# Model 2: VAE with full covariance Gaussian posteriors

Prashant Shekhar

March 30, 2023

# Table of Contents

# Some statistical background

- With given vector $\mu \in \mathbf{R}^m$ and lower triangular matrix $L \in \mathbf{R}^{m \times m}$, we can defined a random vector $z$ as follows:

$$\epsilon \sim N(0, I)$$

$$z = \mu + L\epsilon$$

- With this way of constructing $z$, we have:

$$\text{Mean of z} : \mathbf{E}[z] = \mu$$

$$\text{Variance of z} : Var(z) = \mathbf{E}[(z - \mathbf{E}[z])(z - \mathbf{E}[z])^T] = \mathbf{E}[L\epsilon(L\epsilon)^T] = L\mathbf{E}[\epsilon\epsilon^T]L^T = LL^T$$

- Hence, we have the following distribution for $z$

$$z \sim N(\mu, \Sigma), \text{ where } \Sigma = LL^T \tag{1}$$

# The model

The factorized Gaussian posterior from Model 1 can be extended to a Gaussian with full covariance:

$$q_\phi(z|x) = N(\mu, \Sigma) \tag{2}$$

where unlike before, $\Sigma$ is now a fully populated matrix.
Hence our new **encoder/inference** model: $q_\phi(z|x)$:

$$EncoderNeuralNet_\phi(x) \rightarrow (\mu, \log\sigma, L^{'})$$

$$L \leftarrow L_{mask} \odot L^{'} + diag(\sigma)$$

$$\epsilon \sim N(0, I)$$

$$z = \mu + L\epsilon, \quad \text{Hence: } z \sim N(\mu, \Sigma = LL^T)$$

Here $L_{mask}$ is a masking matrix with zeros on and above the diagonal, and ones below the diagonal. $\odot$ is an elementwise multiplication operator.
The **generative/decoding** model: $p_\theta(x|z)$

$$DecoderNeuralNet_\theta(z) \rightarrow \hat{x}$$

# Computing ELBO

From previous lectures we know:

$$\mathcal{L}_{\theta,\phi}(x) = \mathbf{E}_{q_\phi(z|x)}[log(p_\theta(x,z)) - log(q_\phi(z|x))]$$

But instead of maximizing ELBO, as before, we prefer to minimize negative of ELBO. Hence we have:

$$\begin{aligned}
\mathcal{U}_{\theta,\phi}(x) &= -\mathcal{L}_{\theta,\phi}(x) \\
&= -\mathbf{E}_{q_\phi(z|x)}[log(p_\theta(x,z)) - log(q_\phi(z|x))] \\
&\approx \underbrace{\mathbf{E}_{q_\phi(z|x)}\left[log\left[\frac{q_\phi(z|x)}{p_\theta(z)}\right]\right]}_{\text{Encoder regularizarion}} + \underbrace{-log(p_\theta(x|z))}_{\text{Decoder reconstruction error}} \qquad ; \text{(From Model 1 slide)} \\
&\approx D_{KL}(q_\phi(z|x)||p_\theta(z)) + (1/nd)\sum_{i=1}^{n}\sum_{j=1}^{d}(x_{ij} - \hat{x}_{ij})^2
\end{aligned}$$

# Computing ELBO

We need to compute: $D_{KL}(q_\phi(z|x)||p_\theta(z))$. We know:

$$q_\phi(z|x) = N(\mu, LL^T) \text{ and } p_\theta(z) = N(0, I)$$

Hence, with $\mu_1 = \mu$ and $\Sigma_1 = LL^T$, $\mu_2 = 0$, $\Sigma_2 = I$: we have:

$$\begin{aligned} D_{KL}(q_\phi(z|x)||p_\theta(z)) &= \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - m + Tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)\right] \\ &= \frac{1}{2}\left[-\sum_{i=1}^{m}\log\sigma_i^2 - m + Tr(LL^T) + \sum_{i=1}^{m}\mu_i^2\right] \end{aligned}$$

# Cost functions

# Kullback-Leibler(KL) distance/divergence

- Kullback–Leibler divergence (also called relative entropy and I-divergence), denoted $D_{KL}(P||Q)$, is a type of statistical distance: a measure of how one probability distribution $P$ is different from a second, reference probability distribution $Q$
- Assuming both $P$ and $Q$ have normal distributions with means $\mu_1$ and $\mu_2$ and variances $\Sigma_1$ and $\Sigma_2$ respectively. Then KL divergence from $Q$ to $P$ is:

$$
\begin{aligned}
D_{KL}(P||Q) &= \mathbf{E}_{P(x)}\left[\log\left[\frac{P(x)}{Q(x)}\right]\right] \\
&= \int [\log(P(x)) - \log(Q(x))]P(x)dx \\
&= \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - d + Tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T\Sigma_2^{-1}(\mu_2 - \mu_1)\right]
\end{aligned}
$$

# Cross-Entropy loss function

- Also referred to as logarithmic loss, log loss or logistic loss.
- Each predicted class probability is compared with actual class label/probability of 0 or 1.
- Cross-entropy is defined as:

$$L_{CE} = -\sum_{i=1}^{m} p_i \log(q_i)$$

  where $p_i$ is the true class label and $q_i$ is the softmax probability of $i^{th}$ class. Also, m is the number of classes.
- For example, if we have 3 classes (1/2/3) and for a sample, the target class is class 2, then the true class label vector can be: [0,1,0] and if at the last layer the predicted probabilities are $[q_1, q_2, q_3]$, then the loss is:

$$L_{CE} = -log(q_2)$$

**This also shows why cross entropy loss is sometimes equivalent to negative log-likelihood**

# Mean Squared/Sum Squared loss function

- Mainly used for regression problems.
- With $n$ samples, if the true target value vector is $y \in \mathbf{R}^n$ and the predicted value vector is $\hat{y} \in \mathbf{R}^n$, then Sum Squared Error (SSE) is:

$$SSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- And, Mean Squared Error (MSE) is:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$