Auto-Encoding Variational Bayes

Jeremy Irvin

November 28, 2016

1 Method

Suppose $\{x^{(1)}, ..., x^{(m)}\}$ are IID samples from some discrete or continuous distribution, and each $x^{(i)}$ is generated as follows: $z^{(i)}$ is generated from some prior $p_{\theta^*}(z)$ and then $x^{(i)}$ is generated from some conditional $p_{\theta^*}(x|z)$. Both $p_{\theta^*}(z)$ and $p_{\theta^*}(x|z)$ come from a parametric family of distributions $p_\theta(z)$ and $p_\theta(x|z)$ respectively with PDFs differentiable w.r.t. θ and z. We assume the marginal $p_\theta(x)$, the true posterior $p_\theta(z|x)$, and any integrals required for any mean-field VB algorithm are all intractable. Moreover we assume large dataset so that sampling based solutions are intractable.

We hope to produce an approximation to the true posterior $p_\theta(z|x)$ called the recognition model, denoted $q_\phi(z|x)$. This approximation will be referred to as the encoder, and $p_\theta(x|z)$ the decoder. This method will jointly learn the recognition model parameters ϕ with the generative model parameters θ.

Since we can express the log marginal likelihood as a sum of marginal likelihoods over individual data points, we can focus on $\log p_\theta(x^{(i)})$ which can be rewritten as

$$\log p_\theta(x^{(i)}) = D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z|x^{(i)})) + \mathcal{L}(\theta, \phi; x^{(i)}).$$

The first term (the KL divergence) is nonnegative and hence $\mathcal{L}(\theta, \phi; x^{(i)})$ provides a lower bound of $\log p_\theta(x^{(i)})$. This term can be rewritten as

$$\mathcal{L}(\theta, \phi; x^{(i)}) = -D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z)) + \mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right] \tag{1}$$

We reparametrize the random variable $\tilde{z} \sim q_\phi(z|x)$ by using a differentiable function $g_\phi(\epsilon, x)$ of a noise variable $\epsilon \sim p(\epsilon)$, so that $\tilde{z} = g_\phi(\epsilon, x)$. Then we form Monte Carlo estimates of the expectation of an arbitrary function $f(z)$ over $q_\phi(z|x)$:

$$\mathbb{E}_{q_\phi(z|x^{(i)})}[f(z)] = \mathbb{E}_{p(\epsilon)}\left[f(g_\phi(\epsilon, x^{(i)}))\right] \approx \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\epsilon^{(l)}, x^{(i)})), \quad \epsilon^{(l)} \sim p(\epsilon).$$

Using $f(z) = \log p_\theta(x^{(i)}|z)$ in (1), we get

$$\tilde{L}^B(\theta, \phi; x^{(i)}) = -D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z)) + \frac{1}{L}\sum_{l=1}^{L} \log p^\theta(x^{(i)}|z^{(i,l)}) \tag{2}$$

with $z^{(i,l)} = g_\phi(\epsilon^{(i,l)}, x^{(i)})$ and $\epsilon^{(l)} \sim p(\epsilon)$ for some appropriate choice of $p(\epsilon)$. Note that $D_{KL}(q_\phi(z|x^{(i)}) \| p_\theta(z))$ can be integrated analytically, so only the reconstruction error $\mathbb{E}_{q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)}|z)\right]$ needs to be estimated using a Monte Carlo sampling technique.

2 Variational Auto-Encoder

Let the latent prior be standard multivariate Gaussian, ie, $p_\theta(z) = \mathcal{N}(z; 0, I)$, and $p_\theta(x|z)$ be multivariate Gaussian whose parameters μ and σ are computed from z using a MLP. Assume the true, intractable posterior is approximately Gaussian with diagonal covariance. Then our variational posterior approximator $q_\phi(z|x^{(i)})$ can be restricted to a family of this type, ie,

$$\log q_\phi(z|x^{(i)}) = \log \mathcal{N}(z; \mu^{(i)}, {\sigma^{(i)}}^2 I)$$

where $\mu^{(i)}$ and $\sigma^{(i)}$ are outputs of the encoder. We use $L = 1$ and sample from the posterior $z^{(i)} \sim q_\phi(z|x^{(i)})$ using $z^{(i)} = g_\phi(x^{(i)}, \epsilon) = \mu^{(i)} + \sigma^{(i)} \odot \epsilon$ and $\epsilon \sim \mathcal{N}(0, I)$. Since $p_\theta(z)$ and $q_\phi(z|x)$ are Gaussian, the KL divergence in (2) can be exactly computed without estimation, yielding the final llower bound estimator

$$\mathcal{L}(\theta, \phi; x^{(i)}) \approx \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log({\sigma_j^{(i)}}^2) - {\mu_j^{(i)}}^2 - {\sigma_j^{(i)}}^2\right) + \log p_\theta(x^{(i)}|z^{(i)})$$

with $z^{(i)} = g_\phi(x^{(i)}, \epsilon) = \mu^{(i)} + \sigma^{(i)} \odot \epsilon$ and $\epsilon \sim \mathcal{N}(0, I)$. More explicitly,

$$\begin{aligned} h^{(i)} &= \tanh(Wx^{(i)} + b) \\ \mu^{(i)} &= W_\mu h^{(i)} + b_\mu \\ \log {\sigma^{(i)}}^2 &= W_\sigma h^{(i)} + b_\sigma \end{aligned}$$

for learned parameters $W, W_\mu, W_\sigma, b, b_\mu, b_\sigma$.

3 Results

I ran three different experiments using a network with a single hidden layer of dimension 500 for both the encoder and decoder, with $N_z = 2$, $N_z = 3$, and $N_z = 50$. The training loss curves are shown in Fig. 1. The test losses are shown in Fig. 2. Examples of the reconstructions can be seen for all encoding dimension sizes in Fig. 3. For $N_z = 2$, the learned data manifold can be seen in Fig. 4. This is simply running the decoder on linearly spaced 2 dimensional vectors in $[-3, 3] \times [-3, 3]$. Additionally, all of the training examples were mapped into the 2 dimensional encoding and plotted with the true label, seen in Fig. 4.

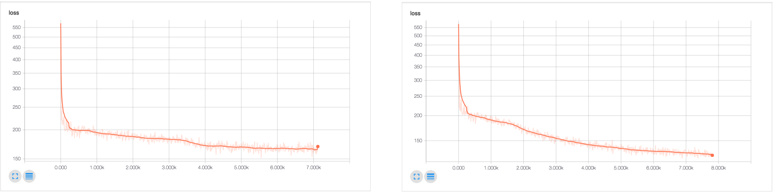

Figure 1: The panes show (from left to right) the training loss for $N_z = 2$, $N_z = 3$, and $N_z = 50$. A network with a single hidden layer of 500 dimension was used for all tests.

Model	Train Loss	Test Loss
$N_z = 2$	147.61	150.92
$N_z = 3$	133.74	136.30
$N_z = 50$	100.47	100.78

Figure 2: Lowest train losses and test losses for the three experiments when running for around 550 epochs.

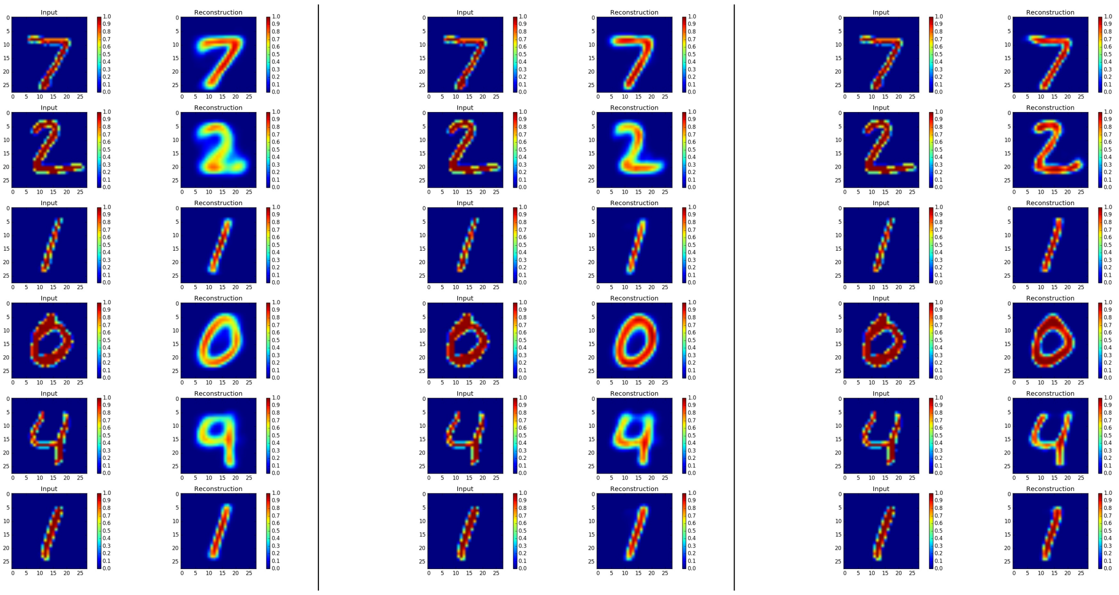

Figure 3: Reconstructed images for six samples for each of the three experiments (from left to right, $N_z = 2$, $N_z = 3$, $N_z = 50$).

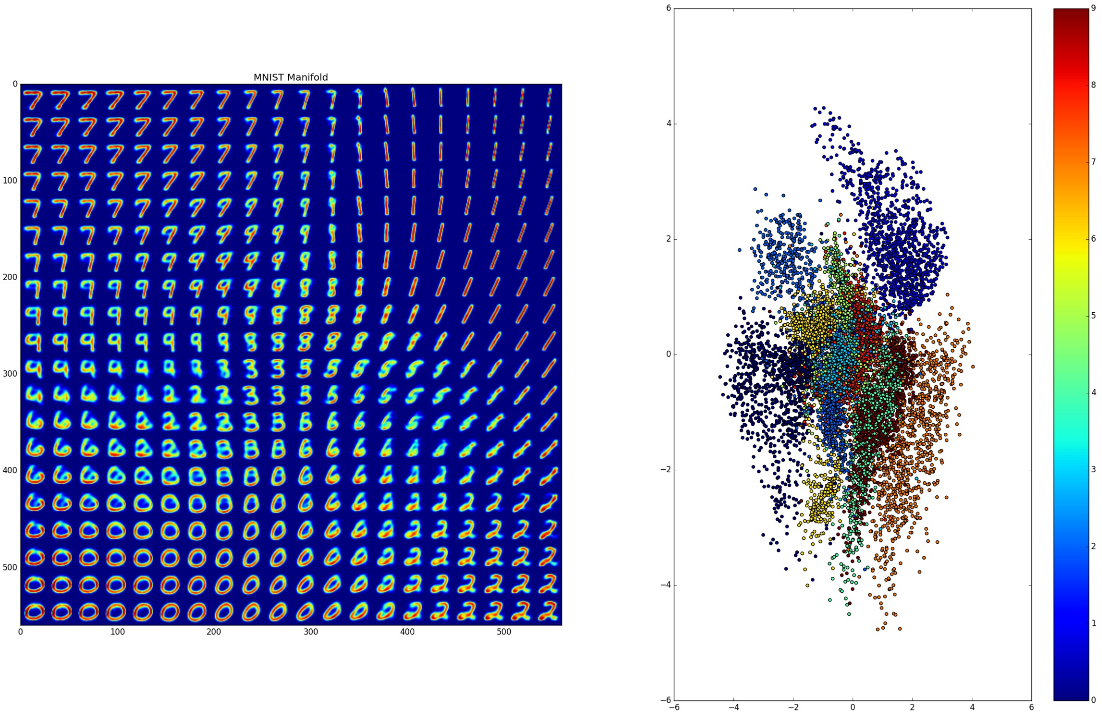

Figure 4: Learned data manifold (left) and scatter plot of 2 dimensional encodings of all training examples with true label (right).