A Noninformative Prior for Neural Networks

Herbert K. H. Lee, Duke University
Box 90251, Durham, NC 27708, herbie@stat.duke.edu

November 9, 2000

Abstract

While many implementations of Bayesian neural networks use large, complex hierarchical priors, in much of modern Bayesian statistics, noninformative (flat) priors are very common. This paper introduces a noninformative prior for feed-forward neural networks, describing several theoretical and practical advantages of this approach. Details of implementation via Markov chain Monte Carlo are included.

Key Words: Bayesian Statistics; Improper Prior; Markov Chain Monte Carlo

1 Introduction

Much of the existing Bayesian neural network literature relies upon complex hierarchical priors for the parameters of the network, for example, MacKay (1992), Neal (1996), Müller and Rios Insua (1998), and Andrieu et al. (1999). In this paper, I propose a fully noninformative prior with several philosophical and practical advantages. I also discuss how to implement model fitting via Markov chain Monte Carlo with this prior.

In the Bayesian context, one's choice of prior is meant to reflect either knowledge from previous experience or data, or to reflect personal and subjective beliefs about the problem (or both). In order to accurately choose a prior, one needs to understand how to interpret the parameters of a neural network model. Unfortunately, few people seem to realize how difficult a problem this is. Section 2 will discuss this issue in more detail. The bottom line is that if one cannot interpret the parameters effectively, it may be unwise to pretend so by using a proper prior. A noninformative prior is a way to acknowledge this high degree of uncertainly about the meaning of the parameters and to let the data more fully guide the model. Such priors were first fully established by Jeffreys (1961). There is now an extensive literature on such reference priors, and Kass and Wasserman (1996) provide a thorough review of this literature.

This noninformative prior is much simpler in structure, and it thus leads to a much simpler final model. This has advantages in reducing the dimension of the parameter space with the attendant benefits of shorter programming and run times. Since the noninformative prior is flat with respect to most parameters, the posterior modes of the parameters will be the same as would be obtained under least-squares fitting. This has the advantage that one's results are directly comparable to those of many other researchers, and that one can directly relate to those doing non-Bayesian analysis. However, one still gains all of the benefits of the Bayesian approach. For example, it is straightforward to get estimates of variability or error probabilities, and one can justifiably interpret the results in terms of probabilities. This approach also allows one to do model averaging (Kass and Raftery, 1995) over several different fitted networks to improve predictive power.

Noninformative priors, including the one of this paper, often have many invariance properties (Jeffreys, 1961; Hartigan, 1964). For example, Jeffreys's prior is invariant with respect to all differentiable transformations of the input variables. The prior of this paper is easily seen to be invariant with respect to a shift in location or scale. Thus one need not be concerned about rescaling the variables (except, perhaps, for computational reasons).

This noninformative prior, as well as many other priors for neural networks, have been shown to be asymptotically consistent for the posterior, in the sense that the posterior probability accumulates in a Hellinger neighborhood of the truth (Lee, 1996).

In many problems, including this one, it is important to consider the issue of model selection. How many hidden nodes are best? Are all of the inputs necessary? Noninformative priors, being not informative, do not provide the same shrinkage as some other priors (or methods such as weight decay), and so the user must keep this in mind. It is advisable to use some model selection technique, such as those in MacKay (1992), Neal (1996), Müller and Rios Insua (1998), Lee (1998), or Andrieu et al. (1999).

In this paper, I focus exclusively on feed-forward neural networks with a single hidden layer of units with logistic activation functions, use linear output units, and do not allow direct connections from the inputs to the outputs. However, all of these methods and results are fully generalizable to any variety of feed-forward networks. The particular form of the model I will work with is

$$y_i = \beta_o + \sum_{j=1}^{k} \beta_j \Psi(\boldsymbol{\gamma}_j' \mathbf{x}_i) + \varepsilon_i, \qquad \varepsilon_i \overset{iid}{\sim} N(0, \sigma^2) \tag{1}$$

where x_i is a (possibly vector-valued) input, y_i is the target, γ are the weights from the inputs to the hidden layer, β are the weights from the hidden layer to the outputs, and the activation function is

$$\Psi(z) = \frac{1}{1 + \exp(-z)} \tag{2}$$

2 Interpretation of the Parameters

This section will demonstrate the infeasibility of trying to interpret the parameters of a neural network in the general case. Before getting to the general case, let us first look at some cases where the parameters can be interpreted, so that we can see what happens to cause a loss of interpretability.

The case of a single hidden node is straightforward. With a single input variable, the model for the fitted values, $\hat{y}_i$, is

$$\hat{y}_i = \beta_0 + \frac{\beta_1}{1 + \exp(-\gamma_0 - \gamma_1 x_i)} . \tag{3}$$

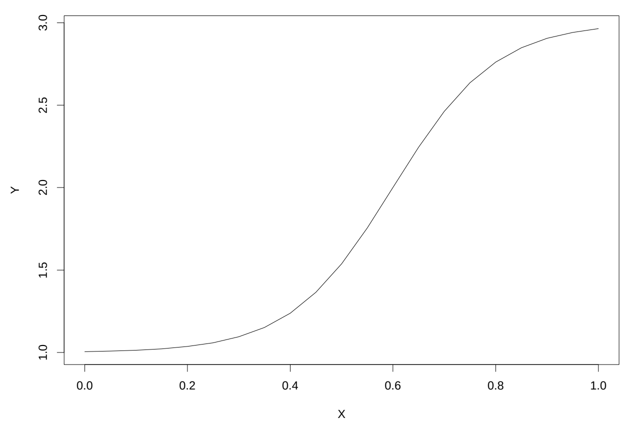

Figure 1: A One-Node Function

Figure 1 shows this fitted function for $\beta_0 = 1$, $\beta_1 = 2$, $\gamma_0 = -6$, and $\gamma_1 = 10$ over x in the unit interval. In the one-hidden node case, the parameters are easily interpreted. β_0 represents the overall location of y, a sort of intercept, in that $y = \beta_0$ when the logistic function is close to zero (which it is here for values of x near or below zero). β_1 is the overall scale factor for y, in that the logistic function ranges from 0 to 1, so β_1 is like the range of y, which in this case is $3 - 1 = 2$. The γ parameters control the location and scale of the logistic function. The center of the logistic function occurs at $-\frac{\gamma_o}{\gamma_1}$, here 0.6. The larger the value of γ_1, the steeper the increase in y as one moves away from the center. If γ_1 is positive, then the logistic rises from left to right. If γ_1 is negative, then the logistic will decrease as x increases.

This interpretation works fine for a single hidden node. For well-separated nodes (in terms of their centers and slopes, so that for any particular value of x, at most one node is near its center and the rest are far enough from their centers that a small change in x does not produce a noticeable change in the logistic function), one can continue to use this interpretation. Two nodes with centers near each other, but with opposite signs on γ_1, can combine to produce a peak, similar in spirit to a kernel or a radial basis function. However, when the nodes effectively overlap, which is typically the case in a real problem, the parameters can no longer be interpreted as above.

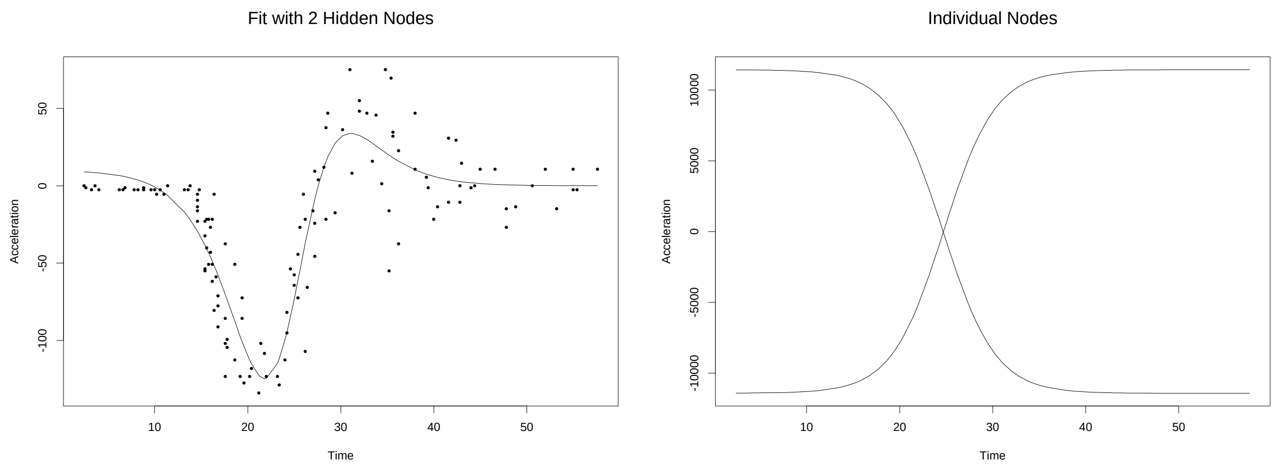

Figure 2: Logistic Basis Functions

For example, the left plot of Figure 2 results from maximum likelihood (least-squares) fitting of a two-node network to the motorcycle accident data of Silverman (1985). The x-axis is the time in milliseconds after the crash, and the y-axis the acceleration force on the head of the rider. The two-node model does a good job of capturing the main features of the data, although perhaps it over-smooths for small x values. Keep in mind that this fit is achieved with only two hidden nodes. It is impossible to use any of the above interpretations of the nodes for this fit, because there are more points of inflection in the fit than there are hidden nodes. The two individual nodes are shown in the right plot of the figure, and they have very similar centers and slopes, but combine to produce a highly non-linear fit. Note that the scale of the basis functions is two orders of magnitude larger than the original data. Looking at only the data, it is not at all clear that the particular basis functions of Figure 2 would work so well. In this case, one would not be likely to have any clear beliefs to specify in a prior. As this is an extremely simple case, with only two hidden nodes and a single input variable, one can easily imagine how much worse things can get with larger networks. Hence a noninformative prior is a practical alternative.

3 An Improper Prior

The prior I propose for feed-forward neural networks is

$$\pi(\beta, \gamma, \sigma^2) \propto \frac{1}{\sigma^2} \,. \tag{4}$$

Notice that this prior is only defined up to a constant. Since it has an infinite integral, it does not matter what the constant is. Priors with an infinite integral, such as this one, are called *improper*. While the thought of using an improper prior may sound odd, improper priors have found to be very useful in practice. Many of Jeffreys's (1961) original reference priors were improper. The key is to ensure that the posterior is proper (finite). Below are some details to enforce propriety for this prior.

This choice of prior is well-established in least-squares regression (see for example, Gelman et al. (1995)). It places uniform prior density over the real line for all parameters except the variance of the error. For the variance, it is uniform with respect to the log of the variance, since variances must be positive and are right-skewed. By using a prior such as this one, we make no assumptions about the location or scale of any of the parameters in the model. Gelman et al. (1995) present some theoretical advantages of this family of priors.

In the case of simple least-squares regression, the posterior is necessarily proper (assuming at least as many data points as parameters). In the case of a neural network, propriety of the posterior is not guaranteed. Hence we need to make a few minor modifications to the prior. The basic idea is to guarantee that the logistic basis functions of the hidden layer are linearly independent. We do this by putting certain restrictions on the parameters during the MCMC fitting process, a method now common in the mixture model literature (Diebolt and Robert, 1994; Wasserman, 1998).

First, a piece notation is helpful. Denote the outputs of the hidden layer as

$$z_{ij} = \left[1 + \exp\left(-\gamma_{j0} - \sum_{h=1}^{p} \gamma_{jh} x_{ih}\right)\right]^{-1} \tag{5}$$

and let Z be the matrix with elements (z_{ij}). The fitting of the vector β is merely a least-squares regression on the design matrix Z. What is necessary is a restriction on Z to ensure the linear independence of the columns of Z. A sufficient condition for linear independence is that the determinant of Z^tZ is larger than some positive value C_n. We can let $C_n \to 0$ as $n \to \infty$ as long as $C_n > 0$ for all n (n is the sample size). To ensure propriety of the posterior, we also need to bound the individual γ's such that $|\gamma_{jh}| < D_n$ for all j, h. A full theoretical justification of these restrictions follows in Section 5.

It is computationally useful to also bound the other parameters (namely $|\beta_j| < D_n$; σ does not need to be bounded). This can help with numerical stability in the MCMC fitting. It is also theoretically useful in that this prior can be shown to be asymptotically consistent for the posterior (Lee, 1996). The consistency proof requires that $D_n = o(\exp(n^r))$ for all $r > 0$.

The basic idea of the constraining values C_n and D_n is that as the sample size n increases, C_n and D_n can become more extreme (closer to zero for C_n, arbitrarily large for D_n). Asymptotically, the rate does not matter for C_n, and the rate for D_n required by the consistency theorem is exponentially fast. In practice, one can just pick a large constant value for D_n such as 100,000, and a small constant value for C_n such as 0.0001. What matters is the numerical stability of one's computational routines, and numbers such as these usually suffice for double precision arithmetic.

Thus the restricted prior is

$$\pi_n(\beta, \gamma, \sigma^2) = \pi(\beta, \gamma, \sigma^2)\mathrm{I}_{\{\theta \in \Omega_n\}} \propto \frac{1}{\sigma^2}\mathrm{I}_{\{\theta \in \Omega_n\}} . \tag{6}$$

where $\mathrm{I}_{\{\}}$ is an indicator function (equal to one when its argument is true, zero otherwise), and Ω_n is the parameter space restricted such that:

1. $\left|\boldsymbol{Z}^T\boldsymbol{Z}\right| > C_n$
2. $|\gamma_{jh}| < D_n$ for all j, h
3. $|\beta_j| < D_n$ for all j

4 Fitting the Model with MCMC

Fitting the model is straightforward using Markov chain Monte Carlo. The full posterior is proportional to the likelihood times the prior:

$$f(\beta, \gamma, \sigma^2|y) \propto \frac{1}{\sigma^2}(2\pi\sigma^2)^{-n/2} \exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{k}\beta_j z_{ij}\right)^2\right]\mathrm{I}_{\{\theta \in \Omega_n\}} . \tag{7}$$

The complete conditional distributions for σ^2 and β given the other parameters and the data are inverse-gamma and normal, respectively:

$$\sigma^2|\gamma, y \sim \Gamma^{-1}\left(\frac{n-k-1}{2},\ \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \hat{\beta}_0 - \sum_{j=1}^{k}\hat{\beta}_j z_{ij}\right)^2\right) \tag{8}$$

$$\beta|\gamma, \sigma^2, y \sim N\left((Z^tZ)^{-1}Z^ty,\ (Z^tZ)^{-1}\sigma^2\right), \tag{9}$$

where $\hat{\beta}$ are the fitted coefficients from a least-squares regression of y on Z. Thus the MCMC algorithm is as follows:

1. Start with arbitrary initial values for γ.
2. Compute Z.
3. Draw σ^2 via Equation (8).
4. Draw β via Equation (9).
5. For each j from $1, \ldots, k$, do the following Metropolis step:
   (a) Generate a candidate $\tilde{\gamma}_j \sim N(\gamma_j, 0.05^2)$.
   (b) Re-compute Z with $\tilde{\gamma}_j$ and compute $|Z^tZ|$.
   (c) If $|Z^tZ| > 0.0001$ and $|\gamma_{jh}| < 100,000$ for all j, h, accept $\tilde{\gamma}_j$ with probability $\min\left(1, \frac{f(\beta,\tilde{\gamma},\sigma^2|y)}{f(\beta,\gamma,\sigma^2|y)}\right)$; otherwise, reject the candidate and keep the current value of γ_j.
6. Repeat steps 2 though 5 for the required number of iterations.

The resulting draws will be a sample from the joint posterior distribution of the parameters. An explanation of why the determinant restriction keeps the posterior from being improper follows in the next section. Note that the choice of $C_n = 0.0001$ and $D_n = 100,000$ are arbitrary and can be adjusted to match individual computational requirements. The choice of 0.05 in the proposal function of step 5(a) should be treated as a tuning parameter, and can be adjusted to aim for an acceptance probability of approximately forty percent. Note that there exist many more sophisticated methods of generating Metropolis-Hastings proposals to achieve higher acceptance probabilities and better mixing of the chain. My personal approach is to recognize the trade-off between programming time and run time, in that more complicated approaches can decrease the necessary run time, but may take much more time and effort to program. I prefer to use a simpler algorithm and just let it run for longer. Both approaches tend to produce similar answers in the long run.

5 Theoretical Justification for the Restrictions

This section contains first a heuristic explanation of the restrictions, followed by more rigorous theory. To understand why the restrictions force propriety on the posterior, first look at a simplified scenario. Consider a single input x with n observations. Instead of using logistic functions, suppose the activation functions are step (Heaviside or indicator) functions of the form

$$\Psi_j(x) = I_{\{x \le a_j\}} \quad \text{or} \quad \Psi_j(x) = I_{\{x > a_j\}}. \tag{10}$$

Suppose for now that network contains only two hidden nodes. Then $z_j = (z_{1j}, \ldots, z_{nj}) = (1, \ldots, 1, 0, \ldots, 0)$ or $(0, \ldots, 0, 1, \ldots, 1)$ and z_0 is a vector of ones (the bias input). Let

$$r_i = \sum_i z_{ij} \quad \text{for } i = 1, 2 \tag{11}$$

$$r_{12} = \sum_i z_{i1}z_{2i}. \tag{12}$$

r_1 is the number of cases with input 1 larger (or smaller depending on the direction of inequality in the indicator) than the first threshold a_1. r_{12} is the number of cases for which both hidden units give outputs of 1. The determinant of interest is

$$\left|Z^t Z\right| = \begin{vmatrix} r_1 & r_{12} & r_1 \\ r_{12} & r_2 & r_2 \\ r_1 & r_2 & n \end{vmatrix} = r_1 r_2 n + 2 r_1 r_2 r_{12} - r_1^2 r_2 - r_1 r_2^2 - r_{12}^2 n. \tag{13}$$

This determinant will be zero if any of the following happen: $r_i = 0$, $r_i = n$, $r_1 = r_2 = r_{12}$, or $\{r_1 = n - r_2$ and $r_{12} = 0\}$. Essentially one needs to ensure that the thresholds for the activation functions are all separated by data points (i.e., there is a data point x_i that is between the thresholds of the two indicators), and that no threshold occurs outside the range of the data. These are exactly the conditions that prevent linear dependence. If we choose a small enough C_n (for indicator functions, all we need is $0 < C_n < 1$), then the same conditions guarantee that $|Z^t Z| > C_n$. The same logic applies to datasets with more inputs and to networks with more hidden nodes.

For indicator function hidden nodes, the requirement on the determinant also prevents impropriety in the posterior. Consider the case of trying to fit an indicator function where the parameter is the threshold. Suppose one tries to fit a threshold smaller than the smallest data point, x_{min}. Then the data do not provide any information for distinguishing between putting the threshold at $x_{min} - b_1$ and $x_{min} - b_2$ for any positive numbers b_1 and b_2. Since this equivalence set has infinite mass under the prior, the posterior is improper. On the other hand, when fitting a threshold inside the range of the data, the data force propriety on the posterior.

The indicator functions relate to logistic functions in that indicator functions are a limiting case of logistic functions. Let

$$\Psi(x) = \frac{1}{1 + \exp(-\gamma_0 - \gamma_1 x)} \tag{14}$$

be a logistic function. If $\gamma_0, \gamma_1 \to \infty$ such that $\frac{\gamma_0}{\gamma_1} \to a$ for some constant a, then $\Psi(x) \to I_{\{x > -a\}}$. We would then need the determinant condition to guarantee linear independence. However, this is not the only condition we need to guarantee a proper posterior. The triangular region where $\gamma_0, \gamma_1 \to \infty$ such that $\frac{\gamma_0}{\gamma_1} \to a$ has infinite area. Over this region, as the parameters get large, the likelihood converges to some non-zero constant (in most problems, the likelihood converges to zero in the tails). Thus the posterior over this region alone is improper. For this reason, we also need to bound the individual parameters.

The logistic functions allow values between zero and one, so that two columns of $Z^t Z$ could be very similar, but not identical. In regression this is called multicollinearity. The near-linear dependence causes instability in the parameter estimates. It is computationally desirable to avoid this case, which we can do by requiring the determinant to be larger than some small positive number C_n rather than merely requiring it to be non-zero.

Now for the the full proof of the propriety of the prior. Denote the likelihood by L_n. Let π be the noninformative prior of equation (4), π_n be the restricted prior of equation (6), and Ω_n the restricted parameter space described with equation (6). Let C_n decrease to 0 (for example, $C_n = 1/n$) and let D_n increase with n (for example, $D_n = 100,000 + n$). First the likelihood is re-written so that $\boldsymbol{\beta}$ can be integrated out, which involves completing the square for $\boldsymbol{\beta}$.

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

where $\hat{\mathbf{Y}}$ is the vector of fitted values, $\hat{\mathbf{Y}} = E[\mathbf{Y}|\mathbf{X}]$. Note that $f(\boldsymbol{\beta}|\boldsymbol{\gamma},\sigma,\mathbf{y})$ 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 $\boldsymbol{\gamma}$ parameters. Then the posterior is proper:

$$\begin{aligned}\int L_n\pi_n &= \int_{\Omega_n} f(\boldsymbol{\beta}|\boldsymbol{\gamma},\sigma,\mathbf{y})f(\boldsymbol{\gamma},\sigma|\mathbf{y})\left[\frac{1}{\sigma^2}\right]d\boldsymbol{\beta}d\sigma d\boldsymbol{\gamma}\\ &= \int_{\Gamma_n}\int\left[\int f(\boldsymbol{\beta}|\boldsymbol{\gamma},\sigma,\mathbf{y})d\boldsymbol{\beta}\right]\frac{1}{\sigma^2}f(\boldsymbol{\gamma},\sigma|\mathbf{y})d\sigma d\boldsymbol{\gamma}\end{aligned}$$

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

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,

$$\int_\kappa |\pi_n - \pi| = \int_\kappa |\pi I_{\Omega_n} - \pi| \rightarrow 0 \text{ as } n \rightarrow 0 \tag{15}$$

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 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,

$$\left|\frac{\partial\log\pi_n}{\partial\theta_0} - \frac{\partial\log\pi}{\partial\theta_0}\right| = O_p\left(\frac{1}{\sqrt{n}}\right) \tag{16}$$

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

6 Example

The canonical dataset in the Bayesian neural network literature 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

$$y_1 = 2.0\cos(x_1) + 1.3\cos(x_1 + x_2) + \varepsilon_1 \tag{17}$$
$$y_2 = 2.0\sin(x_1) + 1.3\sin(x_1 + x_2) + \varepsilon_2\,, \tag{18}$$

where $\varepsilon_i \overset{iid}{\sim} N(0, 0.05^2)$. 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.

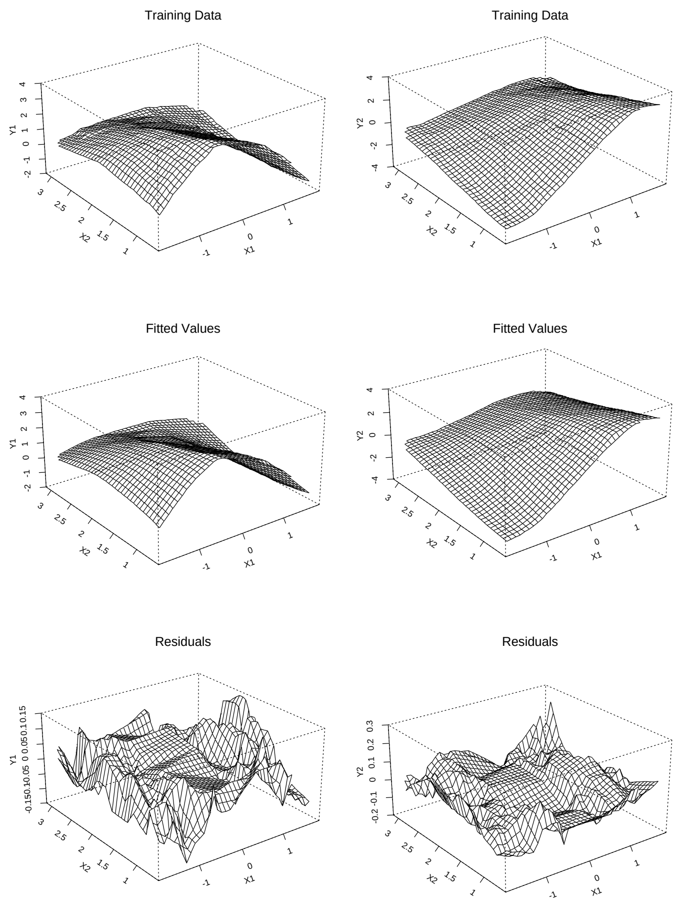

Figure 3: A Six-Node Model for the Robot Arm Data. The left column is the first response variable and the right column is the second.

Using Bayesian Random Searching (BARS) (Lee, 1998) for model selection, a network with six hidden nodes is found to be optimal (with about 98% of the posterior probability in the model space). Plots of the original data, the fitted values, and the residuals are shown in Figure 3. In this 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 surfaces have been interpolated to get a viewable image.)

The MCMC output was then used 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 4. The small mean square error shows that this model does indeed fit quite well.

This dataset has also been analyzed by several others in the Bayesian neural network literature, in particular MacKay (1992), Neal (1996), Müller and Rios Insua (1998a) and Andrieu, de Freitas, and Doucet (1999) . Table 1 shows the MSE achieved on the test data by the methods of each of the above competing models. All of these methods have an error rate on the same order of magnitude, although the methods of this paper do well within this group of methods.

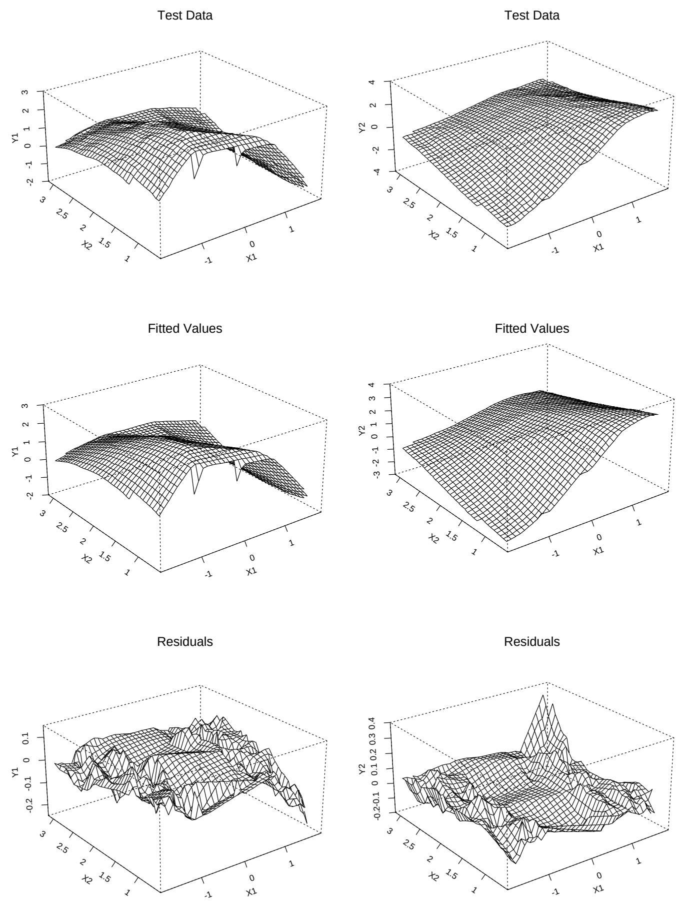

Figure 4: Validation Data for the Robot Arm Problem. The left column is the first response variable and the right column is the second.

Method	Mean Square Error on Test Data
MacKay	0.00557
Neal	0.00549
Müller and Insua	0.00620
Andrieu, de Freitas, and Doucet	0.00502
This paper	0.00501

Table 1: Comparison of Methods on the Robot Arm Data

7 Conclusions

Modern Bayesian methods include noninformative priors in the toolbox. This paper is meant to help bring this perspective and practice to the neural network community. Noninformative priors have both philosophical and practical advantages over both complicated hierarchical priors and over non-Bayesian methods. They simplify the choice of prior, the model, the computational details, and the interpretation of the results.

The methods of this paper are easily extended to other varieties of feed-forward networks, as well as to networks with multiple outputs or networks for classification (Lee, 1998). I hope that these ideas will be found useful in other areas of the machine learning community as well.

References

Andrieu, C., de Freitas, J. F. G., and Doucet, A. (1999). "Robust Full Bayesian Learning for Neural Networks." Tech. Rep. 343, Cambridge University, Engineering Department.

Diebolt, J. and Robert, C. (1994). "Estimation of Finite Mixture Distributions Through Bayesian Sampling." *Journal of the Royal Statistical Society B*, 56, 363–375.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian Data Analysis*. London: Chapman & Hall.

Hartigan, J. (1964). "Invariant Prior Distributions." *Annals of Mathematical Statistics*, 35, 2, 836–845.

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.

Kass, R. E. and Wasserman, L. (1996). "The Selection of Prior Distributions by Formal Rules." *Journal of the American Statistical Association*, 91, 435, 1343–1370.

Lee, H. K. H. (1996). "Model Selection for Consumer Loan Application Data." Tech. Rep. 650, Carnegie Mellon University, Department of Statistics.

— (1998). "Model Selection and Model Averaging for Neural Networks." Ph.D. thesis, Carnegie Mellon University, Department of Statistics.

MacKay, D. J. C. (1992). "Bayesian Methods for Adaptive Methods." Ph.D. thesis, California Institute of Technology.

Müller, P. and Rios Insua, D. (1998). "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.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. New York: Springer.

Silverman, B. W. (1985). "Some Aspects of the Spline Smoothing Approach to Non-Parametric Curve Fitting." *Journal of the Royal Statistical Society B*, 47, 1–52.

Wasserman, L. (1998). "Asymptotic Inference for Mixture Models Using Data Dependent Priors." Tech. Rep. 677, Carnegie Mellon University, Department of Statistics.