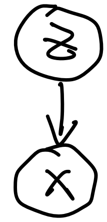
$\boxed{\text{Why VAE}}$: traditional autoencoders encode the input into a single vector, which is a single point in the high-dim space. when generating new, unseen data, they fail to interpolate between seen data, so cannot generate those variations. In contrast, VAE maps inputs into a distribution (Gaussian), which forces in the circle area around μ to have data which makes data embedding in the high-dim space denser and easier to interpolate.

(VAE is like Gaussian Mixture Model, traditional autoencoder is like K-means)

Derivation



What we want:

generate data x' which are similar to the training data X

$\Rightarrow$ we need to model $p(z|X)$ which we can sample z from and generate x: $p(x'|X) = \int p(z|X)\, p(x'|z)\, dz$

$\Rightarrow$ how to model $p(z|X)$

$$p(z|X) = \frac{p(X|z)\, p(z)}{p(X)}$$

$\Rightarrow$ problem: $p(X) = \int p(X|z)\, p(z)\, dz$ is intractable due to continuous z

$\because$ cannot iterate all z

$\therefore$ cannot compute integral using computer.

$\Rightarrow$ solution: use another dist $\boxed{q(z|X)}$ to replace $p(z|X)$ in order to get samplings.

Easier distribution e.g. diagonal Gaussian.

$\Rightarrow$ Back to our goal: we need to generate X' from $p(z|X)$ which are similar to X. So we need $q(z|X)$ is as close to $p(z|X)$ as possible.

$\Rightarrow$ $KL(q(z|X) \| p(z|X))$

$$= \int_z q(z|X) \log \frac{q(z|X)}{p(z|X)} dz$$

$$= \int_z q(z|X) \log q(z|X)\, dz - \boxed{\int_z q(z|X) \log p(z|X)\, dz}$$

$$= \int_z q(z|X) \log \frac{p(X, z)}{p(X)} dz$$

$$= \int_z q(z|X) \log p(X, z)\, dz$$

$$- \boxed{\int_z q(z|x) \log p(x)\, dz}$$

$$= \log p(x)$$

$$\therefore KL(q(z|x) \| p(z|x))$$

$$= \int_z q(z|x) \log q(z|x)\, dz$$

$$- \int_z q(z|x) \log p(x, z)\, dz$$

$$+ \log p(x)$$

$\Rightarrow$ problem: $p(x)$ cannot be computed !! $\int p(x|z)\, p(z)\, dz$.

$$\Rightarrow \boxed{\log p(x) - KL(q(z|x) \| p(z|x))}$$

what we want to maximize

$$= \int_z q(z|x) \log p(x, z)\, dz$$

$$- \int_z q(z|x) \log q(z|x)\, dz$$

① $\log p(x) \uparrow$

$$= \boxed{\mathbb{E}_{q(z|x)}[\log p(x, z)] - \mathbb{E}_{q(z|x)}[\log q(z|x)]}$$

$\Rightarrow$ similar to real data

ELBO

② $q(z|x) \Rightarrow p(z|x)$

$\Rightarrow$ Equivalently, we need to maximize ELBO

Understand ELBO:

$$\text{ELBO} = \mathbb{E}_{q(z|x)}[\log p(x,z)] - \mathbb{E}_{q(z|x)}[\log q(z|x)]$$

$$= \mathbb{E}_{q(z|x)}[\log p(z)] + \mathbb{E}_{q(z|x)}[\log p(x|z)]$$

$$- \mathbb{E}_{q(z|x)}[\log q(z|x)]$$

$$= \boxed{\mathbb{E}_{q(z|x)}[\log p(x|z)]} - \boxed{KL(q(z|x) \| \underbrace{p(z)}_{N(0,I)})}$$

encourage $q(z|x)$ to put as much weights as possible on the mode of $\log p(x|z)$ (or $p(x|z)$)

encourage $q(z|x)$ to stretch a little bit to approximate $p(z)$

If $q(z|x)$ is so narrow, we can add more weights to $KL(q(z|x)\,||\,p(z))$

In VAE:

Encoder Network to represent $q_\theta(z|X)$

Generate (Decoder) Network to represent $P_\phi(X|z)$

Make Everything differentiable:

Reparameterization Trick.

sample $z \sim \mathcal{N}(\mu, \sigma^2)$

$\Leftarrow \quad z = \mu + \sigma \odot \varepsilon \qquad \varepsilon \in \mathcal{N}(0, I)$