Assignment 4 - Theoretical part

Exercise 1 *Given that the data is high-dimensional (in our case, images), what could be the motivation for training a latent variable model, as opposed to directly trying to model $p_\theta(x)$?*

Explanation. *The motivation for training a latent variable model are:*

- *The model doesn't just learn the distribution of the input image X but the parameters that rule this distribution. That will allow us to create new images which are more generalized and vary from the original*
- *The model can be trained in the latent space, which has a smaller dimensionality than the feature space, so less parameters need to be learned*
- *The latent variable model is much simpler than the feature variable model to work with and we will get the same efficiency without reducing the flexibility of the model.*
- *A latent variable model expresses the uncertainty of the model, this is representing the effect of unobservable covariates, factors and inner structure of our data.*
- *Latent variable model summarizes different measurements of the same (directly) unobservable characteristics.*

Exercise 2 *Write out log p(x|z) for this discrete model, and simplify the expression as much as possible. Can you relate this expression to a commonly used loss function for neural networks?*

$$p(x|z) = \prod_{d=1}^{D} {\rho_d}^{x_d}(1-\rho_d)^{(1-x_d)} \tag{1}$$

Explanation. *Taking log on both sides:*

$$\log(p(x|z)) = \log(\prod_{d=1}^{D} {\rho_d}^{x_d}(1-\rho_d)^{(1-x_d)})$$

$$Or, \log(p(x|z)) = \log({\rho_1}^{x_1}(1-\rho_1)^{(1-x_1)} \times {\rho_2}^{x_2}(1-\rho_2)^{(1-x_2)} \times ...\rho^{x_D}(1-\rho_D)^{(1-x_D)})$$

For convenience, we will derive only the first two terms and translate afterwards.

$$R.H.S. = \log(\rho_1^{x_1}(1-\rho_1)^{(1-x_1)}) + \log({\rho_2}^{x_2}(1-\rho_2)^{(1-x_2)})$$

$$R.H.S. = x_1 log(\rho_1) + (1-x_1)log(1-\rho_1) + x_2 log(\rho_2) + (1-x_2)log(1-\rho_2)$$

$$R.H.S. = x_1 log(\rho_1) + x_2 log(\rho_2) + (1-x_1)log(1-\rho_1) + (1-x_2)log(1-\rho_2)$$

$$\therefore \log(p(x|z)) = \sum_{i=1}^{D} x_i \log(\rho_i) + \sum_{i=1}^{D} (1-x_i)\log(1-\rho_i) \tag{2}$$

We know that binary cross entropy loss function is:

$$H(p,q) = -\sum_i p_i \log(q_i)$$

$$H(p,q) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$$

Equation 2 is equivalent to the above equation with $\hat{y}$ translating to the output from the variational autoencoder ρ, where the value of ρ can be either 0 or 1, since the distribution we choose here is Bernoulli's distribution.

Exercise 3 *Write out log p(x|z) for this continuous model, and simplify the expression as much as possible. Can you relate this expression to a commonly used loss function for neural networks?*
(Hint: note that terms that are constant w.r.t. the learned parameters μ will not affect the learning, as their derivative will be zero.)

$$p(x|z) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_d^2}} e^{-\frac{(x_d-\mu_d)^2}{\sigma_d^2}} \tag{3}$$

Explanation. *Taking log on both sides and taking only first two terms for convenience:*

$$\log(p(x|z) = \log(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x_1-\mu_1)^2}{\sigma_1^2}} \times \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x_2-\mu_2)^2}{\sigma_2^2}})$$

$$RHS = \log(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x_1-\mu_1)^2}{\sigma_1^2}}) + \log(\frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x_2-\mu_2)^2}{\sigma_2^2}})$$

$$RHS = [\log(\frac{1}{\sqrt{2\pi\sigma_1^2}}) + \log(\frac{1}{\sqrt{2\pi\sigma_2^2}})] + [(-\frac{(x_1-\mu_1)^2}{\sigma_1^2}) + (-\frac{(x_2-\mu_2)^2}{\sigma_2^2})]$$

Both the square bracketed terms in the above equation continue D times. So the equation translates to the following:

$$\therefore \log(p(x|z)) = \log(\frac{1}{\sqrt{2\pi\sigma_d^2}}^{D}) - \sum_{i=1}^{D} \frac{(x_i-\mu_i)^2}{2\sigma^2} \tag{4}$$

We know that the MSE loss function as:

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

Equation 4 is equivalent to the above equation of MSE loss, given the first part of equation 4 is negligible in value and can be ignored. X_i *is the input to the model whereas* μ *is the output since the distribution we choose the data to be forced into is Gaussian in the variational autoencoder. So, w.r.t* μ *we compute loss and rectify the distribution of the data.*