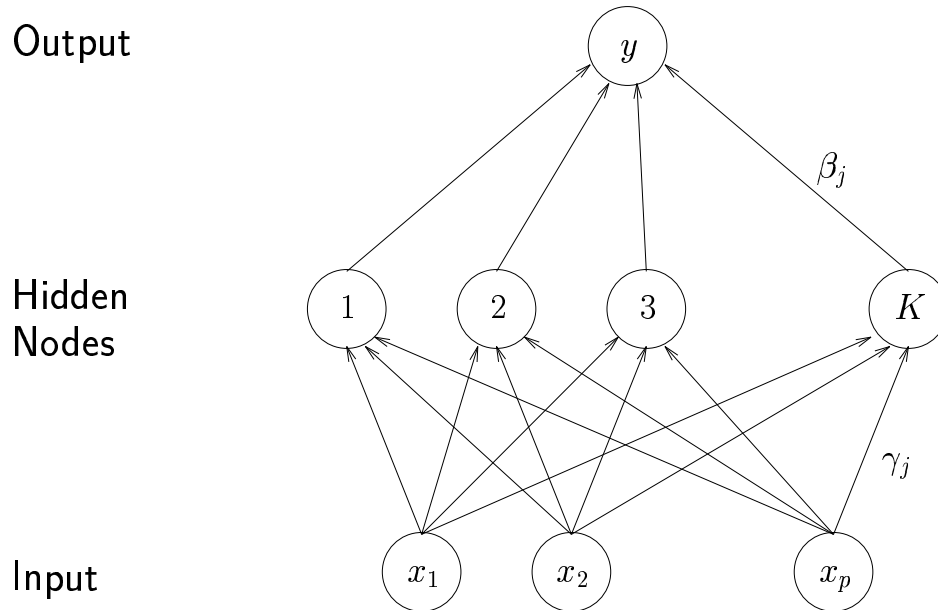


Figure 3: Neural Network Model Diagram

Neural networks are a class of nonparametric regression models that originated as an attempt to model the act of thinking by modeling neurons in the brain. A diagram of a neural network model is shown in Figure 3. The underlying statistical idea of a neural network is that it uses logistic functions to form a basis over the space of continuous functions, with each hidden node corresponding to a basis function. With an infinite number of hidden nodes, one can match any function arbitrarily closely. In practice, only a finite number of hidden units are used, forming an approximation to the best regression function. Neural network

models can be viewed as statistical models of the form

$$Y_i = \beta_0 + \sum_{j=1}^K \beta_j \Psi(\gamma'_j \mathbf{X}_i) + \varepsilon_i, \quad (1)$$

where K is the number of hidden nodes, the β_j 's are the weights of the basis functions, the $\gamma_{j,h}$'s are location and scale parameters (with h indexing covariates), $\varepsilon_i \stackrel{iid}{\sim} N(0, \sigma^2)$ is the error term, and Ψ is the logistic function:

$$\Psi(z) = \frac{1}{1 + \exp(-z)}.$$

Here I consider only feed-forward networks (no directed cycles in Figure 3) with one hidden layer of nodes with logistic activation functions and with one linear output (as per Equation (1)). Such models have been shown by several people (e.g., Hornik, Stinchcombe, and White 1989) to be able to approximate a function arbitrarily closely as the number of hidden nodes gets large. Furthermore, the posterior distributions for these neural network models are asymptotically consistent for the true regression function, as will be described in Section 7. Neural networks work well in approximating both smooth functions and those that have breakpoints in them, since a breakpoint can be modeled effectively with a single extra hidden node. Such discontinuities can be difficult for other nonparametric methods to handle.

As more hidden nodes are used, one can get a better fit to the data at hand. For a finite dataset, enough hidden nodes would give a perfect fit. Similarly, adding explanatory variables to a regression improves the fit to the particular data at hand. In both cases, however, overfitting the data may result in poor predictive performance. To take an extreme case, if a variable is uncorrelated with the response variable, it should not be included in the model because while it may give a slight reduction in the error, it will not improve predictive performance at all. Another important case is when explanatory variables are highly correlated; a subset of these variables will typically give better predictive performance than all of them taken together because using all of them will lead to overfitting. Thus there is the need to balance the desire to use more hidden nodes and more explanatory variables for increased accuracy and the desire to limit them for increased predictive performance. In the Bayesian approach, one specifies a prior over the space of possible models, then chooses the model with highest posterior probability (see e.g., Kass and Raftery 1995). This paper will address the problem of estimating these probabilities and of searching the model space for models of high posterior probability. A benefit of the Bayesian approach is that one gets estimates on the uncertainty associated with the choice of a particular model, in addition to the standard estimates of the uncertainty for the fit of the model.

A standard method of fitting neural network models is backpropagation, which is essentially a gradient-descent algorithm. It has been discovered independently multiple times and was made popular by Rumelhart, Hinton, and Williams (1986). In practice, backpropagation has some problems, such as frequent convergence to a local maximum, rather than a global maximum. Statistically, it is useful to treat this problem as merely a likelihood maximization problem, and to use a numerical routine from one's favorite statistical package. Within the Bayesian approach, Markov Chain Monte Carlo (MCMC) methods can be used to get a sample from the posterior which then can be used to estimate any posterior quantities of interest. The approach of this paper is to use an MCMC algorithm that is described in the next section.

3 Priors for Neural Network Models

In the Bayesian paradigm, one needs to specify a prior for the parameters in the model and then use Bayes' Theorem to get the posterior. A typical approach is to choose a prior which reflects one's personal beliefs. In the case of a neural network, the parameters can be extremely difficult to interpret. The effect of the coefficients of the logistic basis functions (β in equation (1)) can be difficult to visualize because the logistic functions are nonlinear and can combine in unexpected ways. The coefficients inside the logistic function are even less interpretable.

An alternative approach is to use a noninformative prior distribution, one which does not contain any information about the parameters (such as Lebesgue measure for a real-valued parameter) or Jeffreys' prior (Jeffreys 1961; Bernardo 1979). Such priors are typically "improper" in that they do not have a finite

integral. However, in many problems, the posterior will still be proper. Thus an improper prior can be a way to do Bayesian inference without having to specify a proper prior (in a sense, one is specifying that one’s beliefs are equally spread out over all possible values).

A further advantage of the noninformative prior approach is to avoid the complicated hierarchical priors that are typically used for neural network models (MacKay 1992; Neal 1996; Müller and Rios Insua 1998b). Trying to accurately express a prior belief through a complex hierarchy can be difficult, and all of the aforementioned authors fail to advocate a fully informative prior for general use. Both Neal and Müller and Rios Insua use vague priors at the top level of the hierarchy (as an approximation to a truly noninformative prior). MacKay uses a data-dependent prior at the top of his hierarchy.

In contrast, I propose the much simpler prior

$$\pi(\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2) \propto \frac{1}{\sigma^2}, \quad (2)$$

which puts equal mass everywhere along the parameter space (or for the logarithm in the case of the variance). This prior is the same as the standard prior for linear regression. However, this prior is improper and leads to an improper posterior. In practice, I use the restricted version

$$\pi_n(\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2) \propto \frac{1}{\sigma^2} \mathbf{I}_{\{(\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2) \in \Omega_n\}}, \quad (3)$$

where Ω_n is a particular subspace of the parameter space that will be described shortly (equation (4)). This is a data-dependent prior where the data dependence will go to zero asymptotically. It approximates (2) with a sequence of flat priors on increasing compact sets. The idea of using such a restricted prior comes from the mixture model literature (Diebolt and Robert 1994; Wasserman 1998). A heuristic justification is as follows: one can approximate a neural network with linear combinations of indicator functions, instead of logistic functions. In order to fit the model, there must be at least one data point between the changepoints of the indicator functions. If two indicator functions do not have a data point between them, then the model is not uniquely determined, analogously to linear regression when the explanatory variables are linearly dependent. The restriction set Ω_n is analogous to the set of indicator functions with at least one data point between them.

The key here is to ensure that the logistic basis functions are linearly independent. To make this more formal, define

$$z_{ij} = \left[1 + \exp \left(-\gamma_{j0} - \sum_{h=1}^p \gamma_{jh} x_{ih} \right) \right]^{-1}$$

and let \mathbf{Z} be the matrix with elements (z_{ij}) . The fitting of the vector $\boldsymbol{\beta}$ is merely a multiple linear regression on the design matrix \mathbf{Z} . To ensure the linear independence of the columns of \mathbf{Z} , one needs a restriction on \mathbf{Z} . A sufficient condition for linear independence is that the determinant of $\mathbf{Z}^t \mathbf{Z}$ is larger than some positive value D_n . One can let $D_n \rightarrow 0$ as $n \rightarrow \infty$ as long as $D_n > 0$ for all n . To ensure propriety of the posterior, one also needs to bound the individual γ ’s such that $|\gamma_{jh}| < C_n$ for all j, h (where C_n can increase with n), as there are triangular regions of infinite area where the logistic functions converge to indicator functions and thus the likelihood converges to a non-zero value (unlike most parametric problems where the likelihood converges to zero in the tail areas). Thus

$$\Omega_n = \{(\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma) : |\gamma_{jh}| < C_n, |\mathbf{Z}^t \mathbf{Z}| > D_n\} \quad (4)$$

I will show later (equations (5) and (6)) that these adjustments to the prior result in a prior which is asymptotically equivalent to the unrestricted prior. Furthermore, with this choice of prior, the posterior distribution is asymptotically consistent both in the sense that Hellinger neighborhoods of the true regression function have posterior probability tending to one, and that the expected mean square error of the posterior predictive regression function goes to zero in probability, as will be explained in Section 7

MCMC is used to fit the model. σ^2 and $\boldsymbol{\beta}$ can be sampled directly using Gibbs steps. $\boldsymbol{\gamma}$ can be sampled via Metropolis updates of the separate vectors $\boldsymbol{\gamma}_j$. For each j from $1, \dots, k$, one would do the following Metropolis step:

1. Generate a candidate $\tilde{\gamma}_j \sim N(\gamma_j, \tau^2)$, where τ is a tuning parameter (chosen to give an acceptance rate of around 40%).
2. Re-compute \mathbf{Z} with $\tilde{\gamma}_j$ and compute $|\mathbf{Z}^t \mathbf{Z}|$.
3. If $|\mathbf{Z}^t \mathbf{Z}| > D_n$ and $|\gamma_{jh}| < C_n$ for all j, h , accept $\tilde{\gamma}_j$ with probability $\min\left(1, \frac{f(\beta, \tilde{\gamma}, \sigma^2 | \mathbf{y})}{f(\beta, \gamma, \sigma^2 | \mathbf{y})}\right)$; otherwise, reject the candidate and keep the current value of γ_j .

It is straightforward to show that this method will produce a proper posterior. Denote the likelihood by L_n . Let π be the noninformative prior of equation (2), π_n be the restricted prior of equation (3), and Ω_n the restriction set of equation (4). Let D_n decrease to 0 (for example, $D_n = 1/n$) and let C_n increase with n (for example, $C_n = 10,000 + n$). Here one can clearly see how π_n is a data-dependent prior, as Ω_n depends upon x through \mathbf{Z} . First the likelihood is re-written so that β can be integrated out, which involves completing the square for β .

$$\begin{aligned}
L_n &= f(\beta, \gamma, \sigma | \mathbf{y}) = (2\pi\sigma^2)^{-n/2} \exp \left[-\frac{1}{2\sigma^2} \sum_{i=1}^n \left(\sum_{j=0}^k \beta_j z_j - y_i \right)^2 \right] \\
&= (2\pi\sigma^2)^{-n/2} \exp \left[-\frac{1}{2\sigma^2} (\mathbf{Z}\beta - \mathbf{Y})^t (\mathbf{Z}\beta - \mathbf{Y}) \right] \quad \text{in vector notation} \\
&= (2\pi\sigma^2)^{-\frac{k+1}{2}} |\mathbf{Z}^t \mathbf{Z}|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2\sigma^2} [\beta - (\mathbf{Z}^t \mathbf{Z})^{-1} \mathbf{Z}^t \mathbf{Y}]^t (\mathbf{Z}^t \mathbf{Z}) [\beta - (\mathbf{Z}^t \mathbf{Z})^{-1} \mathbf{Z}^t \mathbf{Y}] \right\} * \\
&\quad (2\pi\sigma^2)^{-\frac{n-(k+1)}{2}} |\mathbf{Z}^t \mathbf{Z}|^{\frac{1}{2}} \exp \left\{ -\frac{1}{2\sigma^2} [\mathbf{Y}^t \mathbf{Y} - \hat{\mathbf{Y}}^t \hat{\mathbf{Y}}] \right\} \\
&= f(\beta | \gamma, \sigma, \mathbf{y}) f(\gamma, \sigma | \mathbf{y}),
\end{aligned}$$

where $\hat{\mathbf{Y}}$ is the vector of fitted values, $\hat{\mathbf{Y}} = E[\mathbf{Y} | \mathbf{X}]$. Note that $f(\beta | \gamma, \sigma, \mathbf{y})$ is a proper density as long as $|\mathbf{Z}^t \mathbf{Z}| > 0$, which is true over Ω_n . Denote by Γ_n the subspace of Ω_n that relates to all of the γ parameters. Then the posterior is proper:

$$\begin{aligned}
\int L_n \pi_n &= \int_{\Omega_n} f(\beta | \gamma, \sigma, \mathbf{y}) f(\gamma, \sigma | \mathbf{y}) \left[\frac{1}{\sigma^2} \right] d\beta d\sigma d\gamma \\
&= \int_{\Gamma_n} \int \left[\int f(\beta | \gamma, \sigma, \mathbf{y}) d\beta \right] \frac{1}{\sigma^2} f(\gamma, \sigma | \mathbf{y}) d\sigma d\gamma \\
&= \int_{\Gamma_n} \int \frac{1}{\sigma^2} (2\pi\sigma^2)^{-\frac{n-(k+1)}{2}} |\mathbf{Z}^t \mathbf{Z}|^{\frac{1}{2}} \exp \left\{ -\frac{1}{2\sigma^2} [\mathbf{Y}^t \mathbf{Y} - \hat{\mathbf{Y}}^t \hat{\mathbf{Y}}] \right\} d\sigma d\gamma \\
&\leq \int_{\Gamma_n} \int \frac{1}{\sigma^2} (2\pi\sigma^2)^{-\frac{n-(k+1)}{2}} |\mathbf{Z}^t \mathbf{Z}|^{\frac{1}{2}} d\sigma d\gamma \\
&= (2\pi)^{-\frac{n-(k+1)}{2}} (n-k+1)^{-1} \int_{\Gamma_n} |\mathbf{Z}^t \mathbf{Z}|^{\frac{1}{2}} d\gamma
\end{aligned}$$

The last integral is finite because Γ_n is a bounded set and the integrand is finite. Thus the posterior is proper.

In addition to showing that the adjusted prior leads to a proper posterior, it is also important to show that the adjusted prior is asymptotically equivalent to the original improper prior, which can be shown in both a global and local sense. First, it is clear that, for any compact set κ ,

$$\int_{\kappa} |\pi_n - \pi| = \int_{\kappa} |\pi I_{\Omega_n} - \pi| \rightarrow 0 \text{ as } n \rightarrow 0 \tag{5}$$

because $|\mathbf{Z}^t \mathbf{Z}|$ must be non-zero for the true function (or else it would have one fewer node), and because for a large enough n , Ω_n will contain all elements of κ that satisfy the determinant condition. This equation

says that, in the limit as the sample size grows, π_n converges to π on all compact sets. In this sense, the two priors are “asymptotically globally equivalent” (Wasserman 1998).

Second is a condition of “asymptotic local equivalence”. It also relates to correct second-order frequentist coverage properties (Wasserman 1998). The key is that the original and adjusted priors have the same local properties (while the adjusted prior is better behaved in the tails). Suppose there exists a true value of the parameters, θ_0 . Then for large n ,

$$\left| \frac{\partial \log \pi_n}{\partial \theta_0} - \frac{\partial \log \pi}{\partial \theta_0} \right| = O_p \left(\frac{1}{\sqrt{n}} \right) \quad (6)$$

because if n is large enough, θ_0 will be contained in Ω_n .

4 Model Selection and Model Averaging

Model selection for a neural network entails both a selection of the number of hidden nodes, as well as a selection of the subset of explanatory variables. In either case, using more than necessary will overfit the data and produce poor predictive performance. Thus one needs to balance the desire to increase the fit to the data with the need to limit overfitting and to increase predictive performance.

Methods for model selection include cross-validation (Stone 1974) as well as the penalized likelihood-based methods such as the Akaike Information Criterion (Akaike 1974) and the Bayesian Information Criterion (BIC) (Schwarz 1978). In Bayesian inference, one can simply pick the model with highest posterior probability. The Bayesian approach also lends itself very nicely to prediction, in the guise of model averaging. A good review article on model averaging is Hoeting et al. (1999). In some cases, one will find more than one model with high posterior probability, and these models will give different predictions (see, for example, the heart attack data in Raftery 1996). Model averaging can also result in dramatic decreases in prediction errors (Raftery, Madigan, and Hoeting 1997). Picking only a single model will grossly underestimate the variability of the estimate, since it ignores the fact that another model with significant posterior probability made a different prediction. Instead, one should calculate predictions (or statistics thereof) by using a weighted average over all models in the search space, where the weights are the posterior probabilities of the models (Leamer 1978; Kass and Raftery 1995). Let Y be the response variable being predicted, D the data (the observed values of the explanatory and response variables), and M_i the models of interest for $i \in \mathcal{I}$. Then the posterior predictive distribution of Y is

$$P(Y|D) = \sum_{i \in \mathcal{I}} P(Y|D, M_i) P(M_i|D),$$

where $P(Y|D, M_i)$ is the marginal posterior predictive density given a particular model (with all other parameters integrated out), and $P(M_i|D)$ is the posterior probability of model i .

Estimating the posterior probabilities of the models, $P(M_i|D)$, is a computationally difficult problem. In theory, it is very straightforward; by Bayes’ Theorem:

$$P(M_i|D) = \frac{P(D|M_i)P(M_i)}{\sum_j P(D|M_j)P(M_j)},$$

The trouble is the term $P(D|M_i)$, the marginal probability of the data, which involves integrating out the parameters in the model. For a neural network, this is an analytically intractable integral. The contours of the posterior are irregularly shaped, and the posterior is not unimodal (even after accounting for symmetries induced by the equivalence of some parameters in the model). It turns out that the BIC is useful in approximating the posterior probabilities of the models (Lee 1998) as the BIC is an approximation to the log of the Bayes factor for comparing that model to the null model (Schwarz 1978). Recall that the BIC for model i is defined as

$$BIC_i = L_i - \frac{1}{2} p_i \log n,$$

where L_i is the maximum of the log-likelihood, n is the sample size, and p_i is the number of parameters in model i .

For the prior over the space of models, I use the noninformative prior that puts equal mass on each model, i.e. $P(M_i) = P(M_j)$ for all i and j . The BIC approximation then becomes

$$P(M_i|Y) = \frac{P(Y|M_i)}{\sum_j P(Y|M_j)} \approx \frac{e^{BIC_i}}{\sum_j e^{BIC_j}}. \quad (7)$$

Note that the Bayesian approach automatically takes care of the balance between improving fit and not overfitting, because adding additional variables or nodes that do not sufficiently improve the fit will dilute the posterior, causing a lower posterior probability for the model. This approach, in addition to being conceptually straightforward, also has the advantage that it can be used simultaneously on both the problem of choosing a subset of explanatory variables, and on the problem of choosing the number of hidden nodes for the network. Furthermore, in many cases the BIC has been shown to be asymptotically consistent for choosing the true model. While this has not yet been shown for neural networks, Keribin (1997) has shown it to be true for mixture models, which have many similarities to neural networks. It seems plausible that it would be true for neural networks as well.

This approach utilizes the full posterior in dealing with uncertainty of the optimal number of hidden nodes and the optimal subset of explanatory variables. In contrast, frequentist neural network methods cannot fully measure the uncertainty associated with these methods. Competing Bayesian methods do not use the full posterior. Müller and Rios Insua (1998a) use reversible-jump MCMC to get the full posterior for the number of hidden nodes, but do not tackle the explanatory variable selection problem. MacKay (1992) uses a Gaussian approximation to get an estimate of the posterior probabilities of the models for different numbers of hidden nodes. Neal (1996) advocates using a larger number of nodes than necessary, and using a suitable prior to reduce the effects of the additional nodes to avoid overfitting. For examining the explanatory variables, MacKay (1994) and Neal (1996) advocate a method called Automatic Relevance Detection (ARD), which utilizes an additional layer of hyperparameters in the model where each explanatory variable has an associated hyperparameter that relates to the magnitude of the parameters associated with that explanatory variable. A prior is placed over the hyperparameters, and the full posterior for the model is computed. If any of the variable hyperparameters turns out to be small, it indicates that that variable has little effect on the model, and could be dropped. A variable with a larger effect would necessarily have a larger associated hyperparameter. However, this method leaves open the question of how small a hyperparameter has to be to be dropped, and how to choose the prior for the hyperparameters. My approach of directly estimating the posterior probabilities of the models avoids these additional complications.

5 Bayesian Random Searching

To compare models, one needs to estimate their posterior probabilities. However, for even a moderately sized problem, there are too many possible models to be able to estimate the posteriors for all of them. For example, if there are 9 explanatory variables, there are 2^9 possible subsets of explanatory variables to use, and then one could consider a range of possible numbers of hidden nodes to use for each of these subsets. It would not be feasible to try all of these models. Hence one needs a strategy for searching the model space.

Two traditional search methods are stepwise regression and Leaps and Bounds (Furnival and Wilson 1974). Some Bayesian methods are Occam’s Window (Raftery and Madigan 1994) and Markov Chain Monte Carlo Model Composition (MC^3) (Raftery, Madigan, and Hoeting 1997). I will present a technique, Bayesian Random Searching (BARS), motivated by MC^3 .

The idea behind MC^3 is to create a Markov chain with state space equal to the set of models under consideration and equilibrium distribution equal to the posterior probabilities of the models. Thus if one simulates this chain, the proportion of time that the chain spends visiting each model is a simulation-consistent estimate of the posterior probability of that model. To create such a chain, let the transition probabilities between two models be as follows:

1. The probability of moving from the current model to a model which differs by two or more variables or nodes (differing by inclusion or exclusion) is zero.
2. The probability of moving from the current model to one which differs by exactly one variable or node (either one more or one less) is $\frac{1}{r} \min\{1, \frac{Pr(M'|D)}{Pr(M|D)}\}$, where r is the number of parameters being

considered (i.e., the dimension of the search space), $Pr(M'|D)$ is the posterior probability of the model being moved to, and $Pr(M|D)$ is the posterior distribution of the model being moved from.

3. Otherwise, the chain stays in its current state.

I use the BIC approximation for estimating the probabilities involved in the MCMC step of moving between models.

While MC^3 may be simulation-consistent, there have not been any studies done on how long the simulation needs to run in order to reach its equilibrium state. Furthermore, the BICs of the models visited are computed, but not used directly in the final estimation of posterior probabilities. At some level, it seems wasteful to throw away this information and rely solely on the steady state properties of the chain. Instead, one could use the same Markov chain simulation, but keep a record of all of the BICs for the models visited. To get the posterior probability of each model, one would just use the BIC approximation if the model was visited, and exclude the model if it was not visited. In practice, this may be more accurate than MC^3 because it does not rely on any of the steady state properties of the chain. The chain is merely used as a mechanism for effectively searching through the large model space, hopefully finding all of the models with high posterior probability. I shall refer to this method as Bayesian Random Searching (BARS). This approach is similar to that of Chipman, George, and McCulloch (1998) .

Stochastic searching algorithms such as BARS and MC^3 have a major advantage over deterministic algorithms in that they are less prone to become stuck in local maxima of the model space. It is common to be able to find a model which has a larger BIC than all models that differ by one node or variable, yet is not the model with the largest BIC. Stepwise algorithms cannot generally escape such a local maximum in the BIC. On the other hand, the random component of BARS allows it a chance to escape from any local maximum.

In my implementation of BARS, since I use a BIC approximation for model probabilities, I do not need to get the full posterior via MCMC for each model, but only need the maximum likelihood estimate to compute the BIC. In practice, I use code from Sarle (1994) , of the SAS Institute, that uses proc NLP in SAS to find the maximum likelihood estimates of the parameters in a neural network. At the end of the search, after the models of highest posterior probability have been identified, I then go back and fit those models (typically only a few, or even one model) with MCMC to get the full posterior distribution of the parameters.

A final note about searching the model space needs to be made in the context of neural networks. In many statistical applications, one can fit the model with reasonable confidence (e.g. linear regression). However, fitting a neural network, in either a frequentist or Bayesian framework, involves the use of an iterative algorithm which could find a local maximum rather than a global maximum. One may want to keep in mind that a model visited during a search algorithm may not necessarily be fit correctly. This is a further advantage for search algorithms such as BARS and MC^3 , in that one gets multiple chances to fit each model, and so these algorithms are more robust with respect to difficulties in fitting an individual model.

6 Examples

6.1 Robot Arm Data

A dataset commonly used in Bayesian neural network papers is the robot arm data of MacKay (1992). The idea of the data is to model the relationship between the two joint angles of the arm (x_1 and x_2) and the resulting arm position in Cartesian coordinates (denoted by y_1 and y_2). In this case, the data were actually simulated from the true functions and Gaussian noise was added. The true model is

$$y_1 = 2.0 \cos(x_1) + 1.3 \cos(x_1 + x_2) + \varepsilon_1 \quad (8)$$

$$y_2 = 2.0 \sin(x_1) + 1.3 \sin(x_1 + x_2) + \varepsilon_2, \quad (9)$$

where $\varepsilon_i \stackrel{iid}{\sim} N(0, 0.05^2)$. The values for x_1 were originally generated uniformly from the intervals $[-1.932, -0.453]$ and $[+0.453, +1.932]$, and the values for x_2 were generated independently from a uniform on $[0.534, 3.142]$.

The data are divided into two groups: a training set of 200 observations, and a test set of 200 observations. The models are fit using only the training set, and then the models can be validated on the test set. Since

the true function is known, one could also generate additional data from the true model and test the fitted models on larger sets of data.

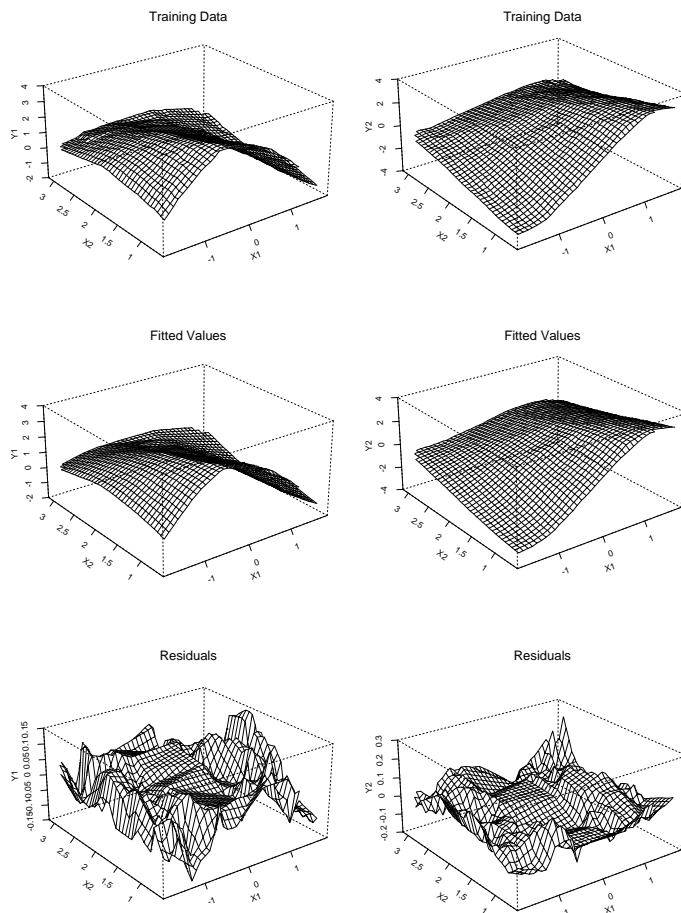


Figure 4: A Six-Node Model for the Robot Arm Data. The left column is the first response variable and the right column is the second.

BARS finds that all models of remotely large posterior probability include both explanatory variables. The model with highest posterior probability uses six hidden nodes, and it has nearly all of the posterior probability. The BIC of the seven-node model is about 4 smaller, which gives the six-node model about 98% of the posterior probability and the seven-node model the other 2%.

I then ran the MCMC code for 20,000 burn-in iterations and 20,000 draws from the posterior distribution. To get fitted values, I averaged the fitted values over each of the 20,000 draws from the posterior. Plots of the original data, the fitted values, and the residuals are shown in Figure 4. In the figure, the left column is y_1 and the right column is y_2 . The top row are plots of the observed data in the training set, with y on the vertical axis and x_1 and x_2 on the bottom axes. The center row shows the fitted values on the vertical axis. The bottom row shows the residuals on the vertical axis. In all cases, the interpolation function of S-Plus has been used to help create the picture, which does involve some small amount of additional smoothing, but is the best way to get a viewable image. Notice that the residuals are reasonably scattered.

I then used the MCMC output to fit the model on the 200 cases in the test data set. The theoretical optimum value for the mean squared error (MSE) is 0.005, and my model has an MSE of 0.00501. Plots of the test data, the fitted values, and the residuals are shown in Figure 5. The small mean square error shows that this model does indeed fit quite well. In order to see if the fit on this particular set of test data is just

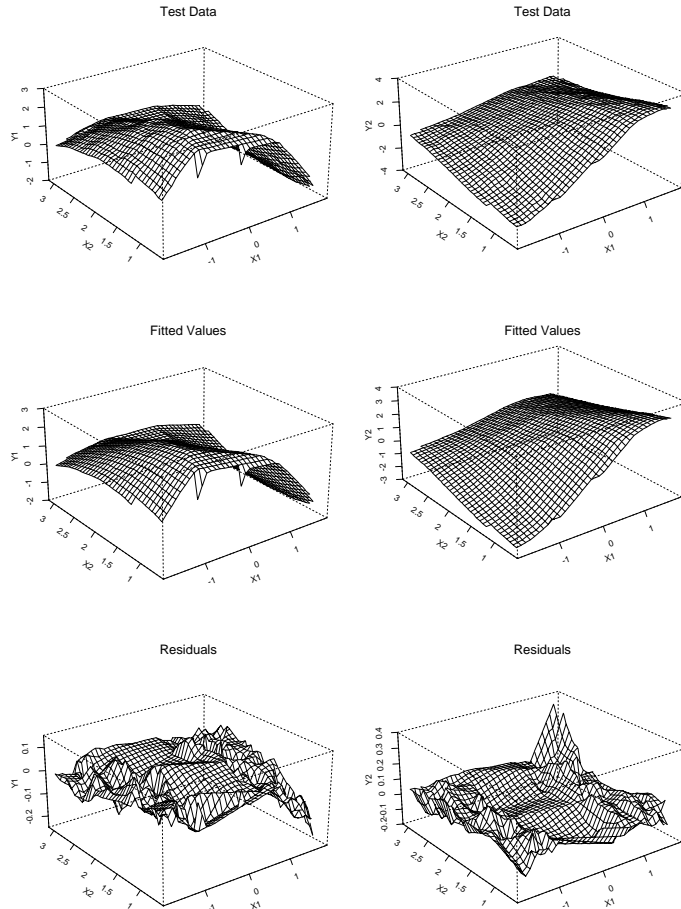


Figure 5: Validation Data for the Robot Arm Problem. The left column is the first response variable and the right column is the second.

a lucky break, I also generated a new test set of 1,000 observations from the original distribution and fit the model to these data. Indeed, the MSE rose to 0.00565. This is still a very good fit, and it is comparable to the fits achieved by competing methods on the test data (described in the next paragraph).

This dataset has also been analyzed by several others in the Bayesian neural network literature, in particular MacKay (1992), Neal (1996), and Müller and Rios Insua (1998a). Table 1 shows the MSE achieved on the test data by the methods of each of the above competing models. MacKay (1992) used what he calls the “evidence” for selecting the number of nodes. This evidence is a Gaussian approximation of a Bayes factor. Neither his model with highest evidence (which has ten nodes), or even his model with lowest MSE on the test data (which has nine nodes), fit as well as the model from this paper. Neal (1996) achieved similar results to the best model of MacKay by using hybrid Monte Carlo methods and averaging over his runs after removing a burn-in period. Neal used a network with sixteen hidden nodes, under the paradigm that choosing a good prior will balance the overfitting, and so one should use more nodes than necessary so that the MCMC mixes well. Neal also tried applying his model selection technique of Automatic Relevance Determination (ARD) (see Section 4). Neal demonstrated the viability of ARD by adding extra “noise” explanatory variables and then fitting a sixteen-node network to the data. In doing so, he managed to improve his fit a little in the sense that his MSE is smaller (see Table 1). In all cases, Neal was not concerned with finding an optimal number of hidden nodes. Finally, Müller and Rios Insua (1998a) applied their reversible-jump MCMC model to these data. They found the posterior distribution on the number of nodes to be the probabilities 0.30,

Method	Mean Square Error on Test Data
MacKay, Gaussian Approximation	
Solution with highest evidence	0.00573
Solution with lowest test error	0.00557
Neal, Hybrid Monte Carlo	
16-Node Model	0.00557
16 Nodes with ARD	0.00549
Müller and Insua	
Posterior Predictions	0.00620
Model from this paper	
Posterior Predictions	0.00501

Table 1: Comparison of Methods on the Robot Arm Data

0.53, 0.15, and 0.02 for six, seven, eight, and nine nodes respectively. Using the full posterior (averaged over all model sizes) gave a set of predictions on the test data that had MSE 0.0062. Compared to the above methods, my methods have found a model which is both more parsimonious (having fewer nodes) as well as better fitting (in that its MSE on the test data is smaller).

6.2 Ozone Data

The methods of this paper were also applied to the groundlevel ozone data of Breiman and Friedman (1985) that was introduced in Section 1 BARS found that the model with nearly all of the posterior probability was one with three nodes and five variables (VH, HUM, DPG, IBT, DAY) having BIC 264. The next best models were one with six nodes and five variables (HUM, DPG, IBT, VIS, DAY) having BIC 260, and one with three nodes and three variables (HUM, IBT, DAY) having BIC 259.

Method	R^2
ACE, 9 variables	0.82
ACE, 4 variables	0.78
GAM	0.80
TURBO	0.80
Box-Tidwell	0.82
Neural Network	0.79

Table 2: Comparison of Methods on the Ozone Data

This dataset has been analyzed by several others in the nonparametric regression literature, and so it is useful for comparing the methods of this paper to other nonparametric regression techniques. Breiman and Friedman (1985) used this dataset in their paper on Alternating Conditional Expectation (ACE). As a goodness-of-fit measure, they use the estimated multiple correlation coefficient, R^2 . They fit the model both using all nine explanatory variables, as well as a subset of only four that were chosen via a stepwise algorithm (the four are TEMP, IBH, DPG, and VIS). The comparison of the R^2 's is shown in Table 2. Hastie and Tibshirani (1984) fit a Generalized Additive Model (GAM) to the data. Friedman and Silverman (1989) fit the data using TURBO. In the discussion of the previous paper, Hawkins (1989) fits the data with linear regression after using Box-Tidwell style transformations on the variables. For comparison, I include the result of my model with three nodes and five explanatory variables. Table 2 shows that all of the above methods have similar goodness-of-fit to the data. All of the methods do manage to find a reasonable fit, but none is clearly better than the others.

Aside from the ACE model with only four variables, the other models in Table 2 all use more explanatory variables than does the neural network, and are thus less parsimonious and subject to increased prediction error. Hastie and Tibshirani (1990) do a comparison of several methods on this dataset in terms of variable

Method	VH	WIND	HUM	TEMP	IBH	DPG	IBT	VIS	DAY
Stepwise ACE				X	X	X		X	
Stepwise GAM	X	X	X	X	X	X		X	X
TURBO	X			X	X	X		X	X
BRUTO				X	X	X		X	X
Optimal Neural Network (3 Nodes)	X		X			X	X		X
Second-best Neural Network (6 Nodes)			X			X	X	X	X
Third-best Neural Network (3 Nodes)			X				X		X

Table 3: Comparison of Variable Selection on the Ozone Data

selection. In addition to some of the above methods, they also include a stepwise algorithm for their GAM models, as well as a response to TURBO which they call BRUTO, which is meant to do automatic variable selection and smoothing parameter selection. Table 3 shows the variables chosen by the models in each of these methods. It is interesting to note that the methods seriously disagree on which variables to select. Partly, this may be because the variables are highly correlated with each other, so that different subsets may give similar predictions. However, TURBO and BRUTO are largely in agreement with each other. And the three neural network models have similar choices of variables, although these are very different from those of TURBO and BRUTO. At the very least, it does seem clear that some variable selection is necessary because of the high level of correlation between the explanatory variables, even if there is dissent about which subset is optimal.

7 Asymptotic Consistency

Neural networks have been shown by many people (e.g., Hornik, Stinchcombe, and White 1989) to be able to approximate a function with arbitrary accuracy. In this section, I present several results on the asymptotic properties of the posterior. I assume that there exists a “true” regression function, and show that neighborhoods of this function have posterior probability tending to one as the sample size grows arbitrarily large. I also show that the mean square error of the posterior predictive function goes to zero in probability. This makes the somewhat frequentist assumption of a “true” regression function. However, it is useful to show that a Bayesian technique has good frequentist properties. First, a frequentist interested in better measures of uncertainty can use this Bayesian method as an approximation to frequentist methods. Second, the Bayesian method can be used by a Bayesian who wants to present the results to a non-Bayesian audience. Finally, a pure Bayesian will be interested to know that the method is asymptotically guaranteed to produce a reasonable answer when one exists.

For the proof, two approaches are taken. First I use a sieve approach, where the number of hidden nodes grows with the sample size, so that, asymptotically, there are an arbitrarily large number of hidden nodes in the model. Later I treat the number of hidden nodes as a parameter.

7.1 Sieve Asymptotics

The first approach is that of sieve asymptotics. A sieve is a series of models which grow with the sample size, so that in the limit, the sieve will be wide enough to encompass models arbitrarily close to the true model (Grenander 1981; Wong and Shen 1995). In this case, the number of hidden nodes is allowed to grow as a function of the sample size. The posterior can then be shown to be consistent over Hellinger neighborhoods (defined in (10) below) and the predictive regression function can be shown to be consistent. Some notation will allow a more precise statement of consistency. Denote the explanatory variables as X and the response variable as Y . Let $f(x, y)$ be a joint density function, and $f(y|x)$ be the corresponding regression function for Y on X . Denote the true function as f_o . Define a family of neighborhoods using the Hellinger distance

by

$$A_\epsilon = \{f; D_H(f, f_o) \leq \epsilon\}, \quad D_H(f, f_o) = \sqrt{\iint (\sqrt{f(x, y)} - \sqrt{f_o(x, y)})^2 dx dy}. \quad (10)$$

Let the number of hidden nodes in the model, k , grow with the sample size such that $k_n \leq n^a$ for any $0 < a < 1$. Let \mathcal{F}_n be the set of all neural networks with each parameter less than C_n in absolute value, where C_n grows with n such that $C_n \leq \exp(n^{b-a})$ for any constant b such that $0 < a < b < 1$ with the a from the bound for k_n . For any $\gamma > 0$, let

$$K_\gamma = \left\{ f; E \left[\log \frac{f_o(x, y)}{f(x, y)} \right] \leq \gamma \right\}.$$

Denote the prior for f by $\pi_n(\cdot)$ and the posterior by $P(\cdot|X^n)$. Denote the predictive density by

$$\hat{f}_n(\cdot) = \int f(\cdot) dP(f|X^n).$$

The predictive density is the Bayes estimate of f . The key result is the following theorem.

Theorem 1 *Suppose that: (i) there exists an $r > 0$ and an N_1 such that $\pi_n(\mathcal{F}_n^c) < \exp(-nr)$, $\forall n \geq N_1$; (ii) for all $\gamma, r > 0$, there exists an N_2 such that $\pi_n(K_\gamma) \geq \exp(-nr)$, $\forall n \geq N_2$. Then $\forall \epsilon > 0$, the posterior is asymptotically consistent for f_o over Hellinger neighborhoods, i.e.*

$$P(A_\epsilon | (X_1, Y_1), \dots, (X_n, Y_n)) \xrightarrow{P} 1. \quad (11)$$

Corollary 1.1 *Let $g_o(x) = E_{f_o}[Y|X = x]$ be the true regression function, and let $\hat{g}_n(x) = E_{\hat{f}_n}[Y|X = x]$ be the regression function from the predictive density using a neural network. Then under the conditions of Theorem 1, \hat{g}_n is asymptotically consistent for g_o , i.e.*

$$\int (\hat{g}_n(x) - g_o(x))^2 dx \xrightarrow{P} 0. \quad (12)$$

The idea of the proof is to use bracketing entropy results from empirical process theory to bound the posterior probability outside the Hellinger neighborhoods of (10). The proof can be found in Lee (2000). The conditions of the theorem can be shown to hold for many standard choices of priors, and in particular, for the noninformative prior of Section 3

7.2 The Number of Hidden Nodes as a Parameter

Instead of allowing the number of hidden nodes, k , to increase as a function of n , one can treat k as another parameter in the model and specify a prior for it. This approach also leads to an asymptotically consistent posterior. Let $\lambda_i = P(k = i)$ be the prior probability that the number of hidden nodes is i , $\sum \lambda_i = 1$. Let π_i be the prior for the parameters of the regression equation, given that $k = i$. The joint prior for all of the parameters is $\sum_i \lambda_i \pi_i$. One can now extend the result of Theorem 1.

Theorem 2 *Suppose that: (i) there exists a sequence $r_i > 0$ and a sequence N_i such that for each i , $\pi_i(\mathcal{F}_n^c) < \exp(-nr_i)$ for all $n \geq N_i$; (ii) for all $\gamma, r > 0$, there exists an I and a sequence M_i such that for any $i \geq I$, $\pi_i(K_\gamma) \geq \exp(-nr)$ for all $n \geq M_i$; (iii) B_n is a bound which grows with n such that for all $r > 0$, there exists a $q > 1$ and an N such that $\sum_{i=B_n+1}^\infty \lambda_i < \exp(-n^q r)$ for $n \geq N$; (iv) for all i , $\lambda_i > 0$. Then for all $\epsilon > 0$,*

$$P(A_\epsilon | (x_1, y_1), \dots, (x_n, y_n)) \xrightarrow{P} 1. \quad (13)$$

The proof can be found in Lee (2000). The conditions on the prior for k hold for most standard choices of priors, including geometric and Poisson distributions. We can also draw analogous conclusions to Corollary 1, in that the mean square error of the predictive regression function goes to zero in probability.

8 Conclusion

In this paper, I have shown how to use a noninformative prior for a neural network, and how to search the model space over both the size of the network and the space of explanatory variables. The new prior is much simpler to use than a complex, hierarchical model, in terms of both specifying the prior and fitting the model. The model selection and model averaging techniques allow one to fully account for uncertainty, both in terms of the estimation variability (i.e., the noise in the data) and the choice of model. Bayesian Random Searching allows one to efficiently search both the subsets of explanatory variables and the number of hidden nodes in order to do a complete data analysis. Tests on several canonical examples show that my approach performs well when compared to other methods. I can often achieve a similar goodness-of-fit or predictive performance while using a smaller, more parsimonious model. Finally, I explain how neural networks methods are asymptotically consistent for the posterior distribution.

9 Acknowledgements

This work was partially supported by National Science Foundation grant DMS-9803433 and National Institutes of Health grant RO1 CA54852-08. The author is grateful to Larry Wasserman for his help and advice.

References

- Akaike, H. (1974). “A New Look at Statistical Model Identification.” *IEEE Transactions on Automatic Control*, AU-19, 716–722.
- Bernardo, J. M. (1979). “Reference Posterior Distributions for Bayesian Inference (with discussion).” *Journal of the Royal Statistical Society B*, 41, 113–147.
- Breiman, L. and Friedman, J. H. (1985). “Estimating Optimal Transformations for Multiple Regression and Correlation.” *Journal of the American Statistical Association*, 80, 580–619.
- Chipman, H., George, E., and McCulloch, R. (1998). “Bayesian CART Model Search (with discussion).” *Journal of the American Statistical Association*, 93, 935–960.
- Diebolt, J. and Robert, C. (1994). “Estimation of Finite Mixture Distributions Through Bayesian Sampling.” *Journal of the Royal Statistical Society B*, 56, 363–375.
- Friedman, J. H. and Silverman, B. W. (1989). “Flexible Parsimonious Smoothing and Additive Modelling (with discussion).” *Technometrics*, 31, 3–39.
- Furnival, G. M. and Wilson, R. W. J. (1974). “Regression by Leaps and Bounds.” *Technometrics*, 16, 499–511.
- Grenander, U. (1981). *Abstract Inference*. New York: Wiley.
- Hastie, T. and Tibshirani, R. (1984). “Generalized Additive Models.” Tech. Rep. 98, Stanford University, Department of Statistics.
- (1990). *Generalized Additive Models*. London: Chapman and Hall.
- Hawkins, D. (1989). “Discussion of ‘Flexible Parsimonious Smoothing and Additive Modelling’ by J. Friedman and B. Silverman.” *Technometrics*, 31, 3–39.
- Hoeting, J. A., Madigan, D., Raftery, A. E., and Volinsky, C. T. (1999). “Bayesian Model Averaging: A Tutorial (with discussion).” *Statistical Science*, 14, 4, 382–417.
- Hornik, K., Stinchcombe, M., and White, H. (1989). “Multilayer Feedforward Networks are Universal Approximators.” *Neural Networks*, 2, 5, 359–366.

- Jeffreys, H. (1961). *Theory of Probability*. Third edition ed. New York: Oxford University Press.
- Kass, R. E. and Raftery, A. E. (1995). “Bayes Factors.” *Journal of the American Statistical Association*, 90, 430, 773–795.
- Keribin, C. (1997). “Consistent Estimation of the Order of Mixture Models.” Tech. rep., Université d’Evry-Val d’Essonne, Laboratoire Analyse et Probabilité.
- Leamer, E. E. (1978). *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. New York: Wiley.
- Lee, H. K. H. (1998). “Model Selection and Model Averaging for Neural Networks.” Ph.D. thesis, Carnegie Mellon University, Department of Statistics.
- (2000). “Consistency of Posterior Distributions for Neural Networks.” *to appear in Neural Networks*.
- MacKay, D. J. C. (1992). “Bayesian Methods for Adaptive Methods.” Ph.D. thesis, California Institute of Technology.
- (1994). “Bayesian Non-Linear Modeling for the Energy Prediction Competition.” *ASHRAE Transactions*, 100, pt. 2, 1053–1062.
- Müller, P. and Rios Insua, D. (1998a). “Feedforward Neural Networks for Nonparametric Regression.” In *Practical Nonparametric and Semiparametric Bayesian Statistics*, eds. D. Dey, P. Müller, and D. Sinha, 181–193. New York: Springer-Verlag.
- (1998b). “Issues in Bayesian Analysis of Neural Network Models.” *Neural Computation*, 10, 571–592.
- Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. New York: Springer.
- Raftery, A. (1996). “Approximate Bayes Factors and Accounting for Model Uncertainty in Generalized Linear Models.” *Biometrika*, 83, 251–266.
- Raftery, A. E. and Madigan, D. (1994). “Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam’s Window.” *Journal of the American Statistical Association*, 89, 1535–1546.
- Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). “Bayesian Model Averaging for Linear Regression Models.” *Journal of the American Statistical Association*, 437, 179–191.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). “Learning Internal Representations by Error Propagation.” In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds. D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, vol. 1, 318–362. Cambridge, MA: MIT Press.
- Sarle, W. S. (1994). “Neural Network Implementation in SAS Software.” In *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, 38–51. SAS Institute, Cary, NC.
- Schwarz, G. (1978). “Estimating the Dimension of a Model.” *The Annals of Statistics*, 6, 2, 461–464.
- Stern, H. S. (1996). “Neural Networks in Applied Statistics.” *Technometrics*, 38, 3, 205–214.
- Stone, M. (1974). “Cross-validatory Choice and Assessment of Statistical Predictions.” *Journal of the Royal Statistical Society B*, 36, 111–147.
- Wasserman, L. (1998). “Asymptotic Inference for Mixture Models Using Data Dependent Priors.” Tech. Rep. 677, Carnegie Mellon University, Department of Statistics.
- Wong, W. H. and Shen, X. (1995). “Probability Inequalities for Likelihood Ratios and Convergence Rates of Sieve MLEs.” *Annals of Statistics*, 23, 339–362.