Efficient Gradient-Based Inference through Transformations between Bayes Nets and Neural Nets

Diederik P. Kingma D.P.KINGMA@UVA.NL
Max Welling M.WELLING@UVA.NL
Machine Learning Group, University of Amsterdam

Abstract

Hierarchical Bayesian networks and neural networks with stochastic hidden units are commonly perceived as two separate types of models. We show that either of these types of models can often be transformed into an instance of the other, by switching between centered and differentiable non-centered parameterizations of the latent variables. The choice of parameterization greatly influences the efficiency of gradient-based posterior inference; we show that they are often complementary to eachother, we clarify when each parameterization is preferred and show how inference can be made robust. In the non-centered form, a simple Monte Carlo estimator of the marginal likelihood can be used for learning the parameters. Theoretical results are supported by experiments.

1. Introduction

Bayesian networks (also called *belief networks*) are probabilistic graphical models where the conditional dependencies within a set of random variables are described by a directed acyclic graph (DAG). Many supervised and unsupervised models can be considered as special cases of Bayesian networks.

In this paper we focus on the problem of efficient inference in Bayesian networks with multiple layers of continuous latent variables, where exact posterior inference is intractable (e.g. the conditional dependencies between variables are nonlinear) but the joint distribution is differentiable. Algorithms for approximate inference in Bayesian networks can be roughly divided into two categories: sampling approaches and parametric approaches. Parametric approaches include Belief Propagation (Pearl, 1982) or the more recent Expectation Propagation (EP) (Minka, 2001). When it is not reasonable or possible to make assumptions about the posterior (which is often the case), one needs to resort to sampling approaches such as Markov Chain Monte Carlo (MCMC) (Neal, 1993). In high-dimensional spaces, gradient-based samplers such as Hybrid Monte Carlo (Duane et al., 1987) and the recently proposed no-U-turn sampler (Hoffman & Gelman, 2011) are known for their relatively fast mixing properties. When just interested in finding a mode of the posterior, vanilla gradient-based optimization methods can be used. The alternative parameterizations suggested in this paper can dramatically improve the efficiency of any of these algorithms.

1.1. Outline of the paper

After reviewing background material in 2, we introduce a generally applicable differentiable reparameterization of continuous latent variables into a differentiable non-centered form in section 3. In section 4 we analyze the posterior dependencies in this reparameterized form. Experimental results are shown in section 6.

2. Background

Notation. We use bold lower case (e.g. $\mathbf{x}$ or $\mathbf{y}$) notation for random variables and instantiations (values) of random variables. We write $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$ to denote (conditional) probability density (PDF) or mass (PMF) functions of variables. With $\boldsymbol{\theta}$ we denote the vector containing all parameters; each distribution in the network uses a subset of $\boldsymbol{\theta}$'s elements. Sets of variables are capitalized and bold, matrices are capitalized and bold, and vectors are written in bold and lower case.

2.1. Bayesian networks

A Bayesian network models a set of random variables $\mathbf{V}$ and their conditional dependencies as a directed acyclic graph, where each variable corresponds to a vertex and each edge to a conditional dependency. Let the distribu-

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

Figure 1. **(a)** The centered parameterization (CP) of a latent variable $\mathbf{z}_j$. **(b)** The differentiable non-centered parameterization (DNCP) where we have introduced an auxiliary 'noise' variable $\boldsymbol{\epsilon}_j \sim p_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_j)$ such that $\mathbf{z}_j$ becomes deterministic: $\mathbf{z}_j = g_j(\mathbf{pa}_j, \boldsymbol{\epsilon}_j, \boldsymbol{\theta})$. This deterministic variable has an interpretation of a hidden layer in a neural network, which can be differentiated efficiently using the backpropagation algorithm.

tion of each variable $\mathbf{v}_j$ be $p_{\boldsymbol{\theta}}(\mathbf{v}_j|\mathbf{pa}_j)$, where we condition on $\mathbf{v}_j$'s (possibly empty) set of parents $\mathbf{pa}_j$. Given the factorization property of Bayesian networks, the joint distribution over all variables is simply:

$$p_{\boldsymbol{\theta}}(\mathbf{v}_1, \ldots, \mathbf{v}_N) = \prod_{j=1}^{N} p_{\boldsymbol{\theta}}(\mathbf{v}_j|\mathbf{pa}_j) \tag{1}$$

Let the graph consist of one or more (discrete or continuous) observed variables $\mathbf{x}_j$ and continuous latent variables $\mathbf{z}_j$, with corresponding conditional distributions $p_{\boldsymbol{\theta}}(\mathbf{x}_j|\mathbf{pa}_j)$ and $p_{\boldsymbol{\theta}}(\mathbf{z}_j|\mathbf{pa}_j)$. We focus on the case where both the marginal likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}) = \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\, d\mathbf{z}$ and the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ are intractable to compute or differentiate directly w.r.t. $\boldsymbol{\theta}$ (which is true in general), and where the joint distribution $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ is at least once differentiable, so it is still possible to efficiently compute first-order partial derivatives $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ and $\nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$.

2.2. Conditionally deterministic variables

A *conditionally deterministic variable* $\mathbf{v}_j$ with parents $\mathbf{pa}_j$ is a variable whose value is a (possibly nonlinear) deterministic function $g_j(.)$ of the parents and the parameters: $\mathbf{v}_j = g_j(\mathbf{pa}_j, \boldsymbol{\theta})$. The PDF of a conditionally deterministic variable is a Dirac delta function, which we define as a Gaussian PDF $\mathcal{N}(.; \mu, \sigma)$ with infinitesimal σ:

$$p_{\boldsymbol{\theta}}(\mathbf{v}_j|\mathbf{pa}_j) = \lim_{\sigma \to 0} \mathcal{N}(\mathbf{v}_j; g_j(\mathbf{pa}_j, \boldsymbol{\theta}), \sigma) \tag{2}$$

which equals $+\infty$ when $\mathbf{v}_j = g_j(\mathbf{pa}_j, \boldsymbol{\theta})$ and equals 0 everywhere else such that $\int_{\mathbf{v}_j} p_{\boldsymbol{\theta}}(\mathbf{v}_j|\mathbf{pa}_j)\, d\mathbf{v}_j = 1$.

2.3. Inference problem under consideration

We are often interested in performing posterior inference, which most frequently consists of either optimization (finding a mode $\operatorname{argmax}_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$) or sampling from the posterior $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$. Gradients of the log-posterior w.r.t. the latent variables can be easily acquired using the equality:

$$\begin{aligned} \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) &= \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \\ &= \sum_{j=1}^{N} \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{v}_j|\mathbf{pa}_j) \end{aligned} \tag{3}$$

In words, the gradient of the log-posterior w.r.t. the latent variables is simply the sum of gradients of individual factors w.r.t. the latent variables. These gradients can then be followed to a mode if one is interested in finding a MAP solution. If one is interested in sampling from the posterior then the gradients can be plugged into a gradient-based sampler such as Hybrid Monte Carlo (Duane et al., 1987); if also interested in learning parameters, the resulting samples can be used for the *E*-step in Monte Carlo EM (Wei & Tanner, 1990) (MCEM).

Problems arise when strong posterior dependencies exist between latent variables. From eq. (3) we can see that the Hessian $\mathbf{H}$ of the posterior is:

$$\mathbf{H} = \nabla_{\mathbf{z}} \nabla_{\mathbf{z}}^T \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \sum_{j=1}^{N} \nabla_{\mathbf{z}} \nabla_{\mathbf{z}}^T \log p_{\boldsymbol{\theta}}(\mathbf{v}_j|\mathbf{pa}_j) \tag{4}$$

Suppose a factor $\log p_{\boldsymbol{\theta}}(z_i|z_j)$ connecting two scalar latent variables $\mathbf{z}_i$ and $\mathbf{z}_j$ exists, and z_i is strongly dependent on z_j, then the Hessian's corresponding element $\frac{\partial^2 \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}{\partial z_i \partial z_j}$ will have a large (positive or negative) value. This is bad for gradient-based inference since it means that changes in z_j have a large effect on the gradient $\frac{\partial \log p_{\boldsymbol{\theta}}(z_i|z_j)}{\partial z_i}$ and changes in z_i have a large effect on the gradient $\frac{\partial \log p_{\boldsymbol{\theta}}(z_i|z_j)}{\partial z_j}$. In general, strong conditional dependencies lead to ill-conditioning of the posterior, resulting in smaller optimal stepsizes for first-order gradient-based optimization or sampling methods, making inference less efficient.

3. The differentiable non-centered parameterization (DNCP)

In this section we introduce a generally applicable transformation between continuous latent random variables and deterministic units with auxiliary parent variables. In rest of the paper we analyze its ramifications for gradient-based inference.

3.1. Parameterizations of latent variables

Let $\mathbf{z}_j$ be some continuous latent variable with parents $\mathbf{pa}_j$, and corresponding conditional PDF:

$$\mathbf{z}_j|\mathbf{pa}_j \sim p_{\boldsymbol{\theta}}(\mathbf{z}_j|\mathbf{pa}_j) \tag{5}$$

This is also known in the statistics literature as the *centered parameterization* (CP) of the latent variable $\mathbf{z}_j$. Let the

differentiable non-centered parameterization (DNCP) of the latent variable $\mathbf{z}_j$ be:

$$\mathbf{z}_j = g_j(\mathbf{pa}_j, \boldsymbol{\epsilon}_j, \boldsymbol{\theta}) \quad \text{where} \quad \boldsymbol{\epsilon}_j \sim p(\boldsymbol{\epsilon}_j) \tag{6}$$

where $g_j(.)$ is some differentiable function. Note that in the DNCP, the value of $\mathbf{z}_j$ is *deterministic* given both $\mathbf{pa}_j$ and the newly introduced auxiliary variable $\boldsymbol{\epsilon}_j$ which is distributed as $p(\boldsymbol{\epsilon}_j)$. See figure 1 for an illustration of the two parameterizations.

By the change of variables, the relationship between the original PDF $p_{\boldsymbol{\theta}}(\mathbf{z}_j|\mathbf{pa}_j)$, the function $g_j(\mathbf{pa}_j, \boldsymbol{\epsilon}_j)$ and the PDF $p(\boldsymbol{\epsilon}_j)$ is:

$$p(\boldsymbol{\epsilon}_j) = p_{\boldsymbol{\theta}}(\mathbf{z}_j = g_j(\mathbf{pa}_j, \boldsymbol{\epsilon}_j, \boldsymbol{\theta})|\mathbf{pa}_j)\ |det(\mathbf{J})| \tag{7}$$

where $det(\mathbf{J})$ is the determinant of Jacobian of $g_j(.)$ w.r.t. $\boldsymbol{\epsilon}_j$. If z_j is a scalar variable, then ϵ_j is also scalar and $|det(\mathbf{J})| = |\frac{\partial z_j}{\partial \epsilon_j}|$.

In the DNCP, the original latent variable $\mathbf{z}_j$ has become deterministic, and its PDF $p_{\boldsymbol{\theta}}(\mathbf{z}_j|\mathbf{pa}_j, \boldsymbol{\epsilon}_j)$ can be described as a Dirac delta function (see section 2.2).

The joint PDF over the random and deterministic variables can be integrated w.r.t. the determinstic variables. If for simplicity we assume that observed variables are always leaf nodes of the network, and that all latent variables are reparameterized such that the only *random* variables left are the observed and auxiliary variables $\mathbf{x}$ and $\boldsymbol{\epsilon}$, then the marginal joint $p_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\epsilon})$ is obtained as follows:

$$\begin{aligned}
p_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\epsilon}) &= \int_{\mathbf{z}} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}, \boldsymbol{\epsilon})\, d\mathbf{z} \\
&= \int_{\mathbf{z}} \prod_j p_{\boldsymbol{\theta}}(\mathbf{x}_j|\mathbf{pa}_j) \prod_j p_{\boldsymbol{\theta}}(\mathbf{z}_j|\mathbf{pa}_j, \boldsymbol{\epsilon}_j) \prod_j p(\boldsymbol{\epsilon}_j)\, d\mathbf{z} \\
&= \prod_j p_{\boldsymbol{\theta}}(\mathbf{x}_j|\mathbf{pa}_j) \prod_j p(\boldsymbol{\epsilon}_j) \int_{\mathbf{z}} \prod_j p_{\boldsymbol{\theta}}(\mathbf{z}_j|\mathbf{pa}_j, \boldsymbol{\epsilon}_j)\, d\mathbf{z} \\
&= \prod_j p_{\boldsymbol{\theta}}(\mathbf{x}_j|\mathbf{pa}_j) \prod_j p(\boldsymbol{\epsilon}_j) \\
&\quad \text{where} \quad \mathbf{z}_k = g_k(\mathbf{pa}_k, \boldsymbol{\epsilon}_k, \boldsymbol{\theta})
\end{aligned} \tag{8}$$

In the last step of eq. (8), the inputs $\mathbf{pa}_j$ to the factors of observed variables $p_{\boldsymbol{\theta}}(\mathbf{x}_j|\mathbf{pa}_j)$ are defined in terms of functions $\mathbf{z}_k = g_k(.)$, whose values are all recursively computed from auxiliary variables $\boldsymbol{\epsilon}$.

3.2. Approaches to DNCPs

There are a few basic approaches to transforming CP of a latent variable $\mathbf{z}_j$ to a DNCP:

1. Tractable and differentiable inverse CDF. In this case, let $\epsilon_j \sim \mathcal{U}(0, 1)$, and let $g_j(\mathbf{z}_j, \mathbf{pa}_j, \boldsymbol{\theta}) = F^{-1}(\mathbf{z}_j|\mathbf{pa}_j; \boldsymbol{\theta})$ be the inverse CDF of the conditional distribution. Examples: Exponential, Cauchy, Logistic, Rayleigh, Pareto, Weibull, Reciprocal, Gompertz, Gumbel and Erlang distributions.

2. For any "location-scale" family of distributions (with differentiable log-PDF) we can choose the standard distribution (with location $= 0$, scale $= 1$) as the auxiliary variable $\boldsymbol{\epsilon}_j$, and let $g_j(.) = \text{location} + \text{scale} \cdot \boldsymbol{\epsilon}_j$. Examples: Gaussian, Uniform, Laplace, Elliptical, Student's t, Logistic and Triangular distributions.

3. Composition: It is often possible to express variables as functions of component variables with different distributions. Examples: Log-Normal (exponentiation of normally distributed variable), Gamma (a sum over exponentially distributed variables), Beta distribution, Chi-Squared, F distribution and Dirichlet distributions.

When the distribution is not in the families above, accurate differentiable approximations to the inverse CDF can be constructed, e.g. based on polynomials, with time complexity comparable to the CP (see e.g. (Devroye, 1986) for some methods).

For the exact approaches above, the CP and DNCP forms have equal time complexities. In practice, the difference in CPU time depends on the relative complexity of computing derivatives of $\log p_{\boldsymbol{\theta}}(\mathbf{z}_j|\mathbf{pa}_j)$ versus computing $g_j(.)$ and derivatives of $\log p(\epsilon_j)$, which can be easily verified to be similar in most cases below. Iterations with the DNCP form were slightly faster in our experiments.

3.3. DNCP and neural networks

It is instructive to interpret the DNCP form of latent variables as "hidden units" of a neural network. The network of hidden units together form a neural network with inserted noise $\boldsymbol{\epsilon}$, which we can differentiate efficiently using the backpropagation algorithm (Rumelhart et al., 1986).

There has been recent increase in popularity of deep neural networks with stochastic hidden units (e.g. (Krizhevsky et al., 2012; Goodfellow et al., 2013; Bengio, 2013)). Often, the parameters $\boldsymbol{\theta}$ of such neural networks are optimized towards maximum-likelihood objectives. In that case, the neural network can be interpreted as a probabilistic model $\log p_{\boldsymbol{\theta}}(\mathbf{t}|\mathbf{x}, \boldsymbol{\epsilon})$ computing a conditional distribution over some target variable $\mathbf{t}$ (e.g. classes) given some input $\mathbf{x}$. In (Bengio & Thibodeau-Laufer, 2013), stochastic hidden units are used for learning the parameters of a Markov chain transition operator that samples from the data distribution.

For example, in (Hinton et al., 2012) a 'dropout' regularization method is introduced where (in its basic ver-

sion) the activation of hidden units z_j is computed as $z_j = \epsilon_j \cdot f(\mathbf{pa}_j)$ with $\epsilon_j \sim p(\epsilon_j) = \text{Bernoulli}(0.5)$, and where the parameters are learned by following the gradient of the log-likelihood lower bound: $\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\epsilon}} \left[\log p_{\boldsymbol{\theta}}(\mathbf{t}^{(i)} | \mathbf{x}^{(i)}, \boldsymbol{\epsilon}) \right]$; this gradient can sometimes be computed exactly (Maaten et al., 2013) and can otherwise be approximated with a Monte Carlo estimate (Hinton et al., 2012). The two parameterizations explained in section 3.1 offer us a useful new perspective on 'dropout'. A 'dropout' hidden unit (together with its injected noise $\boldsymbol{\epsilon}$) can be seen as the DNCP of latent random variables, whose CP is $z_j | \mathbf{pa}_j \sim p_{\boldsymbol{\theta}}(\mathbf{z}_j = \epsilon_j \cdot f(\mathbf{pa}_j) | \mathbf{pa}_j))$. A practical implication is that 'dropout'-type neural networks can therefore be interpreted and treated as hierarchical Bayes nets, which opens the door to alternative approaches to learning the parameters, such as Monte Carlo EM or variational methods.

While 'dropout' is designed as a regularization method, other work on stochastic neural networks exploit the power of stochastic hidden units for generative modeling, e.g. (Frey & Hinton, 1999; Rezende et al., 2014; Tang & Salakhutdinov, 2013) applying (partially) MCMC or (partically) factorized variational approaches to modelling the posterior. As we will see in sections 4 and 6, the choice of parameterization has a large impact on the posterior dependencies and the efficiency of posterior inference. However, current publications lack a good justification for their choice of parameterization. The analysis in section 4 offers some important insight in where the centered or non-centered parameterizations of such networks are more appropriate.

3.4. A differentiable MC likelihood estimator

We showed that many hierarchical continuous latent-variable models can be transformed into a DNCP $p_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\epsilon})$, where all latent variables (the introduced auxiliary variables $\boldsymbol{\epsilon}$) are root nodes (see eq. (8)). This has an important implication for learning since (contrary to a CP) the DNCP can be used to form a differentiable Monte Carlo estimator of the marginal likelihood:

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) \simeq \log \frac{1}{L} \sum_{l=1}^{L} \prod_{j} p_{\boldsymbol{\theta}}(\mathbf{x}_j | \mathbf{pa}_j^{(l)})$$

where the parents $\mathbf{pa}_j^{(l)}$ of the observed variables are either root nodes or functions of root nodes whose values are sampled from their marginal: $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$. This MC estimator can be differentiated w.r.t. $\boldsymbol{\theta}$ to obtain an MC estimate of the log-likelihood gradient $\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})$, which can be plugged into stochastic optimization methods such as Adagrad for approximate ML or MAP. When performed one datapoint at a time, we arrive at our on-line Maximum Monte Carlo Likelihood (MMCL) algorithm.

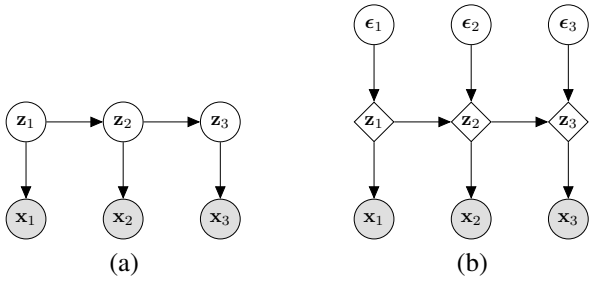


Figure 2. **(a)** An illustrative hierarchical model in its centered parameterization (CP). **(b)** The differentiable non-centered parameterization (DNCP), where $\mathbf{z}_1 = g_1(\boldsymbol{\epsilon}_1, \boldsymbol{\theta})$, $\mathbf{z}_2 = g_2(\mathbf{z}_1, \boldsymbol{\epsilon}_2, \boldsymbol{\theta})$ and $\mathbf{z}_3 = g_3(\mathbf{z}_2, \boldsymbol{\epsilon}_3, \boldsymbol{\theta})$, with auxiliary latent variables $\boldsymbol{\epsilon}_k \sim p_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}_k)$. The DNCP exposes a neural network within the hierarchical model, which we can differentiate efficiently using back-propagation.

Table 1. Limiting behaviour of squared correlations between z and its parent y_i when z is in the centered (CP) and non-centered (DNCP) parameterizaton.

	$\rho^2_{y_i,z}$ (CP)	$\rho^2_{y_i,e}$ (DNCP)
$\lim_{\sigma \to 0}$	1	0
$\lim_{\sigma \to +\infty}$	0	$\frac{\beta w_i^2}{\beta w_i^2 + \alpha}$
$\lim_{\beta \to 0}$	$\frac{w_i^2}{w_i^2 - \alpha\sigma^2}$	0
$\lim_{\beta \to -\infty}$	0	1
$\lim_{\alpha \to 0}$	$\frac{1}{1 - \beta\sigma^2}$	$\frac{\beta\sigma^2}{\beta\sigma^2 - 1}$
$\lim_{\alpha \to -\infty}$	0	0

4. Effects of parameterizations on posterior dependencies

What is the effect of the proposed reparameterization on the efficiency of inference? If the latent variables have linear-Gaussian conditional distributions, we can use the metric of squared correlation between the latent variable and any of its children in their posterior distribution. If after reparameterization the squared correlation is decreased, then in general this will also result in more efficient inference.

For non-linear Gaussian conditional distributions, the log-PDF can be locally approximated as a linear-Gaussian using a second-order Taylor expansion. Results derived for the linear case can therefore also be applied to the non-linear case; the correlation computed using this approximation is a local dependency between the two variables.

Denote by z a scalar latent variable we are going to reparameterize, and by $\mathbf{y}$ its parents, where y_i is one of the parents. The log-PDF of the corresponding conditional dis-

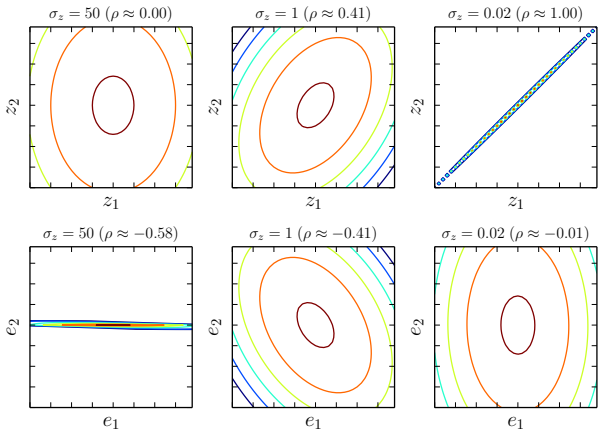


Figure 3. Plots of the log-posteriors of the illustrative linear-Gaussian model discussed in sec. 4.4. Columns: different choices of σ_z, ranging from a low prior dependency ($\sigma_z = 50$) to a high prior dependency ($\sigma_z = 0.02$). First row: CP form. Second row: DNCP form. The posterior correlation ρ between the variables is also displayed. In the original form a larger prior dependency leads to a larger posterior dependency (see top row). The dependency in the *DNCP* posterior is inversely related to the prior dependency between z_1 and z_2 (bottom row).

tribution is

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(z|\mathbf{y}) &= \log \mathcal{N}(z|\mathbf{w}^T\mathbf{y} + b, \sigma^2) \\ &= -(z - \mathbf{w}^T\mathbf{y} - b)^2/(2\sigma^2) + \text{constant}\end{aligned}$$

A reparameterization of z using an auxiliary variable ϵ is $z = g(.) = (\mathbf{w}^T\mathbf{y} + b) + \sigma\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$. With (7) it can be confirmed that this change of variables is correct:

$$\begin{aligned}p_{\boldsymbol{\theta}}(z|\mathbf{y}) \cdot \left|\frac{\partial z}{\partial \epsilon}\right| &= p_{\boldsymbol{\theta}}(z = g(.)|\mathbf{y}) \cdot \left|\frac{\partial z}{\partial \epsilon}\right| \\ &= \mathcal{N}(\mathbf{w}^T\mathbf{y} + b + \sigma\epsilon|\mathbf{w}^T\mathbf{y} + b, \sigma^2) \cdot \sigma_z \\ &= -\exp(\epsilon^2/2)/\sqrt{2\pi} = \mathcal{N}(0, 1) \\ &= p(\epsilon) \end{aligned} \tag{9}$$

First we will derive expressions for the squared correlations between z and its parents, for the CP and DNCP case, and subsequently show how they relate.

The covariance C between two jointly Gaussian distributed variables A and B equals the negative inverse of the Hessian matrix of the log-joint PDF:

$$C = \begin{pmatrix} \sigma_A^2 & \sigma_{AB}^2 \\ \sigma_{AB}^2 & \sigma_B^2 \end{pmatrix} = -\mathbf{H}^{-1} = \frac{1}{det(\mathbf{H})} \begin{pmatrix} -H_B & H_{AB} \\ H_{AB} & -H_A \end{pmatrix}$$

The correlation ρ between two jointly Gaussian distributed variables A and B is given by: $\rho = \sigma_{AB}^2/(\sigma_A\sigma_B)$. Using the equation above, the squared correlation can be computed from the elements of the Hessian matrix:

$$\begin{aligned}\rho^2 &= (\sigma_{AB}^2)^2/(\sigma_A^2\sigma_B^2) \\ &= (H_{AB}/det(\mathbf{H}))^2/((-H_A/det(\mathbf{H}))(-H_B/det(\mathbf{H})) \\ &= H_{AB}^2/(H_A H_B)\end{aligned} \tag{10}$$

Important to note is that derivatives of the log-posterior w.r.t. the latent variables are equal to the derivatives of log-joint w.r.t. the latent variables, therefore,

$$\mathbf{H} = \nabla_{\mathbf{z}}\nabla_{\mathbf{z}}^T \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}}\nabla_{\mathbf{z}}^T \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$$

The following shorthand notation is used in this section:

$$\begin{aligned}L &= \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) \quad \text{(sum of all factors)} \\ z &= \text{the variable to be reparameterized} \\ \mathbf{y} &= z\text{'s parents} \\ L^{(z)} &= \log p_{\boldsymbol{\theta}}(z|\mathbf{y}) \quad (z\text{'s factor}) \\ L^{(\setminus z)} &= L - L^{(z)} \quad \text{(all factors minus } z\text{'s factor)} \\ L^{(z\rightarrow)} &= \text{the factors of } z\text{'s children} \\ \alpha &= \frac{\partial^2 L^{(\setminus z)}}{\partial y_i \partial y_i} \\ \beta &= \frac{\partial^2 L^{(z\rightarrow)}}{\partial z \partial z}\end{aligned}$$

4.1. Squared Correlations

4.1.1. Centered case

In the CP case, the relevant Hessian elements are as follows:

$$\begin{aligned}H_{y_i y_i} &= \frac{\partial^2 L}{\partial y_i \partial y_i} = \alpha + \frac{\partial^2 L^{(z)}}{\partial y_i \partial y_i} = \alpha - w_i^2/\sigma^2 \\ H_{zz} &= \frac{\partial^2 L}{\partial z \partial z} = \beta + \frac{\partial^2 L^{(z)}}{\partial z \partial z} = \beta - 1/\sigma^2 \\ H_{y_i z} &= \frac{\partial^2 L}{\partial y_i \partial z} = \frac{\partial^2 L^{(z)}}{\partial y_i \partial z} = w_i/\sigma^2\end{aligned} \tag{11}$$

Therefore, using eq. (10), the squared correlation between y_i and z is:

$$\rho_{y_i,z}^2 = \frac{(H_{y_i z})^2}{H_{y_i y_i} H_{zz}} = \frac{w_i^2/\sigma^4}{(\alpha - w_i^2/\sigma^2)(\beta - 1/\sigma^2)} \tag{12}$$

4.1.2. NON-CENTERED CASE

In the DNCP case, the Hessian elements are:

$$\begin{aligned} H_{y_i y_i} &= \frac{\partial^2 L}{\partial y_i \partial y_i} = \alpha + \frac{\partial}{\partial y_i}\frac{\partial L^{(z\rightarrow)}}{\partial y_i} \\ &= \alpha + \frac{\partial}{\partial y_i}\left(w_i \frac{\partial L^{(z\rightarrow)}}{\partial z}\right) = \alpha + w_i^2\beta \\ H_{\epsilon\epsilon} &= \frac{\partial^2 L}{\partial \epsilon \partial \epsilon} = \frac{\partial^2 L^{(z\rightarrow)}}{\partial \epsilon \partial \epsilon} + \frac{\partial^2 \log p(\boldsymbol{\epsilon})}{\partial \epsilon \partial \epsilon} = \sigma^2\beta - 1 \\ H_{y_i\epsilon} &= \frac{\partial^2 L}{\partial y_i \partial \epsilon} = \sigma w_i \beta \end{aligned} \tag{13}$$

The squared correlation between y_i and ϵ is therefore:

$$\rho^2_{y_i,\epsilon} = \frac{(H_{y_i\epsilon})^2}{H_{y_i y_i} H_{\epsilon\epsilon}} = \frac{\sigma^2 w_i^2 \beta^2}{(\alpha + w_i^2\beta)(\sigma^2\beta - 1)} \tag{14}$$

4.2. Correlation inequality

We can now compare the squared correlation, between z and some parent y_i, before and after the reparameterization. Assuming $\alpha < 0$ and $\beta < 0$ (i.e. $L^{(\setminus z)}$ and $L^{(z\rightarrow)}$ are concave, e.g. exponential families):

$$\begin{aligned} \rho^2_{y_i,z} &> \rho^2_{y_i,\epsilon} \\ \frac{w_i^2/\sigma^4}{(\alpha - w_i^2/\sigma^2)(\beta - 1/\sigma^2)} &> \frac{\sigma^2 w_i^2 \beta^2}{(\alpha + w_i^2\beta)(\sigma^2\beta - 1)} \\ \frac{w_i^2/\sigma^4}{(\alpha - w_i^2/\sigma^2)(\beta - 1/\sigma^2)} &> \frac{w_i^2 \beta^2}{(\alpha + w_i^2\beta)(\beta - 1/\sigma^2)} \\ \frac{1/\sigma^4}{(\alpha - w_i^2/\sigma^2)} &> \frac{\beta^2}{(\alpha + w_i^2\beta)} \\ \sigma^{-2} &> -\beta \end{aligned} \tag{15}$$

Thus we have shown the surprising fact that the correlation inequality takes on an extremely simple form where the parent-dependent values α and w_i play no role; the inequality only depends on two properties of z: the relative strenghts of σ (its noisiness) and β (its influence on children's factors). Informally speaking, if the noisiness of z's conditional distribution is large enough compared to other factors' dependencies on z, then the reparameterized form is beneficial for inference.

4.3. A beauty-and-beast pair

Additional insight into the properties of the CP and DNCP can be gained by taking the limits of the squared correlations (12) and (14). Limiting behaviour of these correlations is shown in table 1. As becomes clear in these limits, the CP and DNCP often form a beauty-and-beast pair: when posterior correlations are high in one parameterization, they are low in the other. This is especially true in the limits of $\sigma \rightarrow 0$ and $\beta \rightarrow -\infty$, where squared correlations converge to either 0 or 1, such that posterior inference will be extremely inefficient in either CP or DNCP, but efficient in the other. This difference in shapes of the log-posterior is illustrated in figure 3.

Table 2. Effective Sample Size (ESS) for different choices of latent-variable variance σ_z, and for different samplers, after taking 4000 samples. Shown are the results for HMC samplers using the *CP* and *DNCP* parameterizations, as well as a robust HMC sampler.

$\log \sigma_z$	CP	DNCP	ROBUST
-5	2	305	640
-4.5	26	348	498
-4	10	570	686
-3.5	225	417	624
-3	386	569	596
-2.5	542	608	900
-2	406	972	935
-1.5	672	1078	918
-1	1460	1600	1082

4.4. Example: Simple Linear Dynamical System

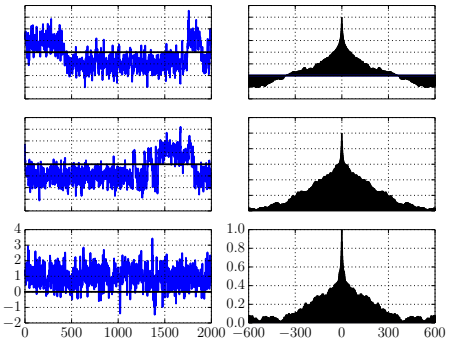
Take a simple model with scalar latent variables z_1 and z_2, and scalar observed variables x_1 and x_2. The joint PDF is defined as $p(x_1, x_2, z_1, z_2) = p(z_1)p(x_1|z_1)p(z_2|z_1)p(x_2|z_2)$, where $p(z_1) = \mathcal{N}(0, 1)$, $p(x_1|z_1) = \mathcal{N}(z_1, \sigma_x^2)$, $p(z_2|z_1) = \mathcal{N}(z_1, \sigma_z^2)$ and $p(x_2|z_2) = \mathcal{N}(z_2, \sigma_x^2)$. Note that the parameter σ_z determines the dependency between the latent variables, and σ_x determines the dependency between latent and observed variables.

We reparameterize z_2 such that it is conditionally deterministic given a new auxiliary variable ϵ_2. Let $p(\epsilon_2) = \mathcal{N}(0, 1)$. let $z_2 = g_2(z_1, \epsilon_2, \sigma_z) = z_1 + \sigma_z \cdot \epsilon_2$ and let $\epsilon_1 = z_1$. See figure 3 for plots of the original and auxiliary posterior log-PDFs, for different choices of σ_z, along with the resulting posterior correlation ρ.

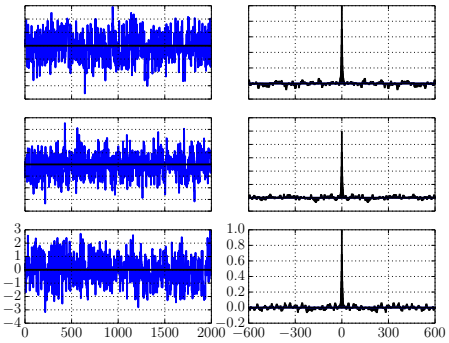
For what choice of parameters does the reparameterization yield smaller posterior correlation? We use equation (15) and plug in $\sigma \leftarrow \sigma_z$ and $-\beta \leftarrow \sigma_x^{-2}$, which results in:

$$\rho^2_{z_1,z_2} > \rho^2_{\epsilon_1,\epsilon_2} \quad \Rightarrow \quad \sigma_z^2 < \sigma_x^2$$

i.e. the posterior correlation in DNCP form $\rho^2_{\epsilon_1,\epsilon_2}$ is smaller when the latent-variable noise parameter σ_z^2 is smaller than the oberved-variable noise parameter σ_x^2. Less formally, this means that the DNCP is preferred when the latent variable is more strongly coupled to the data (likelihood) then to its parents.



(a) Centered Parameterization (CP)



(b) Differentiable Non-Centered Parameterization (DNCP)

Figure 4. Auto-correlation of HMC samples of the latent variables for a DBN in two different parameterizations. Left on each figure are shown 2000 subsequent HMC samples of three randomly chosen variables in the dynamic Bayesian network model. On the right are shown the corresponding HMC sample auto-correlation graphs. The DNCP resulted in much lower posterior dependencies and a dramatic drop in HMC sample auto-correlation.

5. Related work

This is, to the best of our knowledge, the first work to investigate the implications of the different differentiable non-centered parameterizations on the efficiency of gradient-based inference. However, the topic of centered vs non-centered parameterizations has been investigated for efficient (non-gradient based) Gibbs Sampling in work by Papaspiliopoulos et al. (2003; 2007), which also discusses some strategies for constructing parameterization for those cases. There have been some publications for parameterizations of specific models; (Gelfand et al., 1995), for example, discusses parameterizations of mixed models, and (Meng & Van Dyk, 1998) investigate several rules for choosing an appropriate parameterization for mixed-effects models for faster EM. In the special case where Gibbs sampling is tractable, efficient sampling is possible by interleaving between centered and non-centered parameterizations, as was shown in (Yu & Meng, 2011).

Auxiliary variables are used for data augmentation (see (Van Dyk & Meng, 2001) or slice sampling (Neal, 2003)) where, in contrast with our method, sampling is performed in a higher-dimensional augmented space. Auxiliary variables are used in a similar form under the name exogenous variables in Structural Causal Models (SCMs) (Pearl, 2000). In SCMs the functional form of exogenous variables is more restricted than our auxiliary variables. The concept of conditionally deterministic variables has been used earlier in e.g. (Cobb & Shenoy, 2005), although not as a tool for efficient inference in general Bayesian networks with continuous latent variables. Recently, (Raiko et al., 2012) analyzed the elements of the Hessian w.r.t. the parameters in neural network context.

The differentiable reparameterization of latent variables in this paper was introduced earlier in (Kingma, 2013) and independently in (Bengio, 2013), but these publications lack a theoretic analysis of the impact on the efficiency of inference. In (Kingma & Welling, 2013), the reparameterization trick was used in an efficient algorithm for stochastic variational inference and learning.

6. Experiments

6.1. Nonlinear DBN

From the derived posterior correlations in the previous sections we can conclude that depending on the parameters of the model, posterior sampling can be extremely inefficient in one parameterization while it is efficient in the other. When the parameters are known, one can choose the best parameterization (w.r.t. posterior correlations) based on the correlation inequality (15).

In practice, model parameters are often subject to change, e.g. when optimizing the parameters with Monte Carlo EM; in these situations where there is uncertainty over the value of the model parameters, it is impossible to choose the best parameterization in advance. The "beauty-beast" duality from section 4.3 suggests a solution in the form of a very simple sampling strategy: mix the two parameterizations. Let $Q_{CP}(\mathbf{z}'|\mathbf{z})$ be the MCMC/HMC proposal distribution based on $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ (the CP), and let $Q_{DNCP}(\mathbf{z}'|\mathbf{z})$ be the proposal distribution based on $p_{\boldsymbol{\theta}}(\boldsymbol{\epsilon}|\mathbf{x})$ (the DNCP). Then the new MCMC proposal distribution based on the mixture is:

$$Q(\mathbf{z}'|\mathbf{z}) = \rho \cdot Q_{CP}(\mathbf{z}'|\mathbf{z}) + (1-\rho) \cdot Q_{DNCP}(\mathbf{z}'|\mathbf{z}) \tag{16}$$

where we use $\rho = 0.5$ in experiments. The mixing efficiency might be half that of the oracle solution (where the

optimal parameterization is known), nonetheless when taking into account the uncertainty over the parameters, the expected efficiency of the mixture proposal can be better than a single parameterization chosen ad hoc.

We applied a Hybrid Monte Carlo (HMC) sampler to a Dynamic Bayesian Network (DBN) with nonlinear transition probabilities with the same structure as the illustrative model in figure 2. The prior and conditional probabilities are: $\mathbf{z}_1 \sim \mathcal{N}(0, \mathbf{I})$, $\mathbf{z}_t|\mathbf{z}_{t-1} \sim \mathcal{N}(tanh(\mathbf{W}_z\mathbf{z}_{t-1} + \mathbf{b}_z), \sigma_z^2\mathbf{I})$ and $\mathbf{x}_t|\mathbf{z}_t \sim$ Bernoulli$(sigmoid(\mathbf{W}_x\mathbf{z}_{t-1}))$. The parameters were intialized randomly by sampling from $\mathcal{N}(0, \mathbf{I})$. Based on the derived limiting behaviour (see table 1, we can expect that such a network in CP can have very large posterior correlations if the variance of the latent variables σ_z^2 is very small, resulting in slow sampling.

To validate this result, we performed HMC inference with different values of σ_z^2, sampling the latent variables while holding the parameters fixed. For HMC we used 10 leapfrog steps per sample, and the stepsize was automatically adjusted while sampling to obtain a HMC acceptance rate of around 0.9. At each sampling run, the first 1000 HMC samples were thrown away (burn-in); the subsequent 4000 HMC samples were kept. To estimate the efficiency of sampling, we computed the effective sample size (ESS); see e.g. (Kass et al., 1998) for a discussion on ESS.

Results. See table 2 and figure 4 for results. It is clear that the choice of parameterization has a large effect on posterior dependencies and the efficiency of inference. Sampling was very inefficient for small values of σ_z in the CP, which can be understood from the limiting behaviour in table 1.

6.2. Generative multilayer neural net

As explained in section 3.4, a hierarchical model in DNCP form can be learned using a MC likelihood estimator which can be differentiated and optimized w.r.t. the parameters $\boldsymbol{\theta}$. We compare this Maximum Monte Carlo Likelihood (MMCL) method with the MCEM method for learning the parameters of a 4-layer hierarchical model of the MNIST dataset, where $\mathbf{x}|\mathbf{z}_3 \sim$ Bernoulli$(sigmoid(\mathbf{W}_x\mathbf{z}_3 + \mathbf{b}_x))$ and $\mathbf{z}_t|\mathbf{z}_{t-1} \sim \mathcal{N}(\tanh(\mathbf{W}_i\mathbf{z}_{t-1} + \mathbf{b}_i), \sigma_{z_t}^2\mathbf{I})$. For MCEM, we used HMC with 10 leapfrog steps followed by a weight update using Adagrad (Duchi et al., 2010). For MMCL, we used $L \in \{10, 100, 500\}$. We observed that DNCP was a better parameterization than CP in this case, in terms of fast mixing. However, even in the DNCP, HMC mixed very slowly when the dimensionality of latent space become too high. For this reason, $\mathbf{z}_1$ and $\mathbf{z}_2$ were given a dimensionality of 3, while $\mathbf{z}_3$ was 100-dimensional but noiseless ($\sigma_{z_1}^2 = 0$) such that only $\mathbf{z}_3$ and $\mathbf{z}_2$ are random variables that require posterior inference by sampling. The model was trained on a small (1000 datapoints) and large (50000 datapoints) version of the MNIST dataset.

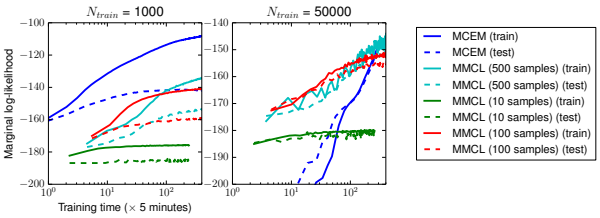


Figure 5. Performance of MMCL versus MCEM in terms of the marginal likelihood, when learning the parameters of a generative multilayer neural network (see section 6.2).

Results. We compared train- and testset marginal likelihood. See figure 5 for experimental results. As was expected, MCEM attains asymptotically better results. However, despite its simplicity, the on-line nature of MMCL means it scales better to large datasets, and (contrary to MCEM) is trivial to implement.

7. Conclusion

We have shown how Bayesian networks with continuous latent variables and generative neural networks are related through two different parameterizations of the latent variables: CP and DNCP. A key result is that the differentiable non-centered parameterization (DNCP) of a latent variable is preferred, in terms of its effect on decreased posterior correlations, when the variable is more strongly linked to its parents than its children. Through theoretical analysis we have also shown that the two parameterizations are complementary to each other: when posterior correlations are large in one form, they are small in the other. We have also illustrated that this theoretical result can be exploited in practice by designing a MCMC strategy that mixes between both parameterizations, making it robust to situations where MCMC can otherwise be inefficient.

Acknowledgments

The authors thank the reviewers for their excellent feedback and Joris Mooij, Ted Meeds and Taco Cohen for invaluable discussions and input.

References

Bengio, Yoshua. Estimating or propagating gradients through stochastic neurons. *arXiv preprint arXiv:1305.2982*, 2013.

Bengio, Yoshua and Thibodeau-Laufer, Éric. Deep generative stochastic networks trainable by backprop. *arXiv preprint arXiv:1306.1091*, 2013.

Cobb, Barry R and Shenoy, Prakash P. Nonlinear deterministic relationships in Bayesian networks. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 27–38. Springer, 2005.

Devroye, Luc. Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*, pp. 260–265. ACM, 1986.

Duane, Simon, Kennedy, Anthony D, Pendleton, Brian J, and Roweth, Duncan. Hybrid Monte Carlo. *Physics letters B*, 195 (2):216–222, 1987.

Duchi, John, Hazan, Elad, and Singer, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2010.

Frey, Brendan J and Hinton, Geoffrey E. Variational learning in nonlinear Gaussian belief networks. *Neural Computation*, 11 (1):193–213, 1999.

Gelfand, AE, Sahu, SK, and Carlin, BP. Efficient parameterisations for normal linear mixed models. *Biometrika*, 82:479–488, 1995.

Goodfellow, Ian J, Warde-Farley, David, Mirza, Mehdi, Courville, Aaron, and Bengio, Yoshua. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Hinton, Geoffrey E, Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

Hoffman, Matthew D and Gelman, Andrew. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *arXiv preprint arXiv:1111.4246*, 2011.

Kass, Robert E, Carlin, Bradley P, Gelman, Andrew, and Neal, Radford M. Markov chain Monte Carlo in practice: A roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

Kingma, Diederik P. Fast gradient-based inference with continuous latent variable models in auxiliary form. *arXiv preprint arXiv:1306.0733*, 2013.

Kingma, Diederik P and Welling, Max. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoff. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pp. 1106–1114, 2012.

Maaten, Laurens, Chen, Minmin, Tyree, Stephen, and Weinberger, Kilian Q. Learning with marginalized corrupted features. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 410–418, 2013.

Meng, X-L and Van Dyk, David. Fast EM-type implementations for mixed effects models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(3):559–578, 1998.

Minka, Thomas P. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 362–369. Morgan Kaufmann Publishers Inc., 2001.

Neal, Radford M. Probabilistic inference using Markov Chain Monte Carlo methods. 1993.

Neal, Radford M. Slice sampling. *Annals of statistics*, pp. 705–741, 2003.

Papaspiliopoulos, Omiros, Roberts, Gareth O, and Sköld, Martin. Non-centered parameterisations for hierarchical models and data augmentation. In *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*, pp. 307. Oxford University Press, USA, 2003.

Papaspiliopoulos, Omiros, Roberts, Gareth O, and Sköld, Martin. A general framework for the parametrization of hierarchical models. *Statistical Science*, pp. 59–73, 2007.

Pearl, Judea. *Reverend Bayes on inference engines: A distributed hierarchical approach*. Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982.

Pearl, Judea. *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press, 2000.

Raiko, Tapani, Valpola, Harri, and LeCun, Yann. Deep learning made easier by linear transformations in perceptrons. In *International Conference on Artificial Intelligence and Statistics*, pp. 924–932, 2012.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic back-propagation and variational inference in deep latent gaussian models. *arXiv preprint arXiv:1401.4082*, 2014.

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.

Tang, Yichuan and Salakhutdinov, Ruslan. Learning stochastic feedforward neural networks. In *Advances in Neural Information Processing Systems*, pp. 530–538, 2013.

Van Dyk, David A and Meng, Xiao-Li. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10 (1), 2001.

Wei, Greg CG and Tanner, Martin A. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85 (411):699–704, 1990.

Yu, Yaming and Meng, Xiao-Li. To Center or Not to Center: That Is Not the Question–An Ancillarity-Sufficiency Interweaving Strategy (ASIS) for Boosting MCMC Efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570, 2011. doi: 10.1198/jcgs.2011.203main. URL `http://amstat.tandfonline.com/doi/abs/10.1198/jcgs.2011.203main`.