CS 4641 Final Project Appendix

Patrick Astorga

3/30/2024

1 ELBO Derivation

1.1 Introduction

In this generative model, we assume the existence of some underlying set of continuous latent variables Z with a given prior $p(z)$, such that the conditional distribution $p_{X|Z}$ can be modeled with a neural network. Our goal is to optimize the latent variables Z and the distribution $p_{X|Z}$ simultaneously. Previously, we have used the Expectation Maximization (EM) algorithm to do this, which requires us to calculate the distribution $p_{Z|X}$ during the E step. To do this requires Bayes Theorem

$$p(z \mid x) = \frac{p(z \mid x)p(z)}{\int_z p(z \mid x)p(z)dz}$$

In this problem, we assume, Z is continuous and $p_{X|Z}$ is a neural network, so the integral in the denominator is intractable. So instead we must find a new way to optimize the latent variables Z and the distribution $p_{X|Z}$.

1.2 Evidence Lower Bound

1.2.1 Derivation

Introduce a approximation $q_{Z|X} \sim p_{Z|X}$, modeled by a neural network. Now we can see that

$$\begin{aligned}
\log p(x^{(i)}) &= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)})\right] \\
&= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)}, z) - \log p(z \mid x^{(i)})\right] \\
&= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)}, z) - \log p(z \mid x^{(i)}) + \log q(z \mid x^{(i)}) - \log q(z \mid x^{(i)})\right] \\
&= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)}, z) - \log q(z \mid x^{(i)})\right] + \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log q(z \mid x^{(i)}) - \log p(z \mid x^{(i)})\right] \\
&= \mathcal{L}\left(x^{(i)}; \theta, \phi\right) + KL\left(q(z \mid x^{(i)}) \mid\mid p(z \mid x^{(i)})\right)
\end{aligned}$$

and since KL-divergence is nonnegative you can see that

$$\mathcal{L}\left(x^{(i)}; \theta, \phi\right) \leq \log p(x^{(i)})$$

and so $\mathcal{L}\left(x^{(i)}; \theta, \phi\right)$ is called the Evidence Lower Bound (ELBO).

1.2.2 Alternate Form

Further rewriting the ELBO wee see that

$$\begin{aligned}
\mathcal{L}\left(x^{(i)}; \theta, \phi\right) &= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)}, z) - \log q(z \mid x^{(i)})\right] \\
&= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z) + \log p(z) - \log q(z \mid x^{(i)})\right] \\
&= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z)\right] - \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log q(z \mid x^{(i)}) - \log p(z)\right] \\
&= \mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z)\right] - KL\left(q(z \mid x^{(i)}) \mid\mid p(z)\right)
\end{aligned}$$

The first term is known as the "reconstruction loss" and the second term is the regularization term.

1.3 ELBO as a Training Objective

Our goal is the train the $p_{X|Z}$ and $q_{Z|X}$ networks with backpropagation. In our first example (MNIST), we will take the output of $p(x_i \mid z)$ as Bernoulli and the output of $q(z \mid x)$ as multivariate Gaussian. Furthermore we will let $p(z)$ be a standard normal distribution.

1.3.1 KL-Divergence Term

Since both $q(z \mid x)$ and $p(z)$ are multivariate Gaussian, their KL divergence can be computed analytically. If we let

$$q(z \mid x^{(i)}) \sim \mathcal{N}\left(\mu^{(i)}, \sigma^{(i)}\right)$$

such that $\mu^{(i)} = q_\mu(x^{(i)})$ and $\sigma^{(i)} = q_\sigma(x^{(i)})$, then the KL divergence term simplifies to

$$KL\left(q(z \mid x^{(i)}) \mid\mid p(z)\right) = -\frac{1}{2}\sum_{j=1}^{k}\left[1 + \log(\sigma_j^{(i)})^2 - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right]$$

1.3.2 Reconstruction Loss

Calculating the reconstruction loss is more difficult, since it involves an expectation, which is intractable. However, we can estimate it using sampling. If $z^{(i,1)}, \ldots, z^{(i,L)}$ are L samples from $q(z \mid x^{(i)})$, and if we let $\hat{x}^{(i,l)}$ be the output of the $p_{X|Z}$ network such that

$$\hat{x}_j^{(i,l)} = \mathbb{P}(x_j = 1, z^{(i,l)})$$

then we have

$$\begin{aligned}
\mathbb{E}_{z\sim q(z|x^{(i)})}\left[\log p(x^{(i)} \mid z)\right] &\approx \frac{1}{L}\sum_{l=1}^{L}\log p(x^{(i)} \mid z^{(i,l)}) \\
&= \frac{1}{L}\sum_{l=1}^{L}\sum_{j=1}^{N}\left[x_j^{(i)}\log\hat{x}_j^{(i,l)} + (1 - x_j^{(i)})\log(1 - \hat{x}_j^{(i,l)})\right]
\end{aligned}$$

There is one last problem with the way this is formulated. Our goal is to use gradient descent to train the parameters of the q network, but the act of drawing samples from

$$q(z \mid x^{(i)}) \sim \mathcal{N}\left(q_\mu(x^{(i)}), q_\sigma(x^{(i)})\right)$$

does not yield a well defined gradient. However, we can instead let $\epsilon^{(i,1)}, \ldots, \epsilon^{(i,L)}$ be L samples from $\mathcal{N}(0, I)$, and therefore we can explicitly define

$$z = \mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(i)}$$

where again $\mu^{(i)} = q_\mu(x^{(i)})$ and $\sigma^{(i)} = q_\sigma(x^{(i)})$ and $\odot$ represents element-wise multiplication. With this formulation, the gradient with respect to q is well defined. This is known as the "reparameterization trick".

1.3.3 Final Formulation

Let $\epsilon^{(i,1)}, \ldots, \epsilon^{(i,L)}$ be L samples from $\mathcal{N}(0, I)$. Then

$$\mathcal{L}\left(x^{(i)}; \theta, \phi\right) = \frac{1}{L}\sum_{l=1}^{L}\sum_{j=1}^{N}\left[x_j^{(i)}\log\hat{x}_j^{(i,l)} + (1 - x_j^{(i)})\log(1 - \hat{x}_j^{(i,l)})\right] + \frac{1}{2}\sum_{j=1}^{k}\left[1 + \log(\sigma_j^{(i)})^2 - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right]$$

such that $\mu^{(i)}, \sigma^{(i)}$ are the two outputs of the q network evaluated with the input $x^{(i)}$, and $\hat{x}^{(i,l)}$ is the output of the p network evaluated with the input $\mu^{(i)} + \sigma^{(i)} \odot \epsilon^{(i,l)}$.

References

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014.