DS-GA 1008 Homework 4 Q1

Zian Jiang (zj444)

November 12, 2020

1 ELBO

1.1

Firstly, we have

$$ELBO(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] = \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z},$$

and

$$KL\left[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\right] = \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z}.$$

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}) d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] d\mathbf{z} + \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left[\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] d\mathbf{z} \\
&= ELBO(\theta, \phi; \mathbf{x}) + KL\left[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\right]
\end{aligned}$$

■

1.2

$$\log p_\theta(\mathbf{x}) \geq ELBO(\theta, \phi; \mathbf{x})$$

because the KL divergence term is non-negative: $KL\left[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\right] \geq 0$.

$$\log p_\theta(\mathbf{x}) = ELBO(\theta, \phi; \mathbf{x})$$

only when $KL\left[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\right] = 0$, which is only true when $q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$, in other words $q_\phi(\mathbf{z}|\mathbf{x})$ is equal to the true posterior distribution.

2 ELBO surgery

2.1

$$
\begin{aligned}
ELBO(\theta, \phi; \mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z})} \frac{p_\theta(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z})} \div \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z})}\right] d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z})} d\mathbf{z} - \int q_\phi(\mathbf{z}|\mathbf{x}) \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z})} d\mathbf{z} \\
&= \int q_\phi(\mathbf{z}|\mathbf{x}) \log p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z} - KL\left[q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL\left[q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z})\right]
\end{aligned}
$$

■

2.2

The first term is the reconstruction term. It is trying to reconstruct $\mathbf{x}$ given the latent $\mathbf{z}$. When decoder $p_\theta(\mathbf{x}|\mathbf{z})$ assigns high probability to the original $\mathbf{x}$, this term is maximized. The second term is the regularizer that minimizes the divergence between approximation $q_\phi(\mathbf{z}|\mathbf{x})$ and prior $p_\theta(\mathbf{z})$, which we fix to be a unit Normal distribution. Thus, the second term encourages the latent space to look Gaussian and prevents encoder $q_\phi(\mathbf{z}|\mathbf{x})$ from simply encoding an identity mapping, and instead forces it to learn some more interesting representation.

3 Reconstruction loss

3.1

$$
\begin{aligned}
-\log p(\mathbf{x}|\tilde{\mathbf{z}}) &= -\sum_{d=1}^{D} \log Bern(x_d; \tilde{x}_d) \\
&= -\sum_{d=1}^{D} \log \left[(\tilde{x}_d)^{x_d}(1-\tilde{x}_d)^{1-x_d}\right] \\
&= -\sum_{d=1}^{D} \left[x_d \log \tilde{x}_d + (1-x_d)\log(1-\tilde{x}_d)\right] \\
&= BCELoss(\tilde{\mathbf{x}}, \mathbf{x}) \quad \text{summed over D dimensions}
\end{aligned}
$$

■

3.2

$$
\begin{aligned}
-\log p(\mathbf{x}|\tilde{\mathbf{z}}) &= -\sum_{d=1}^{D} \log \mathcal{N}(x_d; \tilde{x}_d, \sigma^2) \\
&= -\sum_{d=1}^{D} \log \left[\frac{1}{\sigma\sqrt{2\pi}} e^{-(x_d - \tilde{x}_d)^2 / 2\sigma^2} \right] \\
&= -\sum_{d=1}^{D} -(x_d - \tilde{x}_d)^2 \Big/ 2\sigma^2 + D\log(\sigma\sqrt{2\pi}) \\
&= MSE(\tilde{\mathbf{x}}, \mathbf{x})/2\sigma^2 + D\log(\sigma\sqrt{2\pi}) \quad \text{summed over D dimensions}
\end{aligned}
$$

■

4 Short answer

4.1 Reparameterization

VAEs use reparameterization because we cannot use back propagation through the sampler, which is a random node. Instead reparameterization allows us to use back propagation through deterministic nodes.

4.2 Overlapping latents

If there is overlapping, the reconstruction loss may be very big because the model cannot reconstruct back to the original input.

4.3 Missing labels

- Train a VAE with all images. Do k-means clustering on the latent space and assign pseudo-labels to data with missing labels. Then we can train a classifier using supervised approaches.
- Use semi-supervised VAE with 2 heads where one of them infers labels.
- Use classification restricted Boltzmann machine.

4.4 Bonus: Discrete latent variables

The reconstruction term. We can use Gumbel-Softmax, which samples a reparametrizable continuous distribution, to represent samples from a discrete distribution.