Variational Inference in Deep Directed Latent Variable Models

Riashat Islam & Jiameng Gao & Vera Johne
University of Cambridge

UNIVERSITY OF CAMBRIDGE

Overview

We provide a unifying overview to compare methods for efficient training of directed latent variable models. Directed latent variable models can represent complicated marginal distributions over observed variables. Additionally, probabilistic flexible deep neural networks can represent complex multi-modal distributions. In all the approaches, these two areas are combined together. Due to intractable posterior distribution, variational inference methods are used to approximate the posterior distribution.

Key Idea: A neural network parameterized model $q_\phi(z)$ or $q_\phi(z|x)$ is used to infer posterior. The auxiliary neural network is trained to perform inference by maximizing the variational lower bound. We compare the different approaches of performing variational inference based on optimizing the lower bound.

Variational Inference

Considering large datasets and latent space z, the exaxt posterior distribution $p_\theta(z|x)$ is intractable. Considering the latent variable model $P_\theta(x,z) = P_\theta(x|z)P_\theta(z)$, the goal is to learn the maximum likelihood parameters θ and infer the latent variables z for observations x.

The log marginal likelihood to the lower bounded as follows:

$$\log P_\theta(x) \geq E_Q[\log P_\theta(x,z) - \log Q_\phi(z|x)] = L_{\theta,\phi}(x) \tag{1}$$

In variational inference, a non-gradient based optimisation technique, the goal is to minimize the KL divergence $D_{KL}(Q_\phi(z|x)||P_\theta(z|x))$, by alternating between maximizing the lower bound $L_{\theta,\phi}(x)$ w.r.t variational posterior $Q_\phi(z|x)$ and the model parameters θ.

Auto-Encoding Variational Bayes (AEVB)

By considering the lower bound objective function from equation 1 as $L = E_{Q_\phi(z|x)}[f_\phi(x,z)]$, the **stochastic gradient-based variational inference** method considers stochastic gradient ascent by finding the gradient $\nabla_\phi L$.

$$\nabla_\phi L = E_{Q_\varphi(z|x)}[(\nabla_\phi \log q_\phi(z|x)) f_\phi(x,z)] = (\nabla_\phi \log q_\phi(z^l|x)) f_\phi(x,z^l) \tag{2}$$

where $z^l \sim q_\phi(z|x)$. This means that we take a single sample z^l from $q_\phi(z|x)$ and then can approximate $\nabla_\phi L \approx \nabla_\phi f_\phi(x,z^l)$.

Instead of considering variance reduction techniques (as shown in NVIL), and other good gradient estimators using multiple samples (IWAE, RWS), AEVB considers a reparameterization technique to reduce variance of gradient estimator.

The reparameterized gradient estimator can be obtained by first sampling ϵ^l from $p(\epsilon)$. Considering the hidden variable z parameterized by $z \sim g_\phi(\epsilon, x)$ such that the transformation $g_\phi(\epsilon, x)$ is differentiable. This means z^l can be sampled from $g_\phi(\epsilon)$, such that $z^l \sim q_\phi(z|x)$.

Considering the variational distribution $q_\phi(z|x) = N(\mu, \sigma^2)$ as a Gaussian distribution, the parameters μ and σ can be otained from a multi-layer neural network. The lower bound $L_{\phi,\theta}(x)$ of $\log p(x)$ can therefore be jointly optimized w.r.t θ and ϕ.

Neural Variational Inference (NVIL)

NVIL similarly considers using an inference network with a feedforward model $Q_\phi(h|x)$ and the generative model and the inference network is trained jointly by updating the parameters θ and ϕ to increase the variational lower bound $L_{\theta,\phi}(x)$. Instead of using the reparameterisation trick, NVIL considers finding gradients directly and then using a neural network parameterized baseline network as a variance reduction technique. The gradients w.r.t θ and ϕ are given as:

$$\nabla_\theta L_{\theta,\phi}(x) = E_Q[\nabla_\theta \log P_\theta(x,z)] \tag{3}$$

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

Reducing Variance Using Baseline: NVIL considers choosing an input-dependent baseline $b_\psi(x)$ which is also a neural network that can be seen as capturing $\log P_\theta(x)$. A baseline function is chosen that can reduce the variance of the gradient estimator $\nabla_\phi L_{\theta,\phi}(x)$ such that the resulting learning signal $\log P_\theta(x,z) - \log Q_\phi(z|x) - b_\psi(x)$ is close to zero. The unbiased estimator of $\nabla_\phi L_{\theta,\phi}(x)$ for any baseline is therefore given as:

$$\nabla_\phi L_{\theta,\phi}(x) = (\log P_\theta(x,z) - \log Q_\phi(z|x) - b_\psi(x))\frac{d}{d\phi}\log Q_\phi(z|x) \tag{5}$$

Reweighted Wake-Sleep (RWS)

The reweighted wake-sleep algorithm extended from wake-sleep algorithm considers an importance sampling estimate due to sampling of latent variables multiple times from recognition model. The estimator of the parameter gradient is:

$$\nabla_\theta L_p = \sum_{k=1}^{K} \hat{w_k} \frac{d}{d\theta}\log p(x, z^k) \tag{6}$$

with $z^k \sim q(z|x)$ and $w_k = \frac{p(x,h^k)}{q(z^k|x)}$. In the sleep phase q-update, consider $x, z \sim p(x,z)$ and then calculate the gradient $\nabla_\phi L_q$. The wake-phase q-update is:

$$\frac{d}{d\phi} L_q = \sum_{k=1}^{K} \hat{w_k} \frac{d}{d\phi}\log q(x, h^k) \tag{7}$$

The reweighted wake-sleep algorithm is based on $\arg\min_\phi KL(p_\phi(z|x)|q_\phi(z|x))$ and obtains a lower bias lower variance gradient estimation for maximizing the lower bound.

Importance Weighted Auto-Encoder (IWAE)

IWAE derives the lower bound gradient estimator from importance weighting, and uses multiple samples from the inference network instead of single samples. IWAE can obtain better flexible approximate posterior distribution to model the true posterior.

$$L_k = E_Q[\log \frac{1}{k}\sum_{i=1}^{k} w_i] \leq \log E_Q[\frac{1}{k}\sum_{i=1}^{k} w_i] \tag{8}$$

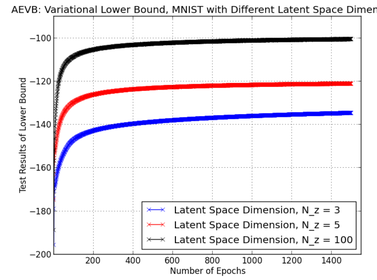
where the weights w are $w(x,h,\theta) = \frac{p(x,h|\theta)}{q(h|x,\theta)}$. To optimize the lower bound, the gradient estimate considering importance weighting is:

$$\nabla_\theta L(x) = \frac{1}{k}\sum_{i=1}^{k} \nabla_\theta \log w(x, h(\epsilon_i; x, \theta), \theta) \tag{9}$$

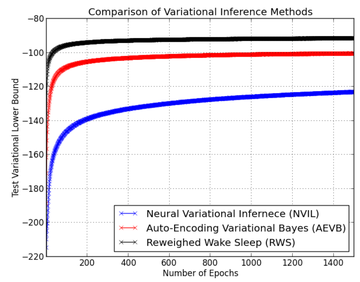
where the mapping h is represented as a neural network, and equation 9 is a Monte-Carlo estimator for maximizing the lower bound based on the importance weighted autoencoder algorithm.

Experimental Results

Figure below shows test set performance dor different dimensionality of latent space. More latent variables does not cause overfitting (due to regularizing effect of lower bound). It improves performance of AEVB, and is independent of NVIL.



Comparison of maximization of lower bound of variational inference, using an inference network for neural network parameterized variational distribution. Comparison between **AEVB vs NVIL vs RWS** for directed generative modelling on MNIST dataset.



Summary

- Perform efficient inference in directed probabilistic models with both continuous and discrete latent variables
- Compared variance reduction techniques for the gradient estimator for comparing the marginal log-likelihood lower bound
- Multiple sample-based objectives using RWS generalises better for maximizing variational lower bound
- RWS and IWAE implements effective variance reduction compared to reparameterisation trick in AEVB and using baseline network in NVIL