Modern Variational Learning

Arun Drelich Mitchell Wortsman
Brown University

May 2, 2018

Abstract

Computer Vision and Information Theory have been linked since the 1960s. In this paper we explore the theory and implementation of recent methods which lie in their intersection. Specifically, we examine the effectiveness of deep learning in performing variational Bayesian inference with Variational Auto-encoders (VAEs).

I. Introduction

In a seminal 2012 paper, three researchers from the University of Toronto changed the landscape of Computer Vision. Using deep convolutions neural nets, [KSH12] performed remarkably better than the previous state-of-the-art in the Imagenet Large Scale Visual Recognition Challenge (ILSVRC). This sparked the beginning of an artificial intelligence boom driven by deep learning.

The mathematics of deep learning is not novel. As introduced by [RHW86], the crucial theory has existed for 30 years. The advent of deep learning has been driven instead by unprecedented computing power, specifically of Graphics Processing Units (GPUs). Accordingly, researchers have been able to use deep learning to pursue problems where solutions were previously thought to be out of reach. One such area of renewed attention is variational Bayesian methods.

As discussed in [Att00; FR12; KW13], variational Bayesian methods approximate intractable integrals which arise in Bayesian inference. They are an alternative to more traditional techniques such as Markov Chain Monte Carlo (MCMC).

Variational auto encoders were introduced by [KW13], and have since found a myriad of applications including [Esl+16] and [Nez+18]. The defining contribution of the VAE was to exploit the representational capacity of deep neural networks to approximate both the variational distribution and likelihood function.

In this paper we examine the theory and implementation of modern techniques which combine deep learning and variational Bayesian inference. We begin by surveying relevant mathematics and conclude with our own implementation and experiments.

II. Preliminary Mathematics

i. Bayesian Inference

We begin with the classic problem of Bayesian inference. Given only a prior $p(z)$ and likelihood $p(x|z)$ we are interested in the posterior probability $p(z|x)$. Using Bayes' theorem we may express the posterior as follows.

$$p(z|x) = \frac{p(x|z)p(x)}{\int p(x|z')p(z')dz'} \tag{1}$$

However, in most cases the integral in the denominator is intractable. This intractability may occasionally be surmounted using techniques such as MCMC. In this paper we focus instead variational inference methods.

ii. Kullback-Leibler (KL) Divergence

The Kullback-Leibler divergence (also referred to as the relative entropy) measures the divergence of two probability distributions. For densities p and q the KL divergence is defined as

$$D_{\mathrm{KL}}(p(x)\|q(x)) = \int_{\mathbb{R}} \log \frac{p(x)}{q(x)} p(x)\ dx \quad (2)$$

and satisfies the following property.

Lemma 1. $D_{\mathrm{KL}}(p(x)\|q(x)) \geq 0$ with equality if and only if $p = q$.

Proof. This follows from Jensen's inequality. Recall that for any convex function ϕ and random variable X, Jensen's inequality asserts that

$$\mathbb{E}[\phi(X)] \geq \phi\left(\mathbb{E}[X]\right) \quad (3)$$

with equality if and only if ϕ is linear or X is constant. The inequality is reversed when ϕ is concave. Using that $\phi(x) = -\log x$ is a convex function we observe

$$\begin{aligned} &D_{\mathrm{KL}}(p(x)\|q(x)) \\ &= \int_{\mathbb{R}} \log \frac{p(x)}{q(x)} p(x)\ dx && (4) \\ &= \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] && (5) \\ &= \mathbb{E}_{X\sim p}\left[-\log \frac{q(X)}{p(X)}\right] && (6) \\ &\geq -\log \mathbb{E}_{X\sim p}\left[\frac{q(X)}{p(X)}\right] && (7) \\ &= -\log \int_{\mathbb{R}} \frac{q(x)}{p(x)} p(x)\ dx && (8) \\ &= -\log 1 = 0 && (9) \end{aligned}$$

since q is a density and so $\int_{x\in\mathbb{R}} q(x)\ dx = 1$. □

III. Variational Bayesian Methods

i. The Model

Consider a data set of N images $x = \{x_1, ..., x_N\}$. Images are high dimensional with complex structure and so it is useful to consider more tractable latent variables $z = \{z_1, ..., z_N\}$. Intuitively, image x_i is generated by the latent variable $z_i \in \mathbb{R}^k$. z_i may be interpreted as the k-dimensional encoding of x_i. Accordingly, we are often interested in the latent variables rather than the images themselves. In an idealized example where each x_i is a photo of a single handwritten digit, each z_i may encode the digit which appears in x_i.

We assume that the marginal, prior, and likelihood can be parameterized by θ. We then write the marginal as

$$p_\theta(x) = \int p_\theta(x|z) p_\theta(z)\ dz. \quad (10)$$

Moreover, we assume that each (x_i, z_i) is independent of every other (x_j, z_j) and so

$$p_\theta(x|z)p_\theta(z) = \prod_{i=1}^{N} p_\theta(x_i|z_i)p_\theta(z_i). \quad (11)$$

We have two main objectives.

1. Find the parameter θ^* which maximizes the likelihood of the marginal $p_{\theta^*}(x)$. The images x are called the *evidence* and we would like to find a parameter θ^* for which the evidence is likely to be observed.

2. Approximate the posterior $p_{\theta^*}(z|x)$. We will then be able to obtain the latent variable corresponding to an image.

As we have discussed, it is often intractable to compute the posterior directly. We instead make use of variational inference methods.

ii. Variational Inference

Under variational inference we introduce a distribution q parameterized by ϕ which approximates some true posterior. Objectives 1 and 2 may then be achieved by finding θ, ϕ which maximize

$$\mathcal{L}_{\theta,\phi} = \mathbb{E}_{Z\sim q_\phi}\left[\log \frac{p_\theta\left(X, Z\right)}{q_\phi\left(Z|X\right)}\right]. \quad (12)$$

We may also write $\mathcal{L}_{\theta,\phi}$ in terms of the relative entropy as follows.

$$\begin{aligned} \mathcal{L}_{\theta,\phi} &= \mathbb{E}_{Z\sim q_\phi}\left[-\log \frac{q_\phi\left(Z|X\right)}{p_\theta\left(X|Z\right)p_\theta(Z)}\right] && (13) \\ &= -D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z)) \\ &\quad + \mathbb{E}_{Z\sim q_\phi}[\log p_\theta(X|Z)] && (14) \end{aligned}$$

In modern literature, $\mathcal{L}_{\theta,\phi}$ is often referred to as the *evidence lower bound*. By maximizing this lower bound, we may simultaneously accomplish objectives 1 and 2.

Lemma 2. $\log p_\theta(x) \geq \mathcal{L}_{\theta,\phi}$, thus maximizing $\mathcal{L}_{\theta,\phi}$ in θ space will in turn maximize $\log p_\theta(x)$.

Proof.

$$\begin{aligned} &\log p_\theta(x) \\ &= \log \int p_\theta(x,z)\, dz && (15) \\ &= \log \int \frac{p_\theta(x,z)}{q_\phi(z|x)} q_\phi(z|x)\, dz && (16) \\ &= \log \mathbb{E}_{Z\sim q_\phi}\left[\frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right] && (17) \\ &\geq \mathbb{E}_{Z\sim q_\phi}\left[\log \frac{p_\theta\,(X,Z)}{q_\phi\,(Z|X)}\right] && (18) \\ &= \mathcal{L}_{\theta,\phi} && (19) \end{aligned}$$

where equation 18 follows from Jensen's inequality as log is concave. □

Lemma 3. $D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z|x)) \propto -\mathcal{L}_{\theta,\phi}$, thus maximizing $\mathcal{L}_{\theta,\phi}$ in ϕ space will in turn minimize the divergence of $q_\phi(z|x)$ from $p_\theta(z|x)$.

Proof.

$$\begin{aligned} &D_{\mathrm{KL}}(q_\phi(z|x)\|p_\theta(z|x)) \\ &= \mathbb{E}_{Z\sim q_\phi}\left[\log \frac{q_\phi(Z|X)}{p_\theta(Z|X)}\right] && (20) \\ &= \mathbb{E}_{Z\sim q_\phi}\left[\log \frac{q_\phi(Z|X)}{p_\theta(X,Z)}\right] + \log p_\theta(X) && (21) \\ &\propto -\mathcal{L}_{\theta,\phi} && (22) \end{aligned}$$

as $\log p_\theta(X)$ does not depend on ϕ, q. □

And so we have restated the challenge of inference as an optimization problem.

IV. Learning by SGD

In a new approach to variational inference, q_ϕ and p_θ are represented using a neural network. A neural network is a highly parameterized function approximator trained using some loss function L. Stochastic gradient descent (SGD) is used to update the parameters of the network by back-propagation [RHW86] in order to minimize the associated loss. As an optimization, we approximate the true loss L by $\tilde{L}$. Stochasticity arises by computing $\tilde{L}$ from a random subset of observation samples.

Stochastic gradient descent is the process of moving along the negative gradient of the loss surface until convergence to a minimum. At iteration n, let

$$(\theta^{n+1}, \phi^{n+1}) = (\theta^n, \phi^n) - \eta \nabla_{\theta,\phi} \tilde{L} \qquad (23)$$

for some learning rate η (which may depend on n).

From equation 12 and 14 we have a loss function $L = -\mathcal{L}_{\theta,\phi}$ which we wish to minimize. However, back-propagation requires that $\tilde{L}$ is differentiable with respect to the parameters of the network. Accordingly, we must restate the loss to satisfy this condition.

i. The ϕ Derivative

We first consider the problem of taking ϕ derivatives of the form

$$\nabla_\phi \mathbb{E}_{Z\sim q_\phi}[f_\phi(Z,X)] \qquad (24)$$

for some function f_ϕ. The naive Monte Carlo estimate of this expectation is given by

$$\mathbb{E}_{Z\sim q_\phi}[f_\phi(Z,X)] \approx \frac{1}{L}\sum_{\ell=1}^{L} f_\phi(z^{(\ell)}, x) \qquad (25)$$

where $z^{(\ell)} \sim q_\phi$. However, the above expression is not differentiable with respect to ϕ as samples are taken from q_ϕ.

We may rely instead on the so-called *reparameterization trick* discussed in [KW13]. Let us write Z as a deterministic function of X and some auxiliary independent random variable ϵ with density h.

$$Z = g_\phi(\epsilon, X) \qquad (26)$$

Without loss of generality, consider $Z \sim \mathcal{N}(X, \phi^2)$. Then we may write

$$Z = g_\phi(\epsilon, X) = X + \phi * \epsilon \qquad (27)$$

where $\epsilon \sim \mathcal{N}(0,1)$ has zero mean and unit variance.

Using this *reparameterization trick* we may rewrite the expectation term in equation 24 as

$$\mathbb{E}_{\epsilon\sim h}[f_\phi(g_\phi(\epsilon, X), X)] \tag{28}$$

which has Monte Carlo estimate

$$\frac{1}{L}\sum_{\ell=1}^{L} f_\phi(g_\phi(\epsilon^{(\ell)}, x), x) \tag{29}$$

where $\epsilon^{(\ell)} \sim h$. Assuming f_ϕ and g_ϕ are differentiable, this revised Monte Carlo estimate is itself differentiable with respect to ϕ.

ii. The θ Derivative

As x is observed, computing the partial derivative of $\mathcal{L}_{\theta,\phi}$ with respect to θ is less involved. We must compute the θ derivative of the form

$$\nabla_\theta \mathbb{E}_{Z\sim q_\phi}\left[\Gamma_\theta(X, Z)\right] \tag{30}$$

for some function Γ_θ. However, the expectation in the equation above may be approximated by the Monte Carlo estimate

$$\frac{1}{L}\sum_{\ell=1}^{L} \Gamma_\theta(x, z^{(\ell)}) \tag{31}$$

where $z^{(\ell)} \sim q_\phi$. When Γ_θ is sufficiently smooth this Monte Carlo estimate differentiable with respect to θ.

V. Variational Auto-Encoders (VAEs)

Introduced by [KW13], a variational auto-encoder is a deep learning approach to variational inference.

Let I be the identity matrix and $\mathcal{N}(\cdot;\mu,\Sigma)$ denote the density of a multivariate normal distribution with mean μ and covariance matrix Σ. In a variational auto-encoder, the following is assumed for the likelihood, prior, and approximate posterior q_ϕ:

- The likelihood $p_\theta(x|z)$ factors as $\prod_{i=1}^{N} p_\theta(x_i|z_i)$ where each $p_\theta(x_i|z_i)$ follows $\mathcal{N}(x_i; D_\theta^\mu(z_i), D_\theta^\sigma(z_i)^2 * I)$. The function $D_\theta = (D_\theta^\mu, D_\theta^\sigma)$ is represented by a neural network.
  If the output is binary (i.e. the pixel is black or white) then a Bernoulli distribution with learned parameters may be used instead.
- The prior $p_\theta(z)$ factors as $\prod_{i=1}^{N} p_\theta(z_i)$ where $p_\theta(z_i) = \mathcal{N}(z_i; 0, I)$.
- The approximate posterior $q_\phi(z|x)$ factors as $\prod_{i=1}^{N} q_\phi(z_i|x_i)$ where $q_\phi(z_i|x_i) = \mathcal{N}(z_i; E_\phi^\mu(x_i), E_\phi^\sigma * I)$. The function $E_\phi = (E_\phi^\mu, E_\phi^\sigma)$ is similarly represented by a neural network.

We now reexamine the type of loss function L to be minimized. As $L = -\mathcal{L}_{\theta,\phi}$,

$$\begin{aligned} L = {} & D_{\mathrm{KL}}(q_\phi(z|x) \| p_\theta(z)) \\ & + \mathbb{E}_{Z\sim q_\phi}[-\log p_\theta(X|Z)]. \end{aligned} \tag{32}$$

And so L decomposes into two terms

$$L = L_{\text{KL-Divergence}} + L_{\text{Cross-Entropy}} \tag{33}$$

which we may consider separately. Note that we use random samples of the data (batches) to calculate an approximate loss $\tilde{L}$. Since

$$\begin{aligned} L = {} & \mathbb{E}_{Z\sim q_\phi}\left[\log \frac{\prod_{i=1}^{n} q_\phi(Z_i|X_i)}{\prod_{i=1}^{n} p_\theta(Z_i)}\right] \\ & + \mathbb{E}_{Z\sim q_\phi}\left[-\log \prod_{i=1}^{n} p_\theta(X_i|Z_i)\right] \end{aligned} \tag{34}$$

$$\begin{aligned} = {} & \sum_{i=1}^{N} \mathbb{E}_{Z_i\sim q_\phi}\left[\log \frac{q_\phi(Z_i|X_i)}{p_\theta(Z_i)}\right] \\ & + \sum_{i=1}^{N} \mathbb{E}_{Z_i\sim q_\phi}[-\log p_\theta(X_i|Z_i)] \end{aligned} \tag{35}$$

we may let

$$\tilde{L} = \tilde{L}_{\text{KL-Divergence}} + \tilde{L}_{\text{Cross-Entropy}} \tag{36}$$

$$\begin{aligned} = {} & \sum_{i\in\mathcal{I}_m} D_{\mathrm{KL}}(q_\phi(z_i|x_i) \| p_\theta(z_i)) \\ & + \sum_{i\in\mathcal{I}_m} \mathbb{E}_{Z_i\sim q_\phi}[-\log p_\theta(X_i|Z_i)] \end{aligned} \tag{37}$$

for some batch $\mathcal{I}_m$ of $\{1, ..., N\}$.

i. KL-Divergence Loss

Both $q_\phi(z_i|x_i)$ and $p_\theta(z_i)$ are multivariate normal distributions by our assumptions. Thus, $D_{\mathrm{KL}}(q_\phi(z_i|x_i)\|p_\theta(z_i))$ is the KL divergence of $\mathcal{N}(z_i; E^\mu_\phi(x_i), E^\sigma_\phi * I)$ and $\mathcal{N}(z_i; 0, I)$. Therefore

$$\begin{aligned} &D_{\mathrm{KL}}(q_\phi(z_i|x_i)\|p_\theta(z_i)) \\ &= -\frac{1}{2}\sum_{j=1}^{k}\Big(1+\log(\sigma_j^2)-\mu_j^2-\sigma_j^2\Big) \end{aligned} \tag{38}$$

where $\mu = E^\mu_\phi(z_i)$ and $\sigma = E^\sigma_\phi(z_i)$ are both k dimensional. Equation 38 follows from simple algebra and a full derivation may be found in Appendix B of [KW13].

ii. Cross Entropy Loss

The cross entropy loss is more straightforward to compute in practice, as most neural net packages include existing implementations. In most, a single z_i is sampled from $q_\phi(z_i|x_i)$, then $\mathbb{E}_{Z_i \sim q_\phi}[-\log p_\theta(X_i|Z_i)]$ is approximated by $-\log p_\theta(x_i|z_i)$. This works in practice as there are many samples in each batch.

VI. VAE Results (MNIST)

We implement a simple VAE using pytorch and trained it on the MNIST dataset. The MNIST data consists of 60000 handwritten digits, each a 28×28 single-channel image. The code may be found at `https://github.com/mwortsma/modern_variational_learning`.

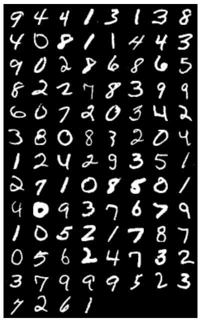

Figure 1: *100 images taken from MNIST*

i. Reconstructions

We first showcase results of reconstructing observations. We generate a reconstruction as follows.

1. Choose x_i at random from $x = \{x_1, ..., x_n\}$.
2. Sample z_i from $q_\phi(\cdot|x_i)$.
3. Sample r_i from $p_\theta(\cdot|z_i)$. We call r_i the reconstruction of x_i.

The following reconstructions were generated from 20-dimensional latent vectors. Figures 2 and 3 are generated after learning θ and ϕ using the given number of images. Figures 4 and 5 are generated after learning θ and ϕ for a given number of epochs. An epoch is defined as one pass through the entire dataset of images.

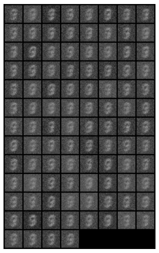

Figure 2: 10^3 *Images*

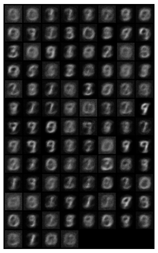

Figure 3: 10^4 *Images*

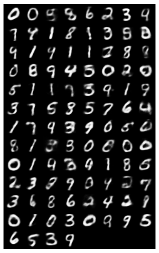

Figure 4: 1 *Epoch*

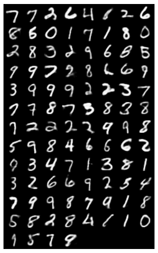

Figure 5: 50 *Epochs*

We similarly generate figures 7 and **??** using 2-dimensional latent vectors.

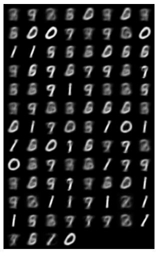

Figure 6: 1 *Epoch*

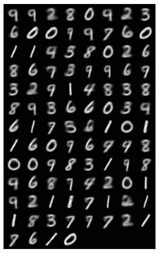

Figure 7: 50 *Epochs*

ii. Random Samples

We consider 20-dimensional latent vectors and showcase random samples. We generate a random sample as follows.

1. Sample z_i from $\mathcal{N}(\cdot; 0, I)$.
2. Sample r_i from $p_\theta(\cdot|z_i)$. We call r_i the random sample.

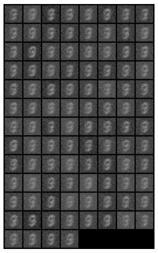

Figure 8: 10^3 *Images*

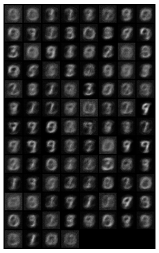

Figure 9: 10^4 *Images*

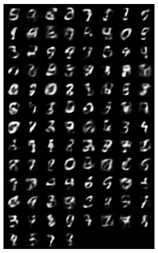

Figure 10: 1 *Epoch*

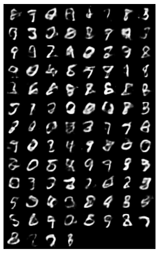

Figure 11: 50 *Epochs*

iii. Visualizing the MNIST Manifold

We can visualize the manifold of MNIST images when the latent vectors are two dimensional.

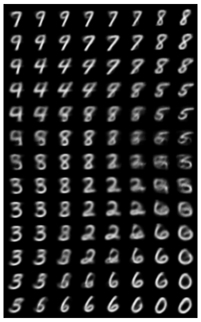

Figure 12: *The MNIST Manifold*

To generate image x_i at row r and column c of Figure 12, we define the latent vector z_i as

$$z_i = \begin{bmatrix} \mathcal{N}^{-1}\left(\frac{r}{R+1}; 0, 1\right) \\ \mathcal{N}^{-1}\left(\frac{c}{C+1}; 0, 1\right) \end{bmatrix} \tag{39}$$

where $\mathcal{N}^{-1}$ is the inverse CDF of the normal distribution, R is the total number of rows, and C is the total number of columns. We then sample x_i from $p_\theta(\cdot|z_i)$.

iv. Random Walks in Latent Space

Finally, we return to a 20 dimensional latent space and consider a random walk of length $M = 100$. We generate the latent vector z_1, corresponding to the first state in the walk, with a random sample from $\mathcal{N}(\cdot; 0, I)$. Each subsequent z_i has a latent vector which is perturbed from z_{i-1} by a sample from $\mathcal{N}(\cdot; 0, I)$ which is scaled by $\sqrt{M} = 0.1$. Each image x_i is then sampled from $p_\theta(\cdot|z_i)$. Figure 13 shows the results of this walk, with initial state in the top-left corner.

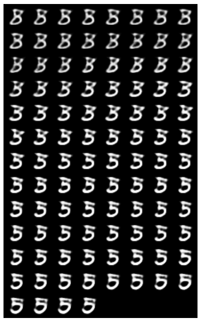

Figure 13: *A Random Walk in Latent Space*

References

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (Oct. 1986), 533 EP -. URL: `http://dx.doi.org/10.1038/323533a0`.

[Att00] Hagai Attias. "A Variational Bayesian Framework for Graphical Models". In: *In Advances in Neural Information Processing Systems 12*. MIT Press, 2000, pp. 209–215.

[FR12] Charles W. Fox and Stephen J. Roberts. "A tutorial on variational Bayesian inference". In: *Artificial Intelligence Review* 38.2 (Aug. 2012), pp. 85–95. ISSN: 1573-7462. DOI: `10.1007/s10462-011-9236-8`. URL: `https://doi.org/10.1007/s10462-011-9236-8`.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

[KW13] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. eprint: `arXiv:1312.6114`.

[Esl+16] S. M. Ali Eslami et al. "Attend, Infer, Repeat: Fast Scene Understanding with Generative Models". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., 2016, pp. 3225–3233. URL: `http://papers.nips.cc/paper/6230-attend-infer-repeat-fast-scene-understanding-with-generative-models.pdf`.

[Nez+18] Milad Zafar Nezhad et al. *A Predictive Approach Using Deep Feature Learning for Electronic Medical Records: A Comparative Study*. 2018. eprint: `arXiv:1801.02961`.