1) Ghazal Kaviani

2.1) The loss that should be used for training standard Autoencoders is the MSE (between input & output).

In some cases, where the input data is between [0, 1], binary cross-entropy loss is also acceptable.

2.2) Vanilla autoencoders don't produce a distribution in the latent/embedding space, rather just a vector, usually without meaningfull entity explanation. Therefore, when using that type of autoencoder, one will only get that from said latent representation, over a whole distribution that VAE encoders are able to generate, thus giving sampling capacity.

Nonetheless, if one is <u>not</u> interested in the samples that can be obtained from the latent space, and rather just shrink all the input information in a smaller representation for other uses, like file compression, denoising, feature extraction, etc., then vanilla autoencoders are sufficient (thus, still usefull).

2.3) $D_{KL}(p(x)||q(x)) = E_{x\sim p(x)}[\log \frac{p(x)}{q(x)}]$

I know that the entropy of a distribution is

$$H(P) = -\mathbb{E}_{x\sim p}\left(\log(P(x))\right) \quad \text{(I)}$$

and that the entropy of a joint distribution is

$$H(P,Q) = -\mathbb{E}_{x\sim p}\left(\log(Q,(x))\right) \quad \text{(II)}$$

If we measure the distance between those distributions (P, Q) assuming that entropy is a good indicator, we can just take the difference of (I) & (II)

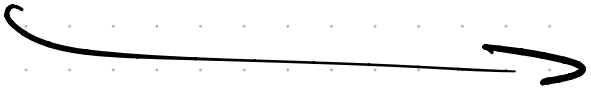
$$H(P,Q) - H(P) = -\mathbb{E}_{x\sim p}\left(\log(Q(x))\right) - \left[-\mathbb{E}_{x\sim p}\left(\log(P(x))\right)\right]$$

$$= -\mathbb{E}_{x\sim p}\left(\log(Q(x))\right) + \left[\mathbb{E}_{x\sim p}\left(\log(P(x))\right)\right]$$

$$= \mathbb{E}_{x\sim p}\left(\log(P(x))\right) - \mathbb{E}_{x\sim p}\left(\log(Q(x))\right)$$

$$= \mathbb{E}_{x\sim p}\left(\log(P(x)) - \log(Q(x))\right) = \boxed{\mathbb{E}_{x\sim p}\left(\log(P(x)/Q(x))\right)}$$

which is the KL divergence



2.3 continued)

if we're trying to draw from a normal distribution:

$P = f_\alpha \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha)$; $Q = g_\beta \sim \mathcal{N}(\mu_\beta, \sigma_\beta)$

then $D_{KL}(P(x) \| Q(x)) = D_{KL}(f_\alpha \| g_\beta) =$

$= \mathbb{E}_{x\sim p}\left(\log\left(\frac{f_\alpha(x)}{g_\beta(x)}\right)\right) = \mathbb{E}_{x\sim p}\left(\log\left(f_\alpha(x) - g_\beta(x)\right)\right)$

$= \mathbb{E}_{x\sim p}\left(\log\left(\frac{1}{\sigma_\alpha\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\mu_\alpha}{\sigma_\alpha}\right)^2} - \frac{1}{\sigma_\beta\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\cdot\left(\frac{x-\mu_\beta}{\sigma_\beta}\right)^2},\right)\right)$

$= \mathbb{E}_{x\sim p}\left(\log(\sigma_\alpha) - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log\left(\frac{x-\mu_\alpha}{\sigma_\alpha}\right) + \frac{1}{2}\log(x^2)\right.$

$\left. + \log(\sigma_\beta) + \frac{1}{2}\log(2\pi) + \frac{1}{2}\log\left(\frac{x-\mu_\beta}{\sigma_\beta}\right) - \frac{1}{2}\log(x^2)\right)$

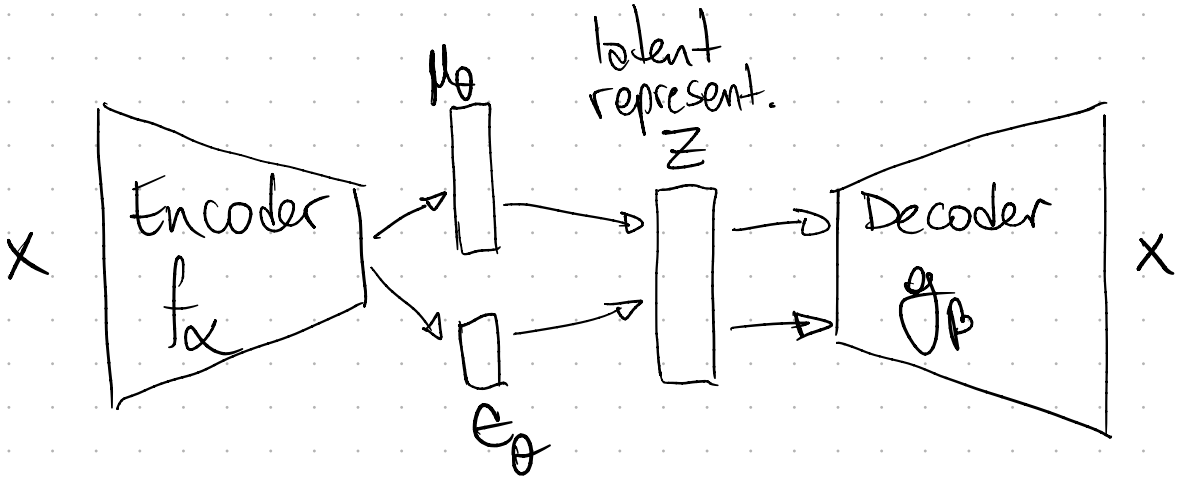
$= \mathbb{E}_{x\sim p}\left(\log(\sigma_\alpha) + \log(\sigma_\beta) - \frac{1}{2}\log\left(\frac{x-\mu_\alpha}{\sigma_\alpha}\right) + \frac{1}{2}\log\left(\frac{x-\mu_\beta}{\sigma_\beta}\right)\right)$

$$= \frac{(\mu_\alpha - \mu_\beta)^2}{\sigma_\alpha^2 \sigma_\beta^2} + \frac{1}{2}\left(\frac{\sigma_\alpha^2}{\sigma_\beta^2} - 1 - \log\left(\sigma_\alpha^2/\sigma_\beta^2\right)\right) = D_{KL}(f_\alpha \| g_\beta)$$

2.4) Gaussian reparametrization trick:

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(x_{t-1}; \mu_q(t), \Sigma_q(t))$$

$$\mu_q = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \right) \epsilon_\theta(x_t, t), \quad \Sigma \text{ assumed known}$$



2.5)

$$ELBO = E_{z\sim q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)]$$

Show that $\log(p_\theta(x)) \geq ELBO$.

For cleanliness, we drop θ and ϕ, since they will be learned and don't play a role here. Therefore:

Expanding ELBO:

$$\mathbb{E}_{z\sim q(z|x)}\left(\log(p(x,z)) - \log(q(z|x))\right)$$

$p(x,z) = p(x|z)\cdot p(z)$ from Bayes

$$= \mathbb{E}_{z\sim q(z|x)} \log\left(\frac{p(x|z)\cdot p(z)}{q(z|x)}\right) \quad (I)$$

Taking $\log(p(x))$ requires knowledge of q:

$$\log(p(x)) = \underbrace{\mathbb{E}_q\left(\log\left(\frac{p(x|z)\cdot p(z)}{q(z|x)}\right)\right)}_{(I) = ELBO} + \underbrace{D_{KL}(q(z|x)\,\|\,p(z|x))}_{\text{positive number or } 0}$$

$\therefore \log(p_\theta(x)) \geq ELBO$ ■

2.6)

$ELBO = E_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - E_{z\sim q_\phi(z|x)}[\log \frac{q_\phi(z|x)}{p_\theta(z)}]$

(Also dropping θ & ϕ)

From 2.5: $ELBO = \mathbb{E}_{z\sim q(z|x)}\left(\log(p(x,z)) - \log(q(z|x))\right)$

$p(x,z) = p(x|z)\cdot p(z)$ from Bayes

$\therefore ELBO = \mathbb{E}_{z\sim q(z|x)} \log\left(\frac{p(x|z)\cdot p(z)}{q(z|x)}\right)$ (I)

manipulating (I):

$$\mathbb{E}_{z\sim q(z|x)} = \left(\log\left(\frac{p(x|z)\cdot p(z)}{q(z|x)}\cdot\frac{p(z)}{p(z)}\right)\right) = \mathbb{E}_{z\sim q}\left(\log\left(\frac{p(x|z)\cdot p(z)}{q(z|x)}\right) + \log p(z) - \log p(z)\right)$$

$$= \mathbb{E}_{z\sim q(z|x)}\left(\log(p(x|z)\cdot p(z)) + \log p(z) - \log(q(z|x)) - \log p(z)\right)$$

$$= \mathbb{E}_{z\sim q(z|x)}\left(\log(p(x|z)) + \log p(z) + \underbrace{\log p(z) - \log(q(z|x))}_{= -\log\left(\frac{q(z|x)}{p(z)}\right)} - \log p(z)\right)$$

$$= \mathbb{E}_{z\sim q(z|x)}\left(\log(p(x|z)) + \log p(z) - \log\left(\frac{q(z|x)}{p(z)}\right) - \log p(z)\right)$$

$$= \mathbb{E}_{z\sim q(z|x)}\left(\log(p(x|z))\right) - \mathbb{E}_{z\sim q(z|x)}\left(\log\left(\frac{q(z|x)}{p(z)}\right)\right)$$

■

3.1) The two-player game played by a GAN is rather simple: The Generator wants to generate samples so good that they could fool the discriminator, which, in its turn, wants to always be able to tell real vs fake apart.

In this case the Nash equilibrium is unclear even locally. Instead, it is searched in the parameter space given some assumptions as presented by Goodfellow, et-al in Generative Adversarial Nets (NeurIPS, 2014) ("the GANs paper").

3.2)

X = reg
y = monet

- The role of:
  - G is to generate regular images that look like Monet paintings
  - D_X is to tell actual real images from images generated by G
  - F is to generate Monet paintings that look like regular images
  - D_Y is to tell actual Monet paintings from the ones generated by F
- There are 2 two-player games: One for G vs. D_X and one for F vs. D_Y.
- there is no obvious Nash equilibria.

3.3)

- The cycle-consistency loss is needed to ensure that the mapping from a domain to another, and then back (full cycle: out from a domain and back to it after 2 mappings/transfers) is consistent to the input.

that is, $F(G(x)) = x$ & $G(F(x)) = x$.

- If not enforced, the final transformation might lose information and the final output could be significantly different than the input.
- I actually call F & G as F & G in the previous question, but regardless, they are the mappings/transformations from one domain to another (real images and Monet paintings).