Neural network based approximations to posterior densities: a class of flexible sampling methods with applications to reduced rank models [*]

Lennart F. Hoogerheide[†], Johan F. Kaashoek & Herman K. van Dijk

Econometric Institute, Erasmus University Rotterdam

Econometric Institute Report EI 2004-19

April 2004

Abstract

Likelihoods and posteriors of econometric models with strong endogeneity and weak instruments may exhibit rather non-elliptical contours in the parameter space. This feature also holds for cointegration models when near non-stationarity occurs and determining the number of cointegrating relations is a nontrivial issue, and in mixture processes where the modes are relatively far apart. The performance of Monte Carlo integration methods like importance sampling or Markov Chain Monte Carlo procedures greatly depends in all these cases on the choice of the importance or candidate density. Such a density has to be 'close' to the target density in order to yield numerically accurate results with efficient sampling. Neural networks seem to be natural importance or candidate densities, as they have a universal approximation property and are easy to sample from. That is, conditionally upon the specification of the neural network, sampling can be done either directly or using a Gibbs sampling technique, possibly using auxiliary variables. A key step in the proposed class of methods is the construction of a neural network that approximates the target density accurately. The methods are tested on a set of illustrative models which include a mixture of normal distributions, a Bayesian instrumental variable regression problem with weak instruments and near non-identification, a cointegration model with near non-stationarity and a two-regime growth model for US recessions and expansions. These examples involve experiments with non-standard, non-elliptical posterior distributions. The results indicate the feasibility of the neural network approach.

Keywords: importance sampling, Markov chain Monte Carlo, neural networks, Bayesian inference. **JEL classification:** C11, C15, C45

[*] A preliminary version of this paper was presented at the 2003 ASA-meeting, ESEM and EC^2. We are indebted to several participants of these meetings for helpful comments.

[†] Correspondence to: L.F. Hoogerheide, Econometric Institute, Erasmus University Rotterdam, P.O. Box 1738, NL-3000 DR Rotterdam, The Netherlands. E-mail: lhoogerheide@few.eur.nl

1 Introduction

Markov Chain Monte Carlo (MCMC) methods like Metropolis-Hastings (MH) and Gibbs sampling are extensively used in Bayesian analyses of econometric and statistical models. The theory of Markov chain samplers starts with Metropolis et al. (1953) and Hastings (1970). An important technical paper on MCMC methods is due to Tierney (1994). Well-known econometric studies are provided by Chib and Greenberg (1996) and Geweke (1999). Indirect independence sampling methods such as importance sampling (IS) have also been successfully applied within Bayesian inference. Importance sampling, see Hammersley and Handscomb (1964), has been introduced in Bayesian inference by Kloek and Van Dijk (1978) and is further developed by Van Dijk and Kloek (1980,1984) and Geweke (1989).

However, in practice, the convergence behavior of Monte Carlo methods is still often uncertain. The complex structure of a model or some extraordinary properties of the data may cause this problem. We mention three cases. A first example of a complex model is a set of equations with a near reduced rank structure for the matrix of coefficients. Then the Hessian of the likelihood function is almost singular. This may be due to strong endogeneity and weak instruments and/or due to near non-stationarity. We refer to the studies by Schotman and Van Dijk (1991) and Kleibergen and Van Dijk (1994, 1998). Convergence problems of importance sampling with a normal or Student t importance density are described by Van Dijk and Kloek (1984) and Geweke (1989). A second example is due to Hobert and Casella (1996). These authors show that the Gibbs sampler does not converge in the case of a hierarchical linear mixed model if the prior is uniform. The reason is that, although all conditional posteriors are proper, the joint posterior is not. Similar problems may occur in dynamic panel data models using diffuse priors. As a third case we mention a multi-modal target density, which one may encounter in mixture processes with a small number of observations around one of the different modes. This may cause problems for all methods. If the MH candidate density is uni-modal, with a low probability of drawing candidate values in one of the modes, then this mode may be completely missed, even if the sample size gets very large. In this case importance sampling with a uni-modal normal or Student t importance density may yield a sample in which most drawings have a negligible weight and only a few drawings almost completely determine the sampling results.

So, an important problem is the choice of the candidate or importance density, especially when one knows little about the shape of the target density.

In this paper we introduce a class of neural network sampling methods which allow for sampling from a target (posterior) distribution that may be multi-modal or skew, or exhibit strong nonlinear correlation among the parameters. That is, a class of methods to sample from non-elliptical distributions.

The basic idea of the neural network sampling algorithms is simple. First, a neural network is constructed that approximates the target density. An important advantage of neural network functions is their 'universal approximation property'. That is, neural network functions can provide approximations of any square integrable function to any desired accuracy, see Gallant and White (1989). As an application of Kolmogorov's general superposition theorem, the neural network approximation property is eluded by Hecht-Nielsen (1987). Proofs concerning neural network approximations for specific configurations can be found in Gallant and White (1989), Hornik et al. (1989), and Leshno et al. (1993). Stinchcombe (1988,1989) shows that it is the presence of intermediate layers with sufficiently many parallel processing elements that is essential for feedforward networks to possess universal approximation capabilities, and that sigmoid activation functions are not necessary for universal approximation. This approximation property implies that the algorithm can handle certain 'strange' target distributions, like multi-modal, extremely skew, strongly correlated or fat-tailed distributions. Second, this neural network is used as an importance function in IS or as a candidate density in MH. Depending on the specification of the neural network, an important advantage of neural network densities is that they are easy to sample from.

The proposed methods are applied on a set of illustrative examples. We start with a mixture of normal densities. Next we perform some experiments with a Bayesian analysis of an instrumental variable regression model and a cointegration model. Finally, we explore a switching model with recessions and expansions for the US real Gross National Product (GNP) growth. Our results indicate that the neural network approach is feasible in cases where a 'standard' MH, Gibbs or IS approach would fail or be rather slow.

The outline of the paper is as follows. In section 2 we discuss how to construct a neural network approximation to a density, how to sample from a neural network density, and how to use these drawings within the IS or MH algorithm. In section 3 we describe a method yielding estimates of moments of the target distribution without requiring a sampling algorithm. Section 4 shows the feasibility of our approach in a simple example of a mixture of bivariate normal distributions. Section 5 illustrates our algorithms in an example with simulated data in an instrumental variable (IV) regression and a cointegration model. Section 6 contains an empirical example concerning a switching model for the quarterly growth rate of the real GNP in the USA. Conclusions are given in section 7 and technical details are given in the appendices.

2 Approximating with and sampling from neural networks

Consider a certain distribution, for example a posterior distribution, with density function $p(x)$ with $x \in \mathbb{R}^n$. The aim is to investigate some of the characteristics of $p(x)$, for example the mean and/or covariance matrix of a random vector $X \sim p(x)$. The approach followed in this paper is:

1. Find a neural network approximation $nn : \mathbb{R}^n \to \mathbb{R}$ to the target density $p(x)$.
2. Obtain a sample of random points from the density (kernel) $nn(x)$.
3. Perform importance sampling or the Metropolis-Hastings algorithm using this sample in order to obtain estimates of the characteristics of $p(x)$.

Consider a 4-layer feed-forward neural network with functional form:

$$nn(x) = eG_2\left(CG_1(Ax+b)+d\right)+f, \qquad x \in \mathbb{R}^n, \tag{1}$$

where A is $H_1 \times n$, b is $H_1 \times 1$, C is $H_2 \times H_1$, d is $H_2 \times 1$, e is $1 \times H_2$ and $f \in \mathbb{R}$. The integers H_1 and H_2 are interpreted as the numbers of cells in the first and second hidden layer of the neural network, respectively. The vector functions $G_1 : \mathbb{R}^{H_1} \to \mathbb{R}^{H_1}$ and $G_2 : \mathbb{R}^{H_2} \to \mathbb{R}^{H_2}$ are defined by

$$G_1(y) = (g_1(y_1), \cdots, g_1(y_{H_1}))' \quad \text{and} \quad G_2(z) = (g_2(z_1), \cdots, g_2(z_{H_2}))' \tag{2}$$

where $g_1 : \mathbb{R} \to \mathbb{R}$ and $g_2 : \mathbb{R} \to \mathbb{R}$ are the activation functions.

A neural network is used because of its well-known universal approximation property, see e.g. Gallant and White (1989) and Hornik et al. (1989). Stinchcombe (1988) poses a sufficient condition for universal approximation capabilities for hidden layer activation functions other than sigmoid; for instance, this condition is satisfied by continuous probability densities. In the following sections, three specifications of (1) will be used:

Type 1 neural network: A standard three-layer feed-forward neural network (in the notation of (1): $H_2 = 1$, $e = 1$, $f = 0$ and g_2 is the identity $g_2(y) = y$). As activation function g_1 in (2), we take the scaled arctangent function:

$$g_1(y) = \frac{1}{\pi} \arctan(y) + \frac{1}{2}, \qquad y \in \mathbb{R}. \tag{3}$$

The reason for this choice is that this activation function can be analytically integrated infinitely many times. We show in subsection 2.2.1, that this property makes the neural network, in the role of a density kernel, easy to sample from.

Type 2 neural network: A simplified four-layer network of which the second hidden layer consists of only one cell ($H_2 = 1$, $e = 1$, $f = 0$) and with g_2 the exponential function:

$$g_2(y) = \exp(y), \qquad y \in \mathbb{R}. \tag{4}$$

In this case, the activation g_1 in (2) is taken to be a piecewise-linear function, called *plin*:

$$plin(y) = \begin{cases} 0 & y < -1/2 \\ y + 1/2 & -1/2 \leq y \leq 1/2 \\ 1 & y > 1/2 \end{cases} \tag{5}$$

With these activation functions, the neural network function can be analytically integrated (once). We show in subsection 2.2.2, that this property makes Gibbs sampling, see e.g. Geman and Geman (1984), possible. To allow for easy sampling it is sufficient to specify a function g_2 which is positive valued and has an analytical expression for its primitive that is analytically invertible; see subsection 2.2.2. Another example of such a function is the logistic function.

Type 3 neural network: A mixture of Student t distributions:

$$nn(x) = \sum_{h=1}^{H} p_h \, t(x|\mu_h, \Sigma_h, \nu), \tag{6}$$

where p_h ($h = 1, \ldots, H$) are the probabilities of the components and where $t(x|\mu_h, \Sigma_h, \nu)$ is a multivariate t density with mode μ_h, scaling matrix Σ_h, and ν degrees of freedom:

$$t(x|\mu_h, \Sigma_h, \nu) = \frac{\Gamma((\nu + n)/2)}{\Gamma(\nu/2)(\pi\nu)^{n/2}} |\Sigma_h|^{-1/2} \left(1 + \frac{(x - \mu_h)'\Sigma_h^{-1}(x - \mu_h)}{\nu}\right)^{-(\nu+n)/2} \tag{7}$$

Note that this mixture of t densities is a four-layer feed-forward neural network (with parameter restrictions) in which we have, in the notation of (1), $H_2 = H$ (the number of t densities), $H_1 = Hn$, activation functions

$$g_1(y) = y^2 \quad \text{and} \quad g_2(z) = z^{-(\nu+n)} \frac{\Gamma((\nu + n)/2)}{\Gamma(\nu/2)(\pi\nu)^{n/2}},$$

and weights $e_h = p_h \, |\Sigma_h|^{-1/2}$ ($h = 1, \ldots, H$), $f = 0$ and:

$$A = \begin{pmatrix} \Sigma_1^{-1/2} \\ \vdots \\ \Sigma_H^{-1/2} \end{pmatrix}, \quad b = \begin{pmatrix} -\Sigma_1^{-1/2}\mu_1 \\ \vdots \\ -\Sigma_H^{-1/2}\mu_H \end{pmatrix}, \quad C = \begin{pmatrix} \iota_n'/\nu & 0 & \cdots & 0 \\ 0 & \iota_n'/\nu & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \iota_n'/\nu \end{pmatrix}, \quad d = \iota_H,$$

where ι_k is a $k \times 1$ vector of ones. Notice that $(x-\mu_h)'\Sigma_h^{-1}(x-\mu_h)$ is the sum of the squared elements of $\Sigma_h^{-1/2}(x-\mu_h)$.

The reason for this choice is that a mixture of t distributions is easy to sample from, and that the Student t distribution has fatter tails than the normal distribution.

Table 1 gives an overview of the reasons for which we have chosen these particular specifications. The implications shown in this table will be clarified in the sequel of this paper.

Table 1: Motivation of the particular neural network specifications

specification of $nn(x)$	special properties of $nn(x)$		consequences of special properties of $nn(x)$
Type 1	- The activation function g is analytically integrable infinitely many times.	$\Rightarrow$	- Direct sampling from $nn(x)$ is possible. - Analytical expressions exist for the moments of the distribution with density $nn(x)$.
Type 2	- The activation function g is piecewise-linear. - The function g_2 is positive valued and analytically integrable, and its primitive is analytically invertible.	$\Rightarrow$	- Gibbs sampling from $nn(x)$ is possible.
	- The function g_2 is the exponential function.	$\Rightarrow$	- Auxiliary variable Gibbs sampling from $nn(x)$ is possible.
Type 3	- The neural network function $nn(x)$ is a mixture of multivariate t densities.	$\Rightarrow$	- Direct sampling from $nn(x)$ is possible.

In the next subsections we discuss the three steps of our approach: construction of a neural network, sampling from it, and using the sample in IS or MH.

2.1 Constructing a neural network approximation to a density

First, we discuss a procedure to obtain a Type 1 or Type 2 neural network approximation. Second, we describe a method to construct a Type 3 neural network.

2.1.1 Constructing a Type 1 or Type 2 neural network approximation

We suggest the following procedure to obtain a Type 1 or Type 2 neural network approximation to a certain target density $p(x)$. First we draw a set of random uniform points x^i $(i = 1, \ldots, N)$ in the bounded region to which we restrict the random variable $X \in \mathbb{R}^n$ to take its values. Then we approximate the target density $p(x)$ with a neural network by minimizing the sum of squared residuals:

$$SSR(A, b, c, d) = \sum_{i=1}^{N} \left(p(x^i) - nn\left(x^i \middle| A, b, c, d\right)\right)^2. \tag{8}$$

We choose the smallest neural network, i.e. the one with the least hidden cells, that still gives a 'good' approximation to the target distribution. One could define a 'good' approximation as one with a high enough squared correlation R^2.

After that, we check the squared correlation R^2 between the neural network and the target density for a (much) larger set of points than the 'estimation set'. If this R^2 is also high enough, then we say that the approximation is accurate and the estimation set is large enough. In that case the network does not only provide a good approximation to the target density in the points x^i $(i = 1, \ldots, N)$ but also in between. Otherwise, we increase the number of points N and start all over again. For example, we make the set twice as large. This process continues until the set is large enough to allow the neural network to 'feel' the shape of the target density accurately.

In the case of our Type 1 (three-layer) neural network, we also have to deal with the problem that the neural network function is not automatically non-negative for each x. In order to prevent this we add a penalty term to (8), and check for non-negativity between the points x^i $(i = 1, \ldots, N)$ afterwards. If $nn(x)$ is negative for some x, we look for its most negative value, and subtract this negative value from the network's constant d. In that way $nn(x)$ becomes non-negative for each x, so that it is a proper density kernel (on the bounded domain to which we restrict it). In our Type 2 (simplified four-layer) neural network the exponential function, or any positive valued function g_2, implies that non-negativity is automatically taken care of.

2.1.2 Constructing a Type 3 neural network approximation

We suggest the following procedure to obtain a Type 3 neural network approximation to a certain target density.

First we compute the mode μ_1 and scaling matrix Σ_1 of the first Student t distribution in our mixture as the maximum likelihood estimator $\mu_1 = \hat{x}_{ML}$ and its estimated covariance matrix:

$$\Sigma_1 = \widehat{\text{cov}}(\hat{x}_{ML}) = \left(- \left. \frac{\delta^2 \log p(x)}{\delta x \delta x'} \right|_{x=\hat{x}_{ML}} \right)^{-1}$$

Then we draw a set of points x^i $(i = 1, \ldots, N)$ from the 'first stage neural network' $nn(x) = t(x|\mu_1, \Sigma_1, \nu)$, with small ν to allow for fat tails. After that we iteratively add components to the mixture by performing the following steps:

Step 1: Compute the importance sampling weights $w(x^i)$ and scaled weights $\tilde{w}(x^i)$:

$$w(x^i) = \frac{p(x^i)}{nn(x^i)} \quad \text{and} \quad \tilde{w}(x^i) = \frac{w(x^i)}{\sum_{i=1}^{N} w(x^i)} \qquad (i = 1, \ldots, N)$$

We make use of simple diagnostics like the weight of the 5% most influential points to determine the number of components of the mixture. If the importance sampling weights pass the diagnostics, then we stop: the current $nn(x)$ is our Type 3 neural network approximation. Otherwise, go to step 2.

Step 2: Add another t distribution with density $t(x|\mu_h, \Sigma_h, \nu)$ to the mixture, where μ_h and Σ_h are estimates of the mean and covariance matrix of the 'residual distribution' with density kernel:

$$res(x) = \max\{p(x) - c\, nn(x), 0\},$$

where c is a constant; we take $\max\{., 0\}$ to make it a (non-negative) density kernel. These estimates of the mean and covariance matrix are easily obtained by importance sampling with the current $nn(x)$ as the candidate density, using the sample x^i $(i = 1, \ldots, N)$ from $nn(x)$ that we already have. The weights $w_{res}(x^i)$ and scaled weights $\tilde{w}_{res}(x^i)$ $(i = 1, \ldots, N)$ are:

$$w_{res}(x^i) = \frac{res(x^i)}{nn(x^i)} = \max\{w(x^i) - c, 0\} \quad \text{and} \quad \tilde{w}_{res}(x^i) = \frac{w_{res}(x^i)}{\sum_{i=1}^{N} w_{res}(x^i)},$$

and μ_h and Σ_h are obtained as:

$$\mu_h = \sum_{i=1}^{N} \tilde{w}_{res}(x^i) x^i \qquad \Sigma_h = \sum_{i=1}^{N} \tilde{w}_{res}(x^i)(x^i - \mu_h)(x^i - \mu_h)'.$$

There are two issues relevant for the choice of c. First, the new t density should appear at places where $nn(x)$ is too small (relative to $p(x)$). Second, there should be enough points with $w(x^i) > c$ in order to make Σ_h nonsingular. A procedure is to calculate Σ_h for c equal to 10 times the average value of $w(x^i)$; if Σ_h is nonsingular, accept c, otherwise lower c.

Step 3: We now choose the probabilities p_h $(h = 1, \ldots, H)$ in the mixture

$$nn(x) = \sum_{h=1}^{H} p_h\, t(x|\mu_h, \Sigma_h, \nu),$$

by minimizing the (squared) coefficient of variation of the importance sampling weights. First we draw N points x_h^i from each component $t(x|\mu_h, \Sigma_h, \nu)$ $(h = 1, \ldots, H)$. Then we minimize $E[w(x)^2]/E[w(x)]^2$, where:

$$E[w(x)^k] = \frac{1}{N} \sum_{i=1}^{N} \sum_{h=1}^{H} p_h\, w(x_h^i)^k \qquad (k = 1, 2)$$

with

$$w(x_h^i) = \frac{p(x_h^i)}{\sum_{h=1}^{H} p_h\, t\left(x_h^i|\mu_h, \Sigma_h, \nu\right)}.$$

Step 4: Draw a sample of N points x^i $(i = 1, \ldots, N)$ from our new mixture of t distributions:

$$nn(x) = \sum_{h=1}^{H} p_h\, t(x|\mu_h, \Sigma_h, \nu) \qquad (9)$$

and go to step 1; in order to draw a point from (9) we first use a drawing from the $U(0, 1)$ distribution to determine which component $t(x|\mu_h, \Sigma_h, \nu)$ is chosen, and then draw from this multivariate t distribution.

In step 3 it may occur that the latest t density $t(x|\mu_H, \Sigma_H, \nu)$ only gets a negligible probability (i.e. $p_H \approx 0$). In that case we start all over again with a larger number of points N. The idea behind this is that the larger N is, the easier it is for the method to 'feel' the shape of the target density, and to specify the t distributions of the mixture adequately.

2.2 Sampling from a neural network density

In the following subsections we discuss sampling from Type 1 and Type 2 networks. In the previous subsection we already remarked that sampling from a Type 3 network, a mixture of t densities, only requires a draw from the $U(0,1)$ distribution to determine which component is chosen, and a draw from the chosen multivariate t distribution.

2.2.1 Sampling from a Type 1 (three-layer) neural network density

Suppose the joint density kernel of a certain $X \in \mathbb{R}^n$ is given by a standard three-layer feed-forward neural network function with an activation function that is analytically integrable infinitely many times. Since the neural network function is a linear combination of these activation functions, the neural network function itself is integrable infinitely many times. Hence one can directly sample from the neural network by iteratively drawing the elements X_i $(i = 1, \dots, n)$ in the following way:

$$\begin{array}{l} \text{Draw } x_1 \text{ from } nn(x_1) \\ \text{Draw } x_2 \text{ from } nn(x_2|x_1) \\ \text{Draw } x_3 \text{ from } nn(x_3|x_1, x_2) \\ \qquad\quad \vdots \\ \text{Draw } x_n \text{ from } nn(x_n|x_1, x_2, x_3, \cdots, x_{n-1}) \end{array} \tag{10}$$

where $nn(x_1)$, $nn(x_2|x_1)$, $nn(x_3|x_1, x_2)$, etc. are the marginal and conditional neural network densities corresponding to the joint density kernel $nn(x)$. The marginal distribution function $CDF_{nn}(x_1)$:

$$CDF_{nn}(x_1) = \frac{\int_{-\infty}^{x_1}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} nn(\tilde{x}_1, x_2, \dots, x_n)dx_n \cdots dx_2 d\tilde{x}_1}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} nn(\tilde{x}_1, x_2, \dots, x_n)dx_n \cdots dx_2 d\tilde{x}_1}, \tag{11}$$

and the conditional distribution function $CDF_{nn}(x_2|x_1)$

$$CDF_{nn}(x_2|x_1) = \frac{\int_{-\infty}^{x_2}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} nn(x_1, \tilde{x}_2, \dots, x_n)dx_n \cdots dx_3 d\tilde{x}_2}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} nn(x_1, \tilde{x}_2, \dots, x_n)dx_n \cdots dx_3 d\tilde{x}_2} \tag{12}$$

etc. can be evaluated analytically.

An example of an activation function that can be analytically integrated infinitely many times, is the scaled arctangent function in (3). Some useful integration formulas for this activation function are given in appendices A.1 and A.2.

So, one can easily sample from the density $nn(x_1)$ or $nn(x_2|x_1)$ in formulas (10) by drawing random variables U_i $(i = 1, \dots, n)$ from the uniform distribution on $[0, 1]$ and then finding the scalars x_i $(i = 1, \dots, n)$ for which $U = CDF_{nn}(x_1)$, $U = CDF_{nn}(x_2|x_1)$, etc. The calculation of x_i $(i = 1, \dots, n)$ is done numerically with an algorithm such as the bisection method.

2.2.2 Sampling from a Type 2 (simplified four-layer) neural network density

Suppose the joint density kernel of a certain $X \in \mathbb{R}^n$ is given by the Type 2 neural network with g_2 the exponential function and g_1 the piecewise-linear function in (5). It is fairly easy to perform Gibbs sampling from this distribution, as one can divide the (bounded) domain of each X_i $(i = 1, \dots, n)$ into a finite number of intervals on which the conditional neural network density is just the exponent of a linear function. Therefore we can analytically integrate the conditional neural network density, and draw from it using the inverse transformation method. Note that the three properties of g_2 mentioned below formula (5) are used here explicitly. Details are given in appendix B.1.

The Gibbs sampling procedure consists of iteratively sampling from one-dimensional conditional distributions:

Specify feasible starting values $x^0 = (x_1^0, \cdots, x_n^0)$.

Do for $j = 1, 2, \dots, m$

$$\begin{array}{ll} x_1^{j+1} & \text{from } nn(x_1 | x_2^j, x_2^j, \cdots, x_n^j) \\ x_2^{j+1} & \text{from } nn(x_2 | x_1^{j+1}, x_3^j, \cdots, x_n^j) \\ x_3^{j+1} & \text{from } nn(x_3 | x_1^{j+1}, x_2^{j+1}, x_4^j, \cdots, x_n^j) \\ \vdots & \\ x_n^{j+1} & \text{from } nn(x_n | x_1^{j+1}, x_2^{j+1}, x_3^{j+1}, \cdots, x_{n-1}^{j+1}) \end{array} \tag{13}$$

Under certain regularity conditions, the sequence $\{x^0, x^1, \cdots, x^j, \cdots\}$ converges to a sample from the distribution with joint density $nn(x_1, \dots, x_n)$.

It is also possible to use a different method to draw from a four-layer neural network density: auxiliary variable Gibbs sampling. Using this method, we do not have to restrict ourselves to the piecewise-linear activation function *plin*. It allows for well-known activation functions such as the logistic and scaled arctangent functions. Auxiliary variable Gibbs sampling is a Gibbs sampling technique, developed by Damien et al. (1999). The method is based on work of Edwards and Sokal (1988). In this method, latent variables are introduced in an artificial way in order to facilitate drawing from the full set of conditional distributions.

Auxiliary variable Gibbs sampling is possible if the density kernel p can be decomposed as follows:

$$p(x) \propto \pi(x) \prod_{k=1}^{K} l_k(x), \tag{14}$$

where π is a density kernel from which sampling is easy, and l_k $(k = 1, \dots, K)$ are non-negative functions of $x \in \mathbb{R}^n$.

Suppose a density kernel of $X \in \mathbb{R}^n$ is given by

$$p_{nn}(x) = \left\{ \begin{array}{ll} nn(x) & \text{if } x_i \in [\underline{x}_i, \bar{x}_i] \;\; \forall i = 1, \dots, n \\ 0 & \text{else} \end{array} \right. \tag{15}$$

where $[\underline{x}_i, \bar{x}_i]$ is the interval to which X_i $(i = 1, \dots, n)$ is restricted. This restriction ensures that (15) is a proper density kernel. The function $nn(x)$ is given by:

$$nn(x) = \exp\left(\sum_{h=1}^{H} c_h \, plin\left(\sum_{i=1}^{n} a_{hi} x_i + b_h \right) + d \right). \tag{16}$$

In Appendix B.2 it is shown that (15) and (16) can be written as (14) with

$$\begin{aligned} \pi(x) &= \prod_{i=1}^{n} I\left\{\underline{x}_i < x_i < \bar{x}_i\right\}, & (17) \\ l_h(x) &= \exp\left(c_h\, plin\left(\sum_{i=1}^{n} a_{hi}x_i + b_h\right)\right) \quad \text{for } h = 1,\ldots,H, & (18) \end{aligned}$$

where $\pi(x)$ is the 'easy' density kernel of n independent variables X_i $(i = 1,\ldots,n)$ with distribution $U(\underline{x}_i, \bar{x}_i)$. This means that we can draw from (15) and (16) using auxiliary variable Gibbs sampling. In the appendix it is shown that in this case auxiliary variable Gibbs sampling only requires sampling from uniform distributions, which is done easily and fast.

2.3 Importance sampling and Metropolis-Hastings

Once we have obtained a sample of random drawings from the neural network density $nn(x)$, we use this sample in order to estimate those characteristics of the target density $p(x)$ that we are interested in. Two methods that we can use for this purpose are importance sampling and the Metropolis-Hastings algorithm.

A discussion of importance sampling can be found in Bauwens et al. (1999). Let X be a random variable with density p. Suppose we are interested in the expectation $E(h(X))$ for a certain function $h : \mathbb{R}^n \to \mathbb{R}$. Then the importance sampling (IS) approach to obtain an estimate of $E(h(X))$ is:

Step 1: Draw a sample of y_i's $(i = 1,\ldots,m)$ from a 'candidate distribution' with density q, the so-called importance function.

Step 2: The estimate of $E(h(X))$ is now given by:

$$h_{IS} = \frac{\sum_{i=1}^{n} w(y_i)h(y_i)}{\sum_{i=1}^{n} w(y_i)}, \tag{19}$$

where $w(x) \equiv p(x)/q(x)$ is the so-called weight function.

The Metropolis-Hastings (MH) algorithm was introduced by Metropolis et al. (1953) and generalized by Hastings (1970). The algorithm samples from a time-reversible Markov chain, converging to the target distribution of the random variable $X \in \mathbb{R}^n$ that we are interested in.

The MH algorithm constructs a Markov chain of m random vectors in the following way:

Initialization: choose feasible vector $x^0 \in \mathbb{R}^n$.

Do for $j = 1, 2, \ldots, m$

Obtain y from a 'candidate' density $q(y|x^{j-1})$, where $q(y|x^{j-1})$ is the 'transition density' that may depend on x^{j-1}.

Compute the 'acceptance probability' $\alpha(x^{j-1}, y)$:

$$\alpha(x^{j-1}, y) \equiv \min\left\{\frac{p(y)q(x^{j-1}|y)}{p(x^{j-1})q(y|x^{j-1})}, 1\right\}$$

Obtain u from the uniform distribution on (0,1).

If $u \leq \alpha(x^{j-1}, y)$ then $x^j = y$ else $x^j = x^{j-1}$.

A realized Markov chain can be used in a number of ways. One way is considering all realizations after a certain burn-in period, and using the sample statistics of these realizations as estimates of the characteristics of the distribution of X that we are interested in.

Note that in the case of a four-layer neural network we need Gibbs sampling in order to obtain the sample, so that the consecutive drawings are not independent. In this case it is not efficient to use the Metropolis-Hastings algorithm. Therefore we have six 'neural network based' algorithms at hand:

- Neural Network Importance Sampling (NNIS) and Neural Network Metropolis-Hastings (NNMH) in which IS or MH is performed using random vectors that are (directly) drawn from a 3-layer neural network;
- Gibbs Neural Network Importance Sampling (GiNNIS) and Gibbs with Auxiliary Variables Neural Network Importance Sampling (GiAuVaNNIS) in which IS is performed using random vectors that are drawn from a 4-layer neural network by Gibbs sampling (possibly with auxiliary variables);
- IS or MH using random vectors that are (directly) drawn from an Adaptive Mixture of t distributions (AdMit-IS or AdMit-MH).

Table 2 gives an overview.

Table 2: Overview of neural network based sampling algorithms

	Importance sampling	Metropolis-Hastings
Type 1 (3-layer) neural network: direct sampling	NNIS	NNMH
Type 2 (4-layer) neural network: (auxiliary variable) Gibbs sampling	Gi(AuVa)NNIS	-
Type 3 neural network (adaptive mixture of t densities): direct sampling	AdMit-IS	AdMit-MH

3 Analytical expressions for moments of the three-layer neural network distribution

There exist analytical expressions for the moments of the 3-layer neural network distribution with the scaled arctangent activation function, just like the expressions for the marginal and conditional distribution functions that make direct sampling possible. The formulas are derived in appendix A.3. This feature of the 3-layer neural network makes the following algorithm possible if one only wants estimates of certain moments of the target distribution:

Step 1: Construct a 3-layer neural network function $nn(x)$ that gives a good approximation to the target density $p(x)$.

Step 2: Compute the moments of the neural network distribution using the formulas in appendix A.3. These moments provide estimates of those moments of the target density $p(x)$ that one is interested in.

In this case no sampling algorithm like MH or IS is needed. As in this case the neural network output is not ‘corrected’ by MH or IS, the neural network has to be a very accurate approximation to the target density. Otherwise its moments are inaccurate approximations. In Hoogerheide, Kaashoek and van Dijk (2003) this method is discussed more elaborately.

4 Example I: Generalizations of the bivariate normal distribution

In order to illustrate the neural network based algorithms in simple examples, we consider two generalizations of the bivariate normal distribution: (1) a mixture of two normal distributions and (2) the conditionally normal distribution of Gelman and Meng (1991).

4.1 A mixture of two normal distributions

Consider the following mixture of normal distributions:

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim 0.5\, N\!\left(\begin{pmatrix} -5 \\ -5 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right) + 0.5\, N\!\left(\begin{pmatrix} 5 \\ 5 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right), \tag{20}$$

We use our algorithms in order to obtain estimates of the mean and standard deviation of X_1 and X_2, and the correlation coefficient $\rho(X_1, X_2)$. For the Type 1 and 2 networks, we restrict the variables X_1 and X_2 to the interval [-10,10], i.e. we only consider the region

$$\{(X_1, X_2) | -10 \leq X_1 \leq 10, -10 \leq X_2 \leq 10\}. \tag{21}$$

This restriction does not affect our estimates, as the probability mass outside this region is negligible.

First we use the approach described in subsection 2.1 in order to construct a Type 1 (3-layer) neural network approximation to the target density. We find a 3-layer neural network with $H = 48$ hidden cells which has a squared correlation $R^2 = 0.985$ with the target density on a set of 2500 points on the region (21), and also $R^2 = 0.985$ on a set of 5000 points.

We also construct a Type 2 (simplified 4-layer) neural network approximation to the target density. We find a 4-layer network with $H = 10$ hidden cells with $R^2 = 0.988$ on the estimation set and $R^2 = 0.984$ on the larger set. Note the large difference between the sizes of the 3-layer network and the 4-layer network. The 3-layer network requires 5 times as many hidden cells. This suggests that the exponential transformation in the 4-layer network makes it much easier to construct an approximation to the target density.

Then we construct a Type 3 (mixture of t distributions) neural network. We choose $\nu = 1$ degree of freedom. We find a mixture of two t distributions with a sample of IS weights in which the 5% most influential points have 11.4% weight.

The contourplots of the neural network approximations are given by Figure 1, together with the contourplot of the target density. These contourplots confirm that the neural networks are good approximations to the target density.

After we have constructed neural network approximations, we sample from these networks and use the samples in IS or MH. For the Type 1 and 2 networks we construct two samples, and we say that convergence has been achieved if the differences between the two estimated means of X_1 and X_2 are both less than 0.05. The results are in Table 3. Note that the six neural network sampling algorithms – NNIS, NNMH, GiNNIS, GiAuVaNNIS, AdMit-IS and AdMit-MH – all yield estimates differing less than 0.05 from the real values. The analytical expressions for the moments of the 3-layer neural network also yield quite good estimates, although not as good as the six neural network sampling algorithms.

NNIS and NNMH require only 50000 drawings, whereas GiNNIS and GiAuVaNNIS require 200000 and 1000000 drawings, respectively. The reason for this is that NNIS and NNMH use a sample of uncorrelated points obtained by direct sampling, whereas GiNNIS and GiAuVaNNIS use Gibbs sequences in which the points are correlated. The first order serial correlations of the Gibbs sequences of X_1's are 0.90 and 0.97 in GiNNIS and GiAuVaNNIS, respectively. Apparently the addition of auxiliary variables increases the serial correlation in the Gibbs sequence, which explains why 1000000 points are needed instead of 200000. In this case AdMit-IS and AdMit-MH only use 25000 independent drawings obtained by direct sampling.

If we look at the computing times (on an AMD Athlon$^{\text{TM}}$ 1.4 GHz processor) required for generating the samples, we conclude that AdMit-IS and AdMit-MH are the winners in this example. In AdMit-IS or AdMit-MH the construction of the network, the sampling, and the IS or MH require altogether 4.5 seconds, whereas the other methods take much more time to generate an adequate sample, and also require time to estimate a network. Among the methods using a Type 1 or 2 network, the GiNNIS algorithm is the fastest. The NNIS and NNMH algorithms are relatively slow, as these methods require a numerical method, such as the bisection method, in order to perform the inverse transformation method.

The total weight of the 5% most influential points is below 15% for the three IS algorithms, confirming the quality of the importance density. The rather high NNMH and AdMit-MH acceptance rates of 67% and 56% indicate the quality of the neural network as a candidate density.

We now compare the performance of the neural network algorithms with the performance of IS and MH with a Student t candidate distribution with 1 degree of freedom, and with the Griddy-Gibbs sampler (see Ritter and Tanner (1992)) using a grid of 10 equidistant points on (21). For the purpose of this example, we ignore the fact that the target distribution is a mixture of normal distributions from which direct sampling and Gibbs sampling are possible. All sampling

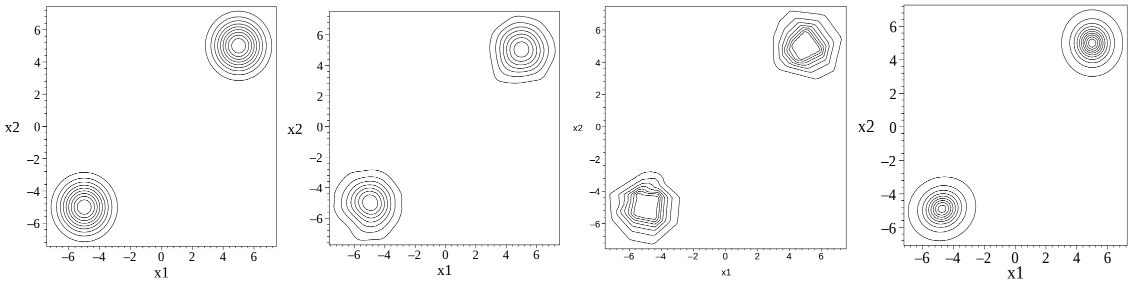

Figure 1: Contourplots: the density of the mixture of two normal distributions in (20) (left), and its Type 1 (second), Type 2 (third), and Type 3 (right) neural network approximation

methods in this example only require that one can evaluate a kernel of the target density.

Again we construct two samples, and we say that convergence has been achieved if the differences between the two estimated means of X_1 and X_2 are both less than 0.05. As the Griddy-Gibbs sequences hardly ever move from one mode to the other, a naive Griddy-Gibbs approach would take extremely much time to converge. Therefore we decide to use a number of Griddy-Gibbs sequences of 100 points (plus 10 burn-in points) with different initial values that are drawn from a uniform distribution on (21): 5000 sequences are needed to reach convergence.

The results are in Table 4. Note the large numbers of drawings required by IS and MH. Although the results of the three methods converge to the real values, more time is needed than in our AdMit-IS and AdMit-MH procedures. We conclude that the neural network approach is feasible, and that AdMit-IS and AdMit-MH are faster than three competing algorithms.

Table 3: Neural network based sampling results for the mixture of two bivariate normal distributions in (20)

	real values	NNIS	NNMH	analytical moments	GiNNIS	GiAuVa NNIS	AdMit IS	AdMit MH
$E(X_1)$	0	0.024	0.011	0.063	0.035	-0.048	-0.024	-0.029
$E(X_2)$	0	0.022	0.012	0.063	0.040	-0.047	-0.015	-0.006
$\sigma(X_1)$	5.099	5.106	5.102	5.088	5.097	5.099	5.084	5.090
$\sigma(X_2)$	5.099	5.103	5.099	5.101	5.104	5.097	5.098	5.104
$\rho(X_1, X_2)$	0.962	0.962	0.962	0.968	0.962	0.962	0.962	0.961
drawings		50000	50000		200000	1000000	25000	25000
time		568 s	568 s		56 s	172 s	4.5 s *	4.5 s *
time/draw		11 ms	11 ms		0.28 ms	0.17 ms	0.18 ms	0.18 ms
5% weights		8.0%			7.4%	7.4%	11.4%	
acc. rate			67%					56%

* The computing times for the AdMit methods concern the whole procedure *including* the construction of the neural networks.

Table 4: Alternative sampling results for the mixture of bivariate normal distributions in (20)

	real values	t_1 IS	t_1 MH	Griddy Gibbs
$E(X_1)$	0	0.033	-0.048	0.017
$E(X_2)$	0	0.032	-0.019	0.021
$\sigma(X_1)$	5.099	5.099	5.097	5.117
$\sigma(X_2)$	5.099	5.101	5.072	5.118
$\rho(X_1, X_2)$	0.962	0.961	0.962	0.954
drawings		1500000	2500000	5000 × 100
time		27.5 s	47.5 s	81.4 s
time/draw		0.02 ms	0.02 ms	0.16 ms
5% weights		54.1%		
acc. rate			34 %	

4.2 A conditionally normal distribution

It may seem a little unfair to use a *mixture of normal distributions* in order to test a method in which the target density is approximated with a *mixture of t densities.* Therefore we also apply our AdMit-IS and AdMit-MH methods to another distribution, a bivariate conditionally normal distribution.

Let X_1 and X_2 be two jointly distributed random variables, for which X_1 is normally distributed given X_2 and vice versa. Then the joint distribution, after location and scale transformations in each variable, can be written as (see Gelman and Meng (1991)):

$$p(x_1, x_2) \propto \exp\left(-\frac{1}{2}\left[Ax_1^2x_2^2 + x_1^2 + x_2^2 - 2Bx_1x_2 - 2C_1x_1 - 2C_2x_2\right]\right) \tag{22}$$

We consider the symmetric case in which $A = 1$, $B = 0$, $C_1 = C_2 = 10$, with conditional distributions

$$X_1|X_2 = x_2 \sim N\left(\frac{10}{1+x_2^2}, \frac{1}{1+x_2^2}\right) \qquad X_2|X_1 = x_1 \sim N\left(\frac{10}{1+x_1^2}, \frac{1}{1+x_1^2}\right).$$

We find a mixture of four t distributions with a sample of IS weights in which the 5% most influential points have 11.5% weight. The contourplots of the neural network approximations are given by Figure 2. The results are in Table 5. We compare the performance of the AdMit algorithms with the performance of IS and MH with a t_1 candidate distribution, and with the Griddy-Gibbs sampler using a grid of 100 equidistant points on $[-5, 15] \times [-5, 15]$. We construct two samples, and we say that convergence has been achieved if the differences between the two estimated means of X_1 and X_2 are both less than 0.10. For the purpose of this example, we ignore the fact that Gibbs sampling from this target distribution is possible; it should be remarked that one would need more than one sequence to avoid extremely slow convergence, as the Gibbs sequences that we generated remained in one mode for 100000000 drawings. So, again all sampling methods only require that one can evaluate a kernel of the target density. The results are in Table 5. Notice the huge numbers of drawings required by IS and MH. Again all results have converged to the real values, but the AdMit-IS and AdMit-MH methods are faster than the three alternative sampling methods.

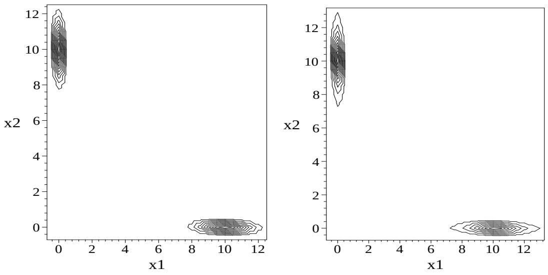

Figure 2: Contourplots: the density of the conditionally normal distribution in (22) with $A = 1$, $B = 0$, $C_1 = C_2 = 10$ (left), and its Type 3 neural network approximation (right)

5 Example II: Bayesian analysis of models with reduced rank

In this section we consider posterior densities for simulated data in two models with reduced rank: an instrumental variable (IV) regression and a vector error correction model (VECM) with cointegration.

5.1 Bayesian analysis of an IV regression

First, we give two examples of well-known IV regressions. Consider the stylized wage regression popular in empirical labor studies:

$$y_1 = \beta y_2 + x_1\gamma + u_1, \tag{23}$$

Table 5: Sampling results for the conditionally normal distribution

	real values	AdMit IS	AdMit MH	t_1 IS	t_1 MH	Griddy Gibbs
$E(X_1)$	4.946	4.907	4.889	4.963	4.865	4.961
$E(X_2)$	4.946	4.986	5.001	4.913	5.022	4.912
$\sigma(X_1)$	4.894	4.893	4.891	4.881	4.896	4.883
$\sigma(X_2)$	4.894	4.894	4.893	4.900	4.902	4.883
$\rho(X_1, X_2)$	-0.979	-0.979	-0.979	-0.979	-0.979	-0.978
drawings		100000	100000	80000000	80000000	10000 × 100
time		237 s *	237 s *	1652 s	1652 s	1415 s
time/draw		2.4 ms	2.4 ms	0.02 ms	0.02 ms	1.4 ms
5% weights		11.5%		59.4 %		
acc. rate			59 %		28 %	

* The computing times for the AdMit methods concern the whole procedure *including* the construction of the neural network.

where y_1 is the log of hourly wage, y_2 denotes education and x_1 captures work experience – all in deviations from their mean values. The structural parameter of interest is β, the rate of return to schooling. However, in order to make inference on β, one should take into account that y_2 is possibly endogenous: y_2 and u_1 may be highly correlated owing to the omission of a variable measuring (unobservable) ability, which is expected to be highly correlated with education. The problem is that potential instruments for y_2 are hard to find as these variables must be correlated with education but uncorrelated with unobserved ability. Angrist and Krueger (1991) suggest using quarter of birth as a dummy variable, as this seems uncorrelated with ability and affects years of schooling weakly, through a combination of the age at which a person begins school and the compulsory education laws in a person's state. Staiger and Stock (1997) show that inference on the rate of return to schooling can be greatly affected by the weak quarter of birth instruments.

As another example, consider the problem of determining the fraction of temporary income consumers spend in a permanent-income-consumption model. Campbell and Mankiw (1989) use the simple regression equation

$$\Delta c = \beta \Delta y + u_1, \tag{24}$$

where c is log consumption and y is log income; β measures the fraction of temporary income consumed. As consumption and income are simultaneously determined, Δy is possibly highly correlated with u_1. In the permanent-income model c and y are cointegrated with cointegrating vector (1,-1) and the error correction model for Δy suggests using lagged values of Δy and Δc and the lagged error correction term, $c - y$, as instruments. However, Δy is poorly predicted from this error correction model, so that the suggested instruments are probably fairly weak. Note that in this example the quality of the instruments is determined by the short-run dynamics in the growth rate of income.

In our example, we consider the following equation

$$y_{1t} = y_{2t}\beta + u_{1t} \quad (t = 1, \ldots, T) \tag{25}$$

where y_{2t} is a possibly endogenous regressor for which we have

$$y_{2t} = x_t \pi + v_{2t} \quad (t = 1, \ldots, T) \tag{26}$$

with $(u_{1t}, v_{2t}) \sim N(0, \Sigma)$ and where x_t is exogenous. We assume a diffuse prior for the parameters β, π and Σ:

$$p(\beta, \pi, \Sigma) \propto |\Sigma|^{-h/2}, \;\; h > 0 \tag{27}$$

The likelihood function for a sample of size T is

$$L(\beta, \pi, \Sigma | y_1, y_2, x) \propto |\Sigma|^{-T/2} \exp\left[-\frac{1}{2}\mathrm{tr}(\Sigma^{-1}U'U)\right], \tag{28}$$

where $U = (\tilde{u}_1(\beta), \tilde{v}_2(\pi))$ with $\tilde{u}_1(\beta) = y_1 - y_2\beta$ and $\tilde{v}_2(\pi) = y_2 - x\pi$. So, the joint posterior based on the diffuse prior is

$$p(\beta, \pi, \Sigma | y_1, y_2, x) \propto |\Sigma|^{-(T+h)/2} \exp\left[-\frac{1}{2}\mathrm{tr}(\Sigma^{-1}U'U)\right]. \tag{29}$$

Using properties of the inverted Wishart distribution (see Zellner (1971) and Bauwens and Van Dijk (1989)), Σ^{-1} can be analytically integrated out of the joint posterior yielding the following joint posterior for (β, π):

$$p(\beta, \pi | y_1, y_2, x) \propto |U'U|^{-(T+h-3)/2}. \tag{30}$$

Choosing $h = 3$ results in the following posterior density

$$p(\beta, \pi | y_1, y_2, x) \propto |U'U|^{-T/2}, \tag{31}$$

which equals the concentrated likelihood function of β and π. In this example we are interested in the (posterior) distribution of the vector (β, π). So, the parameter vector (β, π) plays the role of the random vector X in the previous sections, and $p(\beta, \pi | y_1, y_2, x)$ plays the role of $p(x)$.

Now we simulate $T = 20$ data from the model in (25) and (26) with $\beta = 0$, $\pi = 0.1$, $x_t \sim N(0, 1)$ i.i.d. and

$$\begin{pmatrix} u_{1t} \\ v_{2t} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.99 \\ 0.99 & 1 \end{pmatrix}\right) \quad (t = 1, \ldots, T)$$

Note the extremely high correlation $\rho(u_{1t}, v_{2t}) = 0.99$, causing a very strong endogeneity of the regressor y_2 in equation (25). Also note the low value of $\pi = 0.1$, so that x is a weak instrument for y_2. That is, there is 'weak identification'. Notice that in this IV regression, β is not identified if $\pi = 0$ (unless $\mathrm{cov}(u_{1t}, v_{2t})$ is known). We restrict β to the interval $[-5, 5]$ and π to the interval $[-0.25, 0.25]$; we have to make such a restriction, as otherwise the posterior density would be improper, i.e. its integral would not be finite. Figure 3 shows the contourplot of the posterior density in (31) for our simulated data set. In this case of weak identification and strong endogeneity the contours of the posterior are non-elliptical: we see a bimodal density with two curved ridges. If π is close to zero, large (positive and negative) values of β may occur, which reflects that β is not identified if $\pi = 0$. Van Dijk (2003) shows several cases of simulated IV regressions in which such non-elliptical contours occur.

We use our neural network algorithms to obtain estimates of the posterior means and standard deviations. We find a Type 1 (3-layer) network with $H = 43$ hidden cells with $R^2 = 0.931$ on an estimation set of 2500 points (and $R^2 = 0.930$ on a set of 5000 points), and a Type 2 (simplified 4-layer) network with $H = 10$ hidden cells with $R^2 = 0.933$ on the estimation set (and $R^2 = 0.927$ on the larger set). Note that the 4-layer network is much smaller than the 3-layer network, just like in the first example. We also find a Type 3 (mixture of t distributions) network with $H = 9$ components in which the weight of the 5% most influential points is 23%. The contourplots are given by Figure 3.

After we have constructed neural network approximations, we sample from these networks and use the samples in IS or MH. For the Type 1 and 2 networks we construct two samples, and

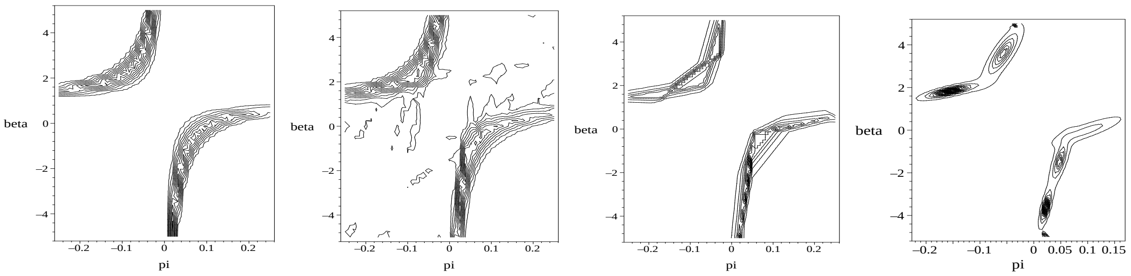

Figure 3: Contourplots: the posterior density in an IV regression in (31) for a simulated data set (left), and its Type 1 (second), Type 2 (third) and Type 3 (right) neural network approximation

Table 6: Neural network based sampling results for the Bayesian IV regression

		NNIS	NNMH	analytical moments	GiNNIS	AdMit IS	AdMit MH
π	mean	-0.01	-0.01	-0.01	-0.01	-0.01	-0.01
	s.d.	0.10	0.10	0.12	0.11	0.11	0.10
β	mean	0.65	0.65	0.49	0.67	0.64	0.69
	s.d.	2.37	2.38	2.49	2.34	2.35	2.39
drawings		25000	25000		100000	10000	10000
time		261 s	261 s		28 s	132.6 s *	132.6 s *
time/draw		10 ms	10 ms		0.28 ms	13 ms	13 ms
5% weights		9%			10 %	23%	
acc. rate			59%				39 %

* The computing times for the AdMit methods concern the whole procedure *including* the construction of the neural network.

Table 7: Maximum likelihood estimates in an IV regression

Parameter:	π	β
MLE:	-0.05	3.36
(std. error)	(0.23)	(11.09)

Table 8: Alternative sampling results for the Bayesian IV regression

		normal candidate		t_1 candidate		Griddy
		IS	MH	IS	MH	Gibbs
π	mean	-0.07	-0.07	-0.01	-0.01	-0.01
	s.d.	0.03	0.03	0.10	0.10	0.10
β	mean	2.96	2.97	0.65	0.65	0.62
	s.d.	0.95	0.94	2.36	2.38	2.37
drawings		100000000	100000000	25000000	60000000	200000
time		2312 s	2312 s	517 s	1288 s	283 s
time/draw		0.02 ms	0.02 ms	0.02 ms	0.02 ms	1.4 ms
5% weights		69%		92%		
acc. rate			6.8%		1.9%	

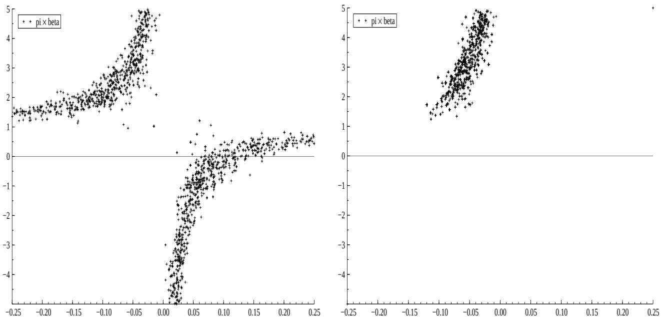

Figure 4: Scatter plots: samples of points (π, β) from the posterior density (31) in the IV regression, obtained by NNMH (left) and MH with a normal candidate distribution (right)

we say that convergence has been achieved if the differences between the two estimated posterior means of β and π are less than 0.05 and 0.005, respectively. The sampling results are in Table 6.

We now compare the performance of the neural network algorithms with the performance of IS and MH with the maximum likelihood estimator's asymptotic distribution as the candidate density. Recall that the asymptotic distribution of $\hat{\theta}_{ML}$ can be approximated with:

$$N\left(\hat{\theta}_{ML}, \hat{I}\left(\hat{\theta}_{ML}\right)^{-1}\right), \quad \hat{I}\left(\hat{\theta}_{ML}\right) = -\left.\frac{\delta^2 \log L(\theta)}{\delta\theta\delta\theta'}\right|_{\theta=\hat{\theta}_{ML}} \tag{32}$$

The maximum likelihood estimates are given by Table 7. The estimated correlation is very high: $\hat{\rho}(\hat{\pi}_{ML}, \hat{\beta}_{ML}) = 0.9985$. The sampling results are in Table 8. Note the large differences between the neural network algorithms and IS or MH with the normal candidate – not even after 100 million drawings the results of the normal candidate have converged to the same values. The scatter plots in Figure 4 of the NNMH and MH samples reveal the reason for these differences: NNMH yields a fine sample showing the contours of the joint posterior density, whereas MH with the normal candidate density completely misses one of the two modes. Table 8 also shows the results of IS and MH with a Student t distribution with 1 degree of freedom around the maximum likelihood estimator, and with Griddy-Gibbs (with 100 grid points). These methods yield approximately the same estimates as the neural network based methods; however, they take more time than the AdMit approach. Note that IS and MH with a Student t candidate density require huge amounts of drawings to reach convergence.

We conclude that the neural network sampling algorithms seem to work well in this example (yielding approximately the same estimates). The analytical expressions for the moments of the 3-layer neural network also yield quite good estimates, although not as good as the other neural network methods. In this example IS or MH with the maximum likelihood estimator's asymptotic distribution as the candidate distribution does not yield reliable estimates. AdMit-IS and AdMit-MH are the fastest among these algorithms.

5.2 Bayesian analysis of a VECM with cointegration

Many economic time series seem to follow random walk processes. A random walk implies that one cannot forecast any future changes of the series. When random walks are independent then the different series will tend to move in different directions. However, in practice it is observed that there exist stationary relationships between series that individually behave as random walks. Some examples are: short and long term interest rates, prices and dividends of stocks, consumption and income. This stylized fact is referred to as cointegration.

First, we mention some well-known cointegration studies that are based on economic theory. Campbell and Shiller (1987) study bubbles in asset prices. Campbell (1987) tests the hypothesis that consumption is determined by permanent income. King, Plosser, Stock and Watson (1991) consider the role of productivity shocks in the postwar US economy. Hall, Anderson and Granger (1992) analyze the term structure of interest rates.

Consider the following two-dimensional VAR(1) model

$$y_t = Ay_{t-1} + \varepsilon_t \qquad (t = 1, \ldots, T), \tag{33}$$

where $y_t = (y_{1t}, y_{2t})'$ and $\varepsilon_t = (\varepsilon_{1t}, \varepsilon_{2t})' \sim N(0, \Sigma)$. Define $\Pi = A - I$, then one can rewrite (33) in error correction form as:

$$\Delta y_t = \Pi y_{t-1} + \varepsilon_t \qquad (t = 1, \ldots, T). \tag{34}$$

Suppose the matrix Π is singular, i.e. $|\Pi| = 0$, and that it has rank one. This implies that the characteristic equation $|I - zA| = 0$ has one unit root and one root z outside the unit circle. The consequence of one unit root and one root outside the unit circle can be shown as follows. Given that the rank of Π is one, one can write $\Pi = \alpha\beta'$ where α and β are 2×1 vectors. Thus we have

$$\Delta y_t = \alpha\beta' y_{t-1} + \varepsilon_t \qquad (t = 1, \ldots, T). \tag{35}$$

It follows directly that the linear combination $\beta' y_t$ is an AR(1) process with

$$\beta' y_t = (1 + \beta'\alpha)\beta' y_{t-1} + \beta'\varepsilon_t \qquad (t = 1, \ldots, T), \tag{36}$$

which is stationary if $|1 + \beta'\alpha| < 1$. It can be shown that this inequality is satisfied if the characteristic equation $|I - zA| = 0$ has one unit root and one root z outside the unit circle. Thus the vector y_t is cointegrated in the sense that a linear combination is stationary. The vector β is defined as the cointegration vector. It is seen from (35) and (36) that the cointegrating vector is defined up to a scale constant, that is, if $\beta' y_t$ is stationary then $\lambda\beta' y_t$ is stationary for any $\lambda \in \mathbb{R}$. A usual normalization is $\beta' = (1, -\beta_2)$. The coefficients α_1 and α_2 are defined as adjustment parameters. Then we can rewrite (35) as:

$$\begin{pmatrix} \Delta y_{1t} \\ \Delta y_{2t} \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} (y_{1,t-1} - \beta_2 y_{2,t-1}) + \begin{pmatrix} \varepsilon_{1t} \\ \varepsilon_{2t} \end{pmatrix} \qquad (t = 1, \ldots, T). \tag{37}$$

In our example we analyze the posterior of the parameters in (37) for a simulated data set. We assume a diffuse prior for the parameters α_1, α_2, β_2 and Σ:

$$p(\alpha_1, \alpha_2, \beta_2, \Sigma) \propto |\Sigma|^{-h/2}, \; h > 0. \tag{38}$$

The likelihood function for a sample of size T is

$$L(\alpha_1, \alpha_2, \beta_2, \Sigma | y_1, y_2) \propto |\Sigma|^{-T/2} \exp\left[-\frac{1}{2}\text{tr}(\Sigma^{-1}E'E)\right], \tag{39}$$

where $E = [\tilde{\varepsilon}_1(\alpha_1, \beta_2), \tilde{\varepsilon}_2(\alpha_2, \beta_2)]$ with $\tilde{\varepsilon}_1(\alpha_1, \beta_2) = \Delta y_1 - \alpha_1(y_1 - \beta_2 y_2)$ and $\tilde{\varepsilon}_2(\alpha_2, \beta_2) = \Delta y_2 - \alpha_2(y_1 - \beta_2 y_2)$. So, the joint posterior based on the diffuse prior is

$$p(\alpha_1, \alpha_2, \beta_2, \Sigma | y_1, y_2) \propto |\Sigma|^{-(T+h)/2} \exp\left[-\frac{1}{2}\text{tr}(\Sigma^{-1}E'E)\right]. \tag{40}$$

Again, Σ^{-1} can be analytically integrated out of the joint posterior yielding the following joint posterior for $(\alpha_1, \alpha_2, \beta_2)$:

$$p(\alpha_1, \alpha_2, \beta_2 | y_1, y_2) \propto |E'E|^{-(T+h-3)/2}. \tag{41}$$

Choosing $h = 3$ results in the following posterior density

$$p(\alpha_1, \alpha_2, \beta_2 | y_1, y_2) \propto |E'E|^{-T/2}, \tag{42}$$

which equals the concentrated likelihood function of α_1, α_2 and β_2. In order to make (42) a proper density kernel, we restrict α_1,α_2 to [-0.2,0.2] and β_2 to [-10,10].

Now we simulate $T = 100$ data from the model in (37) with $\alpha_1 = -0.05$, $\alpha_2 = 0.05$, $\beta_2 = 1$ and ε_{1t}, $\varepsilon_{2t} \sim N(0, 1)$ i.i.d. (and mutually independent). Notice that in this cointegration model β_2 is not identified if $\alpha_1 = \alpha_2 = 0$.

We use AdMit-IS and AdMit-MH to obtain estimates of the posterior means and standard deviations. We find a Type 3 (mixture of t distributions) network with $H = 4$ components in which the weight of the 5% most influential points is 14.8%. The contourplots of the marginal candidate densities of (α_1, α_2), (α_1, β_2) and (α_2, β_2) are given by Figure 5. Figure 6 shows scatter plots of points obtained by AdMit-MH.

We compare the performance of AdMit-IS and AdMit-MH with IS and MH with a Student t distribution with 1 degree of freedom around the maximum likelihood estimator, and with Griddy-Gibbs (with 100 grid points). We construct two samples, and we say that convergence has been achieved if the differences between the three estimated posterior means of α_1, α_2 and β_2 are less than 0.005, 0.005 and 0.05, respectively. Table 9 shows the results. Note that IS and MH with Student t candidate density again require very large amounts of drawings to reach convergence. We conclude that AdMit-IS and AdMit-MH are much faster than the alternative algorithms, while all methods yield approximately the same estimates.

It can be seen from the scatter plots in Figure 6 that if both α_1 and α_2 are close to zero, then a whole spectrum of values of β_2 may occur. This reflects the fact that β_2 is not identified if $\alpha_1 = \alpha_2 = 0$. Also notice the similarity in the mathematical structure of the IV regression and the VAR model with cointegration; in both models the same kind of identification issue may lead to the same sort of non-elliptical contours of the likelihood and posterior density. See also Hoogerheide and Van Dijk (2001) on the similarity of the Anderson-Rubin overidentification test and the Johansen test for cointegration.

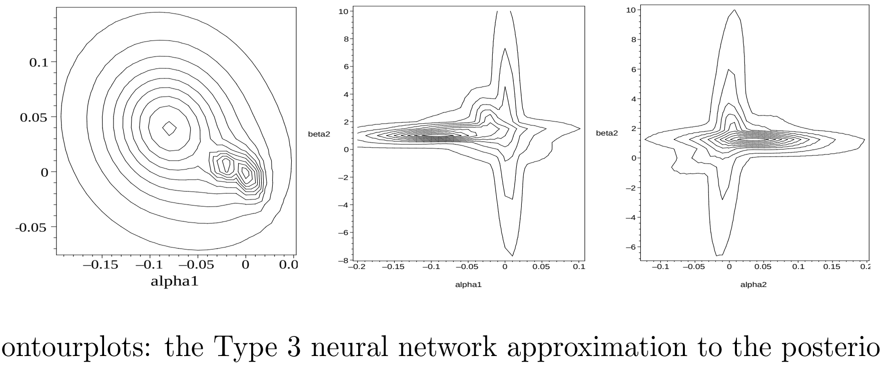

Figure 5: Contourplots: the Type 3 neural network approximation to the posterior density (42) in the VECM in the $\alpha_1 \times \alpha_2$ plane (left), the $\alpha_1 \times \beta_2$ plane (middle), and the $\alpha_2 \times \beta_2$ plane (right).

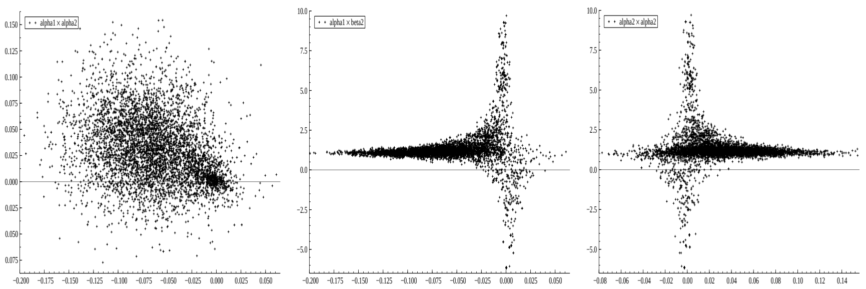

Figure 6: Scatter plots: sample of points $(\alpha_1, \alpha_2, \beta_2)$ from the posterior density (42) in the VECM, obtained by AdMit-MH and displayed in the $\alpha_1 \times \alpha_2$ plane (left), the $\alpha_1 \times \beta_2$ plane (middle), and the $\alpha_2 \times \beta_2$ plane (right)

Table 9: Sampling results for the Bayesian analysis of a VECM

		AdMit IS	AdMit MH	t_1 IS	t_1 MH	Griddy Gibbs
α_1	mean	-0.07	-0.07	-0.07	-0.07	-0.07
	s.d.	0.04	0.04	0.04	0.04	0.04
α_2	mean	0.03	0.03	0.03	0.03	0.03
	s.d.	0.03	0.03	0.03	0.03	0.03
β_2	mean	1.20	1.20	1.21	1.19	1.19
	s.d.	0.64	0.67	0.67	0.64	0.69
drawings		50000	50000	15000000	25000000	50000
time		93 s	93 s	660 s	1116 s	364 s
time/draw		1.9 ms	1.9 ms	0.04 ms	0.04 ms	7.2 ms
5% weights		14.8 %		26.1%		
acc. rate			64%		41%	

* The computing times for the AdMit methods concern the whole procedure *including* the construction of the neural network.

6 Example III: Bayesian analysis of a switching model for the quarterly growth rate of the real US GNP

In models for the growth rate of the gross national product one often allows for separate regimes for periods of recession and expansion. One problem that Bayesian analyses of such models may suffer from is the non-convergence of conventional sampling methods. The reason for this is the possible multi-modality of the posterior distribution. We consider the most simple model, a static 2-regime mixture model. In this model the growth rate y_t has two different mean levels:

$$y_t = \begin{cases} \beta_1 + \varepsilon_t & \text{with probability } p \\ \beta_2 + \varepsilon_t & \text{with probability } 1-p \end{cases}, \tag{43}$$

where $\varepsilon_t \sim N(0, \sigma^2)$. For identification we assume that $\beta_1 < \beta_2$, so that β_1 and β_2 can be interpreted as the mean growth rates during recessions and expansions, respectively. The prior densities of the parameters β_1 and β_2 are taken uniform on the set of values for which $\beta_1 < \beta_2$, and zero elsewhere. The prior on p is taken uniform on the interval $[0, 1]$, while for σ the uninformative prior $\pi(\sigma) \propto 1/\sigma$ is used.

The underlying data we consider are the quarterly growth rates of the real US GNP in the period 1959-2001. The data are shown in Figure 7. The maximum likelihood estimates of the parameters are given by Table 10.

We use the neural network algorithms in order to obtain estimates of the posterior mean and standard deviation of β_1, β_2, σ and p. Looking at the graph of the quarterly growth rate and the maximum likelihood estimates, we choose to restrict the parameters to the following intervals: $\beta_1 \in [-3, 1]$, $\beta_2 \in [0.5, 2]$, $\sigma \in [0.5, 1]$ and $p \in [0, 1]$.

We construct a Type 2 (simplified 4-layer) neural network approximation to the target density; we find a network with $H = 15$ hidden cells with an $R^2 = 0.87$ on an estimation set of 250000 points and $R^2 = 0.86$ on a set of 500000 points. We also find a Type 3 (mixture of t distributions) network with $H = 5$ components in which the weight of the 5% most influential points is 35%.

After we have constructed neural network approximations, we sample from these networks and use the samples in IS or MH. First we compare the results with IS and MH with a normal or t_1 distribution around the maximum likelihood estimator, and with Griddy-Gibbs (with 50 grid points). These sampling methods only require that one can evaluate a kernel of the target density. We construct two samples, and we say that convergence has been achieved if the differences between the four estimated posterior means are less than 0.05. Table 11 and 12 show the results. Even after 25 million drawings IS and MH with the normal candidate distribution yield completely different results than the other algorithms. The other methods yield approximately the same results, where AdMit-IS and AdMit-MH are the fastest.

Estimates of the marginal posterior densities obtained by GiNNIS, GiAuVaNNIS, AdMit-MH and IS with a normal candidate distribution are given by Figures 8, 9, 10 and 11, respectively.

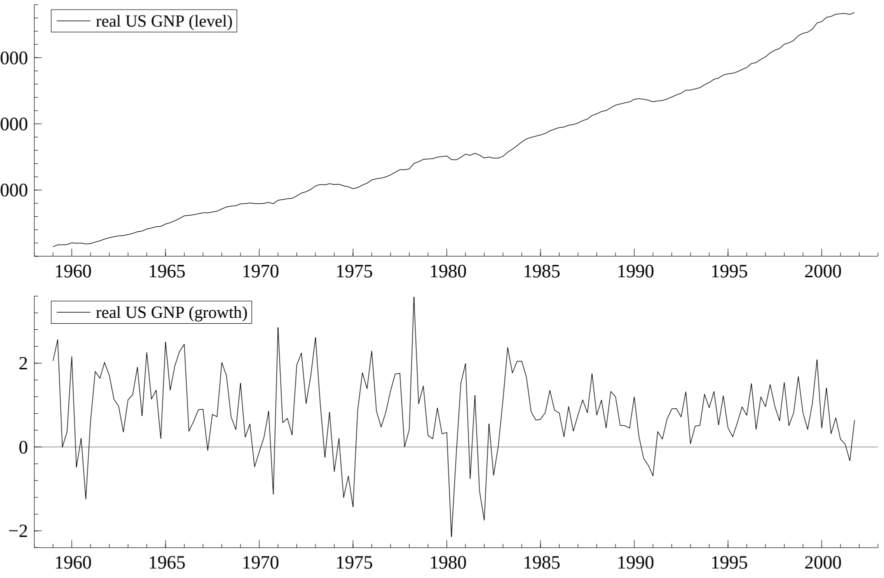

Figure 7: Real GNP of the USA in billions of dollars (above), and its quarterly growth rate in % (below).

Table 10: Maximum likelihood estimates in the 2-regime mixture model (43)

Parameter:	β_1	β_2	σ	p
MLE:	-1.01	0.93	0.79	0.05
(std. error)	(0.51)	(0.08)	(0.06)	(0.04)

Note the large differences between IS with a normal candidate distribution and the other methods, especially in the marginal densities of β_1 and p. The IS estimates indicate a much smaller posterior probability that $\beta_1 \approx 0.8$, and almost zero probability that p exceeds 0.25. This explains why the estimated posterior means of β_1 and p are much smaller according to IS with a normal candidate distribution.

In this model we can also perform Gibbs sampling from the posterior distribution, if we use the method of data augmentation of Tanner and Wong (1987). Data augmentation is used in order to sample from models with latent variables Z, in which sampling the parameters θ seems very difficult, but sampling θ given Z is straightforward. In this algorithm, the parameters θ are drawn conditionally on the latent variables Z, and the latent variables Z are drawn conditionally on θ. Forgetting the values of Z, this procedure yields a valid Markov chain for the parameters θ. In our model we define the latent variables Z_t $(t = 1, \ldots, T)$ as:

$$Z_t = \begin{cases} 1 & \text{if period } t \text{ is a recession period} \\ 0 & \text{if period } t \text{ is an expansion period} \end{cases} . \tag{44}$$

Conditionally on these latent variables Z (and each other), β_1 and β_2 are normally distributed, while σ^2 and p have an inverted gamma and a beta distribution, respectively. Conditionally on the values of the parameters, the latent variables Z_t $(t = 1, \ldots, T)$ have a Bernoulli distribution. The results are in Table 12. Data augmentation estimates of the marginal posterior densities are given by Figure 12. Note that Gibbs sampling with data augmentation requires much more

Table 11: Neural network based sampling results for the 2-regime mixture model (43)

	GiNNIS		GiAuVaNNIS		Admit IS		Admit MH	
	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
β_1	-0.24	0.84	-0.26	0.83	-0.22	0.84	-0.22	0.85
β_2	0.99	0.17	0.98	0.15	1.00	0.19	0.99	0.18
σ	0.84	0.07	0.83	0.07	0.84	0.06	0.84	0.07
p	0.24	0.27	0.22	0.25	0.26	0.29	0.26	0.28
drawings	400000		800000		10000		10000	
time	269 s		421 s		40 s *		40 s *	
time/draw	0.67 ms		0.53 ms		4.0 ms		4.0 ms	
5% weights	31%		32%		34%			
acc. rate							31 %	

* The computing times for the AdMit methods concern the whole procedure *including* the construction of the neural network.

Table 12: Alternative sampling results for the 2-regime mixture model (43)

	normal candidate IS		normal candidate MH		t_1 candidate IS		t_1 candidate MH		Griddy Gibbs		Data Augmentation	
	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
β_1	-0.72	0.67	-0.73	0.67	-0.25	0.84	-0.21	0.85	-0.17	0.84	-0.21	0.84
β_2	0.93	0.08	0.93	0.08	0.99	0.17	0.98	0.15	1.01	0.20	1.00	0.20
σ	0.82	0.06	0.82	0.06	0.83	0.06	0.84	0.07	0.84	0.06	0.84	0.07
p	0.07	0.05	0.07	0.05	0.25	0.28	0.27	0.30	0.27	0.29	0.26	0.29
drawings	25000000		25000000		10000000		15000000		10000		400000	
time	2292 s		2292 s		873 s		1321 s		131 s		42 s	
time/draw	0.09 ms		0.09 ms		0.09 ms		0.09 ms		13 ms		0.11 ms	
5% weights	40%				61%							
acc. rate			45%				18%					

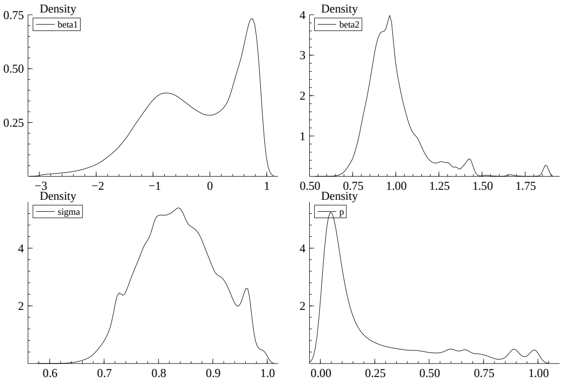

Figure 8: GiNNIS estimates of the marginal posterior densities in the 2-regime mixture model (43)

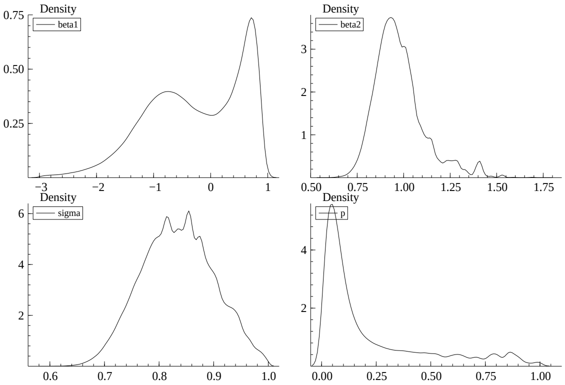

Figure 9: GiAuVaNNIS estimates of the marginal posterior densities in the 2-regime mixture model (43)

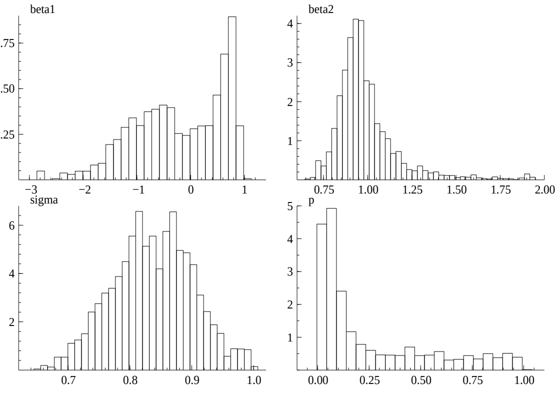

Figure 10: AdMit-MH estimates of the marginal posterior densities in the 2-regime mixture model (43)

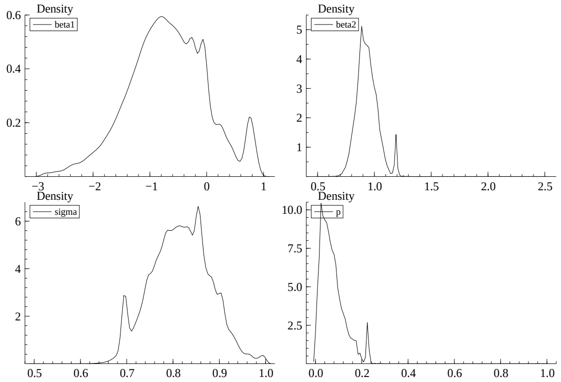

Figure 11: IS estimates (with a normal candidate distribution) of the marginal posterior densities in the 2-regime mixture model (43)

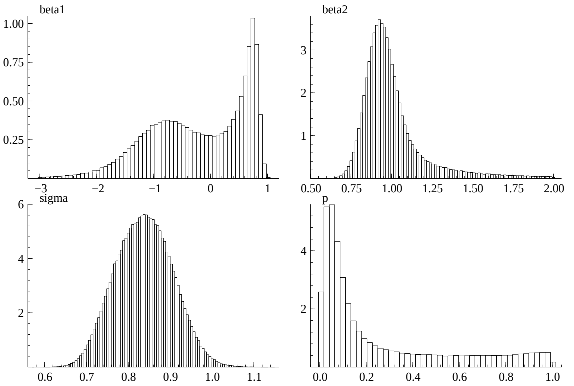

Figure 12: Data augmentation estimates of the marginal posterior densities in the 2-regime mixture model (43)

drawings than the Griddy-Gibbs sampler. The reason for this is that the addition of latent variables increases the serial correlation in the Gibbs sequence: the serial correlations in the Gibbs sequences of p's are 0.992 and 0.854 in data augmentation and Griddy-Gibbs, respectively. We conclude that data augmentation yields about the same results as our neural network methods, while the AdMit procedures are even slightly faster than data augmentation, which requires more knowledge of the model – AdMit only requires that one can evaluate a kernel of the target density.

Finally, consider another 2-regime mixture model for the growth rates y_t of the real US GNP:

$$y_t = \begin{cases} \beta_{11} + \beta_{12} y_{t-1} + \varepsilon_t & \text{with probability } p \\ \beta_{21} + \beta_{22} y_{t-1} + \varepsilon_t & \text{with probability } 1-p \end{cases}, \tag{45}$$

where $\varepsilon_t \sim N(0, \sigma^2)$. For identification we assume that $\beta_{11} < \beta_{21}$. The prior densities of the parameters β_{11}, β_{12}, β_{21} and β_{22} are taken uniform on the set of values for which $\beta_{11} < \beta_{21}$, and zero elsewhere. The prior on p is taken uniform on the interval $[0, 1]$, while for σ the uninformative prior $\pi(\sigma) \propto 1/\sigma$ is used.

In this case we find a Type 3 (mixture of t distributions) network with $H = 7$ components in which the weight of the 5% most influential points is 68.7%. We compare the results with IS and MH with a t_1 candidate distribution. If we perform IS and MH with a t_1 distribution around the maximum likelihood estimator, where we require the estimated posterior means for two samples to differ no more than 0.10, then convergence has not been reached after 25000000 drawings. However, if we draw 2000000 points from the t_1 distribution around the maximum likelihood estimator, and iteratively update the mean and covariance matrix, then we reach convergence after 10 iterations. Table 13 shows the results. The methods yield approximately the same results, where the AdMit procedures require less drawings and time.

Estimates of the marginal posterior densities are given by Figures 13 and 14. Figure 15 shows scatter plots of points obtained by AdMit-MH. Note the bimodality in the marginal distribution of p and the non-elliptical contours in the scatter plots; this causes the slow convergence of IS and MH with a t_1 candidate distribution.

Table 13: Sampling results for the 2-regime mixture model in (45)

	Admit IS		Admit MH		IS		MH	
	mean	s.d.	mean	s.d.	mean	s.d.	mean	s.d.
β_{11}	0.15	0.56	0.17	0.54	0.06	0.71	0.09	0.62
β_{12}	0.44	0.20	0.45	0.20	0.40	0.27	0.43	0.25
β_{21}	1.35	0.76	1.32	0.75	1.30	0.79	1.29	0.75
β_{22}	-0.05	0.41	-0.05	0.40	-0.03	0.40	-0.04	0.38
σ	0.82	0.06	0.82	0.06	0.82	0.06	0.82	0.06
p	0.59	0.36	0.60	0.35	0.55	0.38	0.56	0.37
drawings	25000		25000		10 × 2000000		10 × 2000000	
time	314 s *		314 s *		1320 s		1320 s	
time/draw	13 ms		13 ms		0.07 ms		0.07 ms	
5% weights	68.7 %				98.2 %			
acc. rate			8.2 %				1.5 %	

* The computing times for the AdMit methods concern the whole procedure *including* the construction of the neural network.

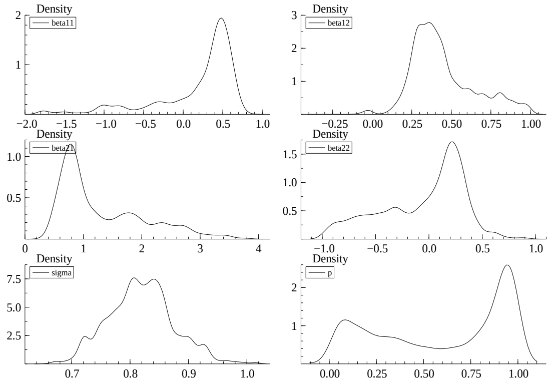

Figure 13: AdMit-MH estimates of the marginal posterior densities in the 2-regime mixture model in (45)

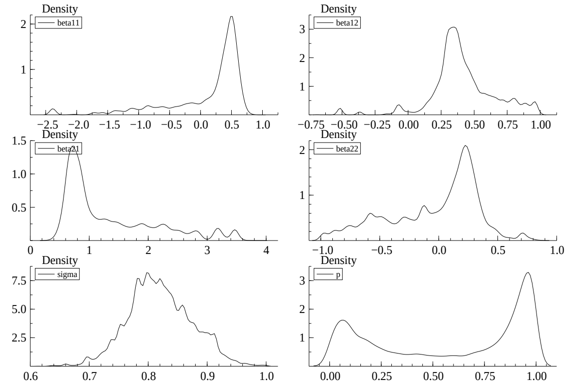

Figure 14: MH estimates of the marginal posterior densities in the 2-regime mixture model (45)

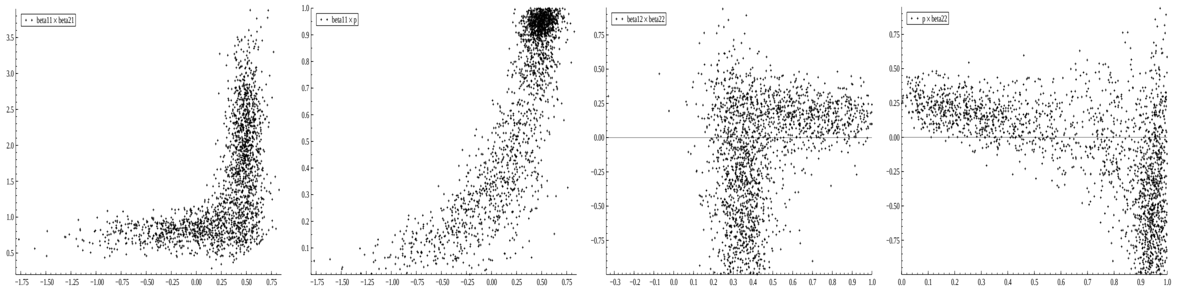

Figure 15: Scatter plots: points $(\beta_{11}, \beta_{12}, \beta_{21}, \beta_{22}, \sigma, p)$ from the posterior distribution in the mixture model (45), obtained by AdMit-MH and displayed in the $\beta_{11} \times \beta_{12}$ plane (left), the $\beta_{11} \times p$ plane (second), the $\beta_{12} \times \beta_{22}$ plane (third) and the $p \times \beta_{22}$ plane (right)

7 Conclusion

In this paper we have introduced a class of neural network sampling algorithms. In these algorithms neural network functions are used as an importance or candidate density in importance sampling or the Metropolis-Hastings algorithm. Neural networks are natural importance or candidate densities, as they have a universal approximation property and are easy to sample from. We have shown how to sample from three types of neural networks. One can sample directly from a certain 3-layer network. Using a 4-layer network one can, depending on the specification of the network, either use a Gibbs sampling approach or sample directly from a mixture of distributions. A key step in the proposed class of methods is the construction of a neural network that approximates the target density accurately. The methods have been tested on a set of illustrative models which include a Bayesian instrumental variable regression problem with weak instruments and near non-identification, a cointegration model and a two-regime growth model for US recessions and expansions. In our examples, involving experiments with non-standard, non-elliptical posterior distributions, the 4-layer network specified as the mixture of t distributions performs the best among the exposed sampling procedures. It is the fastest and moreover the most reliable neural network algorithm, whereas some other algorithms such as the Gibbs sampler, MH and IS fail or are very slow. These results indicate the feasibility and the possible usefulness of the neural network approach. We emphasize that it is naive to expect one sampling method to dominate in all practical cases. We suggested a strategy in which a sophisticated network is specified for complex, non-elliptical densities, while in a relatively simple case of near-elliptical contours a unimodal density or a bimodal mixture may be sufficiently accurate as candidate density. Clearly, more work is needed in this area.

We end this paper with some remarks on how to extend the proposed techniques. First, one may consider other ways of specifying and estimating neural networks. We mention here the following possible extensions. One may pursue the construction of well-behaved neural networks with other activation functions which are more smooth than the piecewise-linear one. We noted in section 2 that it is possible to perform auxiliary variable Gibbs sampling from a 4-layer neural network density with a scaled arctangent instead of the piecewise-linear function. One may also investigate the effects of substituting the exponential function in the second hidden layer by a different function such as the logistic function. One may also, as a first step, transform the posterior density function to a more regular shape. This line of research is recently pursued by, e.g., Bauwens, Bos, Van Dijk and Van Oest (2004) in a class of adaptive direction sampling (ADS) methods. A combination of ADS and neural network sampling may be of interest. In practice, one encounters cases where only part of the posterior density is ill-behaved. Then one may combine the neural network approach for the 'difficult part' with a Gibbs sampling approach for the regular part of the model. In recent work Richard (1998) and Liesenfeld and Richard (2002) constructed an efficient importance sampling technique where the estimation of the parameters of the importance function is done in a sequence of optimization steps.

Second, more experience is needed with empirical econometric models like business cycle models as specified by Hamilton (1989) and Paap and Van Dijk (2002), stochastic volatility models as given by Shephard (1996), and dynamic panel data models; see Pesaran and Smith (1995).

Third, the neural network approximations proposed in this paper may be useful for modelling volatility in financial series, see e.g. Donaldson and Kamstra (1997), and for evaluating option prices, see Hutchinson, Lo and Poggio (1994). We intend to report on this in future research.

A Sampling from a three-layer neural network distribution and computing its moments

Appendix A.1 gives analytical expressions for the integrals of the arctangent function. Appendix A.2 shows how these expressions are used in order to sample from a three-layer neural network distribution. In appendix A.3 these expressions are used to obtain analytical expressions for the moments of a three-layer neural network distribution. For a more elaborate appendix with more detailed derivations we refer to Hoogerheide, Kaashoek and van Dijk (2002).

A.1 A simple analytical expression for the integrals of the arctangent function

Theorem A.1: The n-th integral of the arctangent function $J_n(x)$

$$J_n(x) \equiv \int \cdots \int \arctan(x) dx \cdots dx$$

is given by

$$J_n(x) = p_n(x) \arctan(x) + q_n(x) \ln(1 + x^2) + r_n(x), \tag{46}$$

where p_n and q_n are polynomials of degree n and $n-1$, respectively:

$$\begin{aligned} p_n(x) &= p_{n,0} + p_{n,1}\, x + \cdots + p_{n,n-1}\, x^{n-1} + p_{n,n}\, x^n \\ q_n(x) &= q_{n,0} + q_{n,1}\, x + \cdots + q_{n,n-1}\, x^{n-1} \end{aligned}$$

The coefficients $p_{n,k}$ $(k = 0, 1, \ldots, n)$ and $q_{n,k}$ $(k = 0, 1, \ldots, n-1)$ are given by:

$$p_{n,k} = \begin{cases} \frac{(-1)^{(n-k)/2}}{(n-k)!k!} & \text{if } n-k \text{ is even} \\ 0 & \text{if } n-k \text{ is odd} \end{cases} \qquad q_{n,k} = \begin{cases} \frac{(-1)^{(n-k+1)/2}}{2(n-k)!k!} & \text{if } n-k \text{ is odd} \\ 0 & \text{if } n-k \text{ is even} \end{cases} \tag{47}$$

The polynomial r_n (of degree at most $n-1$) plays the role of the integrating constant.

Proof: We will prove this theorem by induction. First, note that for $n = 1$ the proposition holds, as we have by partial integration:

$$\int \arctan(x) dx = x \arctan(x) - \frac{1}{2} \ln(1 + x^2), \tag{48}$$

Now suppose that our proposition holds for a certain positive integer n. Then we have to show that this implies that the proposition also holds for $n + 1$.

First, note that for any non-negative integer k partial integration yields:

$$\begin{aligned} \int x^k \arctan(x) dx &= \frac{1}{k+1} x^{k+1} \arctan(x) - \frac{1}{k+1} \int \frac{x^{k+1}}{1+x^2} dx, \\ \int x^k \ln(1+x^2) dx &= \frac{1}{k+1} x^{k+1} \ln(1+x^2) - \frac{2}{k+1} \int \frac{x^{k+2}}{1+x^2} dx. \end{aligned} \tag{49}$$

Second, notice that a partial fraction decomposition yields:

$$\int \frac{x^m}{1+x^2} dx = \begin{cases} (-1)^{m/2} \arctan(x) + \sum_{i=0}^{(m-2)/2} \frac{(-1)^i}{m-1-2i}\, x^{m-1-2i} & \text{if } m \text{ is even,} \\ (-1)^{(m-1)/2} \frac{\ln(1+x^2)}{2} + \sum_{i=0}^{(m-3)/2} \frac{(-1)^i}{m-1-2i}\, x^{m-1-2i} & \text{if } m \text{ is odd.} \end{cases} \tag{50}$$

We may omit the polynomials in (50), since these would eventually be absorbed by the irrelevant polynomial r_n in formula (46), anyway. The induction assumption is that for a certain n it holds that:

$$\begin{aligned} J_n(x) &= (p_{n,0} + p_{n,1}\,x + \ldots + p_{n,n}\,x^n)\arctan(x) \\ &\quad + \left(q_{n,0} + q_{n,1}\,x + \ldots + q_{n,n-1}\,x^{n-1}\right)\ln(1+x^2) \end{aligned} \tag{51}$$

where the coefficients $p_{n,k}$ $(k = 0, 1, \ldots, n)$ and $q_{n,k}$ $(k = 0, 1, \ldots, n-1)$ are given by (47). It follows from (49) and (50) that:

$$\begin{aligned} J_{n+1}(x) &= \int J_n(x)dx \\ &= \left(p_{n+1,0} + p_{n,0}\,x + \frac{p_{n,1}}{2}\,x^2 + \ldots + \frac{p_{n,n}}{n+1}\,x^{n+1}\right)\arctan(x) \\ &\quad + \left(q_{n+1,0} + q_{n,0}\,x + \frac{q_{n,1}}{2}\,x^2 + \ldots + \frac{q_{n,n-1}}{n}\,x^n\right)\ln(1+x^2) \end{aligned}$$

Note that $J_{n+1}(x)$ has the shape of formula (46) with $p_{n+1,k} = p_{n,k-1}/k$ $(k = 1, \ldots, n+1)$ and $q_{n+1,k} = q_{n,k-1}/k$ $(k = 1, \ldots, n)$. Combining this with the induction assumption, it is easy to see the validity of the formulas for $p_{n+1,k}$ and $q_{n+1,k}$ for $k \geq 1$. Now we only have to prove that $p_{n+1,0}$ and $q_{n+1,0}$ are also given by (47). From (49) and (50) we have:

$$p_{n+1,0} = \sum_{\{k|1\leq k\leq n; k \text{ odd}\}} -\frac{(-1)^{(k+1)/2}}{k+1}\;p_{n,k} + \sum_{\{k|0\leq k\leq n-1; k \text{ even}\}} -\frac{2(-1)^{(k+2)/2}}{k+1}\;q_{n,k}. \tag{52}$$

If n is even, all $p_{n,k}$'s and $q_{n,k}$'s in the two summations of (52) are equal to zero, so that in that case $p_{n+1,0} = 0$. If n is odd, we have:

$$p_{n+1,0} = \sum_{\{k|1\leq k\leq n; k \text{ odd}\}} -\frac{(-1)^{(n+1)/2}}{(n-k)!(k+1)!} + \sum_{\{k|0\leq k\leq n-1; k \text{ even}\}} -\frac{(-1)^{(n+3)/2}}{(n-k)!(k+1)!}, \tag{53}$$

which can be rewritten as:

$$p_{n+1,0} = \frac{(-1)^{(n+1)/2}}{(n+1)!}\sum_{k=0}^{n}(-1)^k \left(\begin{array}{c} n+1 \\ k+1 \end{array}\right) = \frac{(-1)^{(n+1)/2}}{(n+1)!}, \tag{54}$$

where the last equality of (54) follows from Newton's binomium. The proof for $q_{n+1,0}$ is similar. We conclude that $p_{n+1,0}$ and $q_{n+1,0}$ are also given by (47), so that we have proved the theorem by induction. □

A.2 The marginal and conditional distribution functions corresponding to a three-layer neural network density

Suppose the random vector $X = (X_1, \ldots, X_n)'$ has the following density $p(x_1, \ldots, x_n)$:

$$p(x_1, \ldots, x_n) = \begin{cases} nn(x_1, \ldots, x_n) & \text{if } \underline{x}_i \leq x_i \leq \bar{x}_i \quad \forall\, i = 1, \ldots, n \\ \\ 0 & \text{else} \end{cases} \tag{55}$$

where $[\underline{x}_i, \bar{x}_i]$ is the interval to which the variable x_i $(i = 1, 2, \ldots, n)$ is restricted, and where $nn(x_1, \ldots, x_n)$ is the following three-layer neural network function:

$$nn(x_1, \ldots, x_n) = \sum_{h=1}^{H} \frac{c_h}{\pi} \arctan(a_h' x + b_h) + \frac{1}{2} \sum_{h=1}^{H} c_h + d. \tag{56}$$

Then the cumulative distribution function of X is given by:

$$\begin{aligned} CDF_X(\tilde{x}_1, \ldots, \tilde{x}_n) &= \int_{\underline{x}_n}^{\tilde{x}_n} \cdots \int_{\underline{x}_2}^{\tilde{x}_2} \int_{\underline{x}_1}^{\tilde{x}_1} nn(x_1, \ldots, x_n) dx_1 dx_2 \cdots dx_n \\ &= \sum_{h=1}^{H} \frac{c_h}{\pi} \int_{\underline{x}_n}^{\tilde{x}_n} \cdots \int_{\underline{x}_2}^{\tilde{x}_2} \int_{\underline{x}_1}^{\tilde{x}_1} \arctan(a_h' x + b_h) dx_1 dx_2 \cdots dx_n \\ &\quad + \left(\frac{1}{2} \sum_{h=1}^{H} c_h + d \right) x_1 x_2 \cdots x_n. \end{aligned} \tag{57}$$

Using the fact that $dx_1 = d(a_h' x + b_h)/a_{h1}$ (for constant values of $x_2, \ldots, x_n$), we make the following change of variables:

$$\begin{aligned} \int_{\underline{x}_1}^{\tilde{x}_1} \arctan(a_h' x + b_h) dx_1 &= \frac{1}{a_{h1}} \int_{a_{h1}\underline{x}_1 + a_{h,-1}' x_{-1} + b_h}^{a_{h1}\tilde{x}_1 + a_{h,-1}' x_{-1} + b_h} \arctan(a_h' x + b_h) d(a_h' x + b_h) \\ &= \frac{1}{a_{h1}} \left[J_1(a_{h1}\tilde{x}_1 + a_{h,-1}' x_{-1} + b_h) - J_1(a_{h1}\underline{x}_1 + a_{h,-1}' x_{-1} + b_h) \right], \end{aligned}$$

where we define $a_{h,-1} = (a_{h2}, \ldots, a_{hn})'$ and $x_{-1} = (x_2, \ldots, x_n)'$. If we continue in this way, we obtain the following formula:

$$\begin{aligned} &\int_{\underline{x}_n}^{\tilde{x}_n} \cdots \int_{\underline{x}_2}^{\tilde{x}_2} \int_{\underline{x}_1}^{\tilde{x}_1} \arctan(a_h' x + b_h) dx_1 dx_2 \cdots dx_n = \\ &= \frac{1}{a_{h1} a_{h2} \cdots a_{hn}} \sum_{D_1=0}^{1} \cdots \sum_{D_n=0}^{1} (-1)^{D_1 + D_2 + \cdots + D_n} J_n(a_{h1} x_{1,D_1} + \cdots + a_{hn} x_{n,D_n} + b_h) \end{aligned} \tag{58}$$

where we define $x_{i,0} = \tilde{x}_i$ and $x_{i,1} = \underline{x}_i$ $(i = 1, 2, \ldots, n)$, the upper and lower bounds of the integration intervals. The primitive $J_n(x)$ is given by Theorem A.1 in appendix A.1. Substituting (58) into (57) yields:

$$\begin{aligned} CDF_x(\tilde{x}_1, \ldots, \tilde{x}_n) &= \left(\frac{1}{2} \sum_{h=1}^{H} c_h + d \right) x_1 x_2 \cdots x_n + \\ &+ \sum_{h=1}^{H} \frac{c_h}{\pi a_{h1} a_{h2} \cdots a_{hn}} \sum_{D_1=0}^{1} \cdots \sum_{D_n=0}^{1} (-1)^{D_1 + \cdots + D_n} J_n \left(\sum_{i=1}^{n} a_{hi} x_{i,D_i} + b_h \right). \end{aligned} \tag{59}$$

The marginal distribution functions $CDF_{X_j}(x_j)$ $(j = 1, \ldots, n)$ are now obtained by taking $\tilde{x}_i = \bar{x}_i \; \forall i = 1, \ldots, n; i \neq j$:

$$CDF_{X_j}(x_j) = CDF_x(\bar{x}_1, \ldots, \bar{x}_{j-1}, x_j, \bar{x}_{j+1}, \ldots, \bar{x}_n). \tag{60}$$

The conditional CDF of X_j given $X_{j+1}, \ldots, X_n$ is derived in a similar way, simply by substituting $\sum_{i=j+1}^{n} a_{hi}x_i + b_h$ for b_h and treating the neural network as a function of $x_1, \ldots, x_j$.

As we have explicit formulas for the marginal and conditional distribution functions, it is easy to sample a random vector from a three-layer neural network density with (scaled) arctangent activation function. We can use the numerical inverse transformation method in the following way:

Step 1: Draw n independent U(0,1) variables $U_1, U_2, \ldots, U_n$.

Step 2: Draw X_n from its marginal distribution by computing the value of X_n such that $CDF_{X_n}(X_n) = U_n$ (using, for example, the bisection method).

Step 3: For $j = n-1, n-2, \ldots, 1$ iteratively draw X_j from its conditional distribution on $X_{j+1}, \ldots, X_n$ by computing the value of X_j such that $CDF(X_j | X_{j+1}, \ldots, X_n) = U_j$.

A.3 Analytical expressions for the moments of a three-layer neural network distribution

Suppose the vector $X = (X_1, \ldots, X_n)'$ has the three-layer neural network density $p(x_1, \ldots, x_n)$ given by (55) and (56). Then the expectation of X_n^k $(k = 1, 2, \ldots)$ is given by:

$$\begin{aligned}
&E(X_n^k) = \\
= \;& \int_{\underline{x}_n}^{\bar{x}_n} \int_{\underline{x}_{n-1}}^{\bar{x}_{n-1}} \!\!\cdots \int_{\underline{x}_1}^{\bar{x}_1} x_n^k \;\; nn(x_1, \ldots, x_n) dx_1 \cdots dx_{n-1} dx_n \\
= \;& \sum_{h=1}^{H} \frac{c_h}{\pi a_{h1} \cdots a_{h,n-1}} \sum_{D_1=0}^{1} \cdots \sum_{D_{n-1}=0}^{1} \Big[(-1)^{D_1 + \cdots + D_{n-1}} \times \\
& \qquad\qquad \times \int_{\underline{x}_n}^{\bar{x}_n} x_n^k \, J_{n-1} \left(\sum_{i=1}^{n-1} a_{hi} x_{i,D_i} + a_{hn} x_n + b_h \right) dx_n \Big] \\
& + \left(\frac{1}{2} \sum_{h=1}^{H} c_h + d \right) \frac{1}{k+1} (\bar{x}_1 - \underline{x}_1) \cdots (\bar{x}_{n-1} - \underline{x}_{n-1})(\bar{x}_n^{k+1} - \underline{x}_n^{k+1}),
\end{aligned} \tag{61}$$

where we define $x_{i,0} = \bar{x}_i$ and $x_{i,1} = \underline{x}_i$ $(i = 1, 2, \ldots, n-1)$, the upper and lower bounds of the integration intervals. We now make use of the following theorem:

Theorem A.2: If the n-th integral of a certain function $f : \mathbb{R} \to \mathbb{R}$ is given by $J_n : \mathbb{R} \to \mathbb{R}$, then it holds for $a_h, x \in \mathbb{R}^n$, $b_h \in \mathbb{R}$ and $k = 0, 1, 2, \ldots$ that:

$$\int x_i^k \, J_n(a_h' x + b_h) dx_i = \frac{1}{a_{hi}} \sum_{m=0}^{k} \left(-\frac{1}{a_{hi}} \right)^m \frac{k!}{(k-m)!} \, x_i^{k-m} \, J_{n+1+m}(a_h' x + b_h). \tag{62}$$

Proof: We will prove this theorem by induction with respect to k. First, note that for $k = 0$ we have:

$$\int J_n(a_h' x + b_h) dx_i = \frac{1}{a_{hi}} \int J_n(a_h' x + b_h) d(a_h' x + b_h) = \frac{1}{a_{hi}} J_{n+1}(a_h' x + b_h),$$

which clearly corresponds to Theorem A.2 for $k = 0$. Now suppose that our proposition holds for a certain nonnegative integer k. Then we have to show that this implies that the proposition also holds for $k + 1$.

Partial integration with x_i^{k+1} as the factor to be differentiated yields:

$$\int x_i^{k+1} \, J_n(a'_h x + b_h) dx_i = x_i^{k+1} \, \frac{1}{a_{hi}} \, J_{n+1}(a'_h x + b_h) - \frac{k+1}{a_{hi}} \int x_i^k \, J_{n+1}(a'_h x + b_h) dx_i. \tag{63}$$

The induction assumption is that Theorem A.2 holds for the value k. Using this induction assumption we rewrite the second term of (63) as:

$$\begin{aligned} & -\frac{1}{a_{hi}} \, (k+1) \int x_i^k \, J_{n+1}(a'_h x + b_h) dx_i = \\ = \; & \frac{1}{a_{hi}} \sum_{j=1}^{k+1} \left(-\frac{1}{a_{hi}} \right)^j \frac{(k+1)!}{(k+1-j)!} \, x_i^{k+1-j} \, J_{n+1+j}(a'_h x + b_h) \end{aligned} \tag{64}$$

Adding (64) to the first term of (63) yields:

$$\int x_i^{k+1} \, J_n(a'_h x + b_h) dx_i = \frac{1}{a_{hi}} \sum_{j=0}^{k+1} \left(-\frac{1}{a_{hi}} \right)^j \frac{(k+1)!}{(k+1-j)!} \, x_i^{k+1-j} \, J_{n+1+j}(a'_h x + b_h)$$

which is just equation (62) with $k + 1$ instead of k. We conclude that we have proved Theorem A.2 by induction. □

Substituting equation (62) of Theorem A.2 into (61) now yields $E(X_n^k)$, which can be easily adjusted to the general case of $E(X_i^k)$ $(i = 1, 2, \ldots, n)$ by taking a_{hi} and x_i instead of a_{hn} and x_n:

$$\begin{aligned} E(X_i^k) \;=\; & \sum_{h=1}^{H} \frac{c_h}{\pi a_{h1} \cdots a_{hn}} \sum_{D_1=0}^{1} \cdots \sum_{D_n=0}^{1} \Big[(-1)^{D_1 + \cdots + D_n} \times \\ & \times \sum_{m=0}^{k} \left(-\frac{1}{a_{hi}} \right)^m \frac{k!}{(k-m)!} \, x_i^{k-m} \, J_{n+m} \left(\sum_{i=1}^{n} a_{hi} x_{i,D_i} + b_h \right) \Big] \\ & + \left(\frac{1}{2} \sum_{h=1}^{H} c_h + d \right) \frac{1}{k+1} (\bar{x}_i^{k+1} - \underline{x}_i^{k+1}) \prod_{j=1; j \neq i}^{n} (\bar{x}_j - \underline{x}_j) \end{aligned} \tag{65}$$

In a similar fashion it can be derived that $E(X_i X_j)$ $(i, j = 1, 2, \ldots, n; i \neq j)$ is equal to:

$$\begin{aligned} E(X_i X_j) \quad = \quad & \sum_{h=1}^{H} \frac{c_h}{\pi a_{h1} \cdots a_{hn}} \sum_{D_1=0}^{1} \cdots \sum_{D_n=0}^{1} (-1)^{D_1 + \cdots + D_n} \times \\ & \times \left[x_i x_j J_n \left(\sum_{i=1}^{n} a_{hi} x_{i,D_i} + b_h \right) \right. \\ & - \frac{a_{hi} x_i + a_{hj} x_j}{a_{hi} a_{hj}} J_{n+1} \left(\sum_{i=1}^{n} a_{hi} x_{i,D_i} + b_h \right) \\ & \left. + \frac{1}{a_{hi} a_{hj}} J_{n+2} \left(\sum_{i=1}^{n} a_{hi} x_{i,D_i} + b_h \right) \right] \\ & + \left(\frac{1}{2} \sum_{h=1}^{H} c_h + d \right) \frac{1}{4} (\bar{x}_i^2 - \underline{x}_i^2)(\bar{x}_j^2 - \underline{x}_j^2) \prod_{k=1; k \neq i,j}^{n} (\bar{x}_k - \underline{x}_k). \end{aligned} \tag{66}$$

Using formulas (65) and (66), one can easily compute statistics of a three-layer feed-forward neural network distribution, such as mean, variance, skewness, kurtosis, covariances and correlations.

B Sampling from a four-layer neural network distribution

Appendix B.1 discusses how to draw from a four-layer neural network distribution using Gibbs sampling. Appendix B.2 shows another way to draw from a four-layer neural network: auxiliary variable Gibbs sampling.

B.1 Gibbs sampling from a four-layer neural network distribution

Suppose a density kernel of $X \in \mathbb{R}^n$ is given by

$$p(x) = \begin{cases} nn(x) & \text{if } x_i \in [\underline{x}_i, \bar{x}_i] \;\; \forall i = 1, \ldots, n \\ 0 & \text{else} \end{cases} \tag{67}$$

where $[\underline{x}_i, \bar{x}_i]$ is the interval to which X_i $(i = 1, \ldots, n)$ is restricted. Suppose the function $nn(x)$ corresponds to the following four-layer feed-forward neural network with n inputs x_i $(i = 1, \ldots, n)$, and H hidden cells:

$$nn(x) = \exp\left(\sum_{h=1}^{H} c_h \, plin\left(\sum_{i=1}^{n} a_{hi} x_i + b_h \right) + d \right), \tag{68}$$

where $plin : \mathbb{R} \to \mathbb{R}$ is the following piecewise-linear function:

$$plin(x) = \begin{cases} 0 & x < -1/2 \\ x + 1/2 & -1/2 \leq x \leq 1/2 \\ 1 & x > 1/2 \end{cases} \tag{69}$$

We rewrite the neural network density $nn(x) = nn(x_j, x_{-j})$ as

$$nn(x_j, x_{-j}) \quad \propto \quad \exp\left(\sum_{h=1}^{H} c_h \, plin\left(a_{hj} x_j + \sum_{i=1, i \neq j}^{n} a_{hi} x_i + b_h \right) \right),$$

which is a kernel of the conditional density of x_j given x_{-j}. For each hidden cell h ($h = 1, \ldots, H$) there are two points x_j where its input $a'_h x + b_h$ moves from one of the intervals $(-\infty, -1/2)$, $[-1/2, 1/2]$ and $(1/2, \infty)$ to another one:

$$a_{hj}x_j + \sum_{i=1, i\neq j}^{n} a_{hi}x_i + b_h = \pm\frac{1}{2} \quad \Leftrightarrow \quad x_j = \frac{1}{a_{hj}}\left(\pm\frac{1}{2} - \sum_{i=1, i\neq j}^{n} a_{hi}x_i - b_h\right). \tag{70}$$

Consider only those 'changing points' $\tilde{x}_{j,k}$ ($k = 1, \ldots, m$ with $m \leq 2H$) that are in the interval of interest $[\underline{x}_j, \bar{x}_j]$, and order these m points such that:

$$\tilde{x}_{j,1} < \tilde{x}_{j,2} < \cdots < \tilde{x}_{j,m-1} < \tilde{x}_{j,m}$$

If we define $\tilde{x}_{j,0} = \underline{x}_j$ and $\tilde{x}_{j,m+1} = \bar{x}_j$, we have $m+1$ intervals $[\tilde{x}_{j,k}, \tilde{x}_{j,k+1}]$ ($k = 0, 1, \ldots, m$) on which a kernel of the conditional density of X_j given X_{-j} is given by:

$$nn(x_j, x_{-j}) \propto \exp(\tilde{a}_k x_j + \tilde{b}_k) \tag{71}$$

for certain constants $\tilde{a}_k$ and $\tilde{b}_k$ ($k = 0, 1, \ldots, m$). The primitive of (71) is given by

$$\int \exp(\tilde{a}_k x_j + \tilde{b}_k) dx_j = \begin{cases} \frac{1}{\tilde{a}_k} \exp(\tilde{a}_k x_j + \tilde{b}_k) + C_k & \text{if } \tilde{a}_k \neq 0 \\ \\ \exp(\tilde{b}_k) x_j + C_k & \text{if } \tilde{a}_k = 0. \end{cases}$$

where C_k ($k = 0, 1, \ldots, m$) are integration constants that we specify in such a way that the CDF starts at the value 0 and is continuous in x_j. After this kernel of the conditional CDF has been obtained, X_j is drawn from its conditional distribution using the inverse transformation method: one draws $U \sim U(0,1)$ and computes:

$$X_j = \frac{\log\left[\tilde{a}_k \left(S\,U - C_k\right)\right] - \tilde{b}_k}{\tilde{a}_k} \qquad \text{or} \qquad X_j = \frac{S\,U - C_k}{\exp(\tilde{b}_k)}$$

depending on whether X_j falls in a region with $\tilde{a}_k = 0$ or not; S is the 'scaling constant' of the kernel, which is computed as the value of the kernel of the conditional CDF at $\bar{x}_j$.

Since it is easy to draw X_j conditional on X_{-j} ($j = 1, \ldots, n$), it is easy to perform Gibbs sampling from a four-layer neural network distribution.

B.2 Auxiliary variable Gibbs sampling from a four-layer neural network distribution

Suppose a density kernel of $X \in \mathbb{R}^n$ is given by

$$p(x) = \begin{cases} nn(x) & \text{if } x_i \in [\underline{x}_i, \bar{x}_i] \ \forall i = 1, \ldots, n \\ 0 & \text{else} \end{cases} \tag{72}$$

where $[\underline{x}_i, \bar{x}_i]$ is the interval to which X_i ($i = 1, \ldots, n$) is restricted. Suppose the function $nn(x)$ corresponds to the following four-layer feed-forward neural network with n inputs x_i ($i = 1, \ldots, n$), and H hidden cells:

$$nn(x) = \exp\left(\sum_{h=1}^{H} c_h\, g\left(\sum_{i=1}^{n} a_{hi}x_i + b_h\right) + d\right), \tag{73}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing function taking its values in [0,1], which is invertible on the interval $(\underline{x}, \bar{x})$ where it takes its values in (0,1). We will denote this invertible function by $\tilde{g} : (\underline{x}, \bar{x}) \to (0,1)$ with inverse $\tilde{g}^{-1} : (0,1) \to (\underline{x}, \bar{x})$. Note that the interval $(\underline{x}, \bar{x})$ may be equal to $(-\infty, \infty)$. Examples of such a function g are the logistic, the piecewise-linear and the scaled arctangent function.

Auxiliary variable Gibbs sampling is possible if the density kernel p can be decomposed as follows:

$$p(x) \propto \pi(x) \prod_{k=1}^{K} l_k(x), \tag{74}$$

where π is a density kernel from which sampling is easy, and l_k $(k = 1, \ldots, K)$ are non-negative functions of $x \in \mathbb{R}^n$. The trick is that a set $U = (U_1, \ldots, U_K)$ of auxiliary variables is introduced such that a kernel of the joint density of X and U is given by:

$$p(x, u) \propto \pi(x) \prod_{k=1}^{K} I\left\{0 < u_k < l_k(x)\right\}. \tag{75}$$

It is easily seen that (74) is a marginal density kernel corresponding to the joint density (75). Therefore one can sample $X \sim p(x)$ by sampling both X and U from (75) and forgetting U.

Kernels from the conditional distributions of X and U are easily obtained from the joint density kernel:

$$p(x|u) \propto \pi(x) I\left\{l_k(x) > u_k, k = 1, \ldots, K\right\} \tag{76}$$

$$p(u|x) \propto \prod_{k=1}^{K} I\left\{0 < u_k < l_k(x)\right\} \tag{77}$$

It follows from (76) and (77) that an iteration of the auxiliary variable Gibbs sampler consists of drawing X from a truncated version of an 'easy' distribution with density kernel π, and sampling U_k $(k = 1, \ldots, K)$ from K independent uniform distributions.

We rewrite (72) as:

$$p(x) \propto \prod_{i=1}^{n} I\left\{\underline{x}_i < x_i < \bar{x}_i\right\} \prod_{h=1}^{H} \exp\left(c_h\, g\left(\sum_{i=1}^{n} a_{hi} x_i + b_h\right)\right). \tag{78}$$

which has the shape of (74) with

$$\begin{aligned} \pi(x) &= \prod_{i=1}^{n} I\left\{\underline{x}_i < x_i < \bar{x}_i\right\}, && (79) \\ l_h(x) &= \exp\left(c_h\, g\left(\sum_{i=1}^{n} a_{hi} x_i + b_h\right)\right) \quad \text{for } h = 1, \ldots, H. && (80) \end{aligned}$$

where $\pi(x)$ is the 'easy' density kernel of n independent variables X_i $(i = 1, \ldots, n)$ with distribution $U(\underline{x}_i, \bar{x}_i)$.

Drawing U conditionally on the values of X is straightforward. Combining (77) and (80), it follows that the elements U_h $(h = 1, \ldots, H)$ are drawn independently from the distributions:

$$U_h | X = x \;\sim\; U\left(0, \exp\left[c_h\, g\left(\sum_{i=1}^{n} a_{hi} x_i + b_h\right)\right]\right) \tag{81}$$

Drawing X conditionally on the values of U is a little harder. We choose to break up X and sample the elements X_i ($i = 1, \ldots, n$) conditionally on the values of U and the set of all other elements X_{-i}. Combining (76), (79) and (80), we derive a density kernel of the conditional distribution of X_i given X_{-i} and U:

$$p(x_i|u, x_{-i}) \propto I\left\{\underline{x}_i < x_i < \bar{x}_i\right\} I\left\{l_h(x_i, x_{-i}) > u_h, h = 1, \ldots, H\right\} \tag{82}$$

We now take a closer look at the inequalities $l_h(x_i, x_{-i}) > u_h$ ($h = 1, \ldots, H$). First, we can rule out that $c_h = 0$ or $a_{hi} = 0$ for any h, since in that case we just delete the involved hidden cell. If we consider $l_h(x_i, x_{-i})$ as a function of x_i for given values of x_{-i}, denoted by $l_{h,x_{-i}}(x_i)$, then the inverse $l^{-1}_{h,x_{-i}}$ (if it exists) is given by:

$$l^{-1}_{h,x_{-i}}(u_h) = \frac{1}{a_{hi}}\left(\tilde{g}^{-1}\left(\frac{\log(u_h)}{c_h}\right) - \left(\sum_{j=1, j\neq i}^{n} a_{hj}x_j + b_h\right)\right). \tag{83}$$

Note that this inverse exists only if $\log(u_h)/c_h \in (0, 1)$, and that the cases in which the inverse $l^{-1}_{h,x_{-i}}$ does not exist are the cases in which hidden cell h implies no restriction for x_i. Also notice that this implies an upper bound for x_i if $c_h a_{hi} > 0$ and a lower bound if $c_h a_{hi} < 0$.

We conclude that (82) is a density kernel of the distribution

$$X_i|U = u, X_{-i} = x_{-i} \;\sim\; U(x_{i,LB}(u, x_{-i}), x_{i,UB}(u, x_{-i})), \tag{84}$$

with

$$x_{i,LB}(u, x_{-i}) = \max\left\{\max_{1\leq h\leq H}\left\{l^{-1}_{h,x_{-i}}(u_h)\,\middle|\, c_h a_{hi} > 0, \frac{\log(u_h)}{c_h} \in (0, 1)\right\}, \underline{x}_i\right\}$$

$$x_{i,UB}(u, x_{-i}) = \min\left\{\min_{1\leq h\leq H}\left\{l^{-1}_{h,x_{-i}}(u_h)\,\middle|\, c_h a_{hi} < 0, \frac{\log(u_h)}{c_h} \in (0, 1)\right\}, \bar{x}_i\right\},$$

where $l^{-1}_{h,x_{-i}}(u_h)$ is given by (83), and where $[\underline{x}_i, \bar{x}_i]$ is the interval to which X_i ($i = 1, \ldots, n$) is a priori restricted.

The auxiliary variable Gibbs sampling procedure is now given by:

```
Initialization: Choose feasible $x^0 = (x^0_1, \ldots, x^0_n)$.
Do for $j = 1, 2, \ldots, m$
    Do for $h = 1, 2, \ldots, H$
        Obtain $u^j_h \sim U_h|X = x^{j-1}$ from (81).
    Do for $i = 1, 2, \ldots, n$
        Obtain $x^j_i \sim X_i|U = u^j, X_{-i} = x^{j-1}_{-i}$ from (84).
```

Here x^{j-1}_{-i} denotes

$$x^{j-1}_{-i} = x^j_1, \ldots, x^j_{i-1}, x^{j-1}_{i+1}, \ldots, x^{j-1}_n,$$

the set of all components except x_i at their current values. Note that this procedure only requires drawing from uniform distributions, which is done easily and fast.

References

[1] Angrist, J.D. and A.B. Krueger (1991): "Does Compulsory School Attendance Affect Schooling and Earnings?", *Quarterly Journal of Economics*, 106, 979-1014.

[2] Bauwens, L. and H.K. van Dijk (1989): "Bayesian limited information analysis revisited". In: J. J. Gabszewicz et al. (eds), *Economic Decision-Making: Games, Econometrics and Optimisation*, North-Holland, Amsterdam.

[3] Bauwens, L., M. Lubrano and J.-F. Richard (1999): *Bayesian Inference in Dynamic Econometric Models*, Oxford University Press.

[4] Bauwens, L., C.S. Bos, H.K. van Dijk and R.D. van Oest (2004): "Adaptive Radial-Based Direction Sampling: Some flexible and robust Monte Carlo integration methods", *Journal of Econometrics*, forthcoming.

[5] Campbell, J.Y. (1987): "Does saving anticipate declining labor income? An alternative test of the permanent income hypothesis", *Econometrica*, 55, 1249-1273.

[6] Campbell, J.Y. and R.J. Shiller (1987): "Cointegration and tests of present value models", *Journal of Political Economy*, 95, 1062-1088.

[7] Campbell, J.Y. and G.N. Mankiw (1989): "Consumption, Income, and Interest Rates: Reinterpretinng the Time Series Evidence", *NBER Macroeconomics Annual 1989*, MIT Press, Cambridge.

[8] Chib, S. and E. Greenberg (1996): "Markov Chain Monte Carlo Simulation Methods in Econometrics", *Econometric Theory*, 12(3), 409-431.

[9] Damien, P., J. Wakefield and S. Walker (1999), "Gibbs Sampling for Bayesian Non-conjugate and Hierarchical Models by using Auxiliary Variables", *Journal of the Royal Statistical Society B*, 61, 331-344

[10] Donaldson, R.G. and M. Kamstra (1997), "An Artificial Neural Network-GARCH Model for International Stock Return Volatility", *Journal of Empirical Finance*, 4 (1), 17-46.

[11] Edwards, R.G. and A.D. Sokal (1988), "Generalization of the Fortuin-Kasteleyn-Swendsen-Wang Representation and Monte Carlo Algorithm", *Physical Review D*, 38, 2009-2012

[12] Gallant, A.R. and H. White (1989): "There exists a neural network that does not make avoidable mistakes", in *Proc. of the International Conference on Neural Networks*, San Diego, 1988 (IEEE Press, New York).

[13] Gelman, A. and X. Meng (1991): "A Note on Bivariate Distributions That Are Conditionally Normal", *The American Statistician*, 45, 125-126.

[14] Geman, S. and D. Geman (1984): "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-741.

[15] Geweke, J. (1989): "Bayesian inference in econometric models using Monte Carlo integration", *Econometrica*, 57, 1317-1339.

[16] Geweke, J. (1999): "Using Simulation Methods for Bayesian Econometric Models: Inference, Development, and Communication", *Econometric Reviews*, 18(1), 1-73.

[17] Hall, A.D., H.M. Anderson and C.W.J. Granger (1992): "A cointegration analysis of treasury bill yields", *Review of Economics and Statistics*, 74, 116-126.

[18] Hamilton, J.D. (1989): "A New Approach to the Econometric Analysis of Nonstationary Time Series and Business Cycles", *Econometrica*, 57, 357-384.

[19] Hammersley, J. and D. Handscomb (1964): "Monte Carlo Methods". Chapman and Hall, London.

[20] Hastings, W.K. (1970): "Monte Carlo Sampling Methods using Markov Chains and their Applications", *Biometrika*, 57, 97-109.

[21] Hecht-Nielsen, R. (1987): "Kolmogorov mapping neural network existence theorem", in *Proc. IEEE First International Conference on Neural Networks*, San Diego, 1987, 11-13.

[22] Hobert, J.P. and G. Casella (1996): "The Effect of Improper Priors on Gibbs Sampling in Hierarchical Linear Mixed Models", *Journal of the American Statistical Association*, 91(436), 1461-1473.

[23] Hoogerheide, L.F. and H.K. van Dijk (2001): "Comparison of the Anderson-Rubin test for overidentification and the Johansen test for cointegration", Econometric Institute report 2001-04, Erasmus University Rotterdam.

[24] Hoogerheide, L.F., J.F. Kaashoek and H.K. van Dijk (2002): "Functional Approximations to Posterior Densities: A Neural Network Approach to Efficient Sampling", Econometric Institute report 2002-48, Erasmus University Rotterdam.

[25] Hoogerheide L.F., J.F. Kaashoek and H.K. van Dijk (2003): "Neural Network Approximations to Posterior Densities: An Analytical Approach", 2003 Proceedings of the American Statistical Association, Section on Bayesian Statistical Science [CD-ROM], American Statistical Association, Alexandria, VA.

[26] Hornik, K., M. Stinchcombe, and H. White (1989): "Multilayer feedforward networks are universal approximators", *Neural Networks*, Vol. 2, 359-366.

[27] Hutchinson, J., A. Lo and T. Poggio (1994): "A Nonparametric Approach to the Pricing and Hedging of Derivative Securities Via Learning Networks", *Journal of Finance*, 49, 851-889.

[28] King, R.G., C.I. Plosser, J.H. Stock and M.W. Watson (1991): "Stochastic trends and economic fluctuations", *American Economic Review*, 81, 819-840.

[29] Kleibergen, F.R., and H.K. Van Dijk (1994): "On the Shape of the Likelihood/Posterior in Cointegration Models", *Econometric Theory*, 10(3-4), 514-551.

[30] Kleibergen, F.R., and H.K. Van Dijk (1998): "Bayesian Simultaneous Equations Analysis using Reduced Rank Structures", *Econometric Theory*, 14(6), 701-743.

[31] Kloek, T., and H.K. Van Dijk (1978): "Bayesian estimates of equation system parameters: an application of integration by Monte Carlo", *Econometrica*, 46, 1-19.

[32] Kolmogorov, A.N. (1957): "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition", *American Mathematical Monthly Translation*, Vol. 28, pp 55-59. (Russian original in Doklady Akademii Nauk SSSR, 144, 953-956)

[33] Leshno, M., Lin, V.Y., Pinkus, A. and Schocken, S. (1993): "Multilayer Feedforward Networks With a Nonpolynomial Activation Function Can Approximate Any Function", *Neural networks*, Vol. 6, 861-867.

[34] Liesenfeld, R. and J.-F. Richard (2002): "Univariate and Multivariate Stochastic Volatility Models: Estimation and Diagnostics", Discussion paper, University of Tubingen.

[35] Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller (1953): "Equations of State Calculations by Fast Computing Machines", *Journal of Chemical Physics*, 21, 1087-1091.

[36] Paap, R. and H.K. van Dijk (2002): "Bayes Estimates of Markov Trends in Possibly Cointegrated Series: An Application to US Consumption and Income", Econometric Institute report 2002-42, Erasmus University Rotterdam.

[37] Pesaran, M.H. and R. Smith (1995): "Estimation of Long-Run Relationships from Dynamic Heterogeneous Panels", *Journal of Econometrics*, 68, 79-113.

[38] Richard, J.-F. (2002): "Efficient High-dimensional Monte Carlo Importance Sampling", Discussion paper, University of Pittsburgh.

[39] Ritter, C. and M.A. Tanner (1992): "Facilitating the Gibbs Sampler: The Gibbs Stopper and the Griddy-Gibbs Sampler", *Journal of the American Statistical Association*, 87, 861-868.

[40] Schotman, P.C. and H.K. van Dijk (1991): "A Bayesian Analysis of the Unit Root in Real Exchange Rates", *Journal of Econometrics*, 49, 195-238.

[41] Shephard, N. (1996): "Statistical aspects of ARCH and stochastic volatility", in *Time Series Models with Econometric, Finance and Other Applications*, ed. by D.R. Cox, D.V. Hinkley and O.E. Barndorff-Nielson, Chapman and Hall, London.

[42] Staiger, D. and J.H. Stock (1997): "Instrumental Variable Regression with Weak Instruments", *Econometrica*, 65, 557-586.

[43] Stinchcombe, M. (1989): "Universal Approximation Using Feedforward Networks with Nonsigmoid Hidden Layer Activation Functions", in *Proceedings of the International Joint Conference on Neural Networks, Washington DC*, IEEE Press, New York.

[44] Stinchcombe, M. (1990): "Approximating and Learning Unknown Mappings Using Multilayer Feedforward Networks with Bounded Weights", in *Proceedings of the International Joint Conference on Neural Networks, San Diego*, IEEE Press, New York.

[45] Tanner, M.A. and W.H. Wong (1987): "The Calculation of Posterior Distributions by Data Augmentation" (with discussion), *Journal of the American Statistical Association*, 82, 528-550.

[46] Tierney, L. (1994): “Markov Chains for Exploring Posterior Distributions”, *Annals of Statistics*, 22, 1701-1762.

[47] Van Dijk, H.K., and T. Kloek (1980): “Further experience in Bayesian analysis using Monte Carlo integration”, *Journal of Econometrics*, 14, 307-328.

[48] Van Dijk, H.K., and T. Kloek (1984): “Experiments with some alternatives for simple importance sampling in Monte Carlo integration”, in *Bayesian Statistics 2*, ed. by J. M. Bernardo, M. Degroot, D. Lindley, and A. F. M. Smith, Amsterdam, North-Holland.

[49] Van Dijk, H.K. (2003): “On Bayesian structural inference in a simultaneous equation model”, in *Econometrics and the philosophy of economics*, ed. by B.P. Stigum, Princeton University Press, Princeton, New Jersey.

[50] Zellner, A. (1971): *An introduction to Bayesian inference in econometrics.* Wiley, New York.