# Advances in Variational Inference

Atharv Singh Patlan
Department of Computer Science and Engineering
IIT Kanpur
atharvsp@iitk.ac.in

***Abstract*— The focus of this project is mainly to analyze the development of the the various methods of Variational Inference (VI). We will be analyzing the foundations of VI, and then looking at Mean Field VI, Black Box VI, and finally head to Variational Autoencoders and run a VAE model on the MNIST dataset**

## I. INTRODUCTION

### A. Variational Inference

The main idea of variational methods is to cast inference as an optimization problem.

Suppose we are given an intractable probability distribution *p*. Variational techniques will try to solve an optimization problem over a class of tractable distributions *Q* in order to find a $q \in Q$ that is most similar to *p*. We do this by minimizing the Kullback-Leibler Divergence.

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

### B. Optimization in VI

The main aim of VI is to find the $q(\mathbf{Z}|\phi)$ that is closest to $p(\mathbf{Z}|\mathbf{X})$ by finding the optimal value of $\phi$ by $\phi^* = \arg\min_\phi \text{KL}[q(\mathbf{Z}|\phi)||p(\mathbf{Z}|\mathbf{X})]$

But the problem here is that as $p(\mathbf{Z}|\mathbf{X})$ is intractable, we can't directly solve the KL minimization.

However, it is possible to write the following for any $q$ : $\log p(\mathbf{X}) = \mathcal{L}(q) + \text{KL}(q||p)$ where:

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \log \left[\frac{p(\mathbf{X},\mathbf{Z})}{q(\mathbf{Z})}\right] d\mathbf{Z}$$

$$\text{KL}(q||p) = -\int q(\mathbf{Z}) \log \left[\frac{p(\mathbf{X}|\mathbf{Z})}{q(\mathbf{Z})}\right] d\mathbf{Z}$$

$\mathcal{L}(q)$ is called the Evidence Lower Bound (**ELBO**)

Now as $\log p(\mathbf{X})$ is constant w.r.t. $\mathbf{Z}$, the following holds:

$$\arg\min_q \text{KL}(q||p) = \arg\max_q \mathcal{L}(q)$$

Hence VI finds an approximating distribution $q(\mathbf{Z})$ that maximizes the ELBO

## II. MEAN FIELD VI

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:

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{N} q(\mathbf{Z}_i|\phi_i) = \prod_{i=1}^{N} q_i$$

Where we have partitioned $\mathbf{Z}$ into N groups $\mathbf{Z}_1, ..., \mathbf{Z}_N$ and $q_i = q(\mathbf{Z}_i|\phi_i)$

Hence under the mean field assumption, the ELBO is simplified to

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

For a $q_j$ given all other $q_i$ $(i \neq j)$ we have

$$\mathcal{L}(q) = \int q_j \left[\int \log p(\mathbf{X},\mathbf{Z}) \prod_{i\neq j} q_i d\mathbf{Z}_i\right] d\mathbf{Z}_j - \int q_j \log q_j d\mathbf{Z}_j + c_j$$

where $c_j$ is a constant w.r.t $q_j$

$$= \int q_j \log \hat{p}(\mathbf{X},\mathbf{Z}_j) d\mathbf{Z}_j - \int q_j \log q_j d\mathbf{Z}_j + c_j$$

wherein $\log \hat{p}(\mathbf{X},\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\log p(\mathbf{X},\mathbf{Z})]$

The optimal solution is obtained in the usual way by minimising the KL Divergence.The new distribution of the family hence becomes, $q_j^* = \hat{p}(\mathbf{X},\mathbf{Z}_j)$

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

*However the mean field assumption is a very strong assumption as it divides groups of the latent variables by assuming them to be completely independent of each other which can* ***destroy their structure****. Hence it is not preferred.*

## III. BLACK-BOX VI

Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo. It can work with small minibatches of data rather then the entire dataset, which increases the efficiency of the VI algorithm.

### A. The BBVI Identity

BBVI uses an important identity which can be derived as follows using the dominated convergence theorem:

$$\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\
&= \nabla_\phi \int [\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)]\, q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [\{\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)\}\, q(\mathbf{Z}|\phi)]\, d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \\
&\qquad \int \nabla_\phi q(\mathbf{Z}|\phi) [\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)]\, d\mathbf{Z} \\
&= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \qquad (1)
\end{aligned}$$

(as $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = 0$)

(1) is called the BBVI identity

Now using Monte Carlo on (1), given S samples $\{\mathbf{Z}_s\}_{s=1}^{S}$ from $q(\mathbf{Z}|\phi)$, we have:

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X},\mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi)) \quad (2)$$

### B. Variance Reduction in BBVI

The Monte Carlo estimate calculated by BBVI is very noisy and a high variance due to which the gradients would require very small steps which would lead to slow convergence.

Methods to reduce Variance:

*1) Rao - Blackwellization:* We assume mean field approximation of each of the latent variables and hence calculate the Monte - Carlo approximation for each of the subparts of the latent variables independently, which significantly reduces the variance.

*2) Control Variates:* The Rao-Blackwellized approximations are replaced with control variates, which are functions with the same expectation but with lesser variance.

## IV. VARIATIONAL AUTOENCODERS

- An autoencoder consists of an encoder and a decoder as neural networks and to learn the best encoding-decoding scheme using an iterative optimisation process. However the autoencoder is solely trained to encode and decode with as few loss as possible, no matter how the latent space is organised. This leads to overfitting and due to absence of explicit regularization, some points of the latent space are meaningless once decoded.
- A variational autoencoder can be defined as being an autoencoder whose training is regularised to avoid overfitting and ensure that the latent space has good properties that enable generative process. In order to introduce some regularisation of the latent space, instead of encoding an input as a single point, it is encoded as a distribution over the latent space. The encoder outputs two vectors instead of one, which are the vectors consisting the means and standard deviations of the distribututions in the latent space.
- Subsequently, the network is trained as follows:
  - First, the input is encoded as distribution over the latent space, and the mean and variance is generated.
  - Second, a point from the latent space is sampled from that distribution
  - Third, the sampled point is decoded and the reconstruction error can be computed
  - Finally, the reconstruction error is backpropagated through the network

Now in order to find the values of the encoder parameters $\phi$ and the decoder parameters $\theta$ , we calculate the marginal likelihood of the datapoints given to us. The marginal likelihood is composed of a sum over the marginal likelihoods of individual datapoints as they are i.i.d.

$$\begin{aligned}
\log p_\theta(x) &= \log \prod_i^m p_\theta(x^{(i)}) \\
&= \sum_i^m \log \int_z p_\theta(x^{(i)}, z) dz \\
&= \sum_i^m \log \int_z q_\vartheta(z|x^{(i)}) \frac{p_\theta(x^{(i)}, z)}{q_\vartheta(z|x^{(i)})} dz \\
&\geq \sum_i^m \int_z q_\vartheta(z|x^{(i)}) \left[\log p_\theta(x^{(i)}|z) + \log p(z) - \log q_\vartheta(z|x^{(i)})\right] dz \\
&= \sum_i^m (E_{q_\vartheta(z|x^{(i)})}[\log p_\theta(x^{(i)}|z)] - KL[q_\vartheta(z|x^{(i)})||p(z)])
\end{aligned}$$

The inequality comes due to the presence of a KL divergence term which is intractable, but knowing that it is always greater than or equal to zero, we can straightaway derive a variational lower bound for the marginal likelihoods.

These terms are tractable and are usually derived from simple distributions.

The first term here is responsible for reconstructing the original input data and the second term is responsible for making the approximate distribution close to input distribution.

We use the technique of variational lower bounds to get the 2 SGVB estimators:

$$\hat{\mathcal{L}}^A(x^{(i)}) = \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x^{(i)}, z^{(i,l)}) - \log q_\phi(z^{(i,l)}|x^{(i)})$$

$$\hat{\mathcal{L}}^B(x^{(i)}) = \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x^{(i)}|z^{(i,l)}) - \mathrm{KL}(q_\phi(z|x^{(i)})||p_\theta(z))$$

where $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$ and $\epsilon^{(l)}$ is a noise variable such that $\epsilon^{(l)} \sim p(\epsilon)$ and $g$ is a differentiable transformation on the noise variable $\epsilon$.

This method used reparameterizes the random variable $z$ and is called the reparametrization trick.

## V. EXPERIMENTS

I implemented a Convolutional Variational Autoencoder and ran it to make predictions on the MNIST dataset. The training was done for 50 epochs, accelerated over a Tesla K80 GPU.

A 2-D latent variable space was used, meaning that the mean and variance vectors would have two dimensions.

Subsequently, a continuous stream of points were sampled from the resulting distributions of the mean and variance, and were passed through the decoder to finally produce the output.

Link to the code

Below are the results derived from the experiments.

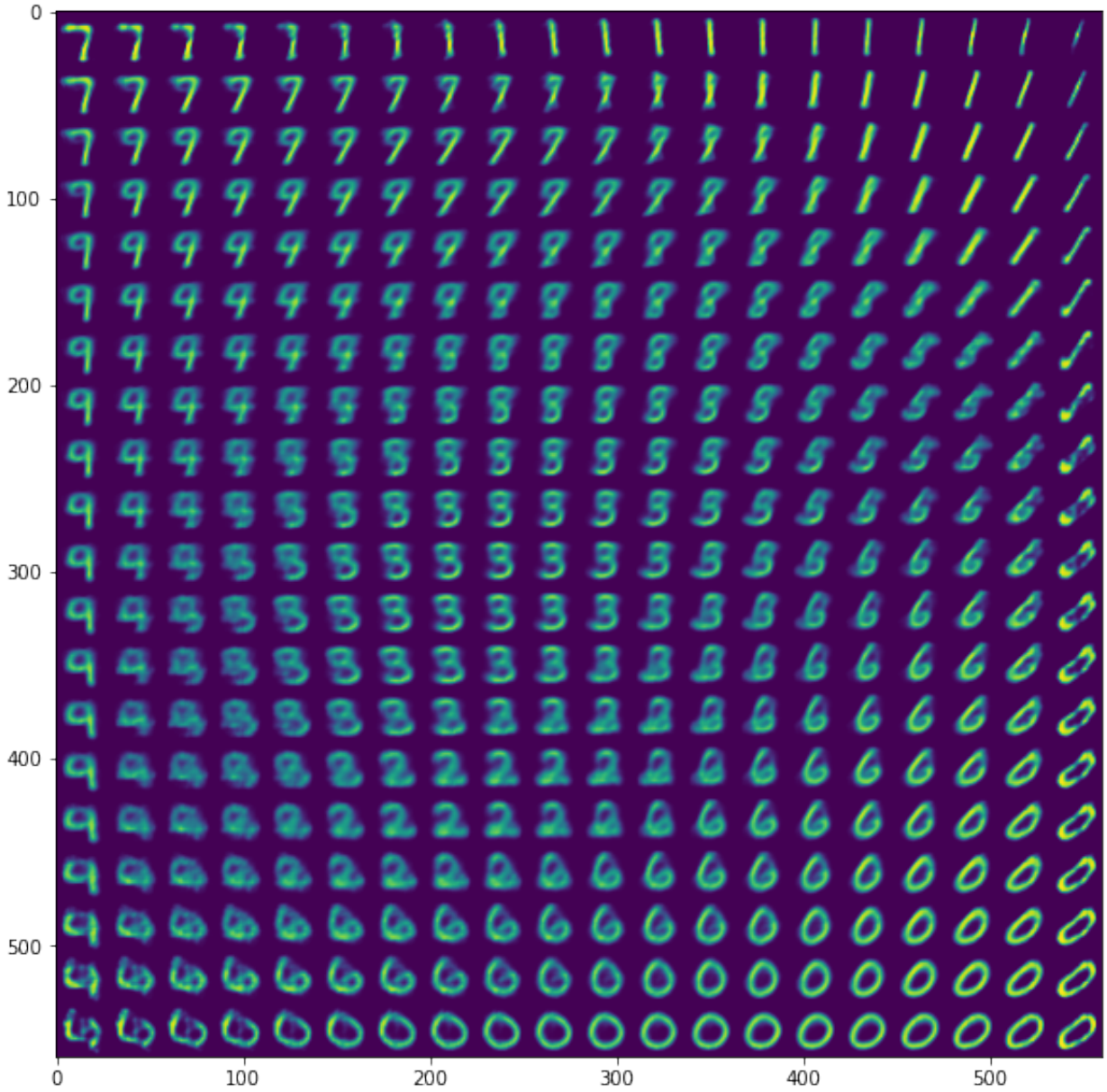


Figure 1: Digits of the MNIST dataset generated using a Variational Autoencoder

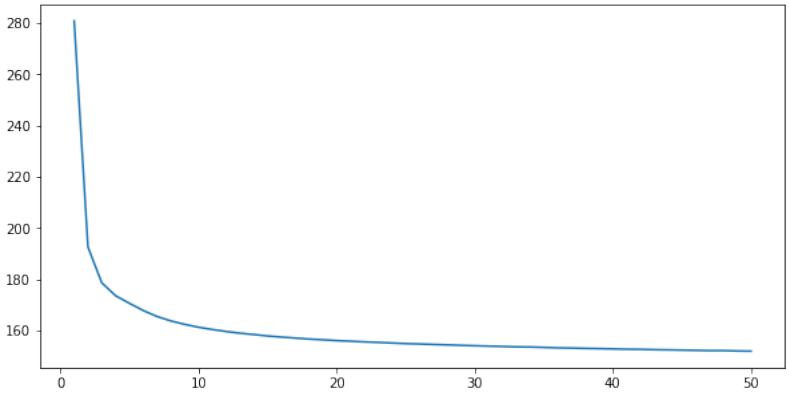


Figure 2: Plot of the loss function while training for 50 epochs