A Unifying Review of Efficient Variational Inference and Learning in Deep Directed Latent Variable Models

Riashat Islam RI258@CAM.AC.UK
University of Cambridge

Jiameng Gao JG632@CAM.AC.UK
University of Cambridge

Vera Johne VGJ21@CAM.AC.UK
University of Cambridge

Abstract

Deep generative latent variable models have recently gained significant interest due to development of efficient and scalable variational inference methods. Variational methods involve maximization of the lower bound on the log-likelihood. However, until recently, directed latent variable model were difficult to train on large datasets. In this work, we provide an overview of several recent methods that have been developed for performing stochastic variational inference on large datasets. We provide an overview of theoretical and experimental results for providing a benchmark comparison of the variational inference methods based on feedforward neural networks. All of the approaches in comparison considers maximizing the variational lower bound by jointly training the model and the inference network. The methods in comparison are all applied on the MNIST dataset as a benchmark comparison. We implemented our own approach to the Auto-Encoding Variational Bayes algorithm (Kingma & Welling, 2013), and compared it with other approaches. Our experimental results show the significance of the different variance reduction techniques for the gradient estimator of the lower bound of the log likelihood.

Submitted as part of the Advanced Machine Learning coursework 2015/16, MPhil Machine Learning, Speech and Language Technology

1. Introduction

In our work we provide a unifying review of efficient inference and learning algorithms in directed generative models with many layers of hidden variables. It is known that directed latent variable models are difficult to train on large datasets since exact inference in such models is intractable. In our work, we compare the different approaches performing inference in deep directed graphical models.

Although directed graphical models are better able to generate observations directly, but there exists a lack of efficient learning algorithms for directed latent variable models. Recent work proposed approximate inference methods based on feedforward neural networks to maximize the variational lower bound on log-likelihood (Mnih & Gregor, 2014). We aim to provide a unifying review of such methods based on feedforward networks, and compare the efficiency of the different learning algorithms. In particular, we focus on the Auto-Encoding Variational Bayes (AEVB) algorithm (Kingma & Welling, 2013)

Recent efforts in machine learning research focused on developing scalable probabilistic models, where using directed graphical models we want to develop generative models that can scale to large datasets. Deep generative models are known to be able to generalize better to unknown data since the directed counterparts can better capture high level abstractions in the dataset. However, efficient inference algorithms for directed generative models has been a major problem.

2. Background

2.1. Variational Inference

The task of probabilistic inference in graphical models is to compute conditional probability distributions over hidden variables. Latent variable models provide a powerful approach to probabilistic modelling. A probabilistic model considers the joint distribution p(x,h) where x are the visible variables, and h are the hidden variables. The goal is to train a latent variable model $P_\theta(x,h)$ parameterized by θ. Since the posterior distribution $p(h|x)$ is complicated to work with, exact inference in such models is intractable, and hence maximum likelihood learning cannot be performed in such models.

Variational methods provide an approach to approximate inference. Since the exact posterior $P_\theta(h|x)$ is intractable, the key idea is to approximate this intractable distribution with a simpler tractable distribution $Q_\phi(h|x)$ with parameters ϕ. The distribution $Q_\phi(h|x)$ serves as an approximation to the exact posterior $P_\theta(h|x)$.

We can lower bound the marginal likelihood using Jensen's inequality:

$$\begin{aligned}\log P_\theta(x) &= \log \sum_h P_\theta(x,h) \\ =\geq \sum_h Q_\phi(h|x) \log \frac{P_\theta(x,h)}{Q_\phi(h|x)} &= E_Q[\log P_\theta(x,h) - \log Q_\phi(h|x)] \\ &= L(x,\theta,\phi)\end{aligned} \tag{1}$$

The lower bound can therefore be written as:

$$L(x,\theta,\phi) = \log P_\theta(x) - KL(Q_\phi(h|x), P_\theta(h|x)) \tag{2}$$

where Kullback-Leibler(KL) divergence is a non-symmetric measure of the differnece between the two probability distributions. The goal of variational infernece is to maximize the variational lower bound w.r.t the approximate distribution $Q_\phi(h|x)$ or equivalently minimize the KL divergence. In other words, the KL divergence is zero when $Q_\phi(h|x) = P_\theta(h|x)$

2.2. Recognition Model

A recognition model or inference network can learn an inverse mapping from the observations x to the hidden variables h. Recognition models can allow for faster convergence during training and test time. Previously in most approaches, the variational posterior $Q_\phi(h|x)$ for each observation was defined using its own set of variational parameters ϕ. However, recent methods are based on defining a *recognition model* or *inference network*. This means, a feedforward neural network will be used to compute the variational distribution from the observation. The inference network can perform the mapping from x to $Q_\phi(h|x)$ with the constraint that it is easy to sample from the inference network. Both the parameters θ of the generative model $p_\theta(x,h)$ and the recognition model $Q_\phi(h|x)$ comes from the neural network.

3. Approach

Having obtained a variational lower bound as shown in equation 2, the task is to optimize the lower bound; ie, we want to train the model by locally maximizing $L(x,\theta,\phi)$ w.r.t the model and inference parameters θ and ϕ. We want to optimize the recognition model in additon to learning the model parameters to perform efficient approximate posterior inference.

In order to optimize the lower bound, we want to differentiate the lower bound L with respect to inference network parameters ϕ and generative model parameters θ. The gradients of the variational bounds are given as:

$$\nabla_\theta L(x) = E_Q[\nabla_\theta \log P_\theta(x,h)] \tag{3}$$

for the model parameters θ, and for the inference network parameters is given as:

$$\nabla_\phi L(x) = E_Q[(\log P_{theta}(x,h) - \log Q_\phi(h|x)) \times \nabla_\phi \log Q_\phi(h|x)] \tag{4}$$

However, since both the gradients in equation 3 and 4 involves expectations, they are intractable. Even though Monte Carlo approximations to the gradients can be used, the variance of such gradient estimators are usually very high. In general, optimisation based approaches are better suited compared to sampling based Monte Carlo methods, and considering that variational methods are usually more efficient. In section 4, we will therefore consider the different approaches that can be taken for maximizing this lower bound on the log likelihood with respect to the parameters, such that a better approximation to the intractable posterior distribution can be obtained, for

both discrete and continuous variables for directed generative models.

4. Methods

4.1. Auto-Encoding Variational Bayes

A variational autoencoder for approximate posterior inference was recently proposed in (Kingma & Welling, 2013) involving a generative network and a recognition network. In this work, both the networks need to trained jointly such as to maximize the variational lower bound on the log likelihood. Since optimization involves finding gradients as shown in equation 3 and 4, in AEVB, it was shown that better optimization can be achieved by considering a reparameterization of the variational lower bound to obtain a reparameterized lower bound estimator. The random hidden variable h can be reparameterized $h \sim g_\phi(\epsilon, x)$ using a transformation that is differentiable $g_\phi(\epsilon, x)$. This trick is therefore applied to the lower bound $L(\theta, \phi; x)$ obtain the Stochastic Gradient Variational Bayes (SGVB) estimator $L_A(\theta, \phi; x)$, such that the reparameterized lower bound is now given as:

$$L_A(\theta, \phi; x) = \frac{1}{L} \sum_{l=1}^{L} \log_\theta(x^i, h^{i,l}) - \log q_\phi(h^{i,l}|x^i) \quad (5)$$

Considering minibatches, the following estimator of marginal likelihood lower bound is therefore given by:

$$L(\theta, \phi; X) \approx L_M(\theta, \phi; X^M) = \frac{N}{M} \sum_{i=1}^{M} \hat{L}(\theta, \phi; x^i) \quad (6)$$

We can therefore obtain gradient estimates $\nabla_{\theta,\phi} \hat{L}(\theta, \phi; x^i)$ and perform stochastic optimization based using gradient descent. Obtaining better estimates of the gradient therefore means that the approximate posterior distribution $Q_\phi(h|x)$ is a better approximation to the true posterior as measured by the KL divergence $KL(Q_\phi(h|x), P_\theta(h|x))$.

In the variational autoencoder (AEVB), the training procedure therefore considered a tradeoff between the data log likelihood $\log p(x)$ and the KL divergence from the true posterior. By doing this, the model learns a representation where it is easy to approximate the posterior inference. Additionally, in (Kingma & Welling, 2013), instead of directly computing the gradient of the log likelihood with respect to recognition

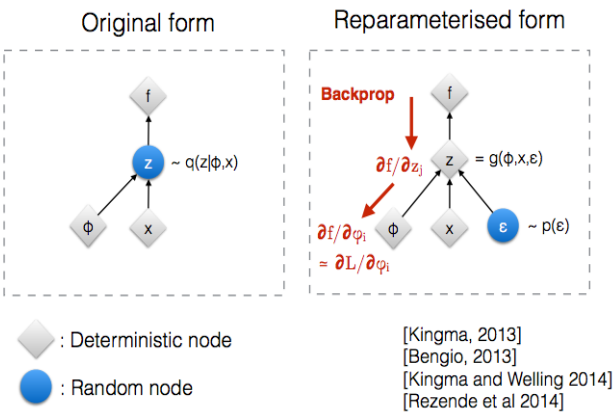


Figure 1. Explaining the Reparameterization Trick in the Auto-Encoding Variational Bayes Method

network parameters, (Kingma & Welling, 2013) considered a reparameterization of the recognition distribution in terms of auxiliary variables ϵ, such that the samples from the recognition model $Q_\phi(h|x)$ are a deterministic function of the inputs and auxiliary variables. This approach is valid for a variety of distributions, although only the Gaussian distribution was considered for experimental purposes. The AEVB algorithm based on stochastic gradients for variational Bayes (SGBV) has been successfully applied in learning to draw images in a realistic manner (Gregor et al., 2015). Figure 1 further shows the significance of the reparameterization trick and the significance of the backpropagation algorithm in the neural net parameterized by ϕ to find the approximate gradient estimator $\nabla_\phi L$.

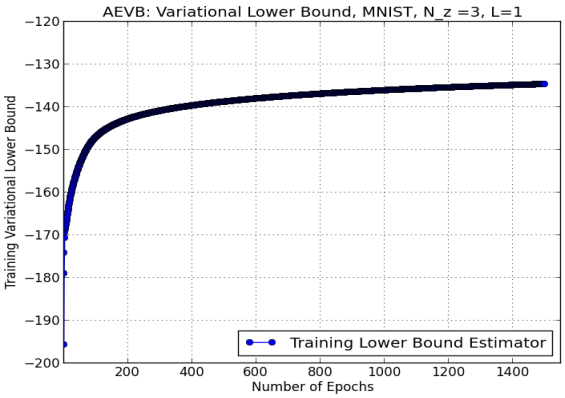
4.1.1. Local Reparameterization Trick

A local reparameterization trick for reducing the variance of the gradient estimators was further proposed in (Kingma et al., 2015). In practice, the performance of stochastic gradient ascent largely depends on the variance of the gradient estimates. Considering the parameters θ and ϕ, the optimization of these parameters for maximizing the lower bound involves uncertainty estimates in the paramters. By considering the local reparameterization trick, the global parameter uncertainty can be translated to local uncertainty per datapoint. Uncertainty in the global model and inference network parameters can be considerd as locan noise which can further be considerd as independent across datapoints in the mini batch sizes. Such parameterizations ensures that the variance in the gradient estimates is inversely proporitonal to the size of mini batches. Therefore, a better gradient estimator of the lower bound, with reduced variance and low computa-

tional complexity can be achieved by using the local reparameterization trick considerd in (Kingma et al., 2015). Such an approach to deep generative models and scalable variational inference was further used to develop a probabilistic model for semi-supervised learning (Kingma et al., 2014).

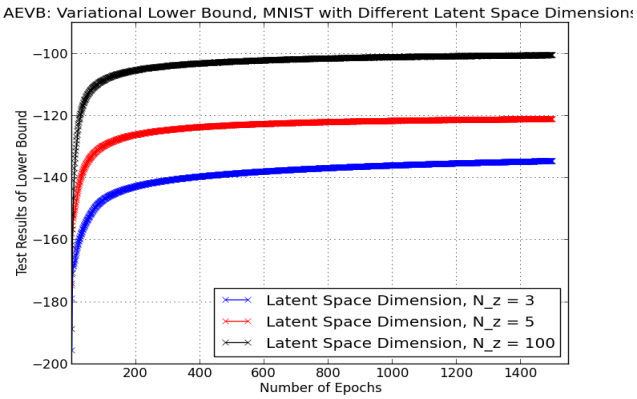
4.1.2. AEVB: Experimental Results and Discussion

Using the MNIST dataset, a generative model of images was trained and we obtained results of the variational lower bound. In our algorithm, the same neural network architecture was used for both the generative model and the recognition model, using 400 hidden units at each layer. Considering the stochastic variational Bayes approach, the parameters θ and ϕ were obtained by maximizing the gradient $\nabla_{\theta,\phi} L(\theta, \phi, X)$ of the lower bound estimator. For all the experiments, we considered mini batch sizes of 100, and a learning rate of 0.01.



In the variational autoencoder, the number of hidden units refers to the number of hidden layers of the neural network that is used in the encoder and decoder. We then analysed how the generalization performance depends on the dimensionality of the latent space N_z. Figure 3 shows that generalisation on MNIST indeed improves with more latent variables, and does not lead to overfitting as the generative model contains higher number of latent variables that needs to be inferred.

Initially, we considered using one sample ($L = 1$) per datapoint. However, for comparison with other methods described below, we further considered using more than one samples per datapoint. However, the AEVB approach does not consider any importance sampling or weighting based approach. The results below shows how the maximization of the lower bound is dependent on the number of samples per datapoint. Interestingly, the test results for different samples are the same, showing that the generalisation performance for AEVB is independent of the number of samples taken from the recognition model. This is further justified since AEVB does not consider any weighted sampling approach, and hence taking $L > 1$ does not affect test set performance.



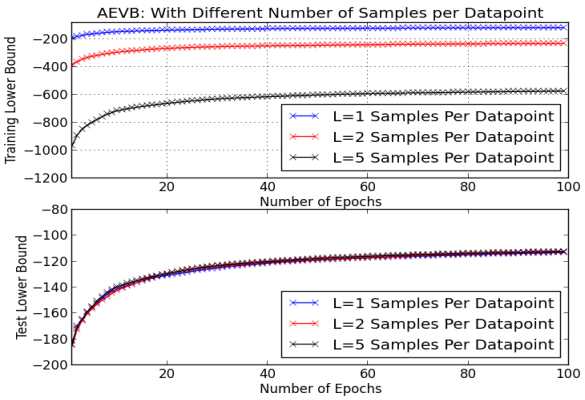


Figure 4. Training lower bound results for different number of samples L per datapoint

4.2. Neural Variational Inference (NVIL)

Another approach to reducing the variance of the gradient estimates for maximizing lower bound using gradient-based optimisation was considerd in Neural Variational Inference (NVIL) (Mnih & Gregor, 2014). Neural Variational Inference and learning (Mnih & Gregor, 2014) also considers training a recognition network parameterized by a neural network to approximate the posterior distribution. However, in addtiion to the model and inference network, NVIL further considers a third network to predict reward baselines in the context of the REINFORCE algorithm (Williams, 1992). It also uses the same variational

objective as the variational autoencoder (AEVB), except that a baseline function is used (inspired from reinforcement learning literature) to reduce variance instead of considering a reparameterization trick. However, one drawback of the NVIL approach is that the estimator requires learning additional parameters from the baseline function for reducing variance of gradient estimates.

Similar to the variational autoencoder (AEVB), NVIL also considers using a feedfoward network for exact sampling from the variational posterior distribution. However, one major difference is that NVIL considers both sampling based and variational methods for training the directed graphical model. Samples from the inference or recognition network are used for obtaining the Monte-Carlo estimates of the gradients, where the inference network is trained jointly with the model by maximizing the variational lower bound.

Considering the gradient w.r.t the inference network parameters, ϕ shown in equation 4, the difference between the two distributions $\log P_\theta(x,h) - \log Q_\phi(h|x)$ can be considered as a learning signal for the inference network parameters:

$$l_\phi(x,h) = \log P_\theta(x,h) - \log Q_\phi(h|x) \tag{7}$$

Reducing the baseline c from the learning signal $l_\phi(x,h)$, the gradient from equation 4 can therefore be written as:

$$\nabla_\phi L(x) = E_Q[(l_\phi(x,h) - c)\nabla_\phi \log Q_\phi(h|x)] \tag{8}$$

The gradient variance can be further reduced by making the baseline a function of the observations $C_\phi(x)$, which is further implemened using a neural network and trained to minimise the expected squared error of the learning signal $E_Q[(l_\phi(x,h) - C_\phi(x) - c)^2]$. This approach incorporating baselines to variance reduction is therefore similar to using control variates.

4.3. NVIL: Experimental Results and Discussion

In this section, we first present the results of the effectiveness of NVIL algorithm on the MNIST dataset. For the initial set of experiments, we considerd using 200 latent variables and used 1500 epochs with a batch size of 100 on the MNIST dataset. The input dependent baseline functions for reducing variance was implemented using a neural network with single hidden layer of 400 tanh units.

We first considered training the generative model along with the inference network using different optimisation approaches. We compared the effectiveness of the stochastic gradient ascent compared to using adaptive learning rate based RMSProp and AdaGrad approaches. Since tuning the learning rate is an expensive process in training large generative models, and hence Adagrad and RMSProp can be effective in automatically adjusting the learning rate parameter. For the stochastic gradient ascent, the learning rate was chosen to be $1e^{-3}$, as given in the original work. Due to time constraints, we did not consider a validation set for fine tuning and selecting the learning rate parameter. Figure 5 shows the dependence in convergence in maximizing the lower bound based on different optimisation techniques. We used adaptive learning rates since fine-tuning the learning rate is a difficult task requiring separate validation set.

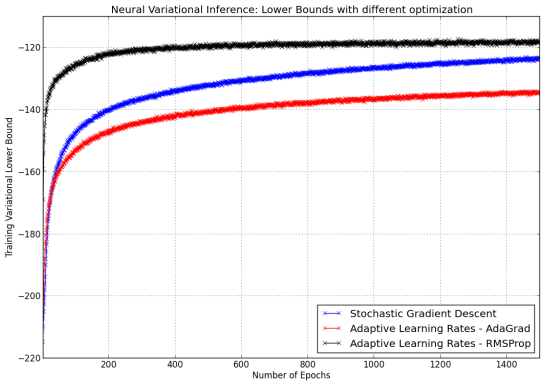


Figure 5. Fixed and Adaptive Learning rate based optimisation of the lower bound estimator using the Neural Variational Inference Algorithm

Based on results in figure 5, it is undestood that while both adaptive learning rates converge significantly more quickly than SGD, in this application RMSProp is able to quickly settle into a significantly higher log-likelihood than AdaGrad, the convergence rate of which quickly slows down. This may be because AdaGrad quickly converges for sparse parameters, even though in a neural network setting many of the parameters would be dense.

We further show the estimates of the lower bound depending on the number of latent variables used in the directed generative model. We considered a single layer with varying number of latent variables. Figure 6 further shows that the optimisation of the lower bound is almost independent of the number of

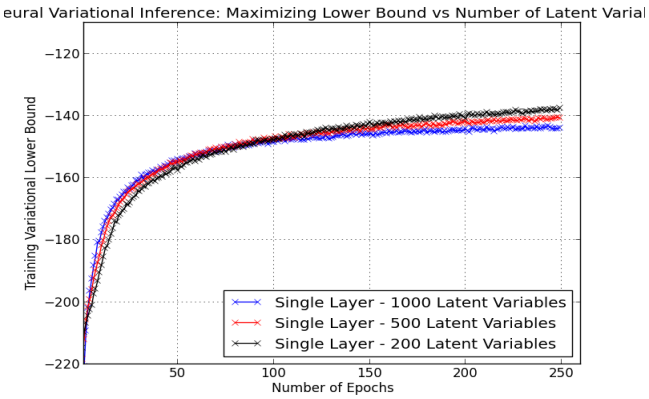


Figure 6. Dependence of NVIL algorithm on the number of latent variables used in the model

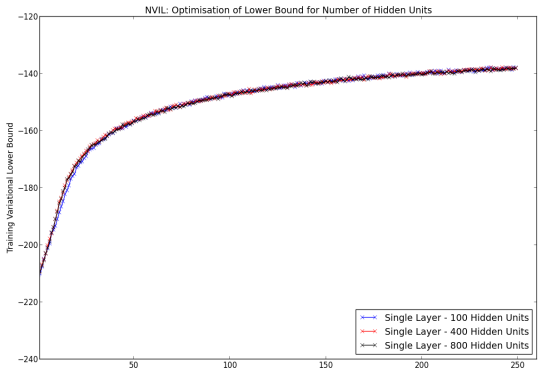


Figure 7. NVIL Optimisation of Lower Bound Estimator depending on the number of hidden units

latent variables used in the model. The generalisation performance (not shown here) also shows no signs of overfitting for higher number of latent variables. This is similar to the results previously observed in figure 3 for the variational auto-encoder. Figure 7 also shows that the performance of the NVIL method is also independent of the number of hidden units used in each layer of the inference network.

The significance of the variance reduction technique using a baseline network can be notably observed when comparing with other variance reduction techniques as discussed later in section 5

4.4. Importance Weighted Auto-Encoders (IWAE)

Perhaps the most closely related to the variational autoencoder approach is the Importance Weighted Autoencoder (IWAE) (Burda et al., 2015). The IWAE uses the same architecutre as VAE, except that the lower bound is derived from importance weighting. However, one difference is that instead of using a single sample from the inference network of the VAE architecure, IWAE uses multiple samples such that the recognition network now produces multiple approximate samples and their weights are then aveaged. By doing so, IWAE can achieve a better flexible approximate posterior distribution to model the true posterior distribution of the generative model.

The IWAE considers multiple stochastic hidden layers for the neural networks, compard to VAE which considerd only one layer of the network. Similar to VAE, the conditional distributions here are also considered as Gaussians. In VAE the means and variances would be computed by a deterministic feedforward neural network; however, in IWAE, the Gaussian recognition distribution $q(h^l|h^{l-1}, \theta)$ whose means and covariances are now computed from the states of the hidden units at the previous layer.

The importance weighted autoencoder also considers a generative network and a recognition network. However, the training procedure for IWAE is based on a different lower bound of $\log p(x)$. Since it considers multiple samples from the recognition network, the lower bound is now based on the k-sample importance weighting of the log likelihood given by:

$$L_k(x) = E_{h_1,h_2,..h_k \sim q(h|x)}[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(x,h_i)}{q(h_i|x)}] \quad (9)$$

where $h_1, h_2, ...h_k$ are now independent samples from the inference network $q(h|x)$. The term inside the sum can be written as w_i since it is the unnormalized importance weights for the joint distribution. The lower bound on the marginal log likelihood, considering Jensen's inequality, can therefore be written as:

$$L_k = E_Q[\log \frac{1}{k} \sum_{i=1}^{k} w_i] \leq \log E_Q[\frac{1}{k} \sum_{i=1}^{k} w_i] \quad (10)$$

However, since the generative model is parameterized by θ, we write the weights w as $w(x, h, \theta) = \frac{p(x,h|\theta)}{q(h|x,\theta)}$. Again, to optimize the lower bound, we need to find the estimate of $\nabla_\theta L(x)$. The gradient estimate, considering importance weighting for optimizing the lower

bound is therefore given as:

$$\nabla_\theta L(x) = \frac{1}{k}\sum_{i=1}^{k} \nabla_\theta \log w(x, h(\epsilon_i; x, \theta), \theta) \qquad (11)$$

where the mapping h is represented as a neural network, and equation 11 is a Monte-Carlo estimator for maximizing the lower bound based on the importance weighted autoencoder algorithm. The IWAE algorithm, being a variant of VAW, has been shown to achieve better generalisation performance compared to the variational autoencoder method due to its ability to learn richer latent feature representations.

4.4.1. IWAE: Experimental Results and Discussion

The importance weighted autoencoder provides a good benchmark for comparing the generative performance between VAE and IWAE using the MNIST dataset. For experimental purposes, all the stochastic hidden layers used Gaussian distributions with a diagonal covariance, and a tanh non-linear activation function was used for each of the hidden units. Similar to the original work as in (Burda et al., 2015), we used Adam for optimisation using minibatches of size 20 and $\epsilon = 10^{-4}$. For our results, we also considered the training process to proceed with 3^i passes over the data with a learning rate of 0.001.

For IWAE, we ran our models with 50 units in the hidden layer, considering 1 or 2 hidden layers in both the generative and recognition model. As done in original work, we also used minibatches of size 20 using Adap optimisation. For the importance sampling, we considered comparing how the number of samples taken from the inference network affects the performance of IWAE using the gradient estimation considered above. As done originally, we compared taking samples of $1, 5 or 50$ for both $L = 1$ and $L = 2$ stochastic hidden layers.

Figure 8 shows that taking more than 1 samples $k > 1$ considerably improved the performance of IWAE on the MNIST dataset. Previously, for AEVB, we showed that generalisation performance does not improve for different number of samples. Using the code available for IWAE, that includes comparison with AEVB, figure 8 further shows that with $k = 1$, both the methods achieve same performance. Indeed, IWAE can improve generalisation performance by taking higher number of samples, while following similar encoder decoder architecture as the AEVB.

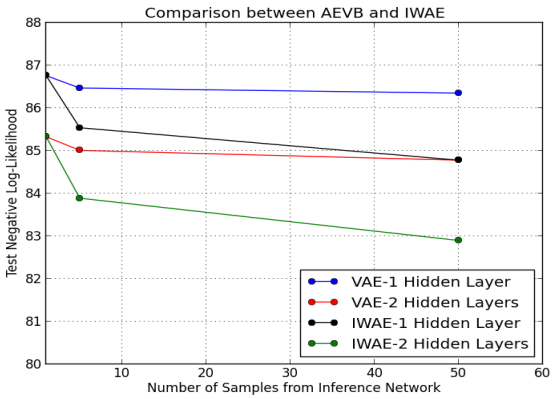


Figure 8. Significance of Number of Samples from Inference Network - Comparing AEVB and IWAE for 1 and 2 stochastic hidden layers

4.5. Reweighted Wake-Sleep

The reweighted wake sleep algorithm (Bornschein & Bengio, 2014) is extended from the wake-sleep algorithm (Hinton et al., 1995), and similar to the importance weighted auto-encoder, it considers obtaining good estimates of the gradient of the lower bound by sampling the latent variables multiple times from the recognition model. Therefore, the updates to the generative network are similar to the gradient estimates based on the lower bound considered in IWAE in equation 11. Therefore, RWS is similar to the importance weighted autoencoder since the estimator of the log likelihood is based on importance sampling.

The wake-sleep algorithm was initially proposed by (Hinton et al., 1995) which was thought of as optimising a biased estimator of the gradient. The wake-sleep algorithm was initially proposed for training generative models like the Helmholtz machines and deep belief network. The original wake-sleep algorithm considered the following variational bound given by:

$$\log p(x) \geq \sum_h q(h|x) \log \frac{p(x,h)}{q(h|x)} \qquad (12)$$

The wake-sleep algorithm considers maximizing this variational bound, where the wake phase corresponds to maximizing w.r.t p and in the sleep phase, the update w.r.t q minimises the reversed KL divergence $KL(p(h|x)||q(h|x))$.

The reweighted wake-sleep algorithm instead considers formulating the likelihood as an importance weighted

average, such that the unbiased estimator of the marginal likelihood is now given as:

$$p(x) = \sum_h q(h|x)\frac{p(x,h)}{q(h|x)} \approx \frac{1}{K}\sum_{k=1}^{K}\frac{p(x,h^k)}{q(h^k|x)} \tag{13}$$

Training of the reweighted wake sleep algorithm consists of two phases. Since both the generative model p and recognition model q are parameterized by θ and ϕ, at first the model p_θ is updated for a given q_ϕ. The gradient $\nabla_\theta L_p(\theta, x)$ of the marginal likelihood $L_p(\theta, x) = \log p_\theta(x)$ is given by:

$$\begin{aligned}\nabla_\theta L_p(\theta,x) &= \frac{1}{p(x)}E_{h\sim q(h|x)}[\frac{p(x,h)}{q(h|x)}\nabla_\theta \log p_\theta(x,h)] \\ \nabla_\theta L_p(\theta,x) &= \sum_{k=1}^{K} w_k \nabla_\theta \log p(x,h^k)\end{aligned} \tag{14}$$

Once the model p_θ is updated, the variance of the estimator can then be reduced by considering updates of q_ϕ for a given p_θ. The recognition network q_ϕ can then be trained by using maximum likelihood learning with the loss $L_q(\phi, x, h) = \log q_\phi(x|h)$. Similar to above, gradient w.r.t ϕ can again be obtained considering importance sampling given by:

$$\nabla_\phi L_q(\phi, x) = \sum_{k=1}^{K} w_k \nabla_\phi \log q_\phi(h^k|x) \tag{15}$$

Additionally, in RWS we consider drawing K samples from the inference network while NVIL considered drawing a single sample. One advance of the reweighted wake sleep algorithm compared to NVIL is that it does not require maintaining any baseline network with additional parameters to reduce variance of gradient estimates.

The reweighted wake sleep algorithm therefore provides a better training procedure for deep generative models and obtains a lower bias lower variance estimator of the log-likelihood gradient at the experience of higher number of samples from the inference network.

4.5.1. RWS: Experimental Results and Discussion

Using code available online, we re-ran the experiments for the reweighted wake-sleep algorithm. Similar to original work, we also used the binarized MNIST dataset and stochastic gradient descent was used with minibatch sizes of 25. The RWS algorithm used $K = 5$ samples during training. The p and q networks in RWS consisted of three hidden layers. In the original work, the significance of the number of samples used during training were analysed. Due to time constraints, in this work, we only analysed how the lower bound estimator on the test set varied with the number of epochs having trained the model with the RWS algorithm. Figure 9 below shows the test set lower bound maximization with the number of iterations using the wake-sleep algorithm. This will later be used in section 5 for comparison with other methods.

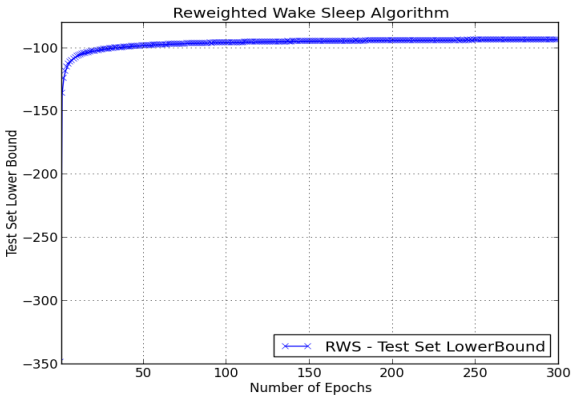


Figure 9. Lower Bound estimator on the MNIST using the Reweighted Wake-Sleep Algorithm

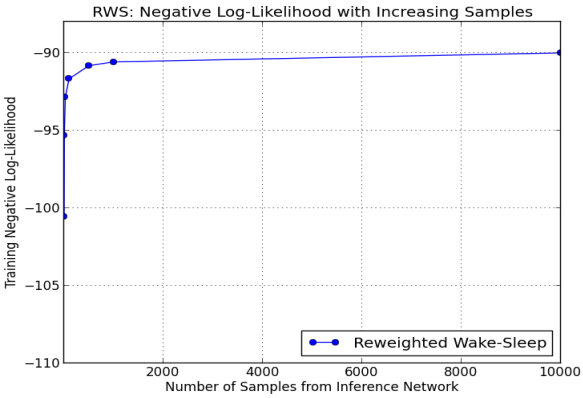


Figure 10. Significance of the number of samples from inference network on final negative log-likelihood

Figure 10 further shows the impact on the final negative log-likelihood as the number of samples are increased from the inference network. Figure 10 suggests that the training of the model, including the optimization of the θ and ϕ parameters are highly dependent on the number of samples taken.

4.6. Related Work

Similar to the algorithms considered above, (Rezende et al., 2014) also considers scalable inference and learning in directed generative models. (Rezende et al., 2014) further developed stochastic backpropagation, for backpropagating the gradients for joint parameterisation of parameters of generative and recognition models. Stochastic Backpropagation (Rezende et al., 2014) can be used for efficient inference as it considers computing gradients involving expectations through random variables. It therefore considers the similar task of inference methods with continuous latent variables by introducing a recognition model and deriving a lower bound estimator of the marginal likelihood, with the only difference being the use of a modified approach to backpropagation algorithm for the inference network.

Another related work is to consider the multiple sample approach of IWAE for generative models with discrete latent variables. Maximization of the lower bound of the intractable marginal log likelihood is often done by estimating gradients using samples from the infernce network or variational posterior distribution. However, the variational posterior is often not flexible. If the bound is based on single sample estimates, then the samples that explains the observations poorly are often heavily penalised. Therefore, the variational posterior distribution covers only the high probability areas of the true posterior distribution. Using VIMCO discussed in (Mnih & Rezende, 2016), multiple samples are considered. This is related to the approach of IWAE. The above effect is minimised by averaging over multiple samples to compute the marginal likelihood estiamtes such that the lower bound is tighter as the number of samples increases. This approach based on averaging over independent samples is called Monte-Carlo objectives as discussed in (Mnih & Rezende, 2016). It introduces an unbiased gradient estimator for multiple-sample objective functions and therefore can reduce variance of the gradient estimator, instead of having to introduce a baseline network such as in NVIL that introduces additional parameters.

5. Comparison of Experimental Results

Figure 11 shows the generalisation performance of the different scalable variational infernece methods on the MNIST dataset. Figure 11 shows the different convergence guarantees of the maximization of the lower bound estimator by jointly optimizing for the parameters θ and ϕ in generative and recognition model.

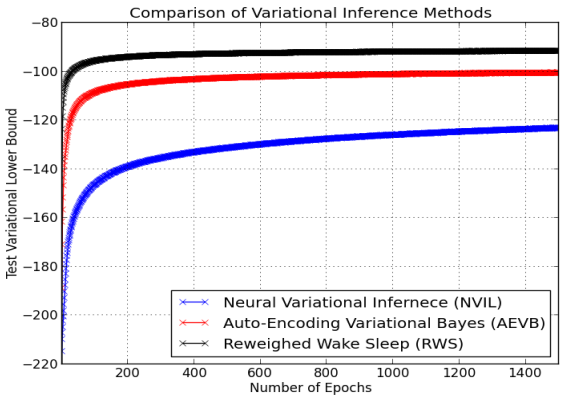


Figure 11. Comparison of Generalisation performance of **AEVB**, **NVIL** and **RWS**

Comparing for the methods above, our results show that the reweighted wake-sleep algorithm, considering optimisation using multiple-samples can outweigh the other methods based on the reparameterisation trick and the baseline network. Since IWAE is a similar multiple-sampling based approach, and due to time constraints, we only consider comparison between **AEVB**, **NVIL** and **RWS** here. The different algorithms in figure 11 have different parameter configurations. In figure 11, AEVB uses a latent space dimension of 100, while NVIL uses 500 latent variables, with a single layer consisting of 400 hidden units, and optimisation performed by stochastic gradient ascent. For the reweighted wake-sleep algorithm, we trained the model with $K = 100$ training samples, we used $K = 5000$ samples with a minibatch of size 25. Previously, for the experimental results comparing IWAE and AEVB as shown in figure 8 previously, we also showed that the importance sampling based approach can obtain better gradient estimates for maximizing the lower bound compared the AEVB method.

6. Discussion and Future Work

In our work, we provided a unifying review of some of the recent approaches for estimating the variational lower bound for efficient approximate inference in deep directed generative models. In AEVB, we demonstrated how it uses a gradient estimator (SGVB) and a reparameterization trick such that the gradient can be found for standard stochastic gradient ascent methods. While AEVB can be used in continuous latent variable models, NVIL, on the other hand, is only applicable to discrete binary latent variables. However, instead of introducing a reparameterization trick, NVIL simply considered including a baseline network in the gradient estimator to reduce variance

and jointly optimize both the inference and generative network. Once considerable drawback, however, is the need to also estimate the parameters of the feedforward baseline network. The NVIL however can approximate the true posterior with more flexible distributions, since it combines both sampling and variational methods.

The importance weighted auto-encoder is an elegant variant of the VAE, as it uses the same architecture, but instead considers multiple samples to maximize a tighter log likelihood lower bound that can be derived using an importance sampling approach. Comparing AEVB with IWAE, it was also shown that for the same number of stochastic hidden layers, IWAE can learn richer latent representations and acheive a better test performance compared to AEVB. The use of multiple samples to approximate the posterior in IWAE gives it better flexibility to model complex posterior distributions.

The comparison of the different methods shown in figure 11 shows that the multiple sampling based approach, considering importance sampling, can generalise better compared to ther other methods that uses a baseline network and the reparameterisation trick. Figure 11 further suggests that the gradient estimates of the lower bound are better estimated considering importance sampling, and even though NVIL and AEVB considered variance reduction techniques, they cannot outweigh benefits of optimisation based on multiple samples. Figures 11 and 8 shown previously suggests that approaches based on multiple-samples can achieve gradient estimates with a lower variance, compared to the other variance reduction techniques.

6.1. Extensions

Variational Inference for Monte-Carlo Objectives (Mnih & Rezende, 2016): VIMCO is another scalable variational inference method that could be considered as a benchmark for comparison. VIMCO is further related to IWAE, where it also considers an importance sampling approach to estiamte the log likelihood. However, VIMCO considers extending IWAE to discrete latent variables. It would be interesting to analyse how VIMCO performs better than IWAE for discrete variables, and why it may provide a more effective gradient estimator compared to single sampling based approaches such as NVIL and AEVB.

Wake-Sleep Algorithm (Hinton et al., 1995): The original AEVB work uses the wake-sleep algorithm as a benchmark for comparison, as it employs the same recognition model as the wake-sleep algorithm. It would be interesting to observe how the NVIL and multiple sampling approaches like IWAE and RWS may compare to the wake-sleep algorithm on the MNIST dataset.

Other Datasets: We only compared the methods above on the full MNIST dataset. If time allows, it would be interesting whether the same trend and comparision in results are also observed on other datasets, such as the Frey Faces dataset or the Omniglot dataset of handwritten characters. By comparing our methods applied on other datasets, a more uniform comparison benchmark can be achieved for all the scalable variational inference methods considered for deep directed generative models.

6.2. Future Work

All the recognition models used in the above methods uses a neural network with weights ϕ to find an approximate to the true posterior. For example, the AEVB algorithm uses the application of the backpropagation algorithm to find an approximate gradient estimator $\nabla_\phi L$. All the weights in these networks are considered as point estimates. In all the above methods, variational inference has been applied to the stochastic hidden units of the autoencoder. One drawback of having point estimates in deep neural networks is that the optimisation is much more difficult on the larger scale, making the network prone to overfitting.

Uncertainty in Weights of Neural Network: An interesting extension of work might be to consider Bayesian neural networks for the parameterisation of the recognition model. In other words, consider the weights ϕ in the inference network as weight distributions and consider Bayes By Backprop as shown in (Blundell et al., 2015). It considers introducing uncertainty in the weights of the network, and perform variational approximation for exact Bayesian updates. It might be interesting to analyse how uncertainty in the weights might affect the methods considered above, as the inference network would now have weight distributions.

Probabilistic Backpropagation: As discussed above, backpropagation needs to be used for the inference network to estimate an approximation to the gradient $\nabla_\phi L$. If we want to consider Bayesian neural networks for the recognition model, we can consider

learning in Bayesian neural networks using the probabilistic backpropagation algorithm (Hernández-Lobato & Adams, 2015). Probabilistic backpropagation works by propagating probabilities forward through the network and then propagate the gradients of the marginal likelihood backwords w.r.t parameters of the posterior approximation. Using PBP, we can therefore consider uncertainty in the weights of the inference network, and analyse how the uncertainty calibration in inference network affects performance in scalabale variational inference in deep directed generative models.

7. Summary

In this work, we have therefore provided a unifying review and comparison of different methods for training directed latent variable models. We compared the variance reduction techniques for obtaining better gradient estimates for maximizing the lower bound estimator of the marginal log-likelihood in deep generative models. All the algorithms considered training an auxiliary neural network to perform inference by optimizing the variational bound. The algorithms considered joint optimisation of the generative and the recognition model. Our experiments were carried out on the MNIST dataset, and we compared the different variance reduction techniques in scalable variational inference methods. We compared the generality and flexibility of the different approaches for performing inference in directed latent variable models. We compared the different with the Auto-Encoding Variational Bayes (AEVB) method for performing approximate inference in directed generative models.

Acknowledgments

We would like to thank the MPhil MLSALT program at University of Cambridge for providing this course on Advanced Machine Learning. We thank Richard Turner and Zoubin Ghahramani, along with other guest lecturers for taking this course. We thank Yinghzen Li and Richard Turner for providing useful feedback and suggestions for directions of work in this project.

References

Blundell, Charles, Cornebise, Julien, Kavukcuoglu, Koray, and Wierstra, Daan. Weight uncertainty in neural networks. *CoRR*, abs/1505.05424, 2015. URL http://arxiv.org/abs/1505.05424.

Bornschein, Jörg and Bengio, Yoshua. Reweighted wake-sleep. *CoRR*, abs/1406.2751, 2014. URL http://arxiv.org/abs/1406.2751.

Burda, Yuri, Grosse, Roger B., and Salakhutdinov, Ruslan. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015. URL http://arxiv.org/abs/1509.00519.

Gregor, Karol, Danihelka, Ivo, Graves, Alex, Rezende, Danilo Jimenez, and Wierstra, Daan. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1462–1471, 2015. URL http://jmlr.org/proceedings/papers/v37/gregor15.html.

Hernández-Lobato, José Miguel and Adams, Ryan. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pp. 1861–1869, 2015. URL http://jmlr.org/proceedings/papers/v37/hernandez-lobatoc15.html.

Hinton, Geoffrey E., Dayan, Peter, Frey, Brendan J., and Neal, Radford M. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1158–1161, 1995.

Kingma, Diederik P. and Welling, Max. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. URL http://arxiv.org/abs/1312.6114.

Kingma, Diederik P., Mohamed, Shakir, Rezende, Danilo Jimenez, and Welling, Max. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3581–3589, 2014. URL http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-model

Kingma, Diederik P., Salimans, Tim, and Welling, Max. Variational dropout and the local reparameterization trick. *CoRR*, abs/1506.02557, 2015. URL http://arxiv.org/abs/1506.02557.

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1791–1799, 2014. URL http://jmlr.org/proceedings/papers/v32/mnih14.html.

Mnih, Andriy and Rezende, Danilo Jimenez. Variational inference for monte carlo objectives. *CoRR*, abs/1602.06725, 2016. URL http://arxiv.org/abs/1602.06725.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1278–1286, 2014. URL http://jmlr.org/proceedings/papers/v32/rezende14.html.

Williams, Ronald J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL http://dx.doi.org/10.1007/BF00992696.