2024-11-02

Unsupervised Deep learning

* L11 : Variational Auto encoders
* L12 : Generative Adversarial Networks (GAN)
* L13 : Diffussion models

Generative AI

Introduction

- So far, **supervised learning**
  - Discriminative methods:

$$\{\mathbf{x}_i, y_i\}_{i=1}^N \rightarrow p(y|\mathbf{x})$$

linear regression
logistic / softmax regression
FNN
CNN
ViT

  - Generative methods:

$$\{\mathbf{x}_i, y_i\}_{i=1}^N \rightarrow P(y), p(\mathbf{x}|y)$$

class size
class characteristics
Gaussian Discriminant analysis

PCA

- Next, **unsupervised learning**:
  - *Finite mixture models* for clustering [Skipped]

$$\{\mathbf{x}_i\}_{i=1}^N \rightarrow P(z), p(\mathbf{x}|z)$$

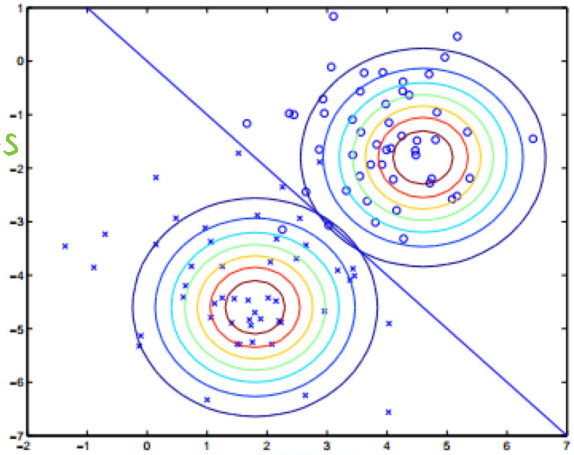
No y_i
cluster size
↑ cluster characteristics

Z : latent variable

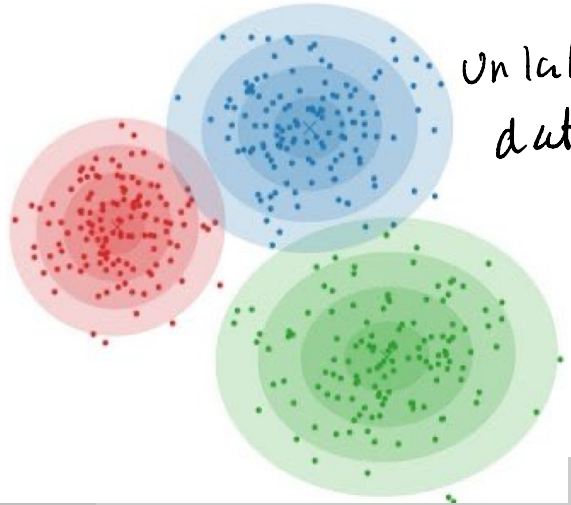
Intelligence
Math grade
physics grade
history grade

Observed variable

Gaussian Mixture Models



labelled data



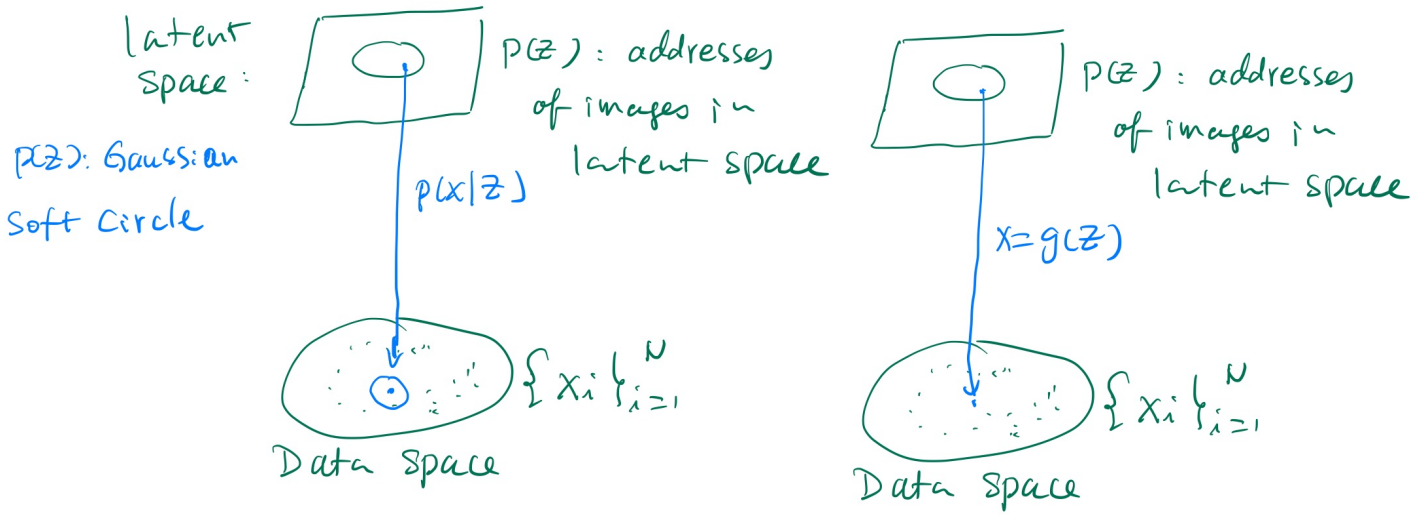
Unlabelled data

- Next, **unsupervised learning**:
  - *Varitional autoencoder* for data generation and representation learning

    $\{\mathbf{x}_i\}_{i=1}^N, p(\mathbf{z}) \rightarrow p(\mathbf{x}|\mathbf{z})$ $q(\mathbf{z}|\mathbf{x})$ used in inference

    * Diffusion Model
  - *Generative adversarial networks* for data generation

    $\{\mathbf{x}_i\}_{i=1}^N, p(\mathbf{z}) \rightarrow \mathbf{x} = g(\mathbf{z})$



Model used to generate new data:

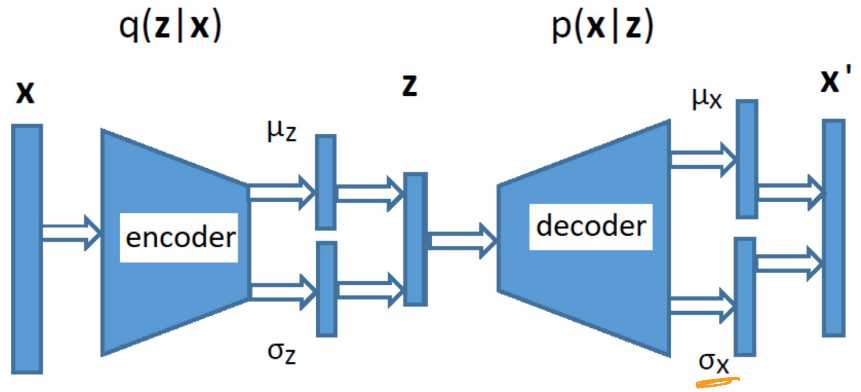
VAE: $z \sim p(z)$ $x \sim p(x|z)$

GAN: $z \sim p(z)$ $x = g(z)$

New image

Lecture 11: Variational Autoencoder (VAE)

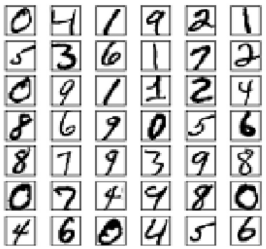
$$\Sigma = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_D^2 \end{bmatrix}$$



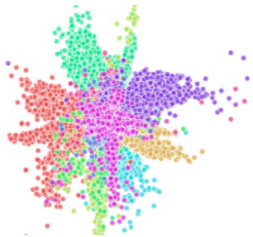
* Want : $p(x|z)$

Decoder : $z \rightarrow x$

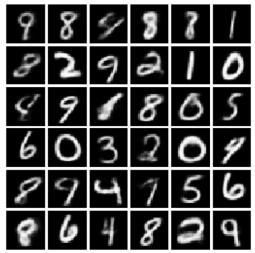
* Introduce encoder $q(z|x)$ in order to train the decoder.



Real



Latent Space



Fake

* General idea

Encoder : $x \longrightarrow z$ (probabilistic)

real data — latent vector

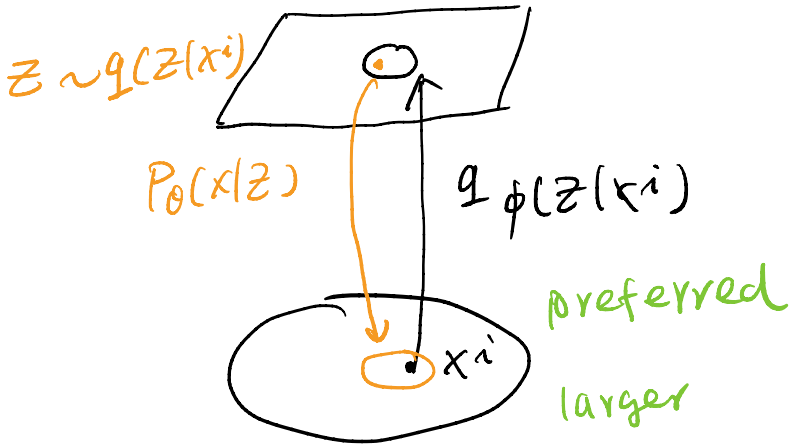
Decoder : $z \longrightarrow x'$

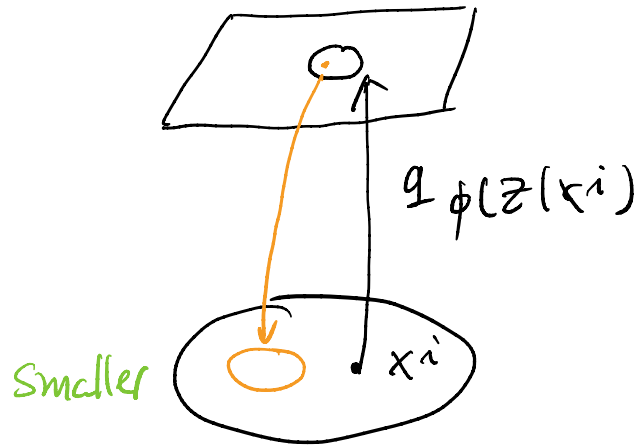
latent vector — fake data

Training Target : $x' \approx x$

The VAE objective function

$\{x^i\}_{i=1}^N$, $p(z)$



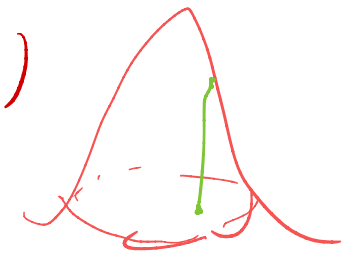


$$L(x^i, \theta, \phi) = E_{z \sim q(z|x^i)}[\log p(x^i|z)] - KL(q(z|x^i)||p(z))$$

reconstruction "error"

regularization

$$\max_{\theta,\phi} L(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} L(x^i, \theta, \phi)$$

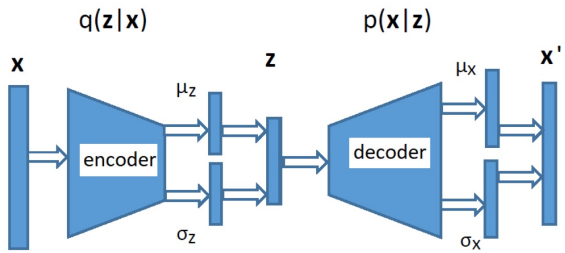


VAE

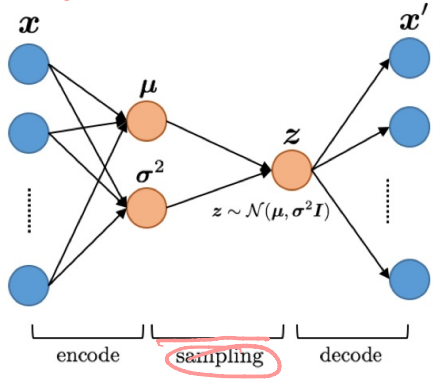
Given : $\{x^i\}_{i=1}^N$ $P(z)$

$$\max_{\theta,\phi} \frac{1}{N} \sum_{i=1}^{N} L(x^i, \theta, \phi)$$

$$\mathcal{L}(\mathbf{x}^{(i)}, \boldsymbol{\theta}, \boldsymbol{\phi}) = E_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right] - \mathcal{D}_{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})]$$



The Need For Reparameterization



- The computation of the first term $\mathcal{L}_1$ of $\mathcal{L}$ requires **sampling**:

$$\mathcal{L}_1 = E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})\right] \approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$$

where $\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$.

The Reparameterization Trick

$\sigma = x^2$

scalar x, z

$q(z|x) = N(x^2+2, x^4)$

$z \sim q(z|x)$ $\quad \frac{dz}{dx} = 0$

$z = x^2 + 2 + x^2 \varepsilon \qquad \varepsilon \sim N(0, 1)$

Gaussian distributed

$E[z] = x^2 + 2$

$\sigma(z) = x^2$

$\frac{dz}{dx} = 2x + 2x\varepsilon$

The Reparameterization Trick

$$\mathcal{L}_1 = E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})\right]$$

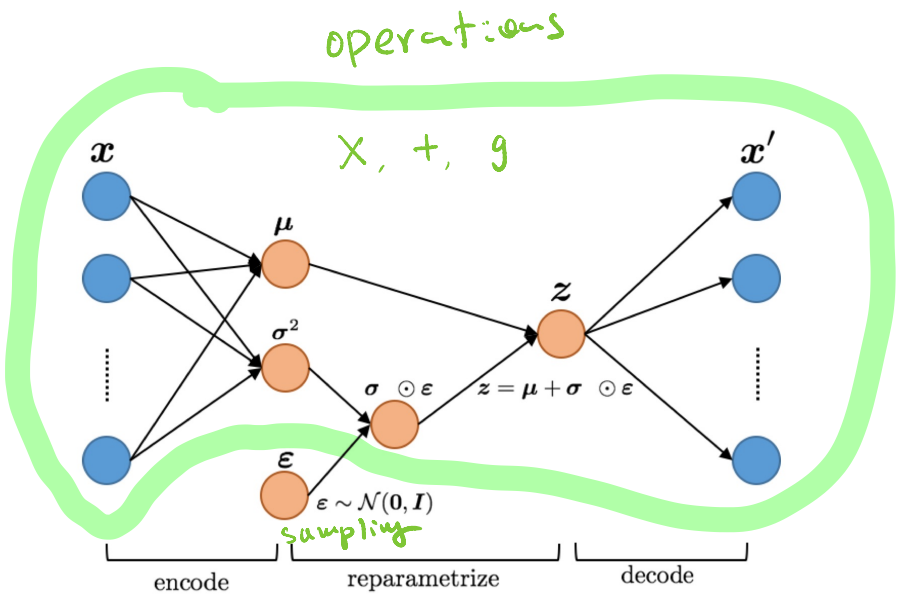
$z(x, \phi, \varepsilon)$

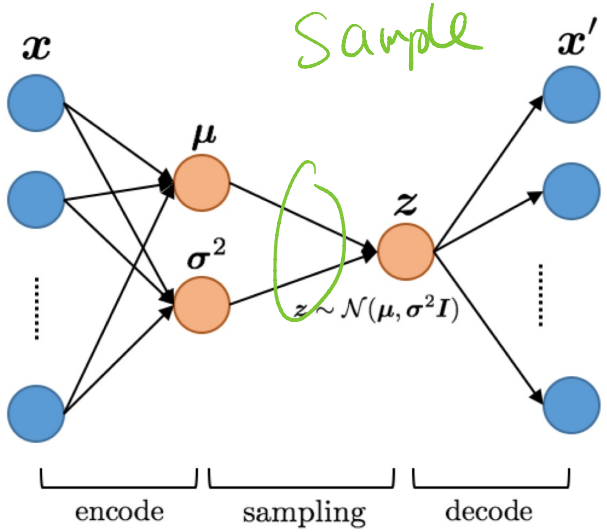
$$\approx \frac{1}{L}\sum_{l=1}^{L} \log p_\theta(x^i | z(x^i, \phi, \varepsilon^l))$$

Depends on θ, ϕ

$$\mathbf{z} = \mu_z(\mathbf{x}, \phi) + \sigma_z(\mathbf{x}, \phi) \odot \epsilon, \ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$\varepsilon^1, \ldots, \varepsilon^L \sim N(0, I)$





The Second term

Gaussian Distribution

max $\mathcal{L}(\mathbf{x}^{(i)}, \theta, \phi) = E_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})\right] - \mathcal{D}_{KL}[q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})]$

max $\mathcal{L}_2 = -\mathcal{D}_{KL}[q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})]$

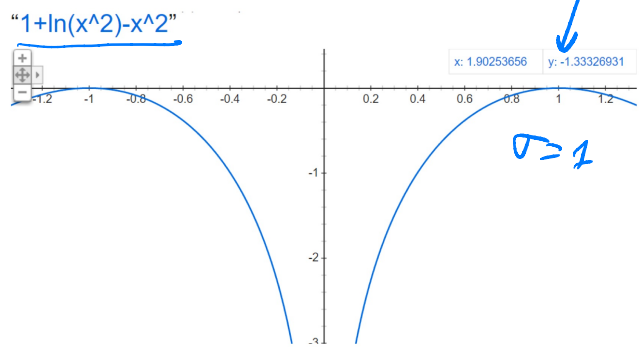
$$= \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right)$$

$q_\phi(z|x^i)$ $\qquad$ $p(z)$

μ $\begin{bmatrix} \mu_1^i \\ \vdots \\ \mu_J^i \end{bmatrix}$ $\xrightarrow{\max L_2}$ $\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

Σ $\begin{bmatrix} (\sigma_1^i)^2 & & 0 \\ & \ddots & \\ 0 & & (\sigma_J^i)^2 \end{bmatrix} \xrightarrow{\max L_2} \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$

max $1 + \log\sigma^2 - \sigma^2$

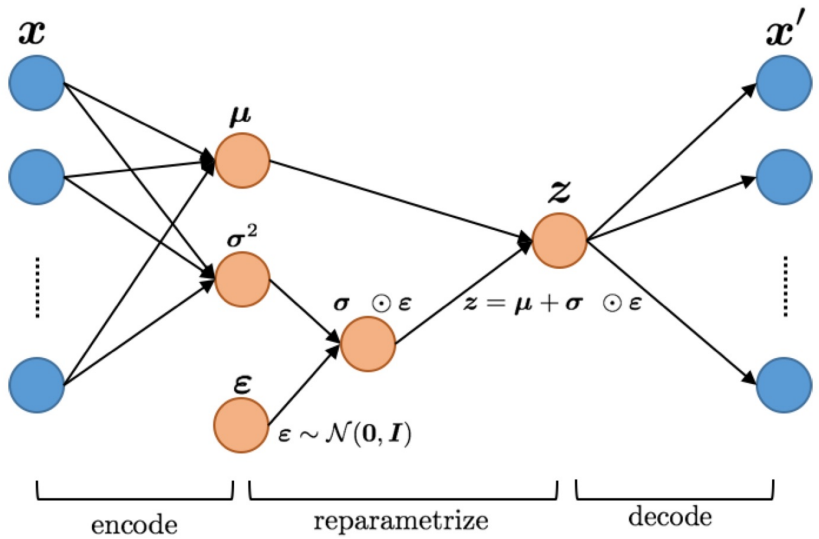


The Final Objective Function

- Putting together, this is the objective function that we maximize using gradient ascent

$$\mathcal{L} \approx \frac{1}{L}\sum_{l=1}^{L} \log p_{\theta}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}) + \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right)$$

where $\mathbf{z}^{(i,l)} = \mu_z(\mathbf{x}^{(i)}, \phi) + \sigma_z^2(\mathbf{x}^{(i)}, \phi) \odot \epsilon^{(l)}$, and $\epsilon^{(l)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

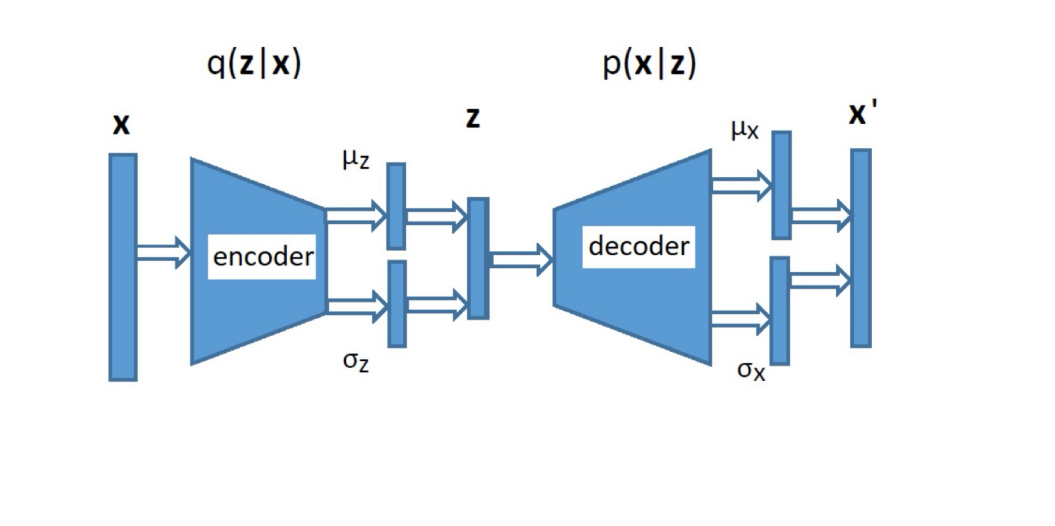


Tutorial 7 VAE

MNIST: images with binary pixels

each pixel: 0, 1

$p(x|z)$: prob of each pixel being 1

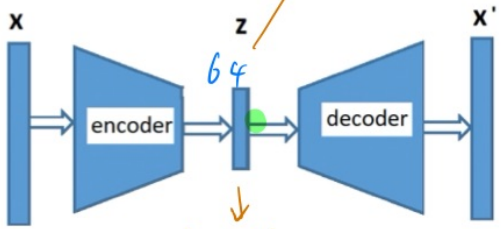


Other Auto encoder models

Auto encoder

AE

28*28



representation

64

bottle neck

$\min \|x - x'\|$

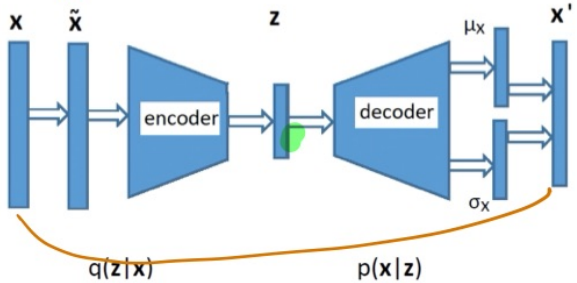
Not regularize

$p(z)$

Denoising AE

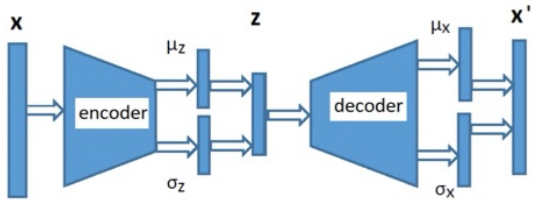
DAE

Add noise



$x \approx x'$

VAE



$p(z)$: $N(0, 1)$

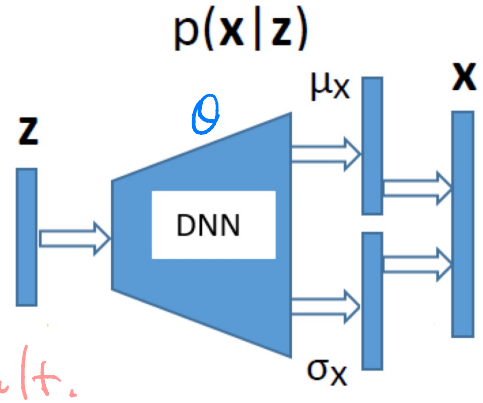
Data Distribution in Latent Space (MNIST)

- VAE: Forces data into a normal distribution in the latent space.
- DAE: Preserves class separation better.

we know where the real images live in the latent space
so we can generate fake images that look like them

Theory of VAE (shared by Diffusion models)

$\{x^i\}_{i=1}^N, \; p(z) \Longrightarrow$



$$p_\theta(x) = \int p_\theta(x|z)\, p(z)\, dx$$

learned, prior

$$\max \ell(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x^i) \quad : \text{Difficult.}$$

$$E_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right]$$

$$= \int q(z|x) \log \frac{p(x,z)}{q(z|x)}\, dz$$

$$\leq \log \int q(z|x) \frac{p(x,z)}{q(z|x)}\, dz$$

Jensen's inequality

$$= \log \int p(x,z)\, dz$$

$$= \log p(x)$$

$$E_{q(z|x)}\left[\log \frac{p(x,z)}{q(z|x)}\right]$$

$$= E_{q(z|x)}\left[\log \frac{p(z)\, p(x|z)}{q(z|x)}\right]$$

$$= E_{q(z|x)}[\log p(x|z)] - E_{q(z|x)}\left[\log \frac{q(z|x)}{p(z)}\right]$$

$$\downarrow$$

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

Therefore:

$$\log p(x) \geq E_{q(z|x)}[\log (x|z)] - KL(q(z|x) \,||\, p(z))$$

variational lower bound,

evidence lower bound (ELBO)