Advances in Variational Inference

Ayush Gupta
Department of Chemistry
IIT Kanpur
gayush@iitk.ac.in

***Abstract*— The focus of this project is on approximate inference in probabilistic machine learning namely Variational Inference (VI). We begin with a trivial assumption in Mean Field Theory, then we move on to other advance variational inference including Black Box VI, Stochastic VI and, then finally head to Variational Autoencoders.**

I. INTRODUCTION

A. Variational Inference

The main idea behind variational methods is to pick a family of distributions over the latent variables with its own variational parameters $q \in Q$ and then, find the setting of the parameters that makes Q close to the posterior, by minimising the Kullback-Leibler Divergence.

We will then query q rather than p in order to get an approximate solution

B. Mean Field Theory

The simplest variational family of distributions to work with is the Mean Field Variational Family,wherein each hidden variable is independent and governed by its own parameter. In mathematical terms:

$$\prod_{i=1}^{N} q(z_i|\phi_i) = q(z|\phi)$$

As it divides the latent variables into N groups, assuming that each latent variable is independent of each other can destroy the structure. Hence it's not preferred.

II. OPTIMIZATION

The main aim of our VI is to get an optimal ϕ that brings $q(Z|X)$ closest to $p(Z|X)$. But the posterior is intractable so we can't directly minimize the KL divergence.

A. Evidence Lower Bound(ELBO)

For which we introduce ELBO. We maximise the evidence lower bound(ELBO), a lower bound on the marginal probability. To derive the Evidence Lower Bound, we introduce Jensens inequality $f(\mathbb{E}[X] \geq \mathbb{E}[f(X)])$.

This gives the bound on the log marginal as:

$$\begin{aligned}\log p(x) &= \log \int p(z,x)dz \\ &= \log \int p(z,x)\frac{q(z)}{q(z)}dz \\ &= \log\left(\mathbb{E}_q\left[\frac{p(x,z)}{q(z)}\right]\right) \\ &= \mathbb{E}_q[\log p(x,z)] - \mathbb{E}_q[\log q(z)] \\ &\equiv \mathcal{L}(q)\end{aligned} \tag{1}$$

Further it can be shown that relation between ELBO and KL divergence is,

$$\log p(x) = \mathcal{L}(q) + \mathrm{KL}(q(z)||p(z|x))$$

Under mean field assumptions ELBO transforms to

$$\mathcal{L}(q) = \int \prod_{i=1}^{N}\left\{\ln p(\mathbf{X},\mathbf{Z}) - \sum_{i=1}^{N}\ln q_i\right\}$$

For a q_j given all other $q_i(i \neq j)$, we get:

$$\begin{aligned}\mathcal{L}(q) &= \int q_j\left\{\int \ln p(\mathbf{X},\mathbf{Z})\prod_{i\neq j} q_i dZ_i\right\}dZ_j - \int q_j \ln q_j dZ_j + \text{const} \\ &= \int q_j \ln \hat{p}(\mathbf{X},\mathbf{Z_j})dZ_j - \int q_j \ln q_j dZ_j + \text{const}\end{aligned} \tag{2}$$

where,

$$\hat{p}(\mathbf{X},\mathbf{Z_j}) = \mathop{\mathbb{E}}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]$$

The optimal solution is obtained in the usual way by minimising the KL Divergence.The new distribution of the family hence becomes,

$$q_j^* = \mathop{\mathbb{E}}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]$$

This is repeated until the ELBO converges to obtain a distribution close to the prior

III. BLACK BOX VARIATIONAL INFERENCE

Black Box Variational Inference(BBVI) works with small minibatches of data rather then the entire dataset, which increases the efficiency of the VI algorithm.It enables applying VB inference for a wide variety of probabilistic models. It approximates ELBO derivatives using Monte-Carlo as,

$$\bigtriangledown_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \bigtriangledown \log q(\mathbf{Z}_\phi)(\log p(\mathbf{X},\mathbf{Z_s}) - \log q(\mathbf{Z_s}|\phi))$$

A. BBVI Identity

Using dominated convergence theorem we can derive identity required for BBVI as follows,

$$\begin{aligned}\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\ &= \int \nabla_\phi[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)]d\mathbf{Z} \\ &= \mathbb{E}_q[-\nabla_\phi\, q(\mathbf{Z}|\phi)] + \\ &\qquad \int \nabla_\phi q(\mathbf{Z}|\phi)[\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)]d\mathbf{Z} \\ &= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log(\mathbf{Z}|\phi))] \\ &\qquad\qquad (\because \mathbb{E}_q[\nabla_\phi q(\mathbf{Z}|\phi)] = 0) \\ &\approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_\phi)(\log p(\mathbf{X},\mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))\end{aligned} \tag{3}$$

B. Variance Reduction in BBVI

Monte Carlo estimate are very noisy and with high variance. The high variance gradients would require very small steps, resulting in slower convergence and even worse approximations.

Methods to reduce variance:

- Rao-Blackwellization : It reduces variance by replacing the function whose expectation is being approximated by Monte Carlo with another function that has the same expectation but smaller variance
- Control Variates : The Rao-Blackwellized approximations are replaced with control variates, which are functions with the same expectation but with lesser variance

IV. Reparametrization Trick

A. In BBVI

For the estimation of ELBO's gradient, we assume deterministic transformation $\mathbf{Z} = g(\epsilon,\phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ does not depend on ϕ.

With reparametrization we can write ELBO's gradient as follows:

$$\begin{aligned}\nabla_\phi\, \mathbb{E}_q[\log p(\mathbf{X}, g(\epsilon,\phi)) - \log q(g(\epsilon,\phi))] = \\ \mathbb{E}_q\, \nabla_\phi\, [\log p(\mathbf{X}, g(\epsilon,\phi)) - \log q(g(\epsilon,\phi))]\end{aligned} \tag{4}$$

The LHS is true due to Law of Unconscious Statistician. From $p(\epsilon)$ we can compute a Monte-Carlo approx, so

$$\begin{aligned}\nabla_\phi\, \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi)] \approx \\ \frac{1}{S}\sum_{s=1}^{S}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon_s,\phi)) - \nabla_\phi \log q(g(\epsilon_s,\phi)|\phi)]\end{aligned} \tag{5}$$

Such gradients are called pathwise gradients.

V. Variational Autoencoders

- Standard autoencoders learn to generate compact representations and reconstruct their inputs well, but asides from a few applications like denoising autoencoders, they are fairly limited. The fundamental problem with autoencoders, for generation, is that the latent space they convert their inputs to and where their encoded vectors lie, may not be continuous, or allow easy interpolation.
- When a neural network is used for the recognition model, we arrive at the variational auto-encoder. A variational autoencoder has a continuous latent space which allows random sampling. This is because its encoder gives two vectors instead of one, which are the mean and standard distribution of latent space. It results in random generated data which is close to the original input data.
- A VAE is made up of two parts - an encoder and a decoder. Encoder is a datapoint x, its output is a hidden representation z, and gives mean and standard deviation for the distribution over the latent space. After which a point from the latent space is chosen and decoded using decoder, for which we get a error which is backpropagated through the network.

In order to find the encoder and decoder parameters for which, we calculate the marginal log likelihood of the datapoints given.

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x})\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right] \\ &= \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x})\ \text{(ELBO)}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[\log\left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})}\right]\right]}_{=D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))}\end{aligned}$$

Fig. 1. Derivation of Marginal Log Likelihood

Here the second term is intractable but as we know that its non-negative or zero. And first term is the ELBO and due to the non-negative KL divergence, the ELBO is the lower bound of the log likelihood of the data. Lower Bound on the log likelihood can be rewritten as,

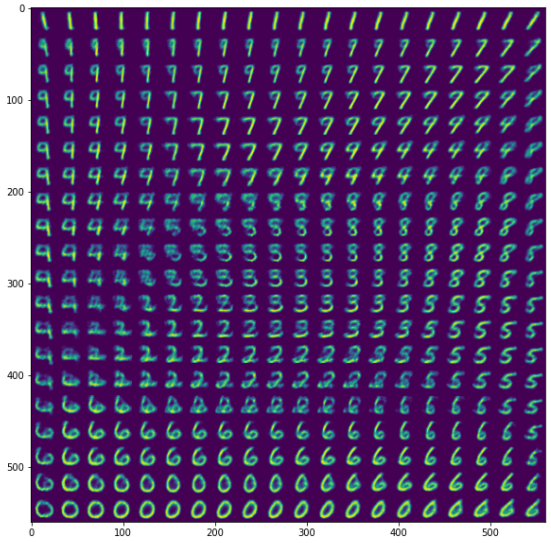
$$\mathcal{L}(\phi,\theta;x^{(i)}) = -D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) + \mathbb{E}_{q_\phi(z|x_{(i)})}\left[\log p_\theta(x_{(i)}|z)\right]$$

In lower bound equation, first term reconstructing the original input data and the second term is responsible for making approximate distribution close to input distribution .

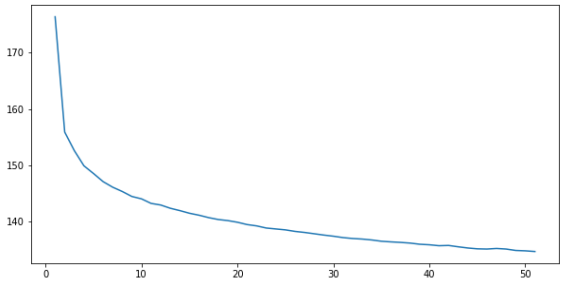
Expriments

I implemented a Variational Autoencoder using PyTorch framework and ran to train on a MNIST dataset for 50 epochs. A 2-D latent space was used.

A continuous stream of points were sampled from the mean and variance and were passed through the decoder, the result is represented in a grid of images below.



: Digits of MNIST dataset generated using a Variational Autoencoder



Plot for loss function while training for 50 epochs