Bayesian Analysis of Non–Linear Autoregression Models Based on Neural Networks

A. Menchero, R. Montes Diez*, D. Ríos Insua

Statistics and Decision Sciences Group.

Rey Juan Carlos University, Madrid, Spain.

P. Müller

M. D. Anderson Cancer Center.

University of Texas, Houston,77030

Abstract

In this paper, we show how Bayesian neural networks can be used for time series analysis. We consider a block based model building strategy to model linear and nonlinear features within the time series. A proposed model is a linear combination of a linear autoregression term and a feedforward neural network (FFNN) with an unknown number of hidden nodes. To allow for simpler models, we also consider these terms separately as competing models to select from. Model identifiability problems arise when FFNN sigmoidal activation functions exhibit almost linear behaviour, or when there are almost duplicate or irrelevant neural network nodes. New reversible jump moves are proposed to facilitate model selection mitigating model identifiability problems. We illustrate this methodology analyzing two time series data examples.

KEYWORDS: Time series, reversible jump MCMC

1 INTRODUCTION

Many neural network models have been applied to time series analysis and forecasting. There has been some interest in using recurrent networks such as Elman networks (Elman, 1990), Jordan networks (Jordan, 1986) and real-time recurrent learning networks (Williams and Zipser, 1989). However, the model most frequently

*Corresponding author: R. Montes Diez, Grupo de Estadística y Ciencias de la Decisión, Departamento de Informática, Estadística y Tecnología. Universidad Rey Juan Carlos. Tulipán s/n. 28933 Móstoles. Madrid. Spain. rmontes@escet.urjc.es

used is the multilayer feedforward neural network (FFNN). Many papers compare FFNNs and standard statistical methods for time series analysis, see e.g. Tang et al. (1991), Foster et al. (1992), Stern (1996), Hill et al. (1996), Faraway and Chatfield (1998). Several papers, have found FFNN superior to linear methods such as ARIMA models for several time series problems and comparable to other nonlinear methods like generalized additive models or projection pursuit regression.

We will consider FFNNs to model non-linear autoregressions. The net output will represent the time series predicted value, when past values of the series are given as net inputs. Fitting a FFNN model requires many choices about the model structure: activation functions, number of hidden layers, number of hidden nodes, inputs, and so on. A non linear optimization algorithm will typically be used to estimate weights to optimize some performance criterion, for example, minimization of mean square error, with weight decay. To fix the model structure, traditionally rules of thumb are used. Remus et al. (1998) suggest appropriate rules for FFNN in time series forecasting.

Alternatively, a Bayesian approach to FFNN modelling (Mackay, 1992; Neal, 1996; Müller and Ríos Insua, 1998; Ríos Insua and Müller, 1998) provides a coherent framework to deal with these issues. FFNN parameters are regarded as random variables whose posterior distribution is inferred in the light of data. Most importantly perhaps, we may include the number of hidden nodes as an additional parameter and model its uncertainty. Predictions are obtained by averaging over all possible models and parameter values according to their posterior distributions.

In this paper we introduce a Bayesian FFNN forecasting model for time series data. We present an inference scheme based on Markov Chain Monte Carlo simulation. To deal with model selection, we analyze why standard birth/death reversible jump moves result in slowly mixing MCMC. Instead, we propose new reversible jump moves to add or delete special kinds of nodes characterized as linearized, irrelevant or duplicate. Two examples are used to illustrate the methodology, the lynx time series data (Priestley, 1988) and the airline passengers data (Box and Jenkins, 1970).

2 MODEL DEFINITION

Consider univariate time series data $\{y_1, y_2, \ldots, y_N\}$. We would like to model the generating stochastic process in an autoregressive fashion,

$$p(y_1, y_2, \ldots, y_N) = p(y_1, \ldots, y_q) \prod_{t=q+1}^{N} p(y_t | y_{t-1}, y_{t-2}, \ldots, y_{t-q})$$

We shall assume that each y_t is modelled by a nonlinear autoregression function of q past values plus a normal error term:

$$\begin{aligned} y_t &= f(y_{t-1}, y_{t-2}, \ldots, y_{t-q}) + \epsilon_t, \ t = q+1, \ldots, N \\ \epsilon_t &\sim N(0, \sigma^2) \end{aligned}$$

so that $y_t|y_{t-1}, y_{t-2}, \ldots, y_{t-q} \sim N\left(f(y_{t-1}, y_{t-2}, \ldots, y_{t-q}), \sigma^2\right),\ t = q+1, \ldots, N$.

We complete model specification using a block-based strategy to describe f. We propose a mixed model as a linear combination of a linear autoregression term and a FFNN. A FFNN model with q input nodes, one hidden layer with M hidden nodes, one output node and activation function φ is a model relating a response variable $\widehat{y}_t$ and q explanatory variables, in our case $x_t = (y_{t-1}, \ldots, y_{t-p})$:

$$\widehat{y}_t(x_t) = \sum_{j=1}^{M} \beta_j \varphi(x_t' \gamma_j + \delta_j)$$

with $\beta_j \in \mathcal{R}$, $\gamma_j \in \mathcal{R}^q$. Biases δ_j may be assimilated to the rest of the γ_j vectors if we consider an additional input with constant value one, say $x_t = (1, y_{t-1}, \ldots, y_{t-q})$, so that, $\gamma_j = (\gamma_{0j}, \gamma_{1j,} \ldots, \gamma_{qj}) \in \mathcal{R}^{q+1}$. To be specific we will in the following assume a logistic activation function: $\varphi(z) = \exp(z)/(1 + \exp(z))$. However, the discussion remains valid for any other sigmoidal functions.

In the proposed model, the linear term accounts for linear features and the FFNN term for a nonlinear correction:

$$f(y_{t-1}, y_{t-2}, \ldots, y_{t-q}) = x_t' \lambda + \sum_{j=1}^{M} \beta_j \varphi(x_t' \gamma_j),\ t = q+1, \ldots, N \tag{1}$$

Initially, the parameters in our model are the linear coefficients $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_q)$ $\in \mathcal{R}^{q+1}$, the hidden to ouput weights $\beta = (\beta_1, \beta_2, \ldots, \beta_M)$, the input to hidden weights $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_M)$ and the error variance σ^2.

As there will be uncertainty about the number of hidden nodes M to include, we shall model this uncertainty considering M as an unknown parameter. Note that the mixed model (1) embeds as a particular case the standard linear autoregression model, when $M = 0$, and the FFNN model, when $\lambda = 0$. We shall assume the autoregressive order q to be known in advance. If desired, one could model uncertainty about q as a problem of variable selection in a model selection context. However in the following discussion we assume q fixed.

To allow for simpler models, we also consider the linear and nonlinear terms separately as competing models to select from: a simple linear autoregression model

$$\widehat{y}_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-q}) = x_t' \lambda,\ t = q+1, \ldots, N \tag{2}$$

and a nonlinear autoregression feedforward neural net model

$$\widehat{y}_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-q}) = \sum_{j=1}^{M} \beta_j \varphi(x_t' \gamma_j),\ t = q+1, \ldots, N \tag{3}$$

for each value of M.

As in Ríos Insua and Müller (1998), we assume a normal/inverse gamma prior

$$\beta_j \sim N(\mu_\beta, \sigma_\beta^2),\quad \lambda \sim N(\mu_\lambda, \sigma_\lambda^2 I),\quad \gamma_j \sim N(\mu_\gamma, \Sigma_\gamma),\quad \sigma^2 \sim InvGamma(a_\sigma, b_\sigma). \tag{4}$$

When there is non-negligible uncertainty about prior hyperparameters, we may extend the prior model with additional hyperpriors. We shall use the following standard conjugate choices in hierarchical models:

$$\begin{aligned} \mu_\beta &\sim N(a_{\mu_\beta}, b_{\mu_\beta}),\ \sigma^2_\beta \sim InvGamma(a_{\sigma_\beta}, b_{\sigma_\beta}) \\ \mu_\lambda &\sim N(a_{\mu_\lambda}, b_{\mu_\lambda}),\ \sigma^2_\lambda \sim InvGamma(a_{\sigma_\lambda}, b_{\sigma_\lambda}) \\ \mu_\gamma &\sim N(a_{\mu_\gamma}, b_{\mu_\gamma}),\ \Sigma_\gamma \sim InvWishart(a_{\Sigma_\lambda}, b_{\Sigma_\lambda}) \end{aligned} \tag{5}$$

Hyperparameters are a priori independent. Given hyperparameters, parameters are a priori independent. Since the likelihood is invariant with respect to relabelings, we include an order constraint to avoid trivial posterior multimodality due to index permutation. For example, we may use $\gamma_{1p} \leq \gamma_{2p} \ldots \leq \gamma_{Mp}$.

3 MCMC POSTERIOR INFERENCE WITH A FIXED MODEL

Consider first the mixed model (1) with fixed M. The complete likelihood for a given data set $D = \{y_1, y_2, \ldots, y_N\}$ is:

$$p(D \mid \lambda, \beta, \gamma, \sigma^2) = p(y_1, \ldots, y_q \mid \lambda, \beta, \gamma, \sigma^2) p(D' \mid y_1, \ldots, y_q, \lambda, \beta, \gamma, \sigma^2)$$

where $D' = \{y_{p+1}, \ldots, y_N\}$ and

$$p(D' \mid y_1, \ldots, y_q, \lambda, \beta, \gamma, \sigma^2) = \prod_{t=q+1}^{N} p(y_t \mid y_{t-1}, y_{t-2}, \ldots, y_{t-q}, \lambda, \beta, \gamma, \sigma^2)$$

is the conditional likelihood given first q values. From here on, we will make inference conditioning on the first q values i.e., assuming they are known without uncertainty (alternatively, we could include an informative prior over first q values in the model and perform inference with the complete likelihood). Together with the prior assumptions (4) and (5), the joint posterior distribution is given by:

$$p(\lambda, \beta, \gamma, \sigma^2, \chi \mid D') \propto p(y_{q+1}, \ldots, y_N \mid y_1, \ldots, y_q, \lambda, \beta, \gamma, \sigma^2) p(\lambda, \beta, \gamma, \sigma^2, \chi) M! \tag{6}$$

where

$$\begin{aligned} p(\lambda, \beta, \gamma, \sigma^2, \chi) &= p\left(\mu_\lambda, \sigma^2_\lambda, \mu_\beta, \sigma^2_\beta, \mu_\gamma, \Sigma_\gamma\right) p(\sigma^2) \\ &\quad p(\lambda \mid \mu_\lambda, \sigma^2_\lambda I) p(\beta \mid \mu_\beta, \sigma^2_\beta I) \prod_{i=1}^{M} p(\gamma_i \mid \mu_\gamma, \Sigma_\gamma) \end{aligned}$$

is the joint prior distribution, $\chi = (\mu_\lambda, \sigma^2_\lambda, \mu_\beta, \sigma^2_\beta, \mu_\gamma, \Sigma_\gamma)$ is the set of hyperparameters and $M!$ appears because of the order constraint on γ .

As in Müller and Ríos Insua (1998), we propose a hybrid, partially marginalized MCMC posterior sampling scheme to implement inference in the fixed mixed model

(1). We use a Metropolis step to update the input to hidden weights γ_j using the marginal likelihood over (β, λ): $p(D' \mid y_1, \ldots, y_q, \gamma, \sigma^2)$ to partly avoid the random walk nature of Metropolis algorithm:

1. Given current values of χ and σ^2 (β and λ are marginalized), for each γ_j, $j = 1, \ldots, M$, generate a proposal $\widetilde{\gamma_j} \sim N(\gamma_j, c\Sigma_\gamma)$, calculate the acceptance probability

$$a = \min\left[1, \frac{p(\widetilde{\gamma} \mid \mu_\gamma, \Sigma_\gamma) p(D' \mid y_1, \ldots, y_q, \widetilde{\gamma}, \sigma^2)}{p(\gamma \mid \mu_\gamma, \Sigma_\gamma) p(D' \mid y_1, \ldots, y_q, \gamma, \sigma^2)}\right]$$

   where $\gamma = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_j, \gamma_{j+1}, \ldots, \gamma_M)$, $\widetilde{\gamma} = (\gamma_1, \ldots, \gamma_{j-1}, \widetilde{\gamma}_j, \gamma_{j+1}, \ldots, \gamma_M)$. With probability a replace γ by $\widetilde{\gamma}$ and rearrange indices if necessary to satisfy order constraint. Otherwise, leave γ_j unchanged.

2. Generate new values for parameters, drawing from their full conditional posteriors:

$$\begin{aligned}
\widetilde{\beta} &\sim p\left(\beta \mid D', \gamma, \lambda, \sigma^2, \chi\right) \text{ is a multivariate normal distribution.} \\
\widetilde{\lambda} &\sim p\left(\lambda \mid D', \gamma, \beta, \sigma^2, \chi\right) \text{ is a multivariate normal distribution.} \\
\widetilde{\sigma^2} &\sim p\left(\sigma^2 \mid D', \gamma, \beta, \lambda\right) \text{ is an inverse Gamma distribution.}
\end{aligned}$$

3. Finally, given current values of $\left(\gamma, \beta, \lambda, \sigma^2\right)$ generate a new value for each hyperparameter by drawing from their complete conditional posterior distributions:

$$\begin{aligned}
\widetilde{\mu_\beta} &\sim p\left(\mu_\beta \mid D', \beta, \sigma_\beta^2\right) \text{ is a normal distribution.} \\
\widetilde{\sigma_\beta^2} &\sim p\left(\sigma_\beta^2 \mid D', \beta, \mu_\beta\right) \text{ is an inverse Gamma distribution.} \\
\widetilde{\mu_\lambda} &\sim p\left(\mu_\lambda \mid D', \lambda, \sigma_\lambda^2\right) \text{ is a multivariate normal distribution.} \\
\widetilde{\sigma_\lambda^2} &\sim p\left(\sigma_\lambda^2 \mid D', \lambda, \mu_\lambda\right) \text{ is an inverse Gamma distribution.} \\
\widetilde{\mu_\gamma} &\sim p\left(\mu_\gamma \mid D', \gamma, \Sigma_\gamma\right) \text{ is a multivariate normal distribution.} \\
\widetilde{\Sigma_\gamma} &\sim p\left(\Sigma_\gamma \mid D', \gamma, \mu_\gamma\right) \text{ is an inverse Wishart distribution.}
\end{aligned}$$

A similar sampling scheme can be used when using a neural net model without linear term (3) but likelihood marginalization would be just over β. Posterior inference with the normal linear autoregression model (2) is straightforward, see e.g. Gamerman (1997).

4 MODELLING UNCERTAINTY ABOUT THE ARCHITECTURE

We now extend the model to include inference about model uncertainty. In fact, the posterior distribution (6) should be written including a reference to the model k considered:

$$p(\lambda, \beta, \gamma, \sigma^2, \chi \mid D', k) = p\left(\theta_k \mid D', k\right)$$

where $\theta_k = \left(\lambda, \beta, \gamma, \sigma^2, \chi\right)$ represents parameters and hyperparameters in model k.

We index models with a pair of indexes m_{hk}, where $h = 0(1)$ indicates absence (presence) of the linear term; $k = 0, 1, \ldots$ indicates the number of hidden nodes in the NN term. Therefore, m_{10} is the linear autoregression model (2), m_{0k} is the $FFNN$ model (3) with k hidden nodes and m_{1k} is the mixed models (1) with k hidden nodes.

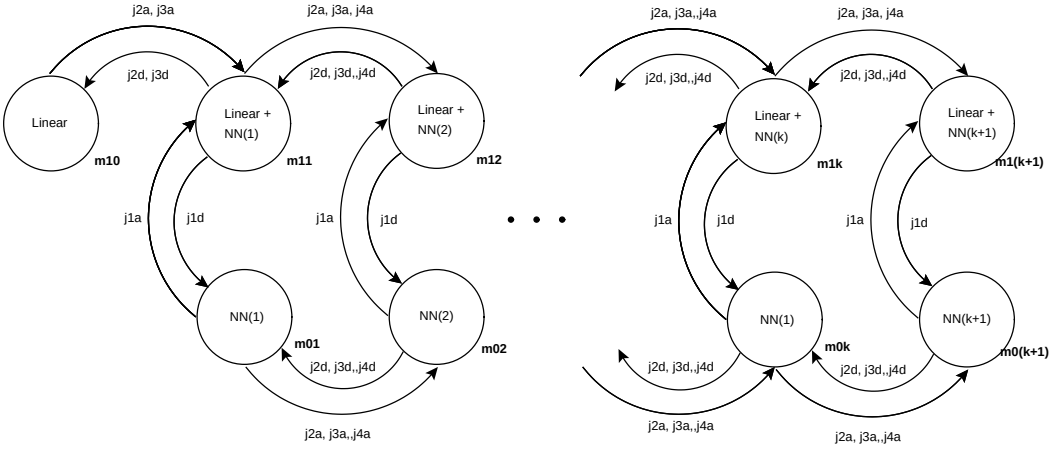

Figure 1: Possible models along with moves available from each model.

Note that models m_{0k}, $k \geq 1$ are nested, as well as m_{1k}, $k \geq 1$. It would be possible to think of the linear model m_{10} as a degenerate case of the mixed model $m_{11}$when $\beta = 0$ and of models m_{0k}, $k \geq 1$, as degenerate m_{1k}, $k \geq 1$ when $\lambda = 0$ and finally consider all models above as nested models. However, given our model exploration strategy outlined below, we prefer to view them as non nested.

We wish to design moves between models to get a good coverage of the model space. When dealing with nested models, it is common to add or delete model components, consequently jumping between models, using *add/delete* or *split/combine* moves pairs (Green, 1995; Richardson and Green, 1997). Similarly, we could define two reversible jump pairs: *add/delete* an arbitrary node selected at random and *add/delete* the linear term. With such strategy, it would be possible to reach a model from any other model. However our experience shows that their acceptance rate is low and model space is not adequately covered.

To avoid this, let us introduce the notion of *LID* (linearized, irrelevant and duplicate) nodes.

Consider a node (β_j, γ_j) with $\gamma_j = (\gamma_{0j}, \gamma_{1j}, \ldots, \gamma_{pj})$ in a model m_{0k} or m_{1k}. We call it:

Definition 1. *(Approximately) linearized if its output $\beta_j \varphi(x_t' \gamma_j)$ is nearly a hyperplane in the input/output space for the input data range.*

Definition 2. *(Approximately) irrelevant if $\beta_j \simeq 0$.*

Definition 3. *We say nodes (β_j, γ_j) and (β_k, γ_k) are (approximately) duplicate nodes if $\|\gamma_j - \gamma_k\| \simeq 0$.*

Note that a model with an (approximately) irrelevant node has (almost) identical likelihood as a model without it. A model with (approximately) duplicate nodes will perform similarly to a model with a single node $(\beta_j + \beta_k, \gamma_j)$ instead of them. Finally the linear behaviour of a linearized node may be assimilated into a linear term or an other linearized nodes. The appearance of *LID* nodes, therefore, cause problems of model identifiability and multimodality in model space.

Recall that the acceptance probability of a proposed model is computed using the marginal likelihood over β and/or λ whenever possible. This marginalization accelerates convergence. But γ is not easily marginalized. Thus, any proposed model structure change is evaluated with the currently imputed value of γ. Adding irrelevant, duplicate or linearized nodes implies less structural change, so that in our simulation it was more common to accept add moves that add actually *LID* nodes. And viceversa, it is very unlikely to accept a jump to a simpler model than proposes an arbitrary node to be deleted. In fact, it is usually easier that a new *LID* node is added before old ones are deleted.

This motivates an MCMC scheme that includes *add/delete* of *LID* nodes as additional moves to a standard reversible jump scheme. Once moves to delete *LID* nodes are defined, we shall propose the corresponding add nodes moves. Note that whereas delete *LID* nodes, would mitigate our problem, add *LID* nodes would seem to complicate our scheme, following our above discussion. In principle, we could propose these add *LID* nodes with low probability. However, besides ensuring balance, add moves have a useful side effect. Note that with add moves, we have control over when and how a *LID* node is added. As mentioned above, adding *LID* nodes will have more probability to be accepted than adding an arbitrary node because it usually implies a smaller structural change. We expect that once a model with a new *LID* node becomes the current model, the next iterations will delete some other *LID* nodes. In this way, more complex models have a chance of being visited, so it can help to get a wide coverage of model space. In a sense, this is related to some global search optimization techniques in which a bad solution is sometimes proposed to escape from a local minima with the hope of reaching a better minima from it. As Bayesian inference implicitly embodies Occam's razor principle and clever delete moves are defined, we expect simpler models to be more common.

In the rest of this section we show how to characterize a linearized node so as to generate it at random or propose for deletion. We also introduce modifications to a basic *thin/seed* move to deal with duplicate nodes and we put an ad hoc distribution over nodes to increase probability of an irrelevant node to be proposed for deletion. Note that there is a decision to make about how much deterministic a move should be. For example, we could look deterministically for two (approximately) duplicate nodes in the set of nodes and, if they exist, delete them but, in general, it is good to allow for some randomness in move definitions (Green, 1995). At the end of this section a complete reversible jump scheme is outlined. A full description of add/delete moves is given in the Appendixes.

4.1 (Approximately) Linearized Nodes

To facilitate explanation, consider a FFNN model in a regression problem:

$$\begin{aligned} \widehat{y}(x_1, \ldots, x_q) &= f(x_1, \ldots, x_q) = \sum_{j=1}^{M} \beta_j \varphi(\zeta_j) \\ \zeta_j(x_1, \ldots, x_q) &= \gamma_{j0} + \gamma_{j1} x_1 + \ldots + \gamma_{jq} x_q \end{aligned}$$

The net output $\widehat{y}$ is given by a linear combination of hidden node outputs. Each hidden node output $\varphi(\zeta_j) = \dfrac{1}{1 + \exp(-\zeta_j)}$, $j = 1, \ldots, M$, represents a sigmoidal surface in $\Re^{q+1}$, which is approximately linear in its middle range (Figure 3). For any data $(x_{i1}, \ldots, x_{iq})$, $i = 1, \ldots, N$, we may find weights $(\gamma_{j0}, \gamma_{j1}, \ldots, \gamma_{jp})$ so that:

$$-\rho \leq \zeta_j(x_1, \ldots, x_q) \leq \rho, i = 1, \ldots, N \tag{7}$$

where ρ and $\varphi(\zeta_j)$ resembles a hyperplane in the input data domain. Note also that when $\zeta_j > \eta$, say $\eta \geq 4$, the sigmoid is saturated (Figure 3) and $\varphi(\zeta_j) \simeq 1$. It is possible to adjust weights so that saturation is verified for the whole data set.

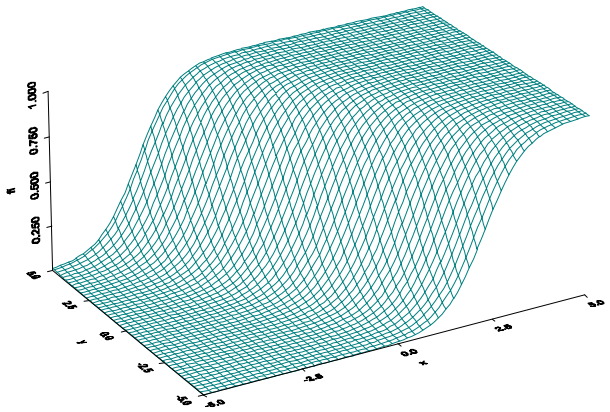

Figure 2: Sigmoidal surface when input space dimension is $q = 2$.

When fitting a FFNN model, some hidden nodes will ocasionally work in the linear zone, behaving approximately as linear regression terms, or in the saturation zone, behaving as a constant term. For example, in Figures 4 and 5 a FFNN with four nodes fits a cosine function. The joint output of three of the four nodes has a linear trend but none of them is working in the linear zone. The fourth node helps to compensate this linear trend so that the cosine is fitted. Also, when using mixed models, some nodes add up their linear behaviour to the linear term.

The previous comments emphasize the need to add and remove linearized nodes to get a good coverage of the model space. When adding a linearized node, we will have to randomly generate weights $(\gamma_{j0}, \gamma_{j1}, \ldots, \gamma_{jq})$ so that the node behaves in the linear zone for the given data set. To show how to achieve this, consider the case in which $q = 2$. For a given node $(\gamma_0, \gamma_1, \gamma_2)$, $\zeta = \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2$ belongs to

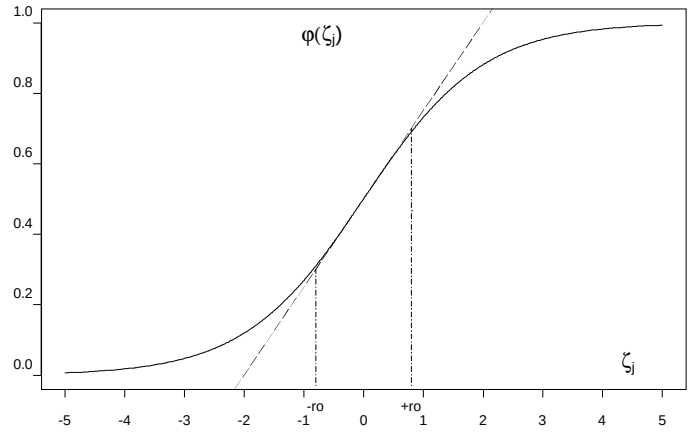

Figure 3: When $-\rho \leq z \leq \rho$, sigmoid function could be well approximated by its tangent in $z = 0$.

a family of straight lines with slope $\dfrac{-\gamma_2}{\gamma_1}$. This node is linearized if (7) is verified. Rewrite (7) as:

$$-\rho - \gamma_0 \leq z\left(x_{i1}, x_{i2}\right) \leq \rho - \gamma_0 \quad \forall \left(x_{i1}, x_{i2}\right), i = 1, \ldots, N \tag{8}$$

where $z\left(x_1, x_2\right) = \gamma_1 x_1 + \gamma_2 x_2 = \kappa\left(s_1 x_1 + v s_2 x_2\right)$ with

$$\begin{aligned} s_i &= sgn(\gamma_i), i = 1, 2 \\ \kappa &= |\gamma_1| \\ v &= \frac{|\gamma_2|}{|\gamma_1|} \end{aligned}$$

For some data points z is maximized or minimized, so the above condition can be summarized as

$$-\rho - \gamma_0 \leq z_{\min} \leq z_{\max} \leq \rho + \gamma_0 \tag{9}$$

where $z_{\min} = z\left(x_1^{z_{\min}}, x_2^{z_{\min}}\right)$ and $z_{\max} = z\left(x_1^{z_{\max}}, x_2^{z_{\max}}\right)$. To find $z_{\min}$ and $z_{\max}$, it is easier to force (9) into a rectangle $\left[\left(x_1^{\min}, x_2^{\min}\right), \left(x_1^{\max}, x_2^{\max}\right)\right]$ containing all input data points:

$$\begin{aligned} x_1^{\min} &= \min_i\left(x_{i1}\right), \qquad x_1^{\max} = \max_i\left(x_{i1}\right) \\ x_2^{\min} &= \min_i\left(x_{i2}\right), \qquad x_2^{\max} = \max_i\left(x_{i2}\right) \end{aligned}$$

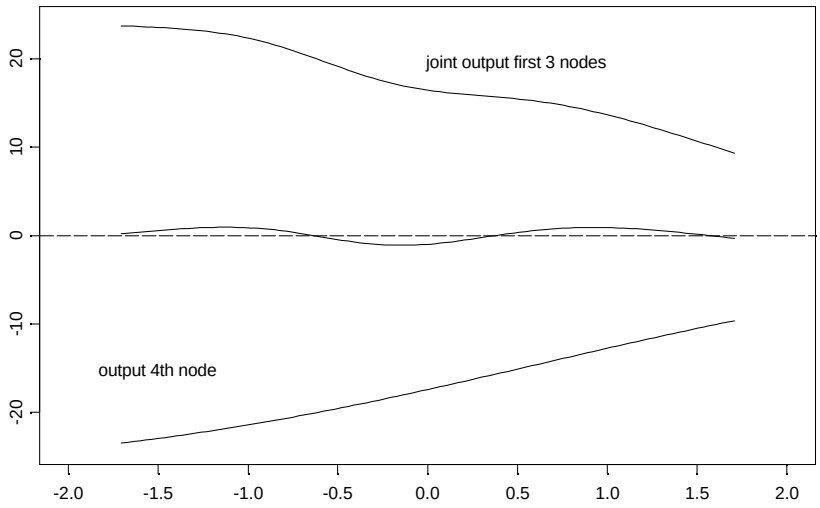

Figure 4: A feedforward neural network with four nodes fitting a cosine function. Three of the four nodes fit the cosine waves with a global up linear trend. The fourth node adds up its linear behaviour to compensate the trend.

We may write then

$$\begin{aligned} x_i^{z_{\max}} &= \frac{1}{2}\left(1+sgn(\gamma_i)\right)x_i^{\max} + \frac{1}{2}\left(1-sgn(\gamma_i)\right)x_i^{\min}, \quad i=1,2 \\ x_i^{z_{\min}} &= \frac{1}{2}\left(1+sgn(\gamma_i)\right)x_i^{\min} + \frac{1}{2}\left(1-sgn(\gamma_i)\right)x_i^{\max}, \quad i=1,2 \end{aligned}$$

and

$$\begin{aligned} z_{\max} &= \gamma_1 x_1^{z_{\max}} + \gamma_2 x_2^{z_{\max}} = \kappa\left(s_1 x_1^{z_{\max}} + v s_2 x_2^{z_{\max}}\right) \\ z_{\min} &= \gamma_1 x_1^{z_{\min}} + \gamma_2 x_2^{z_{\min}} = \kappa\left(s_1 x_1^{z_{\min}} + v s_2 x_2^{z_{\min}}\right) \end{aligned}$$

Let

$$\begin{aligned} L &= z_{\max} - z_{\min} = \frac{\kappa}{\delta} \\ \delta &= \frac{1}{(x_1^{\max} - x_1^{\min}) + v(x_2^{\max} - x_2^{\min})} \end{aligned}$$

Then, rewrite (9) as:

$$\begin{aligned} L &\leq 2\rho \\ -\rho - \gamma_0 + L &\leq z_{\max} \leq \rho - \gamma_0 \end{aligned} \tag{10}$$

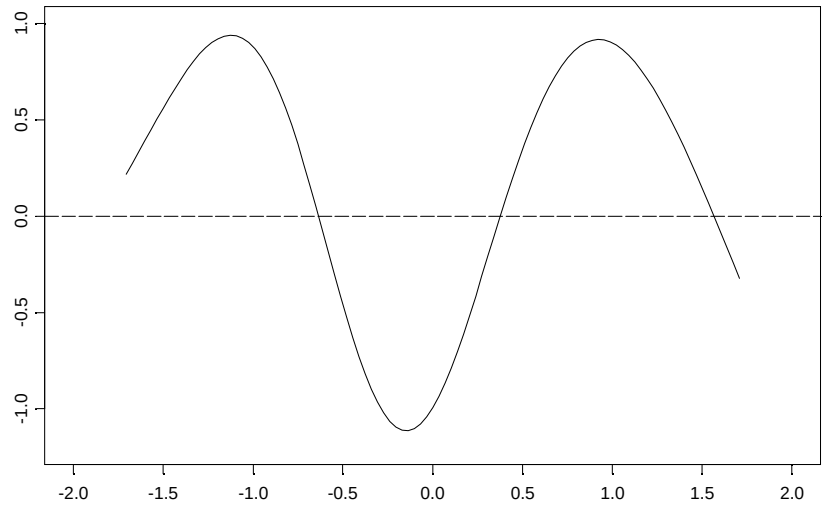

Figure 5: The joint output of the four nodes in the neural net model resembles the target cosine function.

We have three degrees of freedom to verify (10). Setting $v = \dfrac{\gamma_2}{\gamma_1}$ and signs s_i, $i = 1, 2$, fixes the slope of the straight line z which fixes the sigmoidal surface orientation. Once this is defined, we could take a value for κ so that $L \leq 2\rho$, fixing the sigmoidal surface steepness. Finally, we should take γ_0 so that $-\rho - \gamma_0 + L \leq z_{\max} \leq \rho - \gamma_0$, which locates the sigmoidal surface.

Rewrite $\delta = \dfrac{u_1}{x_1^{\max} - x_1^{\min}}$ for some u_1 and

$$\begin{aligned}
L &= \frac{\kappa}{u_1}(x_1^{\max} - x_1^{\min}) = 2\rho b \\
v &= \frac{x_1^{\max} - x_1^{\min}}{x_2^{\max} - x_2^{\min}} \left[\frac{1}{\delta(x_1^{\max} - x_1^{\min})} - 1\right] = \frac{1 - u_1}{u_1} \left[\frac{x_1^{\max} - x_1^{\min}}{x_2^{\max} - x_2^{\min}}\right] \\
\kappa &= \frac{2\rho b u_1}{x_1^{\max} - x_1^{\min}} \\
\gamma_0 &= -\rho + L - z_{\max} + u_0\,(2\rho - L)
\end{aligned}$$

These new quantities may be interpreted as follows:

- u_1 is related with the orientation of the sigmoidal surface, therefore with the slope of the straight lines z. In fact, $\dfrac{1 - u_1}{u_1}$ should take values in $[0, \infty)$.
- b is related with the width L of the interval $(z_{\min}, z_{\max})$ which is related with the steepness of the sigmoidal surface. It should take values in $(0, 1)$.

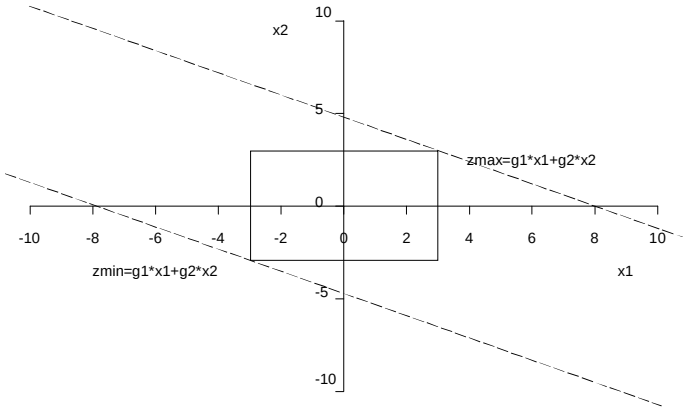

Figure 6: To find $z_{\min}$ and $z_{\max}$, we force $-\rho-\gamma_0 \leq z_{\min} \leq z_{\max} \leq \rho+\gamma_0$ into a rectangle $\left[\left(x_1^{\min}, x_2^{\min}\right), (x_1^{\max}, x_2^{\max})\right]$ containing all input data points.

- u_0 is related with the sigmoidal surface location given by how interval $(z_{\min}, z_{\max})$ is placed within interval $(-\rho-\gamma_0, \rho-\gamma_0)$. It should take values in $(0,1)$.

Let us rewrite γ_0, γ_1 and γ_2 in terms of the above quantities:

$$\begin{aligned}
\gamma_1 &= s_1 \frac{2\rho b u_1}{(x_1^{\max} - x_1^{\min})} \\
\gamma_2 &= s_2 \frac{2\rho b\,(1-u_1)}{(x_2^{\max} - x_2^{\min})} \\
\gamma_0 &= 2\rho u_0\,(1-b) + \rho(2b-1) - (\gamma_1 x_1^{z_{\max}} + \gamma_2 x_2^{z_{\max}})
\end{aligned}$$

Suppose now that we randomly generate weights $(\gamma_0, \gamma_1, \gamma_2)$ from a multivariate normal distribution accepting those verifying (10). The sample distribution of these weights has no recognizable shape, but the distributions of the other quantities above are suggested by their histograms:

$$\begin{aligned}
&p(s_i = +1) = 0.5; \qquad p(s_i = -1) = 0.5 \\
&b \sim Beta(2,2); \qquad u_1 \sim U(0,1)\,; \qquad u_0 \sim U(0,1)
\end{aligned}$$

We can therefore generate at random weights corresponding to linearized nodes.

The generalization to $q > 2$ is straightforward:

$$\begin{aligned}
\gamma_i &= \frac{2\rho\varphi_i}{(x_i^{\max} - x_i^{\min})}, \;\; i = 1, \ldots, q \\
\gamma_0 &= 2\rho u_0\,(1-\varphi_0) + \rho(2\varphi_0 - 1) - (\gamma_1 x_1^{z_{\max}} + \ldots + \gamma_q x_q^{z_{\max}})
\end{aligned}$$

with $\varphi_0 = \sum_{i=1}^{q} |\varphi_i|$ and $x_i^{z_{\max}}$ defined as above.

To generate $\varphi_i, i = 1, .., q$, first generate:

$$\begin{aligned}(\xi_1, \ldots, \xi_{q-1}) &\sim Dirichlet(1, 1, \ldots, 1);\ \xi_q = 1 - \sum_{i=1}^{q-1} \xi_i \\ b &\sim Beta(q, 2) \\ p(s_i &= +1) = p(s_i = -1) = \frac{1}{2}, i = 1, \ldots, q \\ u_0 &\sim U(0, 1)\end{aligned}$$

and define: $\varphi_i = bs_i\xi_i$.

To propose a linearized node for deletion, we simply choose a node at random. Instead of finding out, deterministically, which node, if any, is a linearized one, we let some randomness in the move definition. Once a node is selected, we recover the quantities defined previously:

$$\begin{aligned}\xi_i &= |\varphi_i| = \frac{|\gamma_i| \left(x_i^{\max} - x_i^{\min}\right)}{2\rho} \qquad i = 1, \ldots, q \\ u &= \frac{\gamma_0 + (\gamma_1 x_1^{z_{\max}} + \ldots + \gamma_q x_q^{z_{\max}}) - \rho(2\varphi_0 - 1)}{2\rho(1 - \varphi_0)} \\ b &= \varphi_0 = \sum_{i=1}^{q} |\varphi_i|\end{aligned}$$

Should the node selected be approximately linearized, the following conditions would verify:

$$\begin{aligned}\sum_{i=1}^{q} \xi_i &= 1 \text{ and } \xi_i < 1 \qquad \forall i = 1, \ldots, q \\ u &\in [0, 1] \\ b &\in [0, 1]\end{aligned}$$

The density of any of these parameters, see the Appendix B , would be zero if they do not satisfy that condition. Hence, the probability of deleting a non linearized node with this move is zero, i.e. the move would be rejected.

4.2 (Approximately) Irrelevant nodes

We let chance to deal with generation of an irrelevant node. We modify a basic birth/death move just to put some knowledge when selecting a node for deletion.We choose a node at random with probabilities $p_j = \frac{\psi_j}{\sum_{k=1}^{M+1} \psi_k}$, $j = 1, \ldots, M$ where $\psi_j = \frac{1}{\epsilon\sqrt{2\pi}} \exp\left\{\frac{-1}{2\epsilon^2} x^2\right\}$ so that more weight is given to those nodes whose β_j is almost zero. Full details are given in the Appendix C.

4.3 (Approximately) Duplicate nodes

Consider two (approximately) duplicate nodes (β_j, γ_j) and (β_k, γ_k) in a model m_{0k} or m_{1k}. As we have mentioned we could propose a simpler model collapsing them into a new node $(\beta_j + \beta_k, \gamma_j)$. The proposed add/delete pair is a basic thin/seed move with minor modifications.

To add a duplicate node, take a node (β_j, γ_j) at random and introduce small perturbations to produce two (approximately) duplicate new nodes, hence substituting (β_j, γ_j) by $\left(\widetilde{\beta}_j, \widetilde{\gamma}_j\right)$ and $\left(\widetilde{\beta}_{j+1}, \widetilde{\gamma}_{j+1}\right)$ with:

$$\begin{aligned} \widetilde{\beta}_j &= \beta_j(1-v), \widetilde{\beta}_{j+1} = \beta_j v \\ \widetilde{\gamma}_j &= \gamma_j, \qquad \widetilde{\gamma}_{j+1} = \gamma_j + \delta \end{aligned}$$

Perturbations v and δ are generated from:

$$\begin{aligned} v &\sim Beta(2,2) \\ \delta &\sim N\left(0, c\Sigma_\gamma\right) \end{aligned}$$

where c is a small enough constant (say $c = 0.01$). As we introduced an order constraint on γ, the add move proposed will be rejected if $\widetilde{\gamma}_j$ and $\widetilde{\gamma}_{j+1}$ do not satisfy the order constraint.

Given the order constraint on γ, we expect that for a given (β_j, γ_j), the next node $(\beta_{j+1}, \gamma_{j+1})$ could be an almost duplicate version of it. Then, we propose to remove $(\beta_{j+1}, \gamma_{j+1})$ and transform (β_j, γ_j) as $\widetilde{\beta}_j = \beta_j + \beta_{j+1}$ and $\widetilde{\gamma}_j = \gamma_j$. See Appendix D for full details.

4.4 Our reversible jump scheme

We propose then the following moves:

1. Add/Delete Linear term (j_{1a}, j_{1d})
2. Add/Delete Linearised node (j_{2a}, j_{2d})
3. Add/Delete Arbitrary nodes (j_{3a}, j_{3d})
4. Add/Delete Duplicate nodes (j_{4a}, j_{4d})

Not all of them are reachable from a given model. For example, we cannot delete a linear term when the current model is the linear model, nor can we delete duplicate nodes if we have just one node (see Figure 1).
Specifically:

- From the linear autoregression model m_{10} valid moves are: j_{2a} and j_{4a}.
- From a feedforward neural net model with one node m_{01} valid moves are: j_{1a}, j_{2a}, j_{3a} and j_{4a}.
- From a feedforward neural net model with $k \geq 2$ nodes m_{0k}, valid moves are: j_{1a}, j_{2a}, j_{2d}, j_{3a}, j_{3d}, j_{4a} and j_{4d}.

- From a mixed model with $k \geq 1$ the nodes m_{1k}, valid moves are: j_{1d}, j_{2a}, j_{2d}, j_{3a}, j_{3d}, j_{4a} and j_{4d}.

We will assume that from a given model all reachable moves are equally likely. Formally, unreachable moves are given zero probabilty. Note that the definition of each move depends on the current model where to jump from. For example, adding a linearized node when current model is m_{10} involves proposing neural network parameters and hyperparameters absent in m_{10}. However, when current model is $m_{0k}, k \geq 1$, adding a linearized node is a jump between nested models with shared parameters and hyperparameters. Shared parameters and hyperparameters could be proposed with same values as in previous model. But, non shared parameters and hyperparameters have to be proposed from scratch, for example, from their prior distribution. This usually implies less chance for the move to be accepted. It would be useful for convergence purposes to generate non shared parameters using proposals centered in the values they had the last time the model was visited. We are not going to deal with this matter here and will use priors as proposals. Similar comments apply to other moves. See Appendix for a complete description.
The general reversible jump posterior inference scheme is as follows:

1. Start with an initial model m_{hk}, $h = \{0, 1\}$, $k = 0, 1, \ldots$ and initial values for its parameters (and hyperparameters if needed) (for example, prior means).

   Until convergence is achieved iterate through steps 2 to 4:

2. With probability p_1 (say, $p_1 = 0.5$) decide to stay within the current model, otherwise (with probability $1 - p_1$) decide to move to another model
3. If staying in the current model, perform the MCMC scheme described previously.
4. If moving to another model, select at random a move from the list of reachable moves with probabilities assigned as mentioned and propose a new model accordingly. If accepted, new model is the current model.

5 EXAMPLES

We illustrate the methodology described above with two time series data sets.

5.1 Lynx data

As a first example we consider a time series data giving the annual number of lynx trappings in the Mackenzie River Distric of North–West Canada for the period 1821 to 1934 (Piestley 1988) (see Figure 7).

In NN applications, the data set is often split into two subsets, one for estimation and the other for validation. We split the lynx data set into a training data set (first 72) and a test data set (last 38 observations).

Prior distributions for the unknown parameters and hyperparameters in the model, are chosen as described in Section 2, using the following choices for the

hyperparameters distributions:

$$\begin{array}{lcll} \mu_\beta & \sim & N(0,3), & \sigma^2_\beta \sim InvGamma(9,1) \\ \mu_\lambda & \sim & N((0,0,0),3I), & \sigma^2_\lambda \sim InvGamma(9,1) \\ \mu_\gamma & \sim & N((0,0,0),3I), & \Sigma_\gamma \sim InvWishart(10,2.5I) \\ \sigma^2 & \sim & InvGamma(1,1) & \end{array}$$

Also, the unknown number of nodes M, is given a geometric prior distribution with parameter $\alpha = 2$. Two runs were carried out from different starting points, namely $M = 0$ and $M = 15$. A burn–in of 1000 iterations was used and then a further 9000 iterations were monitored for inference purposes. Figure 8 and Figure 9 show trace plots of the Markov chains for the number of nodes in the FFNN and histogram of the posterior distribution of M, respectively, suggesting $M = 2$ as the most likely number of nodes for the hidden layer of the FFNN term.

Finally Figure 10 shows the time series (log-transformated and detrented) and the one-step ahead forecast values for the test data set, showing good performance of the Bayesian nonlinear autoregression model developed.

5.2 Airline data

As a second example, we will consider the well known international airline data by Box and Jenkins (1970). This time series is one of the most commonly used seasonal ARIMA models. It was modelled by Box and Jenkins (1970) as ARIMA$(011)(011)_{12}$ and in fact, this classic model is usually known as *airline model* due to the time series.

Exploratory data analysis suggests seasonal behaviour of the time series with similar properties exihibited every 12 observations (every 12 months). We shall therefore model each y_t on the basis of the inmediately past values, y_{t-1}, y_{t-2} as well as corresponding seasonal values y_{t-13}, y_{t-14}.

Once more we initialized the MCMC algorithm by two well dispersed values for the number of hidden nodes M and let the Reversible jump MCMC algorithm described to run for a sufficient number of iterations, 2000 burn in and another 18000 iteration for making inference. Similar comments about prior choices to the previous example, apply also here. Figure 13 shows the histogram of the posterior distribution of M, in this case, three hidden nodes should be included in the hidden layer of the FFNN.

It is of interest to look at the reversible jump algorithm development. Table 1 resumes the number of times each node has been propose and accepted, as well as percentage of acceptance for each plausible move. Note that acceptance rates are considerably bigger for irrelevant nodes than for linearized or duplicate nodes, given, perhaps, to the simplicity of the movement.

Finally Figure 14 shows the time series data and the forecasts for the test data set, almost undistinguishable.

Move	Times proposed	Times accepted
j1a	1	1
j1d	840	1
j2a	483	130
j2d	831	36
j3a	731	229
j3d	823	396
j4a	428	68
j4d	837	7

Table 1: Number of times proposed, and times accepted for the different moves.

6 CONCLUSIONS

We have presented a reversible jump algorithm for the analysis of FFNNs viewed as nonlinear autoregression models. The advantages of the Bayesian approach outweighs its additional computational effort: as no local optimization algorithms are used, local minima issues are mitigated; model selection is performed as part of Bayesian methodology without need of deciding explicitly a range of models to select from and having to use ad hoc methods to rank them; Bayesian analysis embodies naturally Occam's razor principle, the principle of preferring simpler models to complex models, also it is possible to include a priori information about parameters and number of hidden nodes in accordance to this principle; the final product of a Bayesian analysis is a predictive distribution that averages over all possible models and parameters values, so uncertainty about predictions is estimated and overfitting risk is reduced.

Many issues remain to be explored. An obvious extension is the application of the reversible jump algorithm based on *LID* nodes to other standard NN applications like regression, classification or density estimation. Also we have confined to FFNNs but many other NN models, analysed from a Bayesian point of view, might prove useful. Relating to time series analysis, here we have assumed autoregressive order q to be known in advance. We could also model uncertainty about q as a problem of model selection.

7 ACKNOWLEDGMENTS

This work has been supported by projects from CICYT, CAM and URJC. One of the authors (RMD) gratefully acknowledges receipt of a postdoctoral fellowship from CAM (Comunidad Autónoma de Madrid).

APPENDIX A: Add/Delete Linear Term

We propose this move with probability B_1

Add Linear Term

$$\begin{aligned} NN(m) &\longrightarrow Linear + NN(m), m \geq 1 \\ (\theta, u_\theta) &\longrightarrow (\phi, u_\phi) \end{aligned}$$

where

$$\begin{aligned} \theta &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma^2_\beta, \mu_\gamma, \Sigma_\gamma\right\} \\ u_\theta &= \{\lambda, \mu_\lambda, \Sigma_\lambda\} \\ \phi &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma^2_\beta, \mu_\gamma, \Sigma_\gamma, \lambda, \mu_\lambda, \Sigma_\lambda\right\} \end{aligned}$$

The Jacobian of the one-to-one transformation $\phi = g(\theta, u_\theta)$ is $J = 1$.
The proposal is given by

$$\begin{aligned} \mu_\lambda &\sim N(m_\lambda, S_\lambda) \\ \Sigma_\lambda &\sim IWishart(w_\lambda, \Omega_\lambda) \\ \lambda &\sim N(\mu_\lambda, \Sigma_\lambda). \end{aligned}$$

Thus

$$\begin{aligned} \frac{p(\phi \mid j)}{p(\theta \mid k)} &= p(\lambda \mid \mu_\lambda, \Sigma_\lambda) p(\mu_\lambda \mid m_\lambda, S_\lambda) p(\Sigma_\lambda \mid w_\lambda, \Omega_\lambda) \\ \frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K \text{ equiprobable models: } p(i) = \frac{1}{K}, \forall i \\ \frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{1d}}{B_{1a}} = 1 \qquad \text{assuming } B_{1d} = B_{1a} = \frac{B_1}{2} \\ \frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= \frac{1}{N(\lambda \mid \mu_\lambda, \Sigma_\lambda) N(\mu_\lambda \mid m_\lambda, S_\lambda) IW(\Sigma_\lambda \mid w_\lambda, \Omega_\lambda)} \end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min\left(1, \frac{p(y \mid \phi)}{p(y \mid \theta)}\right)$$

Delete Linear Term

$$\begin{aligned} Linear + NN(m) &\longrightarrow NN(m), m \geq 1 \\ (\theta, u_\theta) &\longrightarrow (\phi, u_\phi) \end{aligned}$$

where

$$\begin{aligned}
\theta &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma, \lambda, \mu_\lambda, \Sigma_\lambda\right\} \\
\phi &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma\right\} \\
u_\phi &= \{\lambda, \mu_\lambda, \Sigma_\lambda\}
\end{aligned}$$

The Jacobian of the one-to-one transformation $(\phi, u_\phi) = g(\theta)$ is $J = 1$ and

$$\begin{aligned}
\frac{p(\phi \mid j)}{p(\theta \mid k)} &= \frac{1}{p(\lambda \mid \mu_\lambda, \Sigma_\lambda) p(\mu_\lambda \mid m_\lambda, S_\lambda) p(\Sigma_\lambda \mid w_\lambda, \Omega_\lambda)} \\
\frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K\text{equiprobable models } p(i) = \frac{1}{K}, \forall i \\
\frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{1a}}{B_{1d}} = 1 \qquad \text{assuming } B_{1d} = B_{1a} = \frac{B_1}{2} \\
\frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= N(\lambda \mid \mu_\lambda, \Sigma_\lambda) N(\mu_\lambda \mid m_\lambda, S_\lambda) IW(\Sigma_\lambda \mid w_\lambda, \Omega_\lambda)
\end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min\left(1, \frac{p(y \mid \phi)}{p(y \mid \theta)}\right)$$

APPENDIX B: Add/Delete Linearized Node

We propose this move with probability B_2

Add Linearized Node

$$\begin{aligned}
Linear + NN(m) &\longrightarrow Linear + NN(m+1), m \geq 1 \\
(\theta, u_\theta) &\longrightarrow (\phi, u_\phi)
\end{aligned}$$

where

$$\begin{aligned}
\theta &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma\right\} \\
u_\theta &= \{\varpi, \varphi_1, \varphi_2, \ldots, \varphi_p, u\} \\
\phi &= \left\{\widetilde{\beta}_1, \widetilde{\beta}_2, \ldots, \widetilde{\beta}_M, \widetilde{\beta}_{M+1}, \widetilde{\gamma}_1, \widetilde{\gamma}_2, \ldots, \widetilde{\gamma}_M, \widetilde{\gamma}_{M+1}, \widetilde{\sigma}^2, \widetilde{\mu}_\beta, \widetilde{\sigma}_\beta^2, \widetilde{\mu}_\gamma, \widetilde{\Sigma}_\gamma\right\}
\end{aligned}$$

The transformation $\phi = g(\theta, u_\theta)$ is defined by

$$\begin{aligned}
\widetilde{\beta}_j &= \beta_j, \quad \widetilde{\gamma}_j = \gamma_j \qquad j = 1, \ldots, M \\
\widetilde{\beta}_{M+1} &= \varpi, \quad \widetilde{\gamma}_{M+1,i} = \frac{2\rho\varphi_i}{(x_i^{\max} - x_i^{\min})} \qquad i = 1, \ldots, p \\
\widetilde{\gamma}_{M+1,0} &= 2\rho u(1 - \varphi_0) + \rho(2\varphi_0 - 1) - (\widetilde{\gamma}_{M+1,1} x_1^{z_{\max}} + \ldots + \widetilde{\gamma}_{M+1,p} x_p^{z_{\max}}) \\
\widetilde{\sigma}^2 &= \sigma^2, \quad \widetilde{\mu}_\beta = \mu_\beta, \quad \widetilde{\sigma}_\beta^2 = \sigma_\beta^2, \quad \widetilde{\mu}_\gamma = \mu_\gamma, \widetilde{\Sigma}_\gamma = \Sigma_\gamma
\end{aligned}$$

where

$$\begin{aligned}\varphi_0 &= \sum_{i=1}^{p} |\varphi_i| \\ x_i^{z_{\max}} &= \frac{1}{2}\left(1 + sgn(\varphi_i)\right) x_i^{\max} + \frac{1}{2}\left(1 - sgn(\varphi_i)\right) x_i^{\min}, \quad i = 1, \ldots, p.\end{aligned}$$

The Jacobian of the transformation is

$$J = \begin{vmatrix} 2\rho(1-\varphi_0) & 0 & \ldots & 0 & 0 \\ - & \frac{2\rho}{(x_1^{\max} - x_1^{\min})} & \ldots & 0 & 0 \\ - & 0 & \ldots & 0 & 0 \\ - & 0 & \ldots & \frac{2\rho}{(x_{p-1}^{\max} - x_{p-1}^{\min})} & 0 \\ - & 0 & \ldots & 0 & \frac{2\rho}{(x_p^{\max} - x_p^{\min})} \end{vmatrix} = \frac{(2\rho)^p (1-\varphi_0)}{\prod_{i=1}^{p} (x_i^{\max} - x_i^{\min})}$$

The proposal is given by:

$$\begin{aligned} u &\sim U(0,1) \\ \varpi &\sim N(\mu_\beta, \sigma_\beta^2) \\ b &\sim Beta(p,2) \\ (\xi_1, \ldots, \xi_{p-1}) &\sim Dirichlet(1,1,\ldots,1) \\ p(s_i = +1) &= p(s_i = -1) = \frac{1}{2},\ i = 1, \ldots, p, \end{aligned}$$

then, define

$$\varphi_i = s_i b \xi_i,\ i = 1, \ldots, p.$$

and it can be shown that

$$f_\varphi(\varphi_1, \ldots, \varphi_p) = \frac{1}{2^p}\Gamma(p+2)\left(1 - \sum_{i=1}^{p} \varphi_i\right), \quad 0 < |\varphi_i| < 1 - \sum_{k \neq i} |\varphi_k| \qquad i = 1, \ldots, p$$

Thus

$$\begin{aligned} \frac{p(\phi \mid j)}{p(\theta \mid k)} &= p(\beta_{M+1} \mid \widetilde{\mu}_\beta, \widetilde{\sigma}_\beta^2) p\left(\gamma_{M+1} \mid \widetilde{\mu}_\gamma, \widetilde{\Sigma}_\gamma\right) \frac{(M+1)!}{M!} \\ \frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K\text{equiprobable models } p(i) = \frac{1}{K}, \forall i \\ \frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{21d}}{B_{21a}} \frac{\frac{1}{M+1}}{1} = \frac{1}{M+1} \qquad \text{assuming } B_{2d} = B_{2a} = \frac{B_2}{2} \\ \frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= \frac{1}{\frac{1}{2^p}\Gamma(p+2)\left(1 - \sum_{i=1}^{p} \varphi_i\right) N(\varpi \mid \mu_\beta, \sigma_\beta^2)} \end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min\left(1, \frac{p(y \mid \phi)p\left(\gamma_{M+1} \mid \widetilde{\mu}_\gamma, \widetilde{\Sigma}_\gamma\right)}{p(y \mid \theta)\frac{1}{2^p}\Gamma(p+2)\left(1-\sum_{i=1}^p \varphi_i\right)} \frac{(2\rho)^p(1-\varphi_0)}{\prod_{i=1}^p (x_i^{\max} - x_i^{\min})}\right)$$

Delete Linearized Node

$$\begin{array}{rcl} Linear + NN(m+1) & \longrightarrow & Linear + NN(m), m \geq 1 \\ (\theta, u_\theta) & \longrightarrow & (\phi, u_\phi) \end{array}$$

where

$$\begin{aligned} \theta &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \beta_{M+1}, \gamma_1, \gamma_2, \ldots, \gamma_M, \gamma_{M+1}, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma\right\} \\ \phi &= \left\{\widetilde{\beta}_1, \widetilde{\beta}_2, \ldots, \widetilde{\beta}_M, \widetilde{\gamma}_1, \widetilde{\gamma}_2, \ldots, \widetilde{\gamma}_M, \widetilde{\sigma}^2, \widetilde{\mu}_\beta, \widetilde{\sigma}_\beta^2, \widetilde{\mu}_\gamma, \widetilde{\Sigma}_\gamma\right\} \\ u_\phi &= \left\{\varpi, \varphi_1, \varphi_2, \ldots, \varphi_p, u\right\} \end{aligned}$$

Select node $h = 1, \ldots, M+1$ at random, take nodes in $\theta : 1, \ldots, h-1, h+1, \ldots, M+1$ and relabel as $1, \ldots, M$.
The transformation $(\phi, u_\phi) = g(\theta)$ is defined by

$$\begin{aligned} \widetilde{\beta}_j &= \beta_j, \widetilde{\gamma}_j = \gamma_j \qquad j = 1, \ldots, M \\ \widetilde{\sigma}^2 &= \sigma^2, \widetilde{\mu}_\beta = \mu_\beta, \widetilde{\sigma}_\beta^2 = \sigma_\beta^2, \widetilde{\mu}_\gamma = \mu_\gamma, \widetilde{\Sigma}_\gamma = \Sigma_\gamma \\ \varpi &= \beta_h \\ \varphi_i &= \frac{\gamma_{h,i}(x_i^{\max} - x_i^{\min})}{2\rho} \qquad i = 1, \ldots, p \\ u &= \frac{\gamma_{h,0} + (\gamma_{h,1}x_1^{z_{\max}} + \ldots + \gamma_{h,p}x_p^{z_{\max}}) - \rho(2\varphi_0 - 1)}{2\rho(1-\varphi_0)} \end{aligned}$$

where

$$\begin{aligned} \varphi_0 &= \sum_{i=1}^p |\varphi_i|, \\ x_i^{z_{\max}} &= \frac{1}{2}(1 + sgn(\varphi_i))\, x_i^{\max} + \frac{1}{2}(1 - sgn(\varphi_i))\, x_i^{\min}, \ i = 1, \ldots, p. \end{aligned}$$

The Jacobian of the transformation is

$$J = \begin{vmatrix} \frac{1}{2\rho(1-\varphi_0)} & 0 & \ldots & 0 & 0 \\ - & \frac{(x_1^{\max} - x_1^{\min})}{2\rho} & \ldots & 0 & 0 \\ - & 0 & \ldots & 0 & 0 \\ - & 0 & \ldots & \frac{(x_{p-1}^{\max} - x_{p-1}^{\min})}{2\rho} & 0 \\ - & 0 & \ldots & 0 & \frac{(x_p^{\max} - x_p^{\min})}{2\rho} \end{vmatrix} = \frac{\prod_{i=1}^p (x_i^{\max} - x_i^{\min})}{(2\rho)^p(1-\varphi_0)}$$

If γ_h is not a linearised node then the acceptance probability is

$$\alpha = 0$$

otherwise

$$\begin{aligned}
\frac{p(\phi \mid j)}{p(\theta \mid k)} &= \frac{1}{p(\beta_h \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_h \mid \mu_\gamma, \Sigma_\gamma\right)} \frac{M!}{(M+1)!} \\
\frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K\text{equiprobable models } p(i) = \frac{1}{K}, \forall i \\
\frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{21a}}{B_{21d}} \frac{1}{\frac{1}{M+1}} = M+1 \qquad \text{assuming } B_{21d} = B_{21a} = \frac{B_{21}}{2} \\
\frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= \left(\prod_{i=1}^{p} I_{[0,1-\sum_{k \neq i} |\varphi_k|]}(|\varphi_i|) \right) \frac{1}{2^p} \Gamma(p+2) \left(1 - \sum_{i=1}^{p} \varphi_i \right) I_{[0,1]}(u) N(\varpi \mid \mu_\beta, \sigma_\beta^2)
\end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min \left(1, \frac{p(y \mid \phi) \left(\prod_{i=1}^{p} I_{[0,1-\sum_{k \neq i} |\varphi_k|]}(|\varphi_i|) \right) \frac{1}{2^p} \Gamma(p+2) \left(1 - \sum_{i=1}^{p} \varphi_i\right) I_{[0,1]}(u)}{p(y \mid \theta) p\left(\gamma_h \mid \mu_\gamma, \Sigma_\gamma\right)} \frac{\prod_{i=1}^{p} (x_i^{\max} - x_i^{\min})}{(2\rho)^p (1 - \varphi_0)} \right)$$

APPENDIX C: Add/Delete Irrelevant Node

We propose this move with probability B_3

Add Irrelevant Node

$$\begin{aligned}
Linear + NN(m) &\longrightarrow Linear + NN(m+1), m \geq 1 \\
(\theta, u_\theta) &\longrightarrow (\phi, u_\phi)
\end{aligned}$$

where

$$\begin{aligned}
\theta &= \left\{ \beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma \right\} \\
u_\theta &= \left\{ \beta_{M+1}, \gamma_{M+1} \right\} \\
\phi &= \left\{ \beta_1, \beta_2, \ldots, \beta_M, \beta_{M+1}, \gamma_1, \gamma_2, \ldots, \gamma_M, \gamma_{M+1}, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma \right\}
\end{aligned}$$

The proposal is given by

$$\begin{aligned}
\gamma_{M+1} &\sim N(\mu_\gamma, \Sigma_\gamma) \\
\beta_{M+1} &\sim N(\mu_\beta, \sigma_\beta^2)
\end{aligned}$$

rearranging γ to satisfy order constraint $\gamma_{1,p} < \ldots < \gamma_{M+1,p}$.

The Jacobian of the one-to-one transformation $(\phi, u_\phi) = g(\theta)$ is $J = 1$. Also

$$\begin{aligned}
\frac{p(\phi \mid j)}{p(\theta \mid k)} &= \frac{p(\beta_{M+1} \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_{M+1} \mid \mu_\gamma, \Sigma_\gamma\right)}{1} \frac{(M+1)!}{M!} \\
\frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K \text{ equiprobable models } p(i) = \frac{1}{K}, \forall i \\
\frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{3d}}{B_{3a}} \frac{\psi_{M+1}}{\sum_{k=1}^{M+1} \psi_k} = \frac{\psi_{M+1}}{\sum_{k=1}^{M+1} \psi_k} \qquad \text{assuming } B_{3d} = B_{3a} = \frac{B_3}{2} \\
\frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= \frac{1}{p(\beta_{M+1} \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_{M+1} \mid \mu_\gamma, \Sigma_\gamma\right)}
\end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min\left(1, \frac{p(y \mid \phi)}{p(y \mid \theta)} \frac{(M+1)\psi_{M+1}}{\sum_{k=1}^{M+1} \psi_k}\right)$$

Delete Irrelevant Node

$$\begin{aligned}
Linear + NN(m+1) &\longrightarrow Linear + NN(m), m \geq 0 \\
(\theta, u_\theta) &\longrightarrow (\phi, u_\phi)
\end{aligned}$$

where

$$\begin{aligned}
\theta &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \beta_{M+1}, \gamma_1, \gamma_2, \ldots, \gamma_M, \gamma_{M+1}, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma\right\} \\
\phi &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma\right\} \\
u_\phi &= \{\varpi, \delta\}
\end{aligned}$$

Let $\psi_j = \frac{1}{\epsilon\sqrt{2\pi}} \exp\left\{\frac{-1}{2\epsilon^2} x^2\right\}$ $j = 1, \ldots, M+1$ for a given ϵ. Choose node h at random with probabilities $p_j = \frac{\psi_j}{\sum_{k=1}^{M+1} \psi_k}, j = 1, \ldots, M+1$. Remove node h and relabel as $1, \ldots, M$.
The transformation $(\phi, u_\phi) = g(\theta)$ is defined by

$$\begin{aligned}
\varpi &= \beta_h \\
\delta &= \gamma_h
\end{aligned}$$

with Jacobian $J = 1$ and

$$\begin{aligned}
\frac{p(\phi \mid j)}{p(\theta \mid k)} &= \frac{1}{p(\beta_h \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_h \mid \mu_\gamma, \Sigma_\gamma\right)} \frac{M!}{(M+1)!} \\
\frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K \text{ equiprobable models } p(i) = \frac{1}{K}, \forall i \\
\frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{3a}}{B_{3d}} \frac{1}{\frac{\psi_h}{\sum_{k=1}^{M+1} \psi_k}} = \frac{\sum_{k=1}^{M+1} \psi_k}{\psi_h} \qquad \text{assuming } B_{3d} = B_{3a} = \frac{B_3}{2} \\
\frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= \frac{p(\varpi \mid \mu_\beta, \sigma_\beta^2) p\left(\delta \mid \mu_\gamma, \Sigma_\gamma\right)}{1}
\end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min\left(1, \frac{p(y \mid \phi)}{p(y \mid \theta)} \frac{\sum_{k=1}^{M+1} \psi_k}{(M+1)\psi_h}\right)$$

APPENDIX D: Add/Delete Duplicate Node

We propose this move with probability B_4

Add Duplicate Node

$$\begin{aligned}
Linear + NN(m) &\longrightarrow Linear + NN(m+1), m \geq 1 \\
(\theta, u_\theta) &\longrightarrow (\phi, u_\phi)
\end{aligned}$$

where

$$\begin{aligned}
\theta &= \left\{\beta_1, \beta_2, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma\right\} \\
u_\theta &= \{\nu, \delta\} \\
\phi &= \left\{\beta_1, \beta_2, \ldots, \widetilde{\beta}_h, \widetilde{\beta}_{h+1}, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \widetilde{\gamma}_h, \widetilde{\gamma}_{h+1}, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma\right\}
\end{aligned}$$

Choose $h = 1, \ldots, M$ at random and propose $\left(\widetilde{\beta}_h, \widetilde{\gamma}_h\right)$ and $(\widetilde{\beta}_{h+1}, \widetilde{\gamma}_{h+1})$ where

$$\begin{aligned}
\widetilde{\beta}_h &= \beta_h(1-\nu) \\
\widetilde{\gamma}_h &= \gamma_h \\
\widetilde{\beta}_{h+1} &= \beta_h \nu \\
\widetilde{\gamma}_{h+1} &= \gamma_h + \delta
\end{aligned}$$

and reject if $(\gamma_1, \gamma_2, \ldots, \gamma_h, \widetilde{\gamma}_{h+1}, \ldots, \gamma_M)$ is not ordered.
The Jacobian of the transformation $\phi = g(\theta, u_\theta)$ is defined by

$$J = \left| \frac{\partial \left(\widetilde{\beta}_h, \widetilde{\gamma}_h, \widetilde{\beta}_{h+1}, \widetilde{\gamma}_{h+1} \right)}{\partial \left(\gamma_h, \beta_h, \nu, \delta \right)} \right| = \left| \begin{array}{cccc} 0 & 1 & 0 & 1 \\ (1-\nu) & 0 & \nu & 0 \\ -\beta_h & 0 & \beta_h & 0 \\ 0 & 0 & 0 & 1 \end{array} \right| = |\beta_h|$$

The proposal is given by

$$\begin{aligned} \nu &\sim Beta(2,2) \\ \delta &\sim N(0, cS_\gamma). \end{aligned}$$

Thus

$$\begin{aligned} \frac{p(\phi \mid j)}{p(\theta \mid k)} &= \frac{p(\widetilde{\beta}_h \mid \mu_\beta, \sigma_\beta^2) p\left(\widetilde{\gamma}_h \mid \mu_\gamma, \Sigma_\gamma\right) p(\widetilde{\beta}_{h+1} \mid \mu_\beta, \sigma_\beta^2) p\left(\widetilde{\gamma}_{h+1} \mid \mu_\gamma, \Sigma_\gamma\right)}{p(\beta_h \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_h \mid \mu_\gamma, \Sigma_\gamma\right)} \frac{(M+1)!}{M!} \\ \frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K \text{ equiprobable models } p(i) = \frac{1}{K}, \forall i \\ \frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{4d}}{B_{4d}} \frac{\frac{1}{M}}{1} = \frac{1}{M} \qquad \text{assuming } B_{4d} = B_{4a} = \frac{B_4}{2} \\ \frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= \frac{1}{Beta(\nu \mid 2,2) N(\delta \mid 0, cS_\gamma)} \end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min \left(1, \frac{p(y \mid \phi) p(\widetilde{\beta}_h \mid \mu_\beta, \sigma_\beta^2) p\left(\widetilde{\gamma}_h \mid \mu_\gamma, \Sigma_\gamma\right) p(\widetilde{\beta}_{h+1} \mid \mu_\beta, \sigma_\beta^2) p\left(\widetilde{\gamma}_{h+1} \mid \mu_\gamma, \Sigma_\gamma\right) (M+1)}{p(y \mid \theta) p(\beta_h \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_h \mid \mu_\gamma, \Sigma_\gamma\right) \; M \, Beta(\nu \mid 2,2) N(\delta \mid 0, cS_\gamma)} \; |\beta_h| \right)$$

Delete Duplicate Node

$$\begin{aligned} Linear + NN(m+1) &\longrightarrow Linear + NN(m), m \geq 1 \\ (\theta, u_\theta) &\longrightarrow (\phi, u_\phi) \end{aligned}$$

where

$$\begin{aligned} \theta &= \left\{ \beta_1, \beta_2, \ldots, \beta_M, \beta_{M+1}, \gamma_1, \gamma_2, \ldots, \gamma_M, \gamma_{M+1}, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma \right\} \\ \phi &= \left\{ \beta_1, \beta_2, \ldots, \widetilde{\beta}_h, \ldots, \beta_M, \gamma_1, \gamma_2, \ldots, \widetilde{\gamma}_h, \ldots, \gamma_M, \sigma^2, \mu_\beta, \sigma_\beta^2, \mu_\gamma, \Sigma_\gamma \right\} \\ u_\phi &= \{\nu, \delta\} \end{aligned}$$

select $h = 1, \ldots, M$ at random. Remove $(\beta_{h+1}, \gamma_{h+1})$ and propose $\left(\widetilde{\beta}_h, \widetilde{\gamma}_h \right)$ where

$$\begin{aligned} \widetilde{\beta}_h &= \beta_h + \beta_{h+1} \\ \widetilde{\gamma}_h &= \gamma_h \\ \nu &= \frac{\beta_{h+1}}{\beta_h + \beta_{h+1}} \\ \delta &= \gamma_{h+1} - \gamma_h \end{aligned}$$

Then relabel as $1, \ldots, M$.
The Jacobian of the transformation $(\phi, u_\phi) = g(\theta)$ is

$$J = \left| \frac{\partial \left(\widetilde{\beta}_h, \widetilde{\gamma}_h, \nu, \delta \right)}{\partial \left(\gamma_h, \gamma_{h+1}, \beta_h, \beta_{h+1} \right)} \right| = \left| \begin{array}{cccc} 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & \frac{-\beta_{h+1}}{\left(\beta_h + \beta_{h+1}\right)^2} & 0 \\ 1 & 0 & \frac{\beta_h}{\left(\beta_h + \beta_{h+1}\right)^2} & 0 \end{array} \right| = \left| \frac{1}{\beta_h + \beta_{h+1}} \right|$$

and

$$\begin{aligned} \frac{p(\phi \mid j)}{p(\theta \mid k)} &= \frac{p(\widetilde{\beta}_h \mid \mu_\beta, \sigma_\beta^2) p\left(\widetilde{\gamma}_h \mid \mu_\gamma, \Sigma_\gamma\right)}{p(\beta_h \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_h \mid \mu_\gamma, \Sigma_\gamma\right) p(\beta_{h+1} \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_{h+1} \mid \mu_\gamma, \Sigma_\gamma\right)} \frac{M!}{(M+1)!} \\ \frac{p(j)}{p(k)} &= 1 \qquad \text{assuming } K \text{ equiprobable models } p(i) = \frac{1}{K}, \forall i \\ \frac{p_{j \longrightarrow k}}{p_{k \longrightarrow j}} &= \frac{B_{4a}}{B_{4d}} \frac{1}{\frac{1}{M}} = M \qquad \text{assuming } B_{4d} = B_{4a} = \frac{B_4}{2} \\ \frac{q(u_\phi \mid \phi)}{q(u_\theta \mid \theta)} &= I_{(0,1)}(\nu) Beta(\nu \mid 2, 2) N(\delta \mid 0, cS_\gamma) \end{aligned}$$

so that the acceptance probability is given by

$$\alpha = \min \left(1, \frac{p(y \mid \phi) p(\widetilde{\beta}_h \mid \mu_\beta, \sigma_\beta^2) p\left(\widetilde{\gamma}_h \mid \mu_\gamma, \Sigma_\gamma\right) \; M \, I_{(0,1)}(\nu) Beta(\nu \mid 2, 2) N(\delta \mid 0, cS_\gamma)}{p(y \mid \theta) p(\beta_h \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_h \mid \mu_\gamma, \Sigma_\gamma\right) p(\beta_{h+1} \mid \mu_\beta, \sigma_\beta^2) p\left(\gamma_{h+1} \mid \mu_\gamma, \Sigma_\gamma\right) (M+1)} \left| \frac{1}{\beta_h + \beta_{h+1}} \right| \right)$$

Notes:

- Rearrangement of γ to satisfy order constraint should be done once this jump between models has been accepted, to save computational time.
- $p(y \mid \phi)$ and $p(y \mid \theta)$ may be substituted by marginal likelihood in β
- It can be easily demonstrated that the results in appendixes A, B and C, for models containing a linea term $Linear + NN(m)$ also apply for corresponding models without the linear term, $NN(m)$.

References

[1] Box, G. E. P. and Jenkins, G. M. (1970), *Time Series Analysis, Forecasting and Control*, Holden-Day.

[2] Carlin, B. P. and Chib, S. (1995), "Bayesian model choice via Markov chain Monte Carlo methods", *Journal of the Royal Statistical Society B*, 57(3), 473–484.

[3] Dellaportas, P, Forster, J.J., Ntzoufras, I. (1997), "On Bayesian model and variable selection using MCMC", Technical Report.

[4] Elman, J.L. (1990), "Finding structure in time", *Cognitive Science*, 14, 179–211

[5] Faraway, J. and Chatfield C. (1998), "Time series forecasting with neural networks: a comparative study using the airline data", *Journal of the Royal Statistical Society*, Applied Statistics (Series C), 47, Part 2, 231-250

[6] Foster, B., Collopy, F., and Ungar, L.H., (1992), "Neural Network Forecasting of Short Noisy Time Series", *Computers and Chem. Engr.*, 16(4):293-298.

[7] Gamerman, D. (1997), *Markov Chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman & Hall.

[8] Gershenfeld, N.A. and Weigend, A.S. (1993), "The future of time series: learning and understanding", in *Time Series Prediction: Forecasting the future and understanding the past* (eds A.S. Weigend and N.A. Gershenfeld), pp. 1-70, SFI Studies in the Sciences of Complexity, Proc. Vol. XV, Addison Wesley.

[9] Green, P. (1995), "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination." *Biometrika*, 82, 711-732.

[10] Hill, T., O'Connor, M. and Remus, W. (1996), "Neural network models for time serie forecasts", *Mangmnt Sci.*, 42, 1082-1092

[11] Jordan, M. I. (1986), "Attractor dynamics and parallelism in a connectionist sequential machine", in *Proceedings of the Eighth Conference of the Cognitice Science Society.*

[12] Mackay, D.J.C. (1992a), "Bayesian interpolation", *Neural Computation* 4, 415-447.

[13] Mackay, D.J.C. (1992b), "A practical Bayesian framework for backpropagation networks", *Neural Computation* 4, 448-472.

[14] Mackay, D.J.C. (1992c), "Bayesian Methods for Adaptive Models". Ph.D. Thesis. California Institute of Technology.

[15] Müller, P.; Ríos Insua, D. (1998), "Issues in Bayesian Analysis of Neural Network Models", *Neural Computation*, 10, 571-592.

[16] Neal, R.M. (1996), *Bayesian learning for neural networks*, Springer-Verlag. New York

[17] Newton, M. A. and Raftery, A. E. (1994), "Approximate Bayesian inference with the weighted likelihood bootstrap (with discussion)", *Journal of the Royal Statistical Society B*, 56(1):3–48.

[18] Priestley, M.B. (1988) *Non-linear and Non-stationary Time Series Analysis*. Academic Press, London.

[19] Remus, W., O'Connor, M. and K. Griggs, (1998), "The Impact of Information of Unknown Correctness on the Judgmental Forecasting Process", *International Journal of Forecasting* , 14, pp. 313-322.

[20] Richardson, S. and Green, P.J. (1997), "On Bayesian analysis of mixtures with an unknown number of components", *Journal of the Royal Statistical Society B*, 59, 731-792.

[21] Ríos Insua, D; Müller, P. (1998), "Feedforward neural networks for nonparametric regression" in *Practical nonparametric and semiparametric bayesian statistics*. Dey, Müller, Sinha (eds.), Springer.

[22] Sarle, W. (1994), "Neural networks and statistical models", in *Proc. 19th A. SAS Users Group Int. Conf.*, pp. 1538-1550. Cary: SAS Institute

[23] Stern, H. (1996), "Neural networks in applied statistics (with discussion)", *Technometrics*, 38, 205-220

[24] Tang, Z., de Almeida, C. and Fishwick, P.A. (1991), "Time series forecasting using neural networks versus Box-Jenkins methodology", *Simulation*, 57, 303-310

[25] Williams, R. J. and Zipser, D. (1989), "A learning algorithm for continually running fully recurrent neural networks", *Neural Computation*, 1, 270-280.

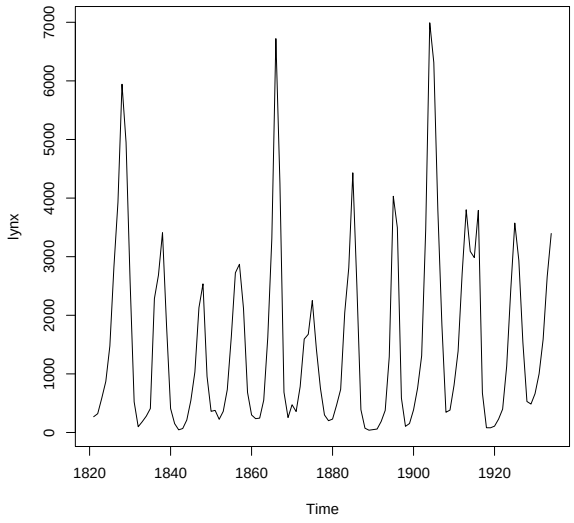

Figure 7: Annual number of lynx trappings in the Mackenzie River Distric of North–West Canada for the period 1821 to 1934 (Piestley 1988)

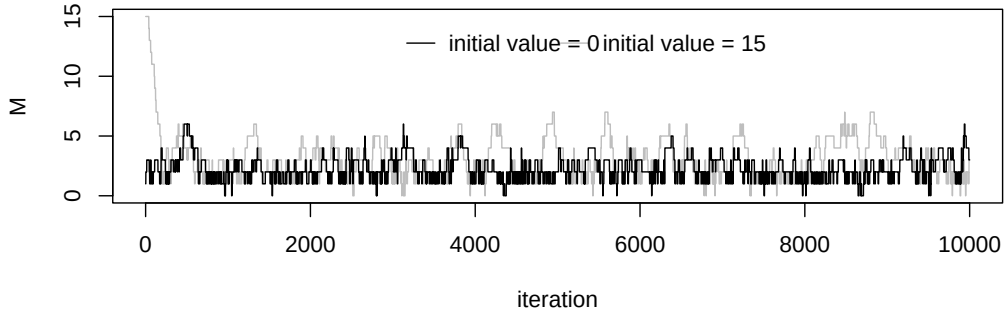

Figure 8: Lynx Data:Trace plots of the Markov chains for the number of hidden nodes, from two well dispersed starting points, $M = 0$ and $M = 15$.

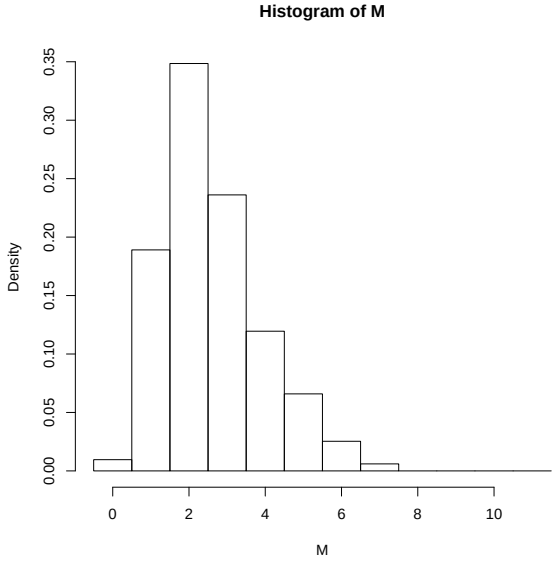

Figure 9: Lynx Data: Histogram of the posterior distribution of M, suggesting $M = 2$ nodes for the hidden layer of the FFNN term.

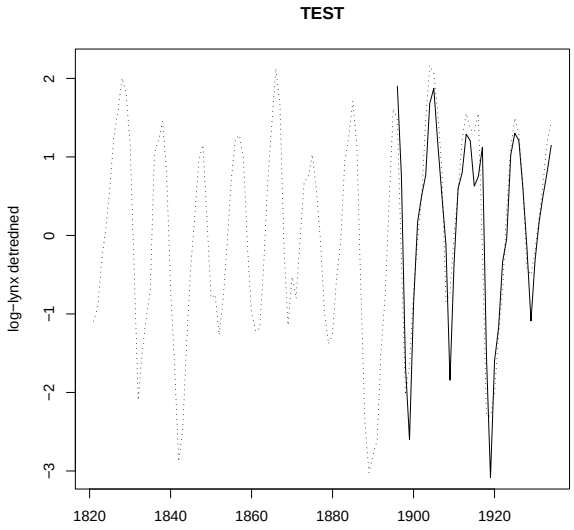

Figure 10: Time series (log-transformed and detrented) and predicted values for the test data set.

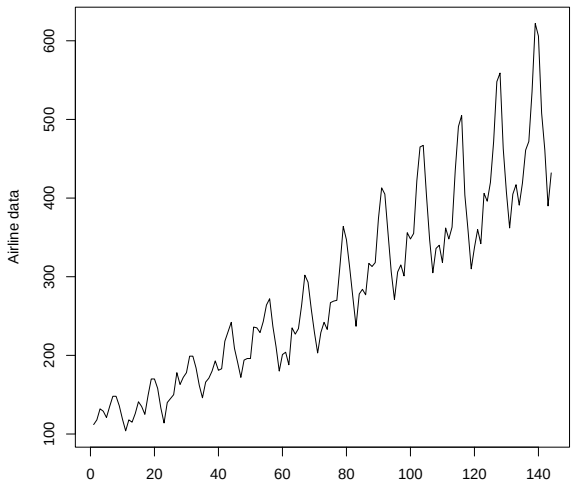

Figure 11:

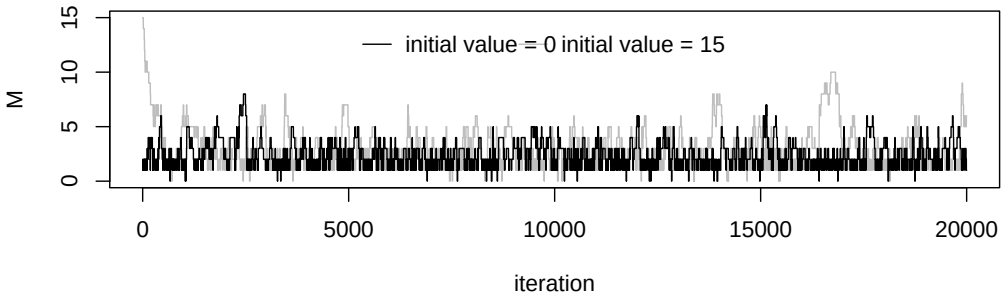

Figure 12:

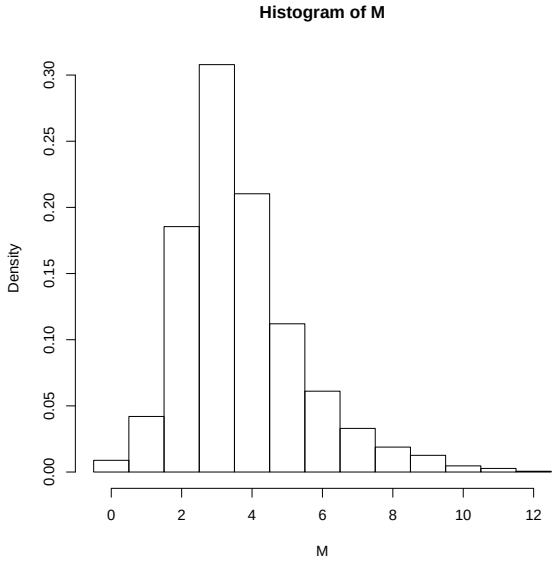

Figure 13: Airline data: Histogram of the posterior distribution of M, suggesting $M = 2$ nodes for the hidden layer of the FFNN term.

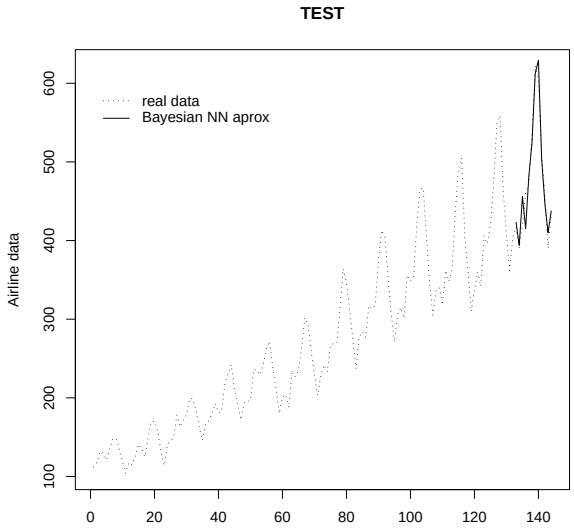

Figure 14: Airline time series data and predicted values for the test data set.