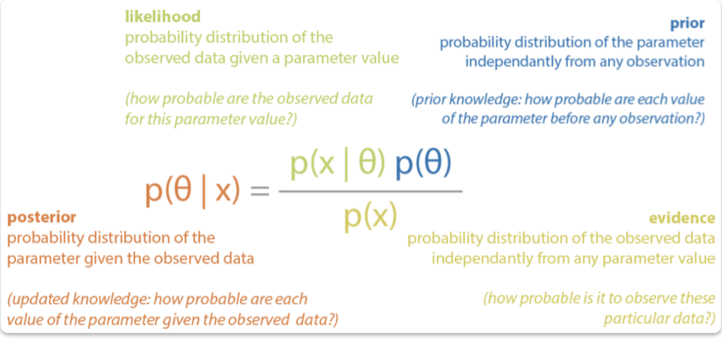
Variational Autoencoder

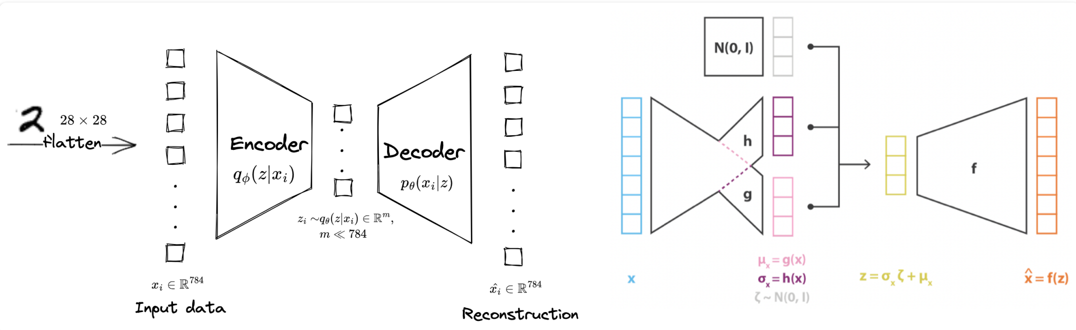
Ref: [Altosaar blog], [Rocca blog], [CMU ppt], [EM], [rose yu slidesCode sample]
Post: [Patacchiola]

0 Bayesian Basics



1 Architecture & Loss Function

1.1 Architecture



1.2 Loss

The loss L_i generated by one training sample x_i is defined as

$$L_i(\theta, \phi) = -\underbrace{\mathop{\mathbb{E}}_{z \sim q_\theta(z|x_i)} \log p_\phi(x_i|z)}_{\text{term1: Recon. loss}} + \underbrace{KL(q_\theta(z|x_i), p(z))}_{\text{term2: Disentanglement}}$$

- $p_\phi(x_i|z)$: *Likelihood* of decoder regenerating the i-th input data x_i, given latent variable z_i
- $q_\theta(z_i)$: *Posterior* probability distribution of latent variable z_i, given i-th input data x_i
- $p(z) = N(0, 1)$: *Prior*. We assume the ground-truth distribution of latent variable z_i to be a standard normal distribution.
- Term1 - Reconstruction loss: Expected neg-log-likelihood of i-the input data x_i. (Given an input data x_i and encoder output z_i, we want to maximize the likelihood of decoder regenerating x_i). This can be effectively replaced by MSE loss when actually implementing the loss:

```
x_hat = model.decoder.forward(model.encoder.forward(x))
loss_recon = nn.functional.mse_loss(x_hat, x)
```
language-python

- Term2 - KL divergence: We want to regularize the latent variable distribution such that it's close to a standard normal distribution.

$$\begin{aligned} KL(q_\theta(z|x_i), p(z)) &= KL(N(\mu,\sigma), N(0,1)) \\ &= \frac{1}{2}\sum_{i=1}^{d}\left(1+\log\sigma_j^2-\mu_j^2-\sigma_j\right) \end{aligned}$$

```
def loss_function(self,*args,**kwargs) -> dict:

        recons = args[0]
        input = args[1]
        mu = args[2]
        log_var = args[3]  # sigma

        kld_weight = kwargs['M_N'] # Account for the minibatch samples from
the dataset
        recons_loss =F.mse_loss(recons, input)
        kld_loss = torch.mean(-0.5 * torch.sum(1 + log_var - mu ** 2 -
log_var.exp(), dim = 1), dim = 0)

        loss = recons_loss + kld_weight * kld_loss

        return {'loss': loss, 'Reconstruction_Loss':recons_loss.detach(),
'KLD':-kld_loss.detach()}
```
language-python

1.3 Re-parameterization Trick

(TODO)

2 ELBO - Evidence Lower Bound

Recall the encoder-decoder view of dimensionality reduction. We'll use the following notation:

- x: Input data (e.g. an image); z: Encoded data
- θ: Decoder model parameters; ϕ: Encoder model parameters
  Out goal is to maximize the likelihood of the decoder regenerating x. Which, expressed in terms of log-likelihood, is:

$$\underset{\theta}{\operatorname{argmax}} \log p_\theta(x)$$

To find $q_\phi(\cdot)$ that best approximates $p_\theta(\cdot)$, we can just minimize their KL divergence, which essentially evaluates how closely two probability distribution functions (PDFs) resembles each other. Therefore we attempt to derive $KL\Big(q_\phi(\cdot), p_\theta(\cdot)\Big)$. From the definition of KL divergence:

$$KL\Big(q_\phi(z|x), p_\theta(z|x)\Big) = \mathbb{E}_{q_\phi}\log\frac{q_\phi(z|x)}{p_\theta(z|x)}$$

Now our goal is to approximate the intractable $p_\theta(z|x)$. We start by evaluating the KL divergence, using expectation of a function of RV along the way:

$$
\begin{aligned}
KL\Big(q_\phi(z|x), p_\theta(z|x)\Big) &= \mathbb{E}_{q_\phi} \log \frac{q_\phi(z|x)}{p_\theta(z|x)} \\
&= \mathbb{E}_{q_\phi} \log q_\phi(z|x) - \mathbb{E}_{q_\phi} \log p_\theta(z|x) && \text{(log property)} \\
&= \mathbb{E}_{q_\phi} \log q_\phi(z|x) - \mathbb{E}_{q_\phi} \log \frac{p_{\theta_1}(z,x)}{p_{\theta_2}(x)} && \text{(Bayes rule)} \\
&= \mathbb{E}_{q_\phi} \log q_\phi(z|x) - \mathbb{E}_{q_\phi} \log p_{\theta_1}(z,x) + \mathbb{E}_{z\sim q_\phi} \underbrace{\log p_{\theta_2}(x)}_{g(z)} \\
&= \mathbb{E}_{q_\phi} \log q_\phi(z|x) - \mathbb{E}_{q_\phi} \log p_{\theta_1}(z,x) + \int \log(p_{\theta_2}(x)) q_\phi(z|x) dz && (\mathbb{E} \text{ of } g(z)) \\
&= \mathbb{E}_{q_\phi} \log q_\phi(z|x) - \mathbb{E}_{q_\phi} \log p_{\theta_1}(z,x) + \log p_{\theta_2}(x) \cancelto{1}{\int q_\phi(z|x) dz}
\end{aligned}
$$

Rearranging, we can express $\log p_{\theta_2}(x)$ as:

$$
\log p_{\theta_2}(x_i) = KL(q_\phi(z|x_i), p_\theta(z|x_i)) + \underbrace{\mathbb{E}_{q_\phi} \log \frac{p_\theta(z,x_i)}{q_\phi(z|x_i)}}_{\text{ELBO}}
$$

Which is intractable since we need exponential time to evaluate $p_{\theta_2}(x) = \int p(x|z)p(z)dz$ (over all configuration of the latent variable z). But, since $KL(q_\phi, p_\theta) \geq 0$ (can prove using Jensen's inequality), we've found a lower bound of $\log p_{\theta_2}(x)$:

$$
\log p_{\theta_2}(x_i) \geq \underbrace{\mathbb{E}_{q_\phi} \log \frac{p_\theta(z,x_i)}{q_\phi(z|x_i)}}_{\text{ELBO}}
$$

The ELBO is short for "*evidence lower bound*", i.e. the lower bound of the approximated posterior $p_\theta(x)$. Therefore we can maximize ELBO in order to maximize $\log p_\theta(x)$:

$$
\begin{aligned}
\text{ELBO} &= \mathbb{E}_{q_\phi} \log \frac{p_\theta(z,x_i)}{q_\phi(z|x_i)} \\
&= \mathbb{E}_{q_\phi} \log \frac{p_\theta(x_i|z)p(z)}{q_\phi(z|x_i)} = \mathbb{E}_{q_\phi} \log p_\theta(x_i|z) - \mathbb{E}_{q_\phi} \frac{q_\phi(z|x_i)}{p(z)} \\
&= \mathbb{E}_{q_\phi} \log p_\theta(x_i|z) - KL(q_\phi(z|x_i)|p(z))
\end{aligned}
$$

$$
L(\theta, \phi;\ x_i) = -\text{ELBO} = -\mathbb{E}_{q_\phi} \log p_\theta(x_i|z) + KL(q_\phi(z|x_i)|p(z))
$$

3 Question

Since $p_\theta(x|z)$ is the decoder output, it makes sense by definition, but what is $p_\theta(z|x)$?

- We can see it as the "ground truth" posterior