Appendices: Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Danilo Jimenez Rezende DANILOR@GOOGLE.COM
Shakir Mohamed SHAKIR@GOOGLE.COM
Daan Wierstra DAANW@GOOGLE.COM
Google DeepMind, London, United Kingdom

A. Additional Model Details

In equation (6) we showed an alternative form of the joint log likelihood that explicitly separates the deterministic and stochastic parts of the generative model and corroborates the view that the generative model works by applying a complex non-linear transformation to a spherical Gaussian distribution $\mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{I})$ such that the transformed distribution best matches the empirical distribution. We provide more details on this view here for clarity.

From the model description in equations (3) and (4), we can interpret the variables $\mathbf{h}_l$ as deterministic functions of the noise variables $\boldsymbol{\xi}_l$. This can be formally introduced as a co-ordinate transformation of the probability density in equation (5): we perform a change of coordinates $\mathbf{h}_l \to \boldsymbol{\xi}_l$. The density of the transformed variables $\boldsymbol{\xi}_l$ can be expressed in terms of the density (5) times the determinant of the Jacobian of the transformation $p(\boldsymbol{\xi}_l) = p(\mathbf{h}_l(\boldsymbol{\xi}_l))|\frac{\partial \mathbf{h}_l}{\partial \boldsymbol{\xi}_l}|$. Since the co-ordinate transformation is linear we have $|\frac{\partial \mathbf{h}_l}{\partial \boldsymbol{\xi}_l}| = |\mathbf{G}_l|$ and the distribution of $\boldsymbol{\xi}_l$ is obtained as follows:

$$
\begin{aligned}
p(\boldsymbol{\xi}_l) &= p(\mathbf{h}_l(\boldsymbol{\xi}_l))|\frac{\partial \mathbf{h}_l}{\partial \boldsymbol{\xi}_l}| \\
p(\boldsymbol{\xi}_l) &= p(\mathbf{h}_L)|\mathbf{G}_L|\prod_{l=1}^{L-1}|\mathbf{G}_l| p_l(\mathbf{h}_l|\mathbf{h}_{l+1}) = \prod_{l=1}^{L}|\mathbf{G}_l||\mathbf{S}_l|^{-\frac{1}{2}}\mathcal{N}(\boldsymbol{\xi}_l) \\
&= \prod_{l=1}^{L}|\mathbf{G}_l||\mathbf{G}_l\mathbf{G}_l^T|^{-\frac{1}{2}}\mathcal{N}(\boldsymbol{\xi}_l|\mathbf{0},\mathbf{I}) = \prod_{l=1}^{L}\mathcal{N}(\boldsymbol{\xi}_l|\mathbf{0},\mathbf{I}). \quad (22)
\end{aligned}
$$

Combining this equation with the distribution of the visible layer we obtain equation (6).

A.1. Examples

Below we provide simple, explicit examples of generative and recognition models.

In the case of a two-layer model the activation $\mathbf{h}_1(\boldsymbol{\xi}_{1,2})$ in equation (6) can be explicitly written as

$$\mathbf{h}_1(\boldsymbol{\xi}_{1,2}) \quad = \mathbf{W}_1 f(\mathbf{G}_2\boldsymbol{\xi}_2) + \mathbf{G}_1\boldsymbol{\xi}_1 + \mathbf{b}_1. \quad (23)$$

Similarly, a simple recognition model consists of a single deterministic layer and a stochastic Gaussian layer with the rank-one covariance structure and is constructed as:

$$
\begin{aligned}
q(\boldsymbol{\xi}_l|\mathbf{v}) &= \mathcal{N}\left(\boldsymbol{\xi}_l|\boldsymbol{\mu}; (\mathrm{diag}(\mathbf{d}) + \mathbf{u}\mathbf{u}^\top)^{-1}\right) && (24)\\
\boldsymbol{\mu} &= \mathbf{W}_\mu \mathbf{z} + \mathbf{b}_\mu && (25)\\
\log \mathbf{d} = \mathbf{W}_d\mathbf{z} + \mathbf{b}_d; &\qquad \mathbf{u} = \mathbf{W}_u\mathbf{z} + \mathbf{b}_u && (26)\\
\mathbf{z} &= f(\mathbf{W}_v\mathbf{v} + \mathbf{b}_v) && (27)
\end{aligned}
$$

where the function f is a rectified linearity (but other non-linearities such as tanh can be used).

B. Proofs for the Gaussian Gradient Identities

Here we review the derivations of Bonnet's and Price's theorems that were presented in section 3.

Theorem B.1 (Bonnet's theorem)**.** *Let $f(\boldsymbol{\xi}) : R^d \mapsto R$ be a integrable and twice differentiable function. The gradient of the expectation of $f(\boldsymbol{\xi})$ under a Gaussian distribution $\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu},\mathbf{C})$ with respect to the mean $\boldsymbol{\mu}$ can be expressed as the expectation of the gradient of $f(\boldsymbol{\xi})$.*

$$\nabla_{\mu_i}\mathbb{E}_{\mathcal{N}(\mu,\mathbf{C})}\left[f(\boldsymbol{\xi})\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[\nabla_{\xi_i} f(\boldsymbol{\xi})\right],$$

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

Proof.

$$\begin{aligned}\nabla_{\mu_i}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[f(\boldsymbol{\xi})\right] &= \int \nabla_{\mu_i}\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu},\mathbf{C})f(\boldsymbol{\xi})d\boldsymbol{\xi} \\ &= -\int \nabla_{\xi_i}\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu},\mathbf{C})f(\boldsymbol{\xi})d\boldsymbol{\xi} \\ &= \left[\int \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu},\mathbf{C})f(\boldsymbol{\xi})d\boldsymbol{\xi}_{\neg i}\right]_{\xi_i=-\infty}^{\xi_i=+\infty} \\ &+ \int \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu},\mathbf{C})\nabla_{\xi_i}f(\boldsymbol{\xi})d\boldsymbol{\xi} \\ &= \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[\nabla_{\xi_i}f(\boldsymbol{\xi})\right], \end{aligned} \tag{28}$$

where we have used the identity

$$\nabla_{\mu_i}\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu},\mathbf{C}) = -\nabla_{\xi_i}\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu},\mathbf{C})$$

in moving from step 1 to 2. From step 2 to 3 we have used the product rule for integrals with the first term evaluating to zero. □

Theorem B.2 (Price's theorem)**.** *Under the same conditions as before. The gradient of the expectation of $f(\boldsymbol{\xi})$ under a Gaussian distribution $\mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{C})$ with respect to the covariance $\mathbf{C}$ can be expressed in terms of the expectation of the Hessian of $f(\boldsymbol{\xi})$ as*

$$\nabla_{C_{i,j}}\mathbb{E}_{\mathcal{N}(\mathbf{0},C)}\left[f(\boldsymbol{\xi})\right] = \frac{1}{2}\mathbb{E}_{\mathcal{N}(\mathbf{0},C)}\left[\nabla_{\xi_i,\xi_j}f(\boldsymbol{\xi})\right]$$

Proof.

$$\begin{aligned}\nabla_{C_{i,j}}\mathbb{E}_{\mathcal{N}(\mathbf{0},\mathbf{C})}\left[f(\boldsymbol{\xi})\right] &= \int \nabla_{C_{i,j}}\mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{C})f(\boldsymbol{\xi})d\boldsymbol{\xi} \\ &= \frac{1}{2}\int \nabla_{\xi_i,\xi_j}\mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{C})f(\boldsymbol{\xi})d\boldsymbol{\xi} \\ &= \frac{1}{2}\int \mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{C})\nabla_{\xi_i,\xi_j}f(\boldsymbol{\xi})d\boldsymbol{\xi} \\ &= \frac{1}{2}\mathbb{E}_{\mathcal{N}(\mathbf{0},\mathbf{C})}\left[\nabla_{\xi_i,\xi_j}f(\boldsymbol{\xi})\right]. \end{aligned} \tag{29}$$

In moving from steps 1 to 2, we have used the identity

$$\nabla_{C_{i,j}}\mathcal{N}(\boldsymbol{\xi}|\mu,\mathbf{C}) = \frac{1}{2}\nabla_{\xi_i,\xi_j}\mathcal{N}(\boldsymbol{\xi}|\mu,\mathbf{C}),$$

which can be verified by taking the derivatives on both sides and comparing the resulting expressions. From step 2 to 3 we have used the product rule for integrals twice. □

C. Deriving Stochastic Back-propagation Rules

In section 3 we described two ways in which to derive stochastic back-propagation rules. We show specific examples and provide some more discussion in this section.

C.1. Using the Product Rule for Integrals

We can derive rules for stochastic back-propagation for many distributions by finding a appropriate non-linear function $B(x;\theta)$ that allows us to express the gradient with respect to the parameters of the distribution as a gradient with respect to the random variable directly. The approach we described in the main text was:

$$\begin{aligned}\nabla_\theta\mathbb{E}_p[f(x)] &= \int \nabla_\theta p(x|\theta)f(x)dx = \int \nabla_x p(x|\theta)B(x)f(x)dx \\ &= \left[B(x)f(x)p(x|\theta)\right]_{supp(x)} - \int p(x|\theta)\nabla_x[B(x)f(x)] \\ &= -\mathbb{E}_{p(x|\theta)}[\nabla_x[B(x)f(x)]] \end{aligned} \tag{30}$$

where we have introduced the non-linear function $B(x;\theta)$ to allow for the transformation of the gradients and have applied the product rule for integrals (rule for integration by parts) to rewrite the integral in two parts in the second line, and the $\text{supp}(x)$ indicates that the term is evaluated at the boundaries of the support. To use this approach, we require that the density we are analysing be zero at the boundaries of the support to ensure that the first term in the second line is zero.

As an alternative, we can also write this differently and find an non-linear function of the form:

$$\nabla_\theta\mathbb{E}_p[f(x)] = -\mathbb{E}_{p(x|\theta)}[B(x)\nabla_x f(x)]. \tag{31}$$

Consider general exponential family distributions of the form:

$$p(x|\theta) = h(x)\exp(\eta(\theta)^\top\phi(x) - A(\theta)) \tag{32}$$

where $h(x)$ is the base measure, θ is the set of mean parameters of the distribution, η is the set of natural parameters, and $A(\theta)$ is the log-partition function. We can express the non-linear function in (30) using these quantities as:

$$B(x) = \frac{[\nabla_\theta\eta(\theta)\phi(x) - \nabla_\theta A(\theta)]}{[\nabla_x\log[h(x)] + \eta(\theta)^T\nabla_x\phi(x)]}. \tag{33}$$

This can be derived for a number of distributions such as the Gaussian, inverse Gamma, Log-Normal, Wald (inverse Gaussian) and other distributions. We show some of these below:

The $B(x;\theta)$ corresponding to the second formulation can also be derived and may be useful in certain situations, requiring the solution of a first order differential equation. This approach of searching for non-linear transformations leads us to the second approach for deriving stochastic back-propagation rules.

Family	θ	$B(x)$
Gaussian	$\begin{pmatrix}\mu\\ \sigma^2\end{pmatrix}$	$\begin{pmatrix}-1\\ \frac{(x-\mu-\sigma)(x-\mu+\sigma)}{2\sigma^2(x-\mu)}\end{pmatrix}$
Inv. Gamma	$\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$	$\begin{pmatrix}\frac{x^2(-\ln x-\Psi(\alpha)+\ln\beta)}{-x(\alpha+1)+\beta}\\ (\frac{x^2}{-x(\alpha+1)+\beta})(-\frac{1}{x}+\frac{\alpha}{\beta})\end{pmatrix}$
Log-Normal	$\begin{pmatrix}\mu\\ \sigma^2\end{pmatrix}$	$\begin{pmatrix}-1\\ \frac{(\ln x-\mu-\sigma)(\ln x-\mu+\sigma)}{2\sigma^2(\ln x-\mu)}\end{pmatrix}$

C.2. Using Alternative Coordinate Transformations

There are many distributions outside the exponential family that we would like to consider using. A simpler approach is to search for a co-ordinate transformation that allows us to separate the deterministic and stochastic parts of the distribution. We described the case of the Gaussian in section 3. Other distributions also have this property. As an example, consider the Levy distribution (which is a special case of the inverse Gamma considered above). Due to the self-similarity property of this distribution, if we draw X from a Levy distribution with known parameters $X \sim \text{Levy}(\mu, \lambda)$, we can obtain any other Levy distribution by rescaling and shifting this base distribution: $kX + b \sim \text{Levy}(k\mu + b, kc)$.

Many other distributions hold this property, allowing stochastic back-propagation rules to be determined for distributions such as the Student's t-distribution, Logistic distribution, the class of stable distributions and the class of generalised extreme value distributions (GEV). Examples of co-ordinate transformations $T(\cdot)$ and the resulsting distributions are shown below for variates X drawn from the standard distribution listed in the first column.

Std Distr.	T($\cdot$)	Gen. Distr.
GEV$(\mu, \sigma, 0)$	$mX+b$	GEV$(m\mu+b, m\sigma, 0)$
Exp(1)	$\mu+\beta\ln(1+\exp(-X))$	Logistic(μ, β)
Exp(1)	$\lambda X^{\frac{1}{k}}$	Weibull(λ, k)

D. Variance Reduction using Control Variates

An alternative approach for stochastic gradient computation is commonly based on the method of control variates. We analyse the variance properties of various estimators in a simple example using univariate function. We then show the correspondence of the widely-known REINFORCE algorithm to the general control variate framework.

D.1. Variance discussion for REINFORCE

The REINFORCE estimator is based on

$$\nabla_\theta \mathbb{E}_p[f(\xi)] = \mathbb{E}_p[(f(\xi) - b)\nabla_\theta \log p(\xi|\theta)], \qquad (34)$$

where b is a baseline typically chosen to reduce the variance of the estimator.

The variance of (34) scales poorly with the number of random variables (Dayan et al., 1995). To see this limitation, consider functions of the form $f(\xi) = \sum_{i=1}^{K} f(\xi_i)$, where each individual term and its gradient has a bounded variance, i.e., $\kappa_l \leq \text{Var}[f(\xi_i)] \leq \kappa_u$ and $\kappa_l \leq \text{Var}[\nabla_{\xi_i} f(\xi_i)] \leq \kappa_u$ for some $0 \leq \kappa_l \leq \kappa_u$ and assume independent or weakly correlated random variables. Given these assumptions the variance of GBP (7) scales as $\text{Var}[\nabla_{\xi_i} f(\xi)] \sim O(1)$, while the variance for REINFORCE (34) scales as $\text{Var}\left[\frac{(\xi_i-\mu_i)}{\sigma_i^2}(f(\xi) - \mathbb{E}[f(\xi)])\right] \sim O(K)$.

For the variance of GBP above, all terms in $f(\xi)$ that do not depend on ξ_i have zero gradient, whereas for REINFORCE the variance involves a summation over all K terms. Even if most of these terms have zero expectation, they still contribute to the variance of the estimator. Thus, the REINFORCE estimator has the undesirable property that its variance scales linearly with the number of independent random variables in the target function, while the variance of GBP is bounded by a constant.

The assumption of weakly correlated terms is relevant for variational learning in larger generative models where independence assumptions and structure in the variational distribution result in free energies that are summations over weakly correlated or independent terms.

D.2. Univariate variance analysis

In analysing the variance properties of many estimators, we discuss the general scaling of likelihood ratio approaches in appendix D. As an example to further emphasise the high-variance nature of these alternative approaches, we present a short analysis in the univariate case.

Consider a random variable $p(\xi) = \mathcal{N}(\xi|\mu, \sigma^2)$ and a simple quadratic function of the form

$$f(\xi) = c\frac{\xi^2}{2}. \qquad (35)$$

For this function we immediately obtain the following variances

$$Var[\nabla_\xi f(\xi)] = c^2\sigma^2 \qquad (36)$$

$$Var[\nabla_{\xi^2} f(\xi)] = 0 \qquad (37)$$

$$Var[\frac{(\xi-\mu)}{\sigma}\nabla_\xi f(\xi)] = 2c^2\sigma^2 + \mu^2c^2 \qquad (38)$$

$$Var[\frac{(\xi-\mu)}{\sigma^2}(f(\xi) - \mathbb{E}[f(\xi)])] = 2c^2\mu^2 + \frac{5}{2}c^2\sigma^2 \qquad (39)$$

Equations (36), (37) and (38) correspond to the variance of the estimators based on (7), (8), (10) respectively whereas equation (39) corresponds to the variance of the REINFORCE algorithm for the gradient with respect to μ.

From these relations we see that, for any parameter configuration, the variance of the REINFORCE estimator is strictly larger than the variance of the estimator based on (7). Additionally, the ratio between the variances of the former and later estimators is lower-bounded by $5/2$. We can also see that the variance of the estimator based on equation (8) is zero for this specific function whereas the variance of the estimator based on equation (10) is not.

E. Estimating the Marginal Likelihood

We compute the marginal likelihood by importance sampling by generating S samples from the recognition model and using the following estimator:

$$p(\mathbf{v}) \approx \frac{1}{S}\sum_{s=1}^{S} \frac{p(\mathbf{v}|\mathbf{h}(\boldsymbol{\xi}^{(s)}))p(\boldsymbol{\xi}^{(s)})}{q(\boldsymbol{\xi}^{s}|\mathbf{v})}; \quad \boldsymbol{\xi}^{(s)} \sim q(\boldsymbol{\xi}|\mathbf{v}) \tag{40}$$

F. Missing Data Imputation

Image completion can be approximatively achieved by a simple iterative procedure which consists of (i) initializing the non-observed pixels with random values; (ii) sampling from the recognition distribution given the resulting image; (iii) reconstruct the image given the sample from the recognition model; (iv) iterate the procedure.

We denote the observed and missing entries in an observation as $\mathbf{v}_o, \mathbf{v}_m$, respectively. The observed $\mathbf{v}_o$ is fixed throughout, therefore all the computations in this section will be conditioned on $\mathbf{v}_o$. The imputation procedure can be written formally as a Markov chain on the space of missing entries $\mathbf{v}_m$ with transition kernel $T^q(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o)$ given by

$$T^q(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o) = \iint p(\mathbf{v}'_m, \mathbf{v}'_o|\xi)q(\xi|\mathbf{v})d\mathbf{v}'_o d\xi, \tag{41}$$

where $\mathbf{v} = (\mathbf{v}_m, \mathbf{v}_o)$.

Provided that the recognition model $q(\xi|\mathbf{v})$ constitutes a good approximation of the true posterior $p(\xi|\mathbf{v})$, (41) can be seen as an approximation of the kernel

$$T(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o) = \iint p(\mathbf{v}'_m, \mathbf{v}'_o|\xi)p(\xi|\mathbf{v})d\mathbf{v}'_o d\xi. \tag{42}$$

The kernel (42) has two important properties: (i) it has as its eigen-distribution the marginal $p(\mathbf{v}_m|\mathbf{v}_o)$; (ii) $T(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o) > 0 \ \forall \mathbf{v}_o, \mathbf{v}_m, \mathbf{v}'_m$. The property (i) can be derived by applying the kernel (42) to the marginal $p(\mathbf{v}_m|\mathbf{v}_o)$ and noting that it is a fixed point. Property (ii) is an immediate consequence of the smoothness of the model.

We apply the fundamental theorem for Markov chains (Neal, 1993, pp. 38) and conclude that given the above properties, a Markov chain generated by (42) is guaranteed to generate samples from the correct marginal $p(\mathbf{v}_m|\mathbf{v}_o)$.

In practice, the stationary distribution of the completed pixels will not be exactly the marginal $p(\mathbf{v}_m|\mathbf{v}_o)$, since we use the approximated kernel (41). Even in this setting we can provide a bound on the L_1 norm of the difference between the resulting stationary marginal and the target marginal $p(\mathbf{v}_m|\mathbf{v}_o)$

Proposition F.1 (L_1 bound on marginal error). *If the recognition model $q(\xi|\mathbf{v})$ is such that for all ξ*

$$\exists \varepsilon > 0 \ s.t. \ \int \left| \frac{q(\xi|\mathbf{v})p(\mathbf{v})}{p(\xi)} - p(\mathbf{v}|\xi) \right| d\mathbf{v} \leq \varepsilon \tag{43}$$

then the marginal $p(\mathbf{v}_m|\mathbf{v}_o)$ is a weak fixed point of the kernel (41) in the following sense:

$$\int \left| \int \left(T^q(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o) - T(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o) \right) p(\mathbf{v}_m|\mathbf{v}_o) d\mathbf{v}_m \right| d\mathbf{v}'_m < \varepsilon. \tag{44}$$

Proof.

$$\begin{aligned}
&\int \left| \int [T^q(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o) - T(\mathbf{v}'_m|\mathbf{v}_m, \mathbf{v}_o)] \, p(\mathbf{v}_m|\mathbf{v}_o) d\mathbf{v}_m \right| d\mathbf{v}'_m \\
&= \int | \iint p(\mathbf{v}'_m, \mathbf{v}'_o|\xi) p(\mathbf{v}_m, \mathbf{v}_o) [q(\xi|\mathbf{v}_m, \mathbf{v}_o) \\
&- p(\xi|\mathbf{v}_m, \mathbf{v}_o)] d\mathbf{v}_m d\xi | d\mathbf{v}'_m \\
&= \int \left| \int p(\mathbf{v}'|\xi) p(\mathbf{v}) [q(\xi|\mathbf{v}) - p(\xi|\mathbf{v})] \frac{p(\mathbf{v})}{p(\xi)} \frac{p(\xi)}{p(\mathbf{v})} d\mathbf{v} d\xi \right| d\mathbf{v}' \\
&= \int \left| \int p(\mathbf{v}'|\xi) p(\xi) [q(\xi|\mathbf{v}) \frac{p(\mathbf{v})}{p(\xi)} - p(\mathbf{v}|\xi)] d\mathbf{v} d\xi \right| d\mathbf{v}' \\
&\leq \int \int p(\mathbf{v}'|\xi) p(\xi) \int \left| q(\xi|\mathbf{v}) \frac{p(\mathbf{v})}{p(\xi)} - p(\mathbf{v}|\xi) \right| d\mathbf{v} d\xi d\mathbf{v}' \\
&\leq \varepsilon,
\end{aligned}$$

where we apply the condition (43) to obtain the last statement. □

That is, if the recognition model is sufficiently close to the true posterior to guarantee that (43) holds for some acceptable error ε than (44) guarantees that the fixed-point of the Markov chain induced by the kernel (41) is no further than ε from the true marginal with respect to the L_1 norm.

G. Variational Bayes for Deep Directed Models

In the main test we focussed on the variational problem of specifying an posterior on the latent variables only. It is natural to consider the variational Bayes problem in which we specify an approximate posterior for both the latent variables and model parameters.

Following the same construction and considering an Gaussian approximate distribution on the model parameters $\boldsymbol{\theta}^g$, the free energy becomes:

$$\begin{aligned}\mathcal{F}(\mathbf{V}) = &- \sum_n \overbrace{\mathbb{E}_q[\log p(\mathbf{v}_n|\mathbf{h}(\boldsymbol{\xi}_n))]}^{\text{reconstruction error}} \\ &+ \underbrace{\frac{1}{2}\sum_{n,l}\left[\|\boldsymbol{\mu}_{n,l}\|^2 + \operatorname{Tr}\mathbf{C}_{n,l} - \log|\mathbf{C}_{n,l}| - 1\right]}_{\text{latent regularization term}} \\ &+ \underbrace{\frac{1}{2}\sum_j \left[\frac{m_j^2}{\kappa} + \frac{\tau_j}{\kappa} + \log\kappa - \log\tau_j - 1\right]}_{\text{parameter regularization term}}, \quad (45)\end{aligned}$$

which now includes an additional term for the cost of using parameters and their regularisation. We must now compute the additional set of gradients with respect to the parameter's mean m_j and variance τ_j are:

$$\nabla_{m_j}\mathcal{F}(\mathbf{v}) = -\mathbb{E}_q\left[\nabla_{\theta_j^g}\log p(\mathbf{v}|\mathbf{h}(\boldsymbol{\xi}))\right] + m_j \quad (46)$$

$$\begin{aligned}\nabla_{\tau_j}\mathcal{F}(\mathbf{v}) = &-\tfrac{1}{2}\mathbb{E}_q\left[\frac{\theta_j - m_j}{\tau_j}\nabla_{\theta_j^g}\log p(\mathbf{v}|\mathbf{h}(\xi))\right] \\ &+ \frac{1}{2\kappa} - \frac{1}{2\tau_j} \quad (47)\end{aligned}$$

H. Additional Simulation Details

We use training data of various types including binary and real-valued data sets. In all cases, we train using mini-batches, which requires the introduction of scaling terms in the free energy objective function (13) in order to maintain the correct scale between the prior over the parameters and the remaining terms (Ahn et al., 2012; Welling & Teh, 2011). We make use of the objective:

$$\begin{aligned}\overline{\mathcal{F}(\mathbf{V})} = &-\lambda\sum_n \mathbb{E}_q\left[\log p(\mathbf{v}_n|\mathbf{h}(\boldsymbol{\xi}_n))\right] + \tfrac{1}{2\kappa}\|\boldsymbol{\theta}^g\|^2 \\ &+ \frac{\lambda}{2}\sum_{n,l}\left[\|\boldsymbol{\mu}_{n,l}\|^2 + \operatorname{Tr}(\mathbf{C}_{n,l}) - \log|\mathbf{C}_{n,l}| - 1\right], \quad (48)\end{aligned}$$

where n is an index over observations in the mini-batch and λ is equal to the ratio of the data-set and the mini-batch size. At each iteration, a random mini-batch of size 200 observations is chosen.

All parameters of the model were initialized using samples from a Gaussian distribution with mean zero and variance 1×10^6; the prior variance of the parameters was $\kappa = 1 \times 10^6$. We compute the marginal likelihood on the test data by importance sampling using samples from the recognition model; we describe our estimator in appendix E.

References

Ahn, S., Balan, A. K., and Welling, M. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *ICML*, 2012.

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The Helmholtz machine. *Neural computation*, 7(5):889–904, September 1995.

Neal, R. M. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 681–688, 2011.