Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Danilo J. Rezende, Shakir Mohamed, Daan Wierstra
{danilor, shakir, daanw}@google.com
Google DeepMind, London

Abstract

We marry ideas from deep neural networks and approximate Bayesian inference to derive a generalised class of deep, directed generative models, endowed with a new algorithm for scalable inference and learning. Our algorithm introduces a recognition model to represent an approximate posterior distribution and uses this for optimisation of a variational lower bound. We develop stochastic backpropagation – rules for gradient backpropagation through stochastic variables – and derive an algorithm that allows for joint optimisation of the parameters of both the generative and recognition models. We demonstrate on several real-world data sets that by using stochastic backpropagation and variational inference, we obtain models that are able to generate realistic samples of data, allow for accurate imputations of missing data, and provide a useful tool for high-dimensional data visualisation.

1. Introduction

There is an immense effort in machine learning and statistics to develop accurate and scalable probabilistic models of data. Such models are called upon whenever we are faced with tasks requiring probabilistic reasoning, such as prediction, missing data imputation and uncertainty estimation; or in simulation-based analyses, common in many scientific fields such as genetics, robotics and control that require generating a large number of independent samples from the model.

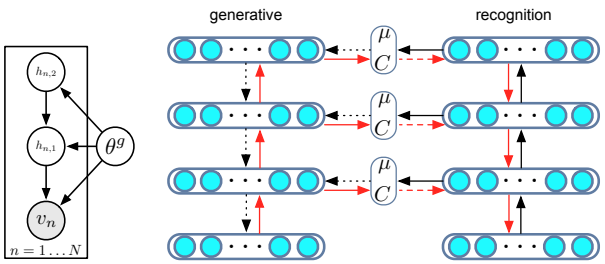
Recent efforts to develop generative models have focused on directed models, since samples are easily obtained by ancestral sampling from the generative process. Directed models such as belief networks and similar latent variable models (Dayan et al., 1995; Frey, 1996; Saul et al., 1996; Bartholomew & Knott, 1999; Uria et al., 2014; Gregor et al., 2014) can be easily sampled from, but in most cases, efficient inference algorithms have remained elusive. These efforts, combined with the demand for accurate probabilistic inferences and fast simulation, lead us to seek generative models that are i) *deep*, since hierarchical architectures allow us to capture complex structure in the data, ii) allow for *fast sampling* of fantasy data from the inferred model, and iii) are computationally *tractable and scalable* to high-dimensional data.

We meet these desiderata by introducing a class of deep, directed generative models with Gaussian latent variables at each layer. To allow for efficient and tractable inference, we use introduce an approximate representation of the posterior over the latent variables using a recognition model that acts as a stochastic encoder of the data. For the generative model, we derive the objective function for optimisation using variational principles; for the recognition model, we specify its structure and regularisation by exploiting recent advances in deep learning. Using this construction, we can train the entire model by a modified form of gradient backpropagation that allows for optimisation of the parameters of both the generative and recognition models jointly.

We build upon the large body of prior work (in section 6) and make the following contributions:

- We combine ideas from deep neural networks and probabilistic latent variable modelling to derive a general class of deep, non-linear latent Gaussian models (section 2).
- We present a new approach for scalable variational inference that allows for joint optimisation of both variational and model parameters by exploiting the properties of latent Gaussian distributions and gradient backpropagation (sections 3 and 4).
- We provide a comprehensive and systematic evaluation of the model demonstrating its applicability to problems in simulation, visualisation, prediction and missing data imputation (section 5).

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



(a) (b)

Figure 1. (a) Graphical model for DLGMs (5). (b) The corresponding computational graph. Black arrows indicate the forward pass of sampling from the recognition and generative models: Solid lines indicate propagation of deterministic activations, dotted lines indicate propagation of samples. Red arrows indicate the backward pass for gradient computation: Solid lines indicate paths where deterministic backpropagation is used, dashed arrows indicate stochastic backpropagation.

2. Deep Latent Gaussian Models

Deep latent Gaussian models (DLGMs) are a general class of deep directed graphical models that consist of Gaussian latent variables at each layer of a processing hierarchy. The model consists of L layers of latent variables. To generate a sample from the model, we begin at the top-most layer (L) by drawing from a Gaussian distribution. The activation $\mathbf{h}_l$ at any lower layer is formed by a non-linear transformation of the layer above $\mathbf{h}_{l+1}$, perturbed by Gaussian noise. We descend through the hierarchy and generate observations $\mathbf{v}$ by sampling from the observation likelihood using the activation of the lowest layer $\mathbf{h}_1$. This process is described graphically in figure 1(a).

This generative process is described as follows:

$$\boldsymbol{\xi}_l \sim \mathcal{N}(\boldsymbol{\xi}_l|\mathbf{0},\mathbf{I}), \quad l = 1,\ldots,L \tag{1}$$

$$\mathbf{h}_L = \mathbf{G}_L\boldsymbol{\xi}_L, \tag{2}$$

$$\mathbf{h}_l = T_l(\mathbf{h}_{l+1}) + \mathbf{G}_l\boldsymbol{\xi}_l, \quad l = 1\ldots L-1 \tag{3}$$

$$\mathbf{v} \sim \pi(\mathbf{v}|T_0(\mathbf{h}_1)), \tag{4}$$

where $\boldsymbol{\xi}_l$ are mutually independent Gaussian variables. The transformations T_l represent multi-layer perceptrons (MLPs) and $\mathbf{G}_l$ are matrices. At the visible layer, the data is generated from any appropriate distribution $\pi(\mathbf{v}|\cdot)$ whose parameters are specified by a transformation of the first latent layer. Throughout the paper we refer to the set of parameters in this generative model by $\boldsymbol{\theta}^g$, i.e. the parameters of the maps T_l and the matrices G_l. This construction allows us to make use of as many deterministic and stochastic layers as needed. We adopt a weak Gaussian prior over $\boldsymbol{\theta}^g$, $p(\boldsymbol{\theta}^g) = \mathcal{N}(\boldsymbol{\theta}|\mathbf{0},\kappa\mathbf{I})$.

The joint probability distribution of this model can be expressed in two equivalent ways:

$$p(\mathbf{v},\mathbf{h}) = p(\mathbf{v}|\mathbf{h}_1,\boldsymbol{\theta}^g)p(\mathbf{h}_L|\boldsymbol{\theta}^g)p(\boldsymbol{\theta}^g)\prod_{l=1}^{L-1}p_l(\mathbf{h}_l|\mathbf{h}_{l+1},\boldsymbol{\theta}^g) \tag{5}$$

$$p(\mathbf{v},\boldsymbol{\xi}) = p(\mathbf{v}|\mathbf{h}_1(\boldsymbol{\xi}_{1\ldots L}),\boldsymbol{\theta}^g)p(\boldsymbol{\theta}^g)\prod_{l=1}^{L}\mathcal{N}(\boldsymbol{\xi}|\mathbf{0},\mathbf{I}). \tag{6}$$

The conditional distributions $p(\mathbf{h}_l|\mathbf{h}_{l+1})$ are implicitly defined by equation (3) and are Gaussian distributions with mean $\boldsymbol{\mu}_l = T_l(\mathbf{h}_{l+1})$ and covariance $\mathbf{S}_l = \mathbf{G}_l\mathbf{G}_l^\top$. Equation (6) makes explicit that this generative model works by applying a complex non-linear transformation to a spherical Gaussian distribution $p(\boldsymbol{\xi}) = \prod_{l=1}^{L}\mathcal{N}(\boldsymbol{\xi}_l|\mathbf{0},\mathbf{I})$ such that the transformed distribution tries to match the empirical distribution. A graphical model corresponding to equation (5) is shown in figure 1(a).

This specification for deep latent Gaussian models (DLGMs) generalises a number of well known models. When we have only one layer of latent variables and use a linear mapping $T(\cdot)$, we recover *factor analysis* (Bartholomew & Knott, 1999) – more general mappings allow for a non-linear factor analysis (Lappalainen & Honkela, 2000). When the mappings are of the form $T_l(\mathbf{h}) = \mathbf{A}_l f(\mathbf{h}) + \mathbf{b}_l$, for simple element-wise non-linearities f such as the probit function or the rectified linearity, we recover the *non-linear Gaussian belief network* (Frey & Hinton, 1999). We describe the relationship to other existing models in section 6. Given this specification, our key task is to develop a method for tractable inference. A number of approaches are known and widely used, and include: mean-field variational EM (Beal, 2003); the wake-sleep algorithm (Dayan, 2000); and stochastic variational methods and related control-variate estimators (Wilson, 1984; Williams, 1992; Hoffman et al., 2013). We also follow a stochastic variational approach, but shall develop an alternative to these existing inference algorithms that overcomes many of their limitations and that is both scalable and efficient.

3. Stochastic Backpropagation

Gradient descent methods in latent variable models typically require computations of the form $\nabla_\theta \mathbb{E}_{q_\theta}[f(\boldsymbol{\xi})]$, where the expectation is taken with respect to a distribution $q_\theta(\cdot)$ with parameters $\boldsymbol{\theta}$, and f is a loss function that we assume to be integrable and smooth. This quantity is difficult to compute directly since i) the expectation is unknown for most problems, and ii) there is an indirect dependency on the parameters of q over which the expectation is taken.

We now develop the key identities that are used to allow for efficient inference by exploiting specific prop-

erties of the problem of computing gradients through random variables. We refer to this computational strategy as *stochastic backpropagation*.

3.1. Gaussian Backpropagation (GBP)

When the distribution q is a K-dimensional Gaussian $\mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}, \mathbf{C})$ the required gradients can be computed using the Gaussian gradient identities:

$$\nabla_{\mu_i}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[f(\boldsymbol{\xi})\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[\nabla_{\xi_i} f(\boldsymbol{\xi})\right], \tag{7}$$

$$\nabla_{C_{ij}}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[f(\boldsymbol{\xi})\right] = \tfrac{1}{2}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[\nabla^2_{\xi_i,\xi_j} f(\boldsymbol{\xi})\right], \tag{8}$$

which are due to the theorems by Bonnet (1964) and Price (1958), respectively. These equations are true in expectation for any integrable and smooth function $f(\boldsymbol{\xi})$. Equation (7) is a direct consequence of the location-scale transformation for the Gaussian (discussed in section 3.2). Equation (8) can be derived by successive application of the product rule for integrals; we provide the proofs for these identities in appendix B.

Equations (7) and (8) are especially interesting since they allow for unbiased gradient estimates by using a small number of samples from q. Assume that both the mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C}$ depend on a parameter vector $\boldsymbol{\theta}$. We are now able to write a general rule for Gaussian gradient computation by combining equations (7) and (8) and using the chain rule:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}[f(\boldsymbol{\xi})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[\mathbf{g}^\top\frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\theta}} + \tfrac{1}{2}\mathrm{Tr}\left(\mathbf{H}\frac{\partial\mathbf{C}}{\partial\boldsymbol{\theta}}\right)\right] \tag{9}$$

where $\mathbf{g}$ and $\mathbf{H}$ are the gradient and the Hessian of the function $f(\boldsymbol{\xi})$, respectively. Equation (9) can be interpreted as a modified backpropagation rule for Gaussian distributions that takes into account the gradients through the mean $\boldsymbol{\mu}$ *and* covariance $\mathbf{C}$. This reduces to the standard backpropagation rule when $\mathbf{C}$ is constant. Unfortunately this rule requires knowledge of the Hessian matrix of $f(\boldsymbol{\xi})$, which has an algorithmic complexity $O(K^3)$. For inference in DLGMs, we later introduce an unbiased though higher variance estimator that requires only quadratic complexity.

3.2. Generalised Backpropagation Rules

We describe two approaches to derive general backpropagation rules for non-Gaussian q-distributions.

Using the product rule for integrals. For many exponential family distributions, it is possible to find a function $B(\boldsymbol{\xi};\boldsymbol{\theta})$ to ensure that

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_{p(\boldsymbol{\xi}|\boldsymbol{\theta})}[f(\boldsymbol{\xi})] = -\mathbb{E}_{p(\boldsymbol{\xi}|\boldsymbol{\theta})}[\nabla_{\boldsymbol{\xi}}[B(\boldsymbol{\xi};\boldsymbol{\theta})f(\boldsymbol{\xi})]].$$

That is, we express the gradient with respect to the parameters of q as an expectation of gradients with respect to the random variables themselves. This approach can be used to derive rules for many distributions such as the Gaussian, inverse Gamma and log-Normal. We discuss this in more detail in appendix C.

Using suitable co-ordinate transformations. We can also derive stochastic backpropagation rules for any distribution that can be written as a smooth, invertible transformation of a standard base distribution. For example, any Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ can be obtained as a transformation of a spherical Gaussian $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, using the transformation $y = \boldsymbol{\mu} + \mathbf{R}\boldsymbol{\epsilon}$ and $\mathbf{C} = \mathbf{R}\mathbf{R}^\top$. The gradient of the expectation with respect to $\mathbf{R}$ is then:

$$\begin{aligned}\nabla_{\mathbf{R}}\mathbb{E}_{\mathcal{N}(\boldsymbol{\mu},\mathbf{C})}\left[f(\boldsymbol{\xi})\right] &= \nabla_{\mathbf{R}}\mathbb{E}_{\mathcal{N}(0,I)}\left[f(\boldsymbol{\mu}+\mathbf{R}\boldsymbol{\epsilon})\right] \\ &= \mathbb{E}_{\mathcal{N}(0,I)}\left[\boldsymbol{\epsilon}\mathbf{g}^\top\right],\end{aligned} \tag{10}$$

where $\mathbf{g}$ is the gradient of f evaluated at $\boldsymbol{\mu} + \mathbf{R}\boldsymbol{\epsilon}$ and provides a lower-cost alternative to Price's theorem (8). Such transformations are well known for many distributions, especially those with a self-similarity property or location-scale formulation, such as the Gaussian, Student's t-distribution, stable distributions, and generalised extreme value distributions.

Stochastic backpropagation in other contexts. The Gaussian gradient identities described above do not appear to be widely used. These identities have been recognised by Opper & Archambeau (2009) for variational inference in Gaussian process regression, and following this work, by Graves (2011) for parameter learning in large neural networks. Concurrently with this paper, Kingma & Welling (2014) present an alternative discussion of stochastic backpropagation. Our approaches were developed simultaneously and provide complementary perspectives on the use and derivation of stochastic backpropagation rules.

4. Scalable Inference in DLGMs

We use the matrix $\mathbf{V}$ to refer to the full data set of size $N \times D$ with observations $\mathbf{v}_n = [v_{n1}, \ldots, v_{nD}]^\top$.

4.1. Free Energy Objective

To perform inference in DLGMs we must integrate out the effect of any latent variables – this requires us to compute the integrated or marginal likelihood. In general, this will be an intractable integration and instead we optimise a lower bound on the marginal likelihood. We introduce an approximate posterior distribution $q(\cdot)$ and apply Jensen's inequality following the variational principle (Beal, 2003) to obtain:

$$\begin{aligned}\mathcal{L}(\mathbf{V}) &= -\log p(\mathbf{V}) = -\log \int p(\mathbf{V}|\boldsymbol{\xi},\boldsymbol{\theta}^g)p(\boldsymbol{\xi},\boldsymbol{\theta}^g)d\boldsymbol{\xi} \\ &= -\log \int \frac{q(\boldsymbol{\xi})}{q(\boldsymbol{\xi})}p(\mathbf{V}|\boldsymbol{\xi},\boldsymbol{\theta}^g)p(\boldsymbol{\xi},\boldsymbol{\theta}^g)d\boldsymbol{\xi} \qquad (11)\\ \leq \mathcal{F}(\mathbf{V}) &= D_{KL}[q(\boldsymbol{\xi})\|p(\boldsymbol{\xi})] - \mathbb{E}_q\left[\log p(\mathbf{V}|\boldsymbol{\xi},\boldsymbol{\theta}^g)p(\boldsymbol{\theta}^g)\right].\end{aligned}$$

This objective consists of two terms: the first is the KL-divergence between the variational distribution and the prior distribution (which acts a regulariser), and the second is a reconstruction error.

We specify the approximate posterior as a distribution $q(\boldsymbol{\xi}|\mathbf{v})$ that is *conditioned* on the observed data. This distribution can be specified as any directed acyclic graph where each node of the graph is a Gaussian conditioned, through linear or non-linear transformations, on its parents. The joint distribution in this case is non-Gaussian, but stochastic backpropagation can still be applied.

For simplicity, we use a $q(\boldsymbol{\xi}|\mathbf{v})$ that is a Gaussian distribution that factorises across the L layers (but not necessarily within a layer):

$$q(\boldsymbol{\xi}|\mathbf{V},\boldsymbol{\theta}^r) = \prod_{n=1}^{N}\prod_{l=1}^{L}\mathcal{N}\left(\boldsymbol{\xi}_{n,l}|\boldsymbol{\mu}_l(\mathbf{v}_n),\mathbf{C}_l(\mathbf{v}_n)\right), \quad (12)$$

where the mean $\boldsymbol{\mu}_l(\cdot)$ and covariance $\mathbf{C}_l(\cdot)$ are generic maps represented by deep neural networks. Parameters of the q-distribution are denoted by the vector $\boldsymbol{\theta}^r$.

For a Gaussian prior and a Gaussian recognition model, the KL term in (11) can be computed analytically and the free energy becomes:

$$\begin{aligned}D_{KL}[\mathcal{N}(\boldsymbol{\mu},\mathbf{C})\|\mathcal{N}(\mathbf{0},\mathbf{I})] &= \tfrac{1}{2}\left[\mathrm{Tr}(\mathbf{C}) - \log|\mathbf{C}| + \boldsymbol{\mu}^\top\boldsymbol{\mu} - D\right],\\ \mathcal{F}(\mathbf{V}) &= -\sum_n \mathbb{E}_q\left[\log p(\mathbf{v}_n|\mathbf{h}(\boldsymbol{\xi}_n))\right] + \tfrac{1}{2\kappa}\|\boldsymbol{\theta}^g\|^2\\ &\quad + \frac{1}{2}\sum_{n,l}\left[\|\boldsymbol{\mu}_{n,l}\|^2 + \mathrm{Tr}(\mathbf{C}_{n,l}) - \log|\mathbf{C}_{n,l}| - 1\right], \quad (13)\end{aligned}$$

where $\mathrm{Tr}(\mathbf{C})$ and $|\mathbf{C}|$ indicate the trace and the determinant of the covariance matrix $\mathbf{C}$, respectively.

The specification of an approximate posterior distribution that is conditioned on the observed data is the first component of an efficient variational inference algorithm. We shall refer to the distribution $q(\boldsymbol{\xi}|\mathbf{v})$ (12) as a *recognition model*, whose design is independent of the generative model. A recognition model allows us introduce a form of amortised inference (Gershman & Goodman, 2014) for variational methods in which we share statistical strength by allowing for generalisation across the posterior estimates for all latent variables using a model. The implication of this generalisation ability is: faster convergence during training; and faster inference at test time since we only require a single pass through the recognition model, rather than needing to perform any iterative computations (such as in a generalised E-step).

To allow for the best possible inference, the specification of the recognition model must be flexible enough to provide an accurate approximation of the posterior distribution – motivating the use of deep neural networks. We regularise the recognition model by introducing additional noise, specifically, bit-flip or drop-out noise at the input layer and small additional Gaussian noise to samples from the recognition model. We use rectified linear activation functions as non-linearities for any deterministic layers of the neural network. We found that such regularisation is essential and without it the recognition model is unable to provide accurate inferences for unseen data points.

4.2. Gradients of the Free Energy

To optimise (13), we use Monte Carlo methods for any expectations and use stochastic gradient descent for optimisation. For optimisation, we require efficient estimators of the gradients of all terms in equation (13) with respect to the parameters $\boldsymbol{\theta}^g$ and $\boldsymbol{\theta}^r$ of the generative and the recognition models, respectively.

The gradients with respect to the jth generative parameter θ^g_j can be computed using:

$$\nabla_{\theta^g_j}\mathcal{F}(\mathbf{V}) = -\mathbb{E}_q\left[\nabla_{\theta^g_j}\log p(\mathbf{V}|\mathbf{h})\right] + \tfrac{1}{\kappa}\theta^g_j. \quad (14)$$

An unbiased estimator of $\nabla_{\theta^g_j}\mathcal{F}(\mathbf{V})$ is obtained by approximating equation (14) with a small number of samples (or even a single sample) from the recognition model q.

To obtain gradients with respect to the recognition parameters $\boldsymbol{\theta}^r$, we use the rules for Gaussian backpropagation developed in section 3. To address the complexity of the Hessian in the general rule (9), we use the co-ordinate transformation for the Gaussian to write the gradient with respect to the factor matrix $\mathbf{R}$ instead of the covariance $\mathbf{C}$ (recalling $\mathbf{C} = \mathbf{R}\mathbf{R}^\top$) derived in equation (10), where derivatives are computed for the function $f(\boldsymbol{\xi}) = \log p(\mathbf{v}|\mathbf{h}(\boldsymbol{\xi}))$.

The gradients of $\mathcal{F}(\mathbf{v})$ in equation (13) with respect to the variational mean $\boldsymbol{\mu}_l(\mathbf{v})$ and the factors $\mathbf{R}_l(\mathbf{v})$ are:

$$\begin{aligned}\nabla_{\mu_l}\mathcal{F}(\mathbf{v}) &= -\mathbb{E}_q\left[\nabla_{\boldsymbol{\xi}_l}\log p(\mathbf{v}|\mathbf{h}(\xi))\right] + \boldsymbol{\mu}_l, \quad (15)\\ \nabla_{R_{l,i,j}}\mathcal{F}(\mathbf{v}) &= -\tfrac{1}{2}\mathbb{E}_q\left[\epsilon_{l,j}\nabla_{\xi_{l,i}}\log p(\mathbf{v}|\mathbf{h}(\boldsymbol{\xi}))\right]\\ &\quad + \tfrac{1}{2}\nabla_{R_{l,i,j}}\left[\mathrm{Tr}\,\mathbf{C}_{n,l} - \log|\mathbf{C}_{n,l}|\right], \quad (16)\end{aligned}$$

where the gradients $\nabla_{R_{l,i,j}}\left[\mathrm{Tr}\,\mathbf{C}_{n,l} - \log|\mathbf{C}_{n,l}|\right]$ are computed by backpropagation. Unbiased estimators of the gradients (15) and (16) are obtained jointly

Algorithm 1 Learning in DLGMs

while hasNotConverged() **do**
  $\mathbf{V} \leftarrow$ getMiniBatch()
  $\boldsymbol{\xi}_n \sim q(\xi_n|\mathbf{v}_n)$ (bottom-up pass) eq. (12)
  $\mathbf{h} \leftarrow \mathbf{h}(\xi)$ (top-down pass) eq. (3)
  updateGradients() eqs (14) – (17)
  $\boldsymbol{\theta}^{g,r} \leftarrow \boldsymbol{\theta}^{g,r} + \Delta\boldsymbol{\theta}^{g,r}$
end while

by sampling from the recognition model $\boldsymbol{\xi} \sim q(\boldsymbol{\xi}|\mathbf{v})$ (bottom-up pass) and updating the values of the generative model layers using equation (3) (top-down pass).

Finally the gradients $\nabla_{\theta_j^r}\mathcal{F}(\mathbf{v})$ obtained from equations (15) and (16) are:

$$\nabla_{\boldsymbol{\theta}^r}\mathcal{F}(\mathbf{v}) = \nabla_{\boldsymbol{\mu}}\mathcal{F}(\mathbf{v})^\top \frac{\partial\boldsymbol{\mu}}{\partial\boldsymbol{\theta}^r} + \mathrm{Tr}\left(\nabla_{\mathbf{R}}\mathcal{F}(\mathbf{v})\frac{\partial\mathbf{R}}{\partial\boldsymbol{\theta}^r}\right). \quad (17)$$

The gradients (14) – (17) are now used to descend the free-energy surface with respect to both the generative and recognition parameters in a single optimisation step. Figure 1(b) shows the flow of computation in DLGMs. Our algorithm proceeds by first performing a forward pass (black arrows), consisting of a bottom-up (recognition) phase and a top-down (generation) phase, which updates the hidden activations of the recognition model and parameters of any Gaussian distributions, and then a backward pass (red arrows) in which gradients are computed using the appropriate backpropagation rule for deterministic and stochastic layers. We take a descent step using:

$$\Delta\theta^{g,r} = -\Gamma^{g,r}\nabla_{\theta^{g,r}}\mathcal{F}(\mathbf{V}), \quad (18)$$

where $\Gamma^{g,r}$ is a diagonal pre-conditioning matrix computed using the RMSprop heuristic[1]. The learning procedure is summarised in algorithm 1.

4.3. Gaussian Covariance Parameterisation

There are a number of approaches for parameterising the covariance matrix of the recognition model $q(\boldsymbol{\xi})$. Maintaining a full covariance matrix $\mathbf{C}$ in equation (13) would entail an algorithmic complexity of $O(K^3)$ for training and sampling per layer, where K is the number of latent variables per layer.

The simplest approach is to use a diagonal covariance matrix $\mathbf{C} = \mathrm{diag}(\mathbf{d})$, where $\mathbf{d}$ is a K-dimensional vector. This approach is appealing since it allows for linear-time computation and sampling, but only allows for axis-aligned posterior distributions.

We can improve upon the diagonal approximation by parameterising the covarinace as a rank-1 matrix with a diagonal correction. Using a vectors $\mathbf{u}$ and $\mathbf{d}$, with $\mathbf{D} = \mathrm{diag}(\mathbf{d})$, we parameterise the precision $\mathbf{C}^{-1}$ as:

$$\mathbf{C}^{-1} = \mathbf{D} + \mathbf{u}\mathbf{u}^\top. \quad (19)$$

This representation allows for arbitrary rotations of the Gaussian distribution along one principal direction with relatively few additional parameters (Magdon-Ismail & Purnell, 2010). By application of the matrix inversion lemma (Woodbury identity), we obtain the covariance matrix in terms of $\mathbf{d}$ and $\mathbf{u}$ as:

$$\mathbf{C} = \mathbf{D}^{-1} - \eta\mathbf{D}^{-1}\mathbf{u}\mathbf{u}^\top\mathbf{D}^{-1}, \qquad \eta = \frac{1}{\mathbf{u}^\top\mathbf{D}^{-1}\mathbf{u}+1},$$
$$\log|\mathbf{C}| = \log\eta - \log|\mathbf{D}|. \quad (20)$$

This allows both the trace $\mathrm{Tr}(\mathbf{C})$ and $\log|\mathbf{C}|$ needed in the computation of the Gaussian KL, as well as their gradients, to be computed in $O(K)$ time per layer.

The factorisation $\mathbf{C} = \mathbf{R}\mathbf{R}^\top$, with $\mathbf{R}$ a matrix of the same size as $\mathbf{C}$ and can be computed directly in terms of $\mathbf{d}$ and $\mathbf{u}$. One solution for $\mathbf{R}$ is:

$$\mathbf{R} = \mathbf{D}^{-\frac{1}{2}} - \left[\frac{1-\sqrt{\eta}}{\mathbf{u}^\top\mathbf{D}^{-1}\mathbf{u}}\right]\mathbf{D}^{-1}\mathbf{u}\mathbf{u}^\top\mathbf{D}^{-\frac{1}{2}}. \quad (21)$$

The product of $\mathbf{R}$ with an arbitrary vector can be computed in $O(K)$ without computing $\mathbf{R}$ explicitly. This also allows us to sample efficiently from this Gaussian, since any Gaussian random variable $\boldsymbol{\xi}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{C} = \mathbf{R}\mathbf{R}^\top$ can be written as $\boldsymbol{\xi} = \boldsymbol{\mu} + \mathbf{R}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a standard Gaussian variate.

Since this covariance parametrisation has linear cost in the number of latent variables, we can also use it to parameterise the variational distribution of all layers jointly, instead of the factorised assumption in (12).

4.4. Algorithm Complexity

The computational complexity of producing a sample from the generative model is $O(L\bar{K}^2)$, where $\bar{K}$ is the average number of latent variables per layer and L is the number of layers (counting both deterministic and stochastic layers). The computational complexity per training sample during training is also $O(L\bar{K}^2)$ – the same as that of matching auto-encoder.

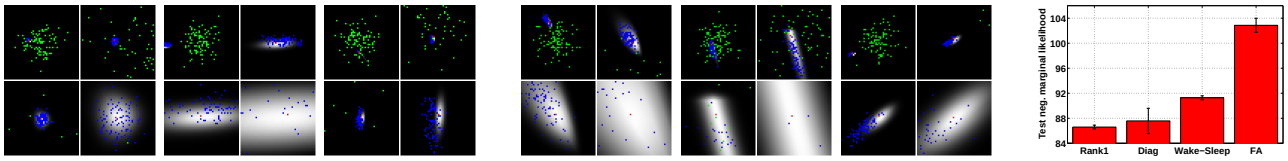
5. Results

Generative models have a number of applications in simulation, prediction, data visualisation, missing data imputation and other forms of probabilistic reasoning. We describe the testing methodology we use and present results on a number of these tasks.

5.1. Analysing the Approximate Posterior

We use sampling to evaluate the true posterior distribution for a number of MNIST digits using the binarised data set from Larochelle & Murray

[1] Described by G. Hinton, 'RMSprop: Divide the gradient by a running average of its recent magnitude', in *Neural networks for machine learning*, Coursera lecture 6e, 2012.



(a) Diagonal covariance (b) Low-rank covariance (c) Performance

Figure 2. (a, b) Analysis of the true vs. approximate posterior for MNIST. Within each image we show four views of the same posterior, zooming in on the region centred on the MAP (red) estimate. (c) Comparison of test log likelihoods.

Table 1. Comparison of negative log-probabilities on the test set for the binarised MNIST data.

Model	$-\ln p(\mathbf{v})$
Factor Analysis	106.00
NLGBN (Frey & Hinton, 1999)	95.80
Wake-Sleep (Dayan, 2000)	91.3
DLGM diagonal covariance	87.30
DLGM rank-one covariance	86.60
Results below from Uria et al. (2014)	
MoBernoullis K=10	168.95
MoBernoullis K=500	137.64
RBM (500 h, 25 CD steps) approx.	86.34
DBN 2hl approx.	84.55
NADE 1hl (fixed order)	88.86
NADE 1hl (fixed order, RLU, minibatch)	88.33
EoNADE 1hl (2 orderings)	90.69
EoNADE 1hl (128 orderings)	87.71
EoNADE 2hl (2 orderings)	87.96
EoNADE 2hl (128 orderings)	85.10

(2011). We visualise the posterior distribution for a model with two Gaussian latent variables in figure 2. The true posterior distribution is shown by the grey regions and was computed by importance sampling with a large number of particles aligned in a grid between -5 and 5. In figure 2(a) we see that these posterior distributions are elliptical or spherical in shape and thus, it is reasonable to assume that they can be well approximated by a Gaussian. Samples from the prior (green) are spread widely over the space and very few samples fall in the region of significant posterior mass, explaining the inefficiency of estimation methods that rely on samples from the prior. Samples from the recognition model (blue) are concentrated on the posterior mass, indicating that the recognition model has learnt the correct posterior statistics, which should lead to efficient learning.

In figure 2(a) we see that samples from the recognition model are aligned to the axis and do not capture the posterior correlation. The correlation is captured using the structured covariance model in figure 2(b). Not all posteriors are Gaussian in shape, but the recognition places mass in the best location possible to provide a reasonable approximation. As a benchmark for comparison, the performance in terms of test log-likelihood is shown in figure 2(c), using the same architecture, for factor analysis (FA), the wake-sleep algorithm, and our approach using both the diagonal and structured covariance approaches. For this experiment, the generative model consists of 100 latent variables feeding into a deterministic layer of 300 nodes, which then feeds to the observation likelihood. We use the same structure for the recognition model.

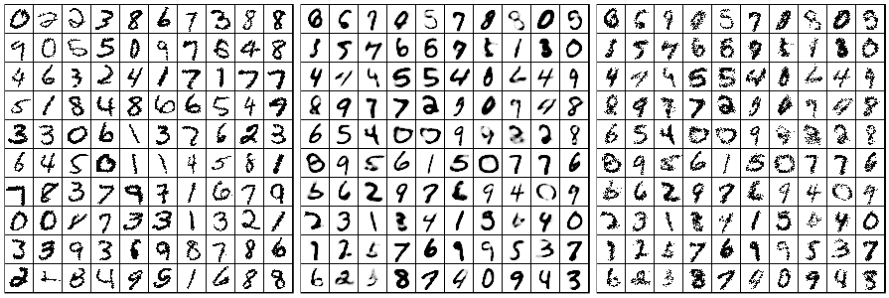
5.2. Simulation and Prediction

We evaluate the performance of a three layer latent Gaussian model on the MNIST data set. The model consists of two deterministic layers with 200 hidden units and a stochastic layer of 200 latent variables. We use mini-batches of 200 observations and trained the model using stochastic backpropagation. Samples from this model are shown in figure 3(a). We also compare the test log-likelihood to a large number of existing approaches in table 1. We used the binarised dataset as in Uria et al. (2014) and quote the log-likelihoods in the lower part of the table from this work. These results show that our approach is competitive with some of the best models currently available. The generated digits also match the true data well and visually appear as good as some of the best visualisations from these competing approaches.

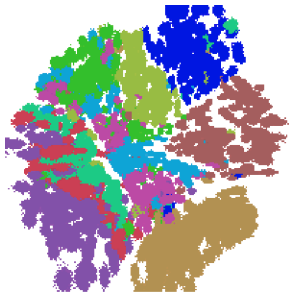
We also analysed the performance of our model on three high-dimensional real image data sets. The NORB object recognition data set consists of $24,300$ images that are of size 96×96 pixels. We use a model consisting of 1 deterministic layer of 400 hidden units and one stochastic layer of 100 latent variables. Samples produced from this model are shown in figure 4(a). The CIFAR10 natural images data set consists of $50,000$ RGB images that are of size 32×32 pixels, which we split into random 8×8 patches. We use the same model as used for the MNIST experiment and show samples from the model in figure 4(b). The Frey faces data set consists of almost $2,000$ images of different facial expressions of size 28×20 pixels.

5.3. Data Visualisation

Latent variable models are often used for visualisation of high-dimensional data sets. We project the MNIST data set to a 2-dimensional latent space and use this 2D embedding as a visualisation of the data – an embedding for MNIST is shown in figure 3(b). The classes separate into different regions, suggesting that such embeddings can be useful in understanding the structure of high-dimensional data sets.

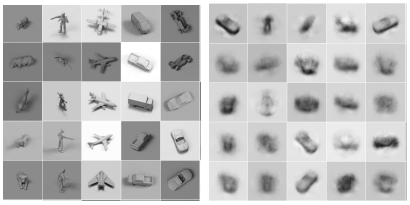


(a) Left: Training data. Middle: Sampled pixel probabilities. Right: Model samples

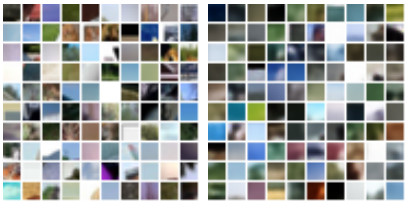


(b) 2D embedding.

Figure 3. Performance on the MNIST dataset. For the visualisation, each colour corresponds to one of the digit classes.



(a) NORB



(b) CIFAR

(c) Frey

Figure 4. Sampled generated from DLGMs for three data sets: (a) NORB, (b) CIFAR 10, (c) Frey faces. In all images, the left image shows samples from the training data and the right side shows the generated samples.

5.4. Missing Data Imputation and Denoising

We demonstrate the ability of the model to impute missing data using the street view house numbers (SVHN) data set (Netzer et al., 2011), which consists of $73,257$ images of size 32×32 pixels, and the Frey faces and MNIST data sets. The performance of the model is shown in figure 5.

We test the imputation ability under two different missingness types (Little & Rubin, 1987): Missing-at-Random (MAR), where we consider 60% and 80% of the pixels to be missing randomly, and Not Missing-at-Random (NMAR), where we consider a square region of the image to be missing. The model produces very good completions in both test cases. There is uncertainty in the identity of the image and this is reflected in the errors in these completions as the resampling procedure is run (see transitions from digit 9 to 7, and digit 8 to 6 in figure 5). This further demonstrates the ability of the model to capture the diversity of the underlying data. We do not integrate over the missing values, but use a procedure that simulates a Markov chain that we show converges to the true marginal distribution of missing given observed pixels. The imputation procedure is discussed in appendix F.

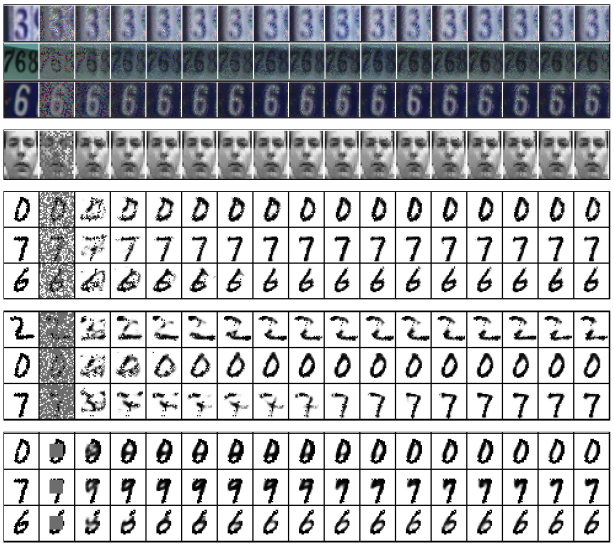


Figure 5. Imputation results: Row 1, SVHN. Row 2, Frey faces. Rows 3–5, MNIST. Col. 1 shows the true data. Col. 2 shows pixel locations set as missing in grey. The remaining columns show imputations for 15 iterations.

6. Discussion

Directed Graphical Models. DLGMs form a unified family of models that includes factor analysis (Bartholomew & Knott, 1999), non-linear factor analysis (Lappalainen & Honkela, 2000), and non-linear Gaussian belief networks (Frey & Hinton, 1999). Other related models include sigmoid belief networks (Saul et al., 1996) and deep auto-regressive networks (Gregor et al., 2014), which use auto-regressive Bernoulli distributions at each layer instead of Gaussian distributions. The Gaussian process latent variable model and deep Gaussian processes (Lawrence, 2005; Damianou & Lawrence, 2013) form the non-parametric analogue of our model and employ Gaus-

sian process priors over the non-linear functions between each layer. The neural auto-regressive density estimator (NADE) (Larochelle & Murray, 2011; Uria et al., 2014) uses function approximation to model conditional distributions within a directed acyclic graph. NADE is amongst the most competitive generative models currently available, but has several limitations, such as the inability to allow for deep representations and difficulties in extending to locally-connected models (e.g., through the use of convolutional layers), preventing it from scaling easily to high-dimensional data.

Alternative latent Gaussian inference. Few of the alternative approaches for inferring latent Gaussian distributions meet the desiderata for scalable inference we seek. The Laplace approximation has been concluded to be a poor approximation in general, in addition to being computationally expensive. INLA is restricted to models with few hyperparameters (< 10), whereas our interest is in 100s-1000s. EP cannot be applied to latent variable models due to the inability to match moments of the joint distribution of latent variables and model parameters. Furthermore, no reliable methods exist for moment-matching with means and covariances formed by non-linear transformations – linearisation and importance sampling are two, but are either inaccurate or very slow. Thus, the the variational approach we present remains a general-purpose and competitive approach for inference.

Monte Carlo variance reduction. *Control variate* methods are amongst the most general and effective techniques for variance reduction when Monte Carlo methods are used (Wilson, 1984). One popular approach is the REINFORCE algorithm (Williams, 1992), since it is simple to implement and applicable to both discrete and continuous models, though control variate methods are becoming increasingly popular for variational inference problems (Hoffman et al., 2013; Blei et al., 2012; Ranganath et al., 2014; Salimans & Knowles, 2014). Unfortunately, such estimators have the undesirable property that their variance scales linearly with the number of independent random variables in the target function, while the variance of GBP is bounded by a constant: for K-dimensional latent variables the variance of REINFORCE scales as $O(K)$, whereas GBP scales as $O(1)$ (see appendix D).

An important family of alternative estimators is based on *quadrature and series expansion methods* (Honkela & Valpola, 2004; Lappalainen & Honkela, 2000). These methods have low-variance at the price of introducing biases in the estimation. More recently a combination of the series expansion and control variate approaches has been proposed by Blei et al. (2012).

A very general alternative is the *wake-sleep algorithm* (Dayan et al., 1995). The wake-sleep algorithm can perform well, but it fails to optimise a single consistent objective function and there is thus no guarantee that optimising it leads to a decrease in the free energy (11).

Relation to denoising auto-encoders. Denoising auto-encoders (DAE) (Vincent et al., 2010) introduce a random corruption to the encoder network and attempt to minimize the expected reconstruction error under this corruption noise with additional regularisation terms. In our variational approach, the recognition distribution $q(\boldsymbol{\xi}|\mathbf{v})$ can be interpreted as a stochastic encoder in the DAE setting. There is then a direct correspondence between the expression for the free energy (11) and the reconstruction error and regularization terms used in denoising auto-encoders (c.f. equation (4) of Bengio et al. (2013)). Thus, we can see denoising auto-encoders as a realisation of variational inference in latent variable models.

The key difference is that the form of encoding 'corruption' and regularisation terms used in our model have been derived directly using the variational principle to provide a strict bound on the marginal likelihood of a known directed graphical model that allows for easy generation of samples. DAEs can also be used as generative models by simulating from a Markov chain (Bengio et al., 2013; Bengio & Thibodeau-Laufer, 2013). But the behaviour of these Markov chains will be very problem specific, and we lack consistent tools to evaluate their convergence.

7. Conclusion

We have introduced a general-purpose inference method for models with continuous latent variables. Our approach introduces a recognition model, which can be seen as a stochastic encoding of the data, to allow for efficient and tractable inference. We derived a lower bound on the marginal likelihood for the generative model and specified the structure and regularisation of the recognition model by exploiting recent advances in deep learning. By developing modified rules for backpropagation through stochastic layers, we derived an efficient inference algorithm that allows for joint optimisation of all parameters. We show on several real-world data sets that the model generates realistic samples, provides accurate imputations of missing data and can be a useful tool for high-dimensional data visualisation.

Appendices can be found with the online version of the paper. `http://arxiv.org/abs/1401.4082`

Acknowledgements. We are grateful for feedback from the reviewers as well as Peter Dayan, Antti Honkela, Neil Lawrence and Yoshua Bengio.

References

Bartholomew, D. J. and Knott, M. *Latent variable models and factor analysis*, volume 7 of Kendall's library of statistics. Arnold, 2nd edition, 1999.

Beal, M. J. *Variational Algorithms for approximate Bayesian inference*. PhD thesis, University of Cambridge, 2003.

Bengio, Y. and Thibodeau-Laufer, É. Deep generative stochastic networks trainable by backprop. Technical report, University of Montreal, 2013.

Bengio, Y., Yao, L., Alain, G., and Vincent, P. Generalized denoising auto-encoders as generative models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1–9, 2013.

Blei, D. M., Jordan, M. I., and Paisley, J. W. Variational Bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pp. 1367–1374, 2012.

Bonnet, G. Transformations des signaux aléatoires a travers les systèmes non linéaires sans mémoire. *Annales des Télécommunications*, 19(9-10):203–220, 1964.

Damianou, A. C. and Lawrence, N. D. Deep Gaussian processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

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

Dayan, P. Helmholtz machines and wake-sleep learning. *Handbook of Brain Theory and Neural Network. MIT Press, Cambridge, MA*, 44(0), 2000.

Frey, B. J. Variational inference for continuous sigmoidal Bayesian networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1996.

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

Gershman, S. J. and Goodman, N. D. Amortized inference in probabilistic reasoning. In *Proceedings of the 36th Annual Conference of the Cognitive Science Society*, 2014.

Graves, A. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24 (NIPS)*, pp. 2348–2356, 2011.

Gregor, K., Mnih, A., and Wierstra, D. Deep autoregressive networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, October 2014.

Hoffman, M., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, May 2013.

Honkela, A. and Valpola, H. Unsupervised variational Bayesian learning of nonlinear models. In *Advances in Neural Information Processing Systems (NIPS)*, 2004.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Lappalainen, H. and Honkela, A. Bayesian non-linear independent component analysis by multi-layer perceptrons. In *Advances in independent component analysis (ICA)*, pp. 93–121. Springer, 2000.

Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Lawrence, N. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

Little, R. J. and Rubin, D. B. *Statistical analysis with missing data*, volume 539. Wiley New York, 1987.

Magdon-Ismail, M. and Purnell, J. T. Approximating the covariance matrix of GMMs with low-rank perturbations. In *Proceedings of the 11th international conference on Intelligent data engineering and automated learning (IDEAL)*, pp. 300–307, 2010.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Opper, M. and Archambeau, C. The variational Gaussian approximation revisited. *Neural computation*, 21 (3):786–92, March 2009.

Price, R. A useful theorem for nonlinear devices having Gaussian inputs. *IEEE Transactions on Information Theory*, 4(2):69–72, 1958.

Ranganath, R., Gerrish, S., and Blei, D. M. Black box variational inference. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, October 2014.

Salimans, T. and Knowles, D. A. On using control variates with stochastic approximation for variational bayes and its connection to stochastic linear regression. *ArXiv preprint. arXiv:1401.1022*, October 2014.

Saul, L. K., Jaakkola, T., and Jordan, M. I. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research (JAIR)*, 4:61–76, 1996.

Uria, B., Murray, I., and Larochelle, H. A deep and tractable density estimator. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229 – 256, 1992.

Wilson, J. R. Variance reduction techniques for digital simulation. *American Journal of Mathematical and Management Sciences*, 4(3):277–312, 1984.

Appendices: Stochastic Backpropagation and Approximate Inference in Deep Generative Models

Danilo J. Rezende, Shakir Mohamed, Daan Wierstra
{danilor, shakir, daanw}@google.com
Google DeepMind, London

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 coordinate 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 $f(\boldsymbol{\xi})$.*

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

Proof.

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

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

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.

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

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:

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

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:

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

Consider general exponential family distributions of the form:

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

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:

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

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:

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	$\begin{pmatrix}\mu\\ \sigma^2\end{pmatrix}$	$\begin{pmatrix}-1\\ \frac{(\ln x-\mu-\sigma)(\ln x-\mu+\sigma)}{2\sigma^2(\ln x-\mu)}\end{pmatrix}$

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.

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(·)	**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.