Imperial College London

Variational AutoEncoders (VAEs)

Introduction to the theory of VAEs

Jiashun Yao

Outline

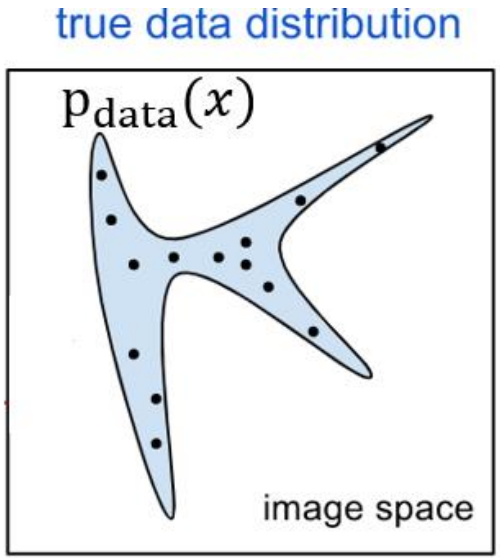
- Generative Model
- Autoencoders
- VAEs
- Mathematics of VAEs

Generative model

Given real (training set) data, we want to use a generator G to generate such data.

→ find the distribution $p_{data}(x)$ that describes where the real data are likely to locate in the high dimensional space

→ Sample from $p_{data}(x)$ to generate realistic samples.

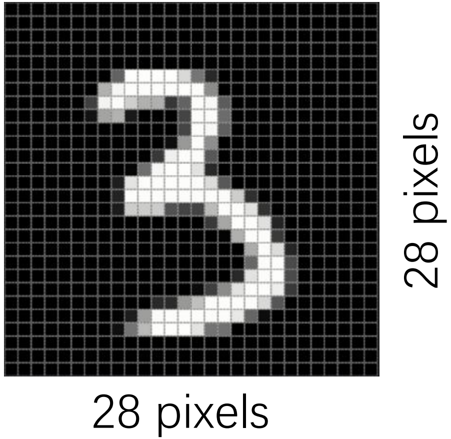


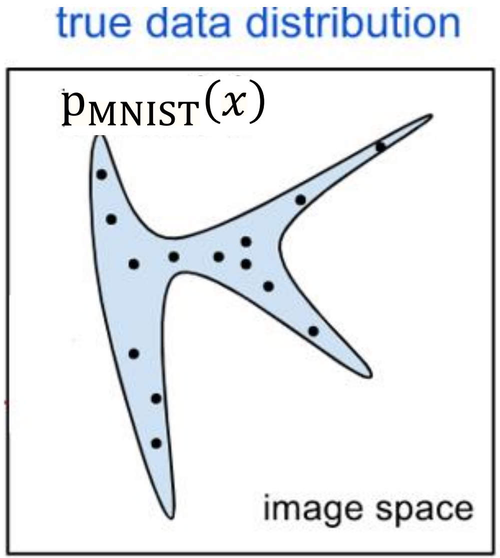
x – image or other data high dimensional vector

Image: https://openai.com/blog/generative-models/

Generative model

Each MNIST image can be seen as a 28x28 length vector, it is a sample in a 784-dimensional space.





Imagine $p_{MNIST}(x)$ is associated with a region on this 784-dimensional space.

Cannot explicitly write the probability function that describes it (too complex, we only have limited number of samples).

Use a model G to learn/approximate $p_{MNIST}(x)$.

Image: https://openai.com/blog/generative-models/

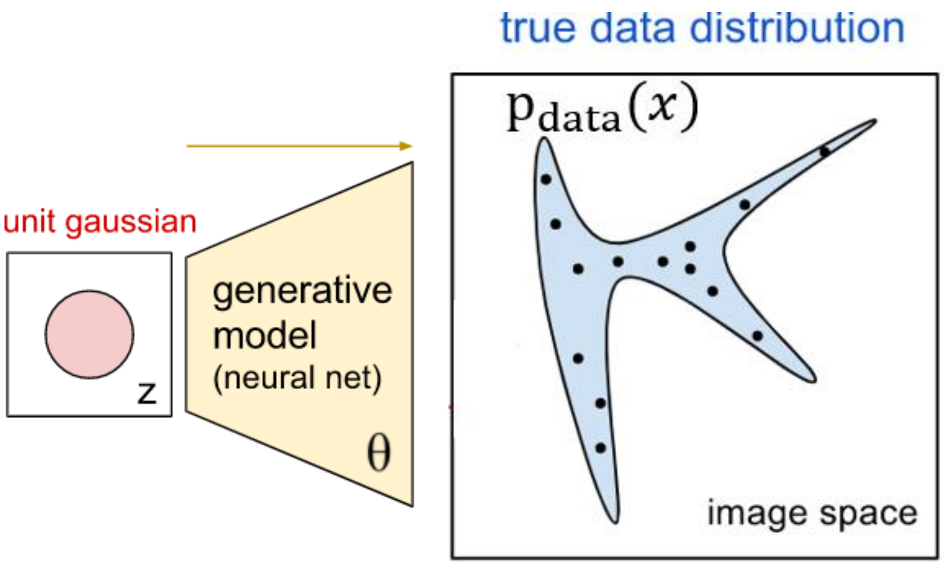
Generative model

To use a model G to generate the data in $p_{data}(x)$, there are different types of methods, two straightforward ones are:

- Assume the p_{data} to be in a certain form, construct it, then sample from it
    - GMM for example
- Use a neural network G_θ to estimate a mapping from a low-dimensional space simple distribution $p(z)$ to the high-dimensional data space real data distribution $p_{data}(x)$: $G_\theta(z) \text{ –> } x$
    - In VAE:
        - The low dimensional space –> latent space
        - The simple distribution $p(z)$ –> normal distribution
        - $z \sim p(z)$ –> latent space sample
        - $x \sim p_\theta(x|z)$

Generative model

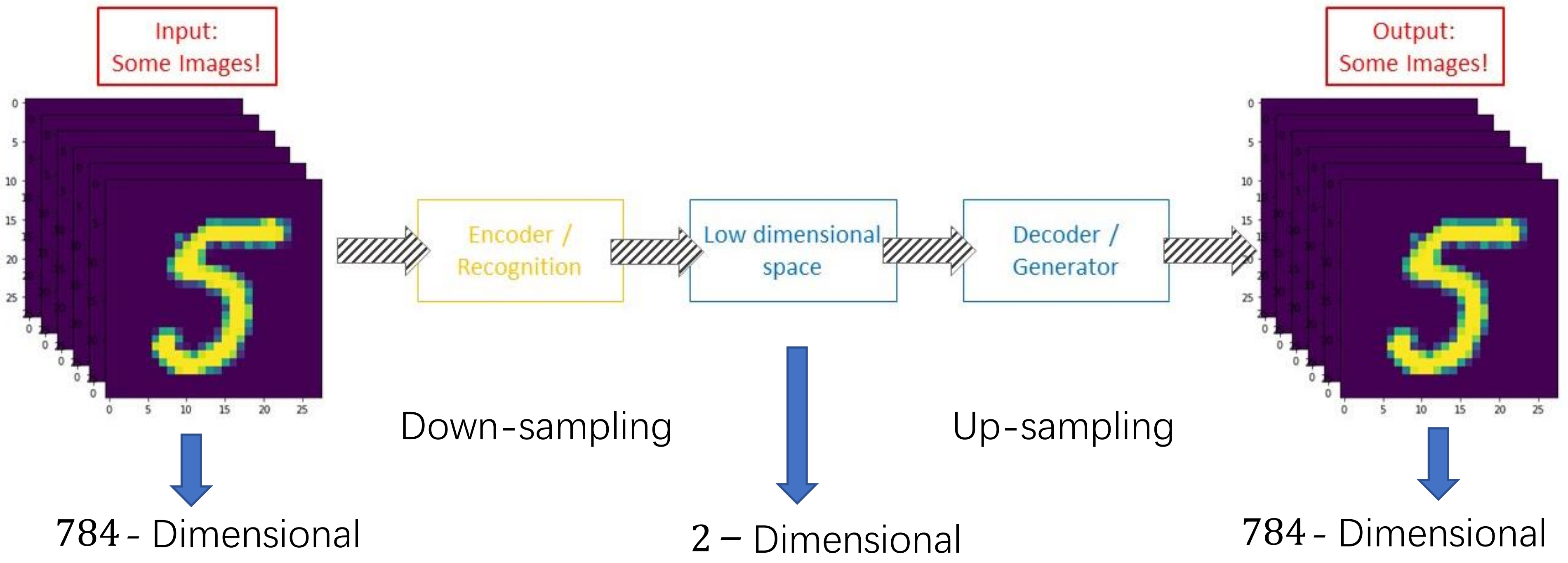
- $z \rightarrow x$
- $z \sim p(z)$
- $x \sim p_\theta(x|z)$
- The joint distribution expressed by the generative model is: $p_\theta(x,z) = p_\theta(x|z)p(z)$



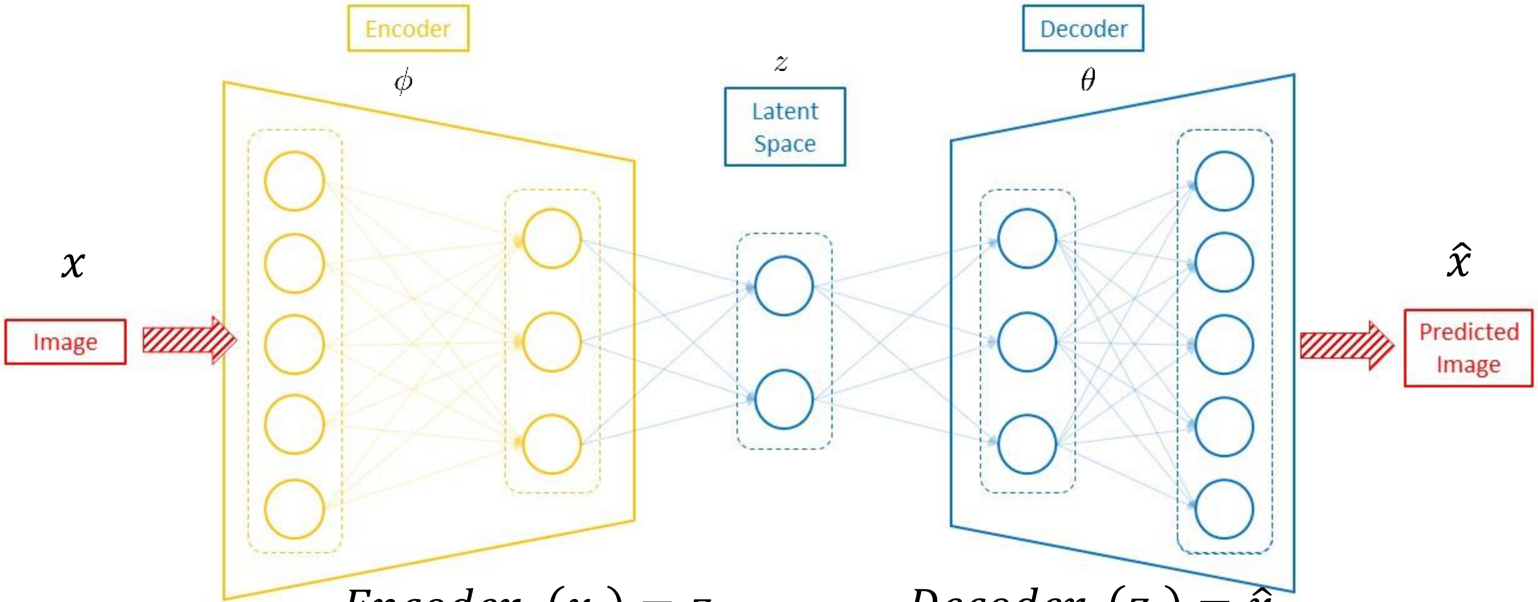
Practically, to generate a data sample, instead of sampling from complex $p(x)$, we sample from a much lower dimensional simple distribution $p(z)$, then use the network G_θ to map it to a high-dimensional data sample.

Image: https://openai.com/blog/generative-models/

Auto-encoder



Auto-encoder

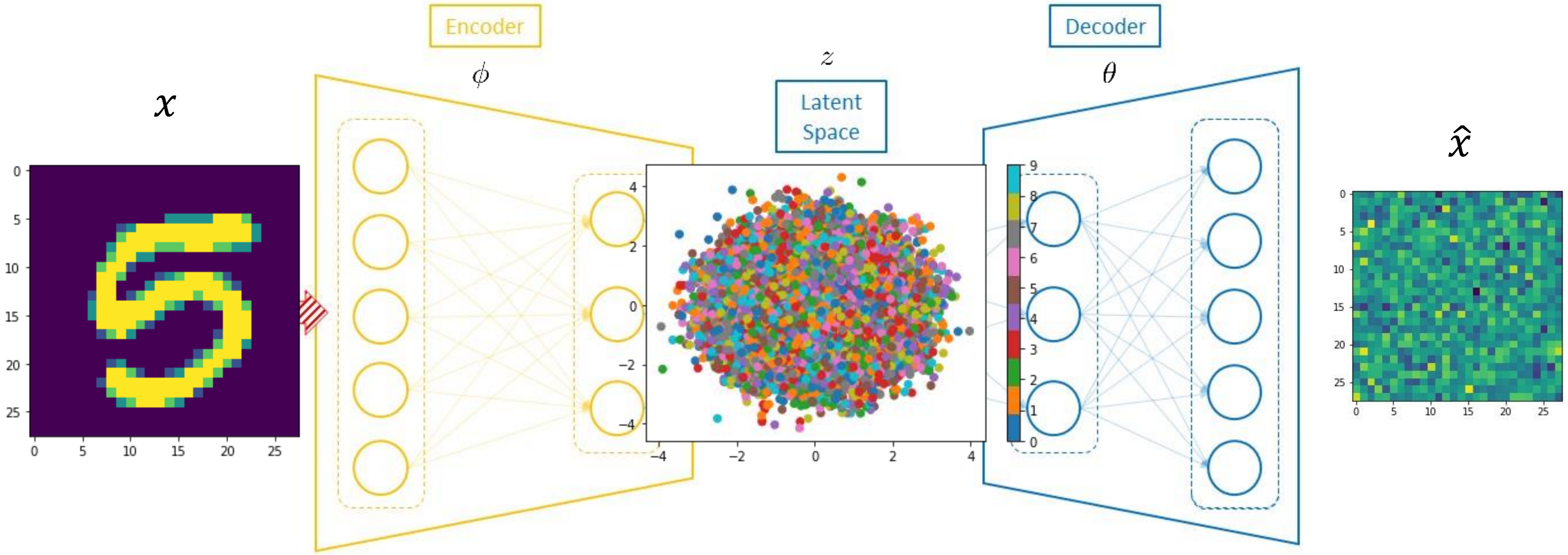


$$Encoder_{\phi}(x_i) = z_i \qquad Decoder_{\theta}(z_i) = \hat{x}_i$$

Can be used as a unsupervised learning tool for dimension reduction.

Using the latent space representations, we can do image classification, regression, property analyses, etc.

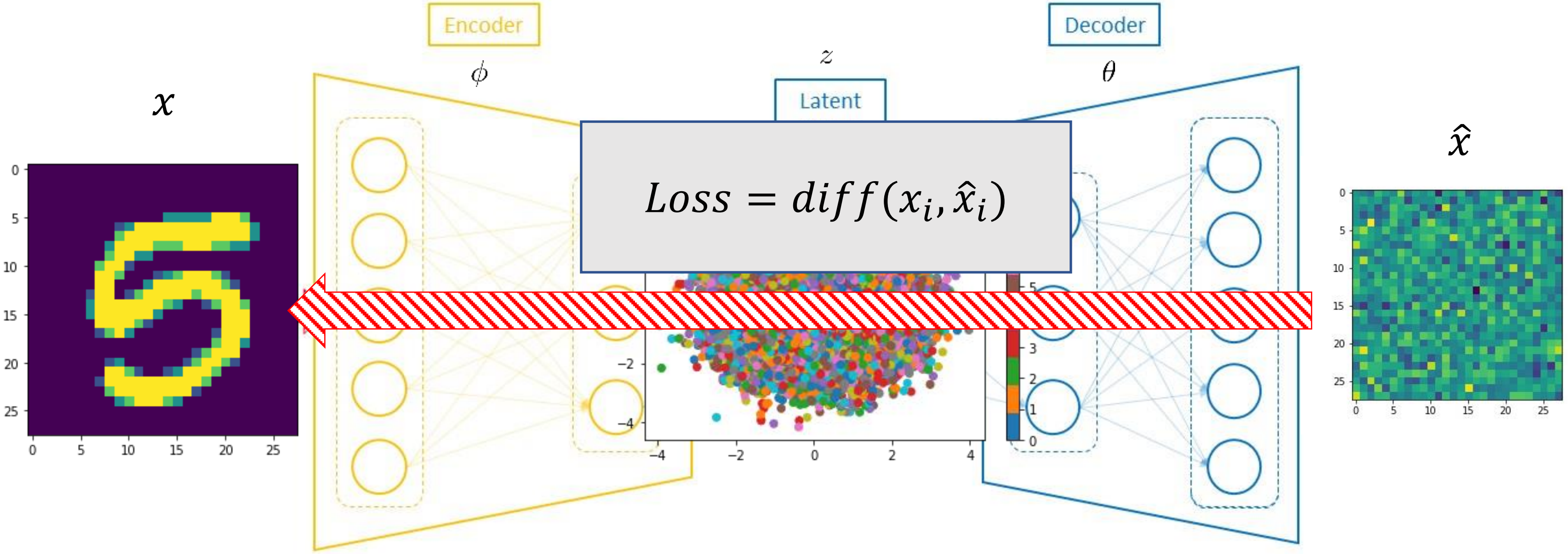
Auto-encoder - Training



Before training – with random ϕ and θ:

- $\hat{x}$ is random
- Latent space representations are randomly distributed

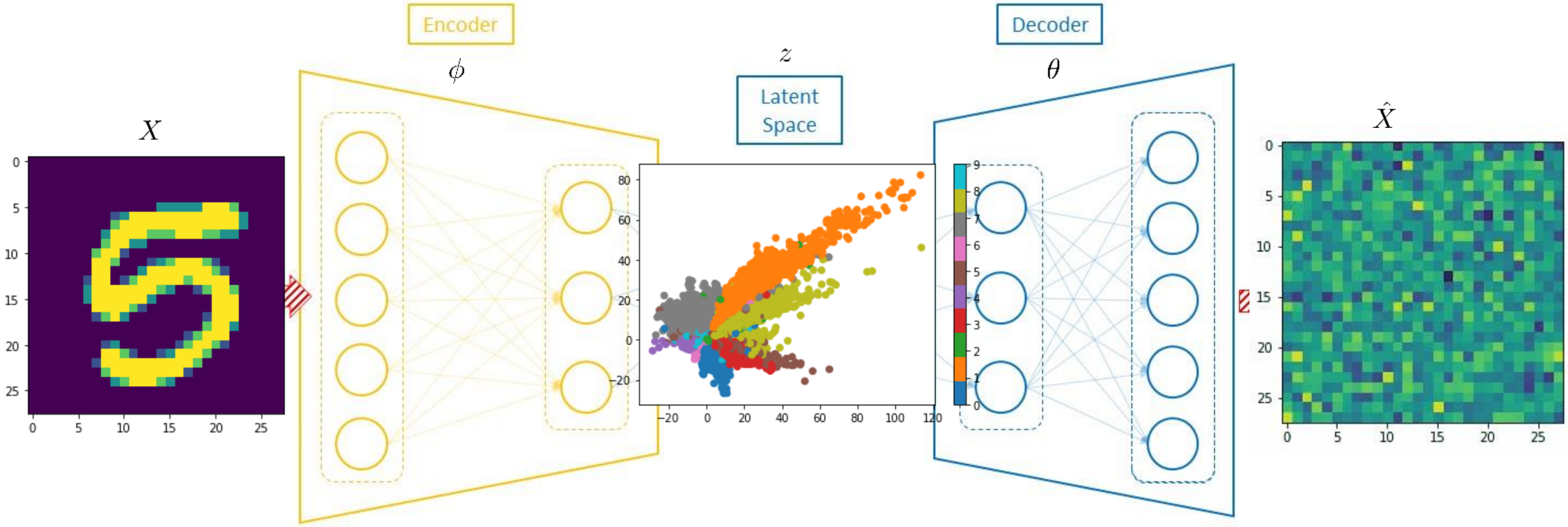
Auto-encoder - Training



$$Loss = diff(x_i, \hat{x}_i)$$

To train, minimise the mismatch (usually pixel-wise differences) between each x_i and $\hat{x}_i$ through back-propagation.

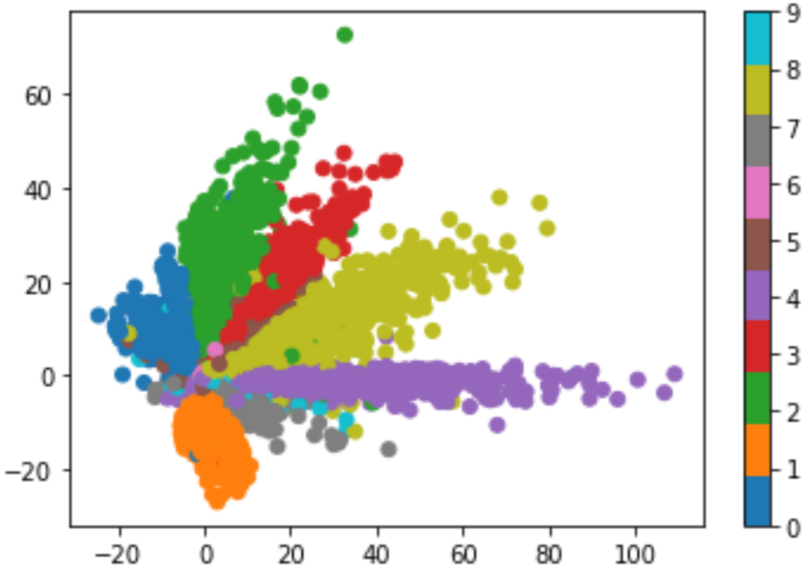
Auto-encoder - Training

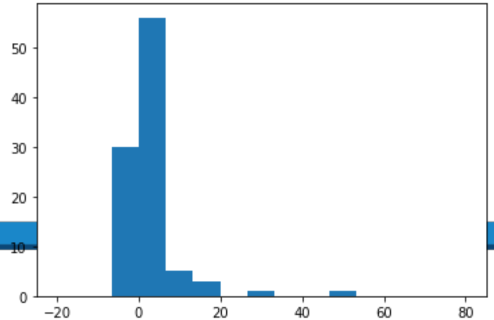


Well trained auto-encoder:

- $\hat{x}$ matches to x
- Latent space representations are in certain patterned clusters

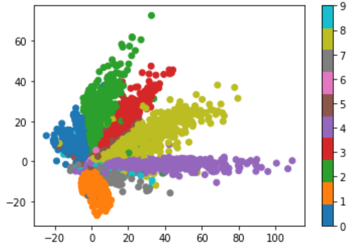
Why not this decoder as a data generator?

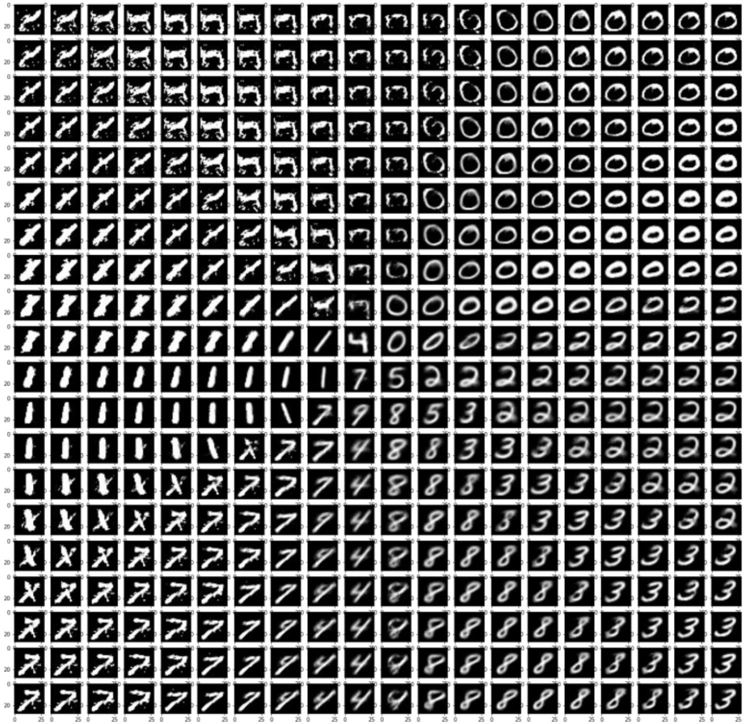




- Hard to get a valid sample from the latent space:
  - When we visualise our latent space, we see that the range of the latent vector is not well bounded
  - It may seem to follow a certain distribution, but we cannot know it easily

Why not this decoder as a data generator?



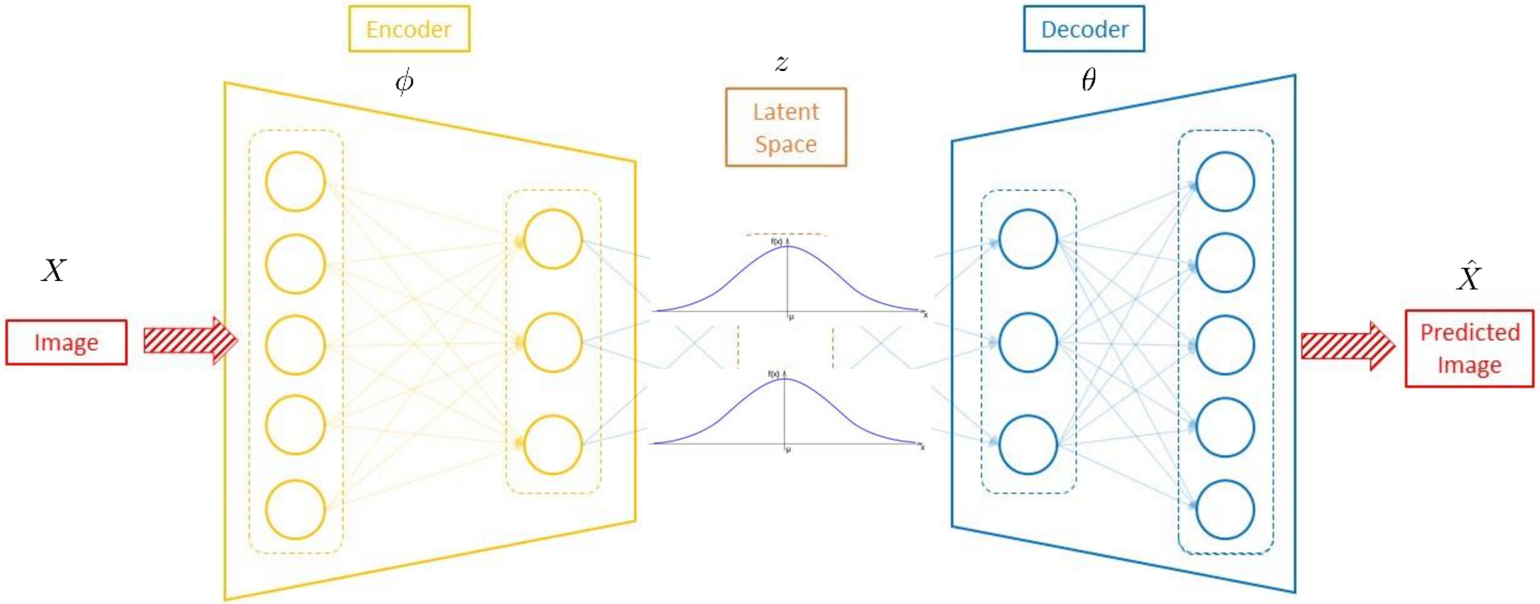


Randomly sampled latent space representations are usually invalid.

How to solve this problem?

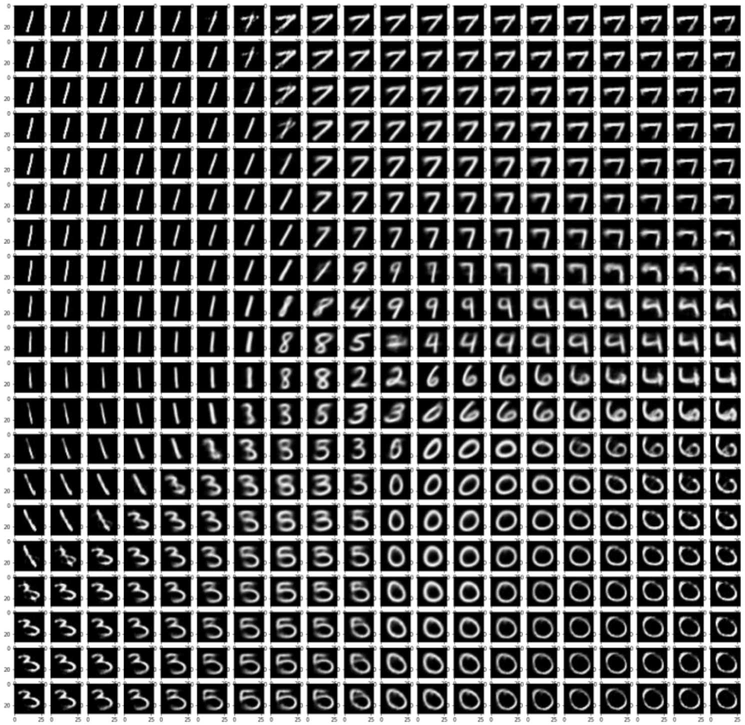
- **Force the valid latent representations to follow a specific distribution – VAEs**

Variational Auto-encoder (VAE)

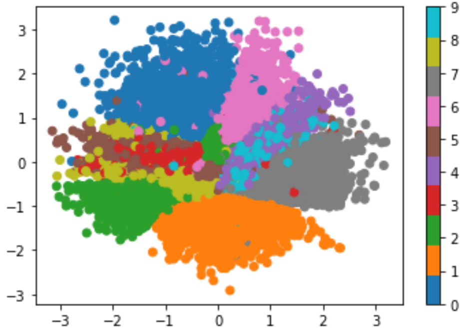


Now the latent space representations are from a distribution of our choice (e.g. multivariate normal distribution).

Variational Auto-encoder (VAE)

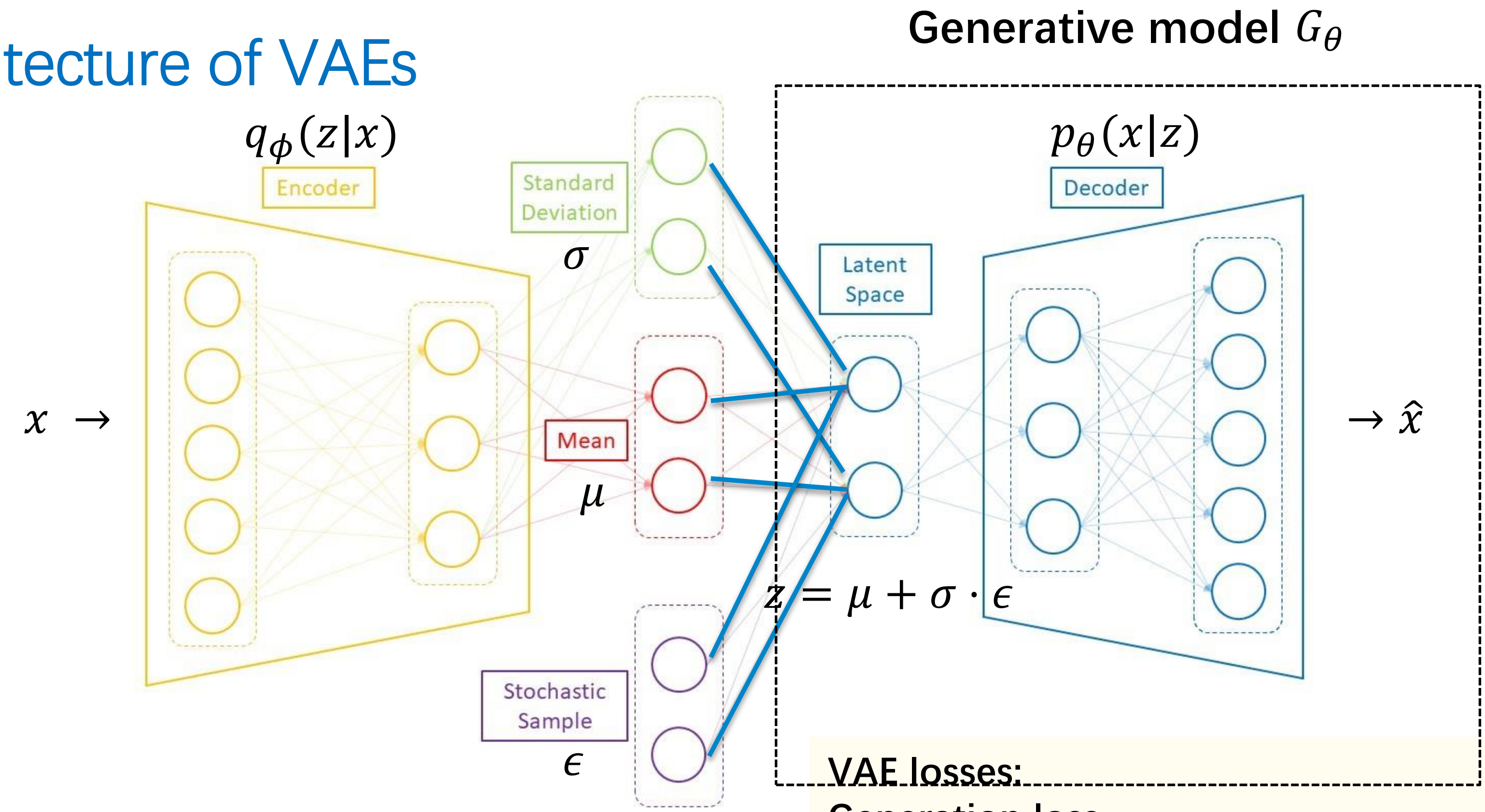


- Intuitively, by doing this, now our valid latent space representations are following a distribution.



- As long as we sample z_i from this distribution, we no longer generate invalid samples.

Architecture of VAEs



VAE losses:

Generation loss

$$diff(x_i, \hat{x}_i)$$

Latent space loss

$$KL\ (latent\ variable\ ||\ unit\ Gaussian)$$

Mathematics of VAEs

- Decoder (generator) network with parameter θ :
  - $z \sim p(z) = N(0, I)$ – z is from normal distribution
  - $x|z \sim p_\theta(x|z)$
    - z → ***Decoder network with parameters*** θ → x

$x \rightarrow NN_\phi \rightarrow \big(\mu(x), \sigma(x)\big) \underline{\rightarrow z}$
$z \rightarrow NN_\theta \rightarrow x$

- Encoder network with parameter ϕ :
  - $q_\phi(z|x)$
  - x → ***Encoder network with parameters*** ϕ → $(\mu(x), \sigma(x))$ → z

- We want to find θ maximizes the likelihood of the training set samples (i.e., given the generator distribution, maximize the probability values of training set samples):

$$L = \log p_\theta(x)$$

Mathematics of VAEs

- $L = \log p_\theta(x)$
- $p_\theta(x) = \int_z p_\theta(x|z)p(z)dz$ -- intractable

$$\int_z f(z|x)dz = \int_z \frac{f(z,x)}{f(x)}dz = \frac{1}{f(x)}\int_z f(z,x)dz = \frac{1}{f(x)} \cdot f(x) = 1$$

- Inserting $q_\phi(z|x)$, and do rearrangements:
  - $\log p_\theta(x) = \int_z q_\phi(z|x) \log p_\theta(x) dz$

$$= \int_z q_\phi(z|x) \log\left(\frac{p_\theta(z,x)}{p_\theta(z|x)}\right) dz = \int_z q_\phi(z|x) \log\left(\frac{p_\theta(z,x)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p_\theta(z|x)}\right) dz$$

$$= \int_z q_\phi(z|x) \log\left(\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right) dz + \int_z q_\phi(z|x) \log\left(\frac{q_\phi(z|x)}{p_\theta(z|x)}\right) dz$$

$KL(q_\phi(z|x) \,||\, p_\theta(z|x))$ – KL divergence

- $= 0$ if $q_\phi(z|x)$ and $p_\theta(z|x)$ are identical
- > 0 if $q_\phi(z|x)$ and $p_\theta(z|x)$ are not identical

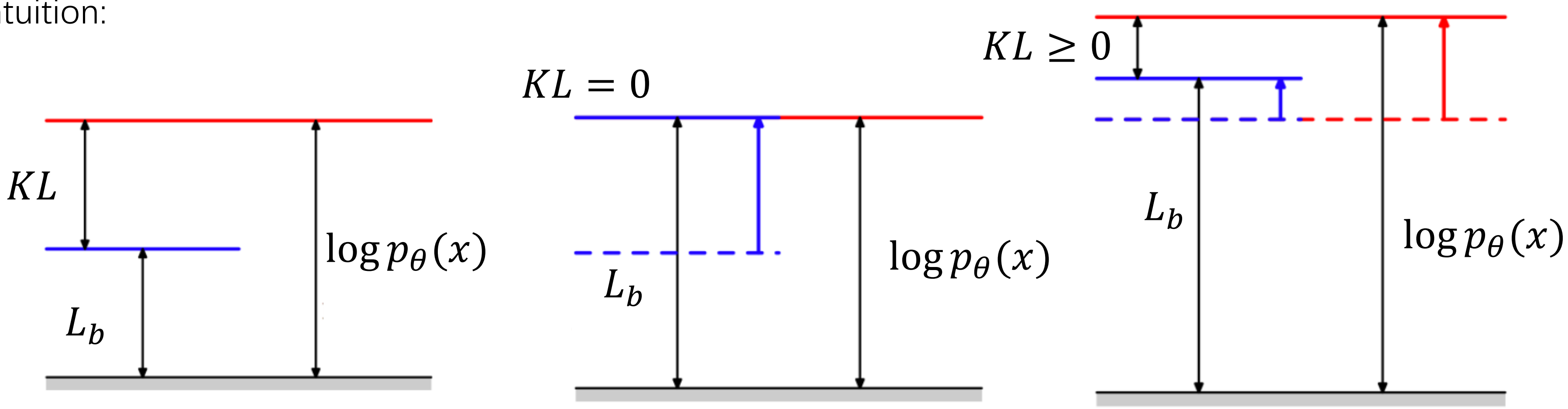
- $L \geq \int_z q_\phi(z|x) \log\left(\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right) dz$ – **Evidence Lower bound** (ELBO) L_b

Mathematics of VAEs

- To maximizes the likelihood $L = \log p_\theta(x)$

→ to find $p_\theta(x|z)$ and $q_\phi(z|x)$ that maximizes the ELBO $L_b = \int_z q_\phi(z|x) \log\left(\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right) dz$

Intuition:



1. Maximize L_b by updating $q_\phi(z|x)$ → L_b approaches $\log p_\theta(x)$
2. Maximize L_b by updating both $q_\phi(z|x)$ and $p_\theta(x|z)$ → maximizes $\log p_\theta(x)$

Mathematics of VAEs

- $$L_b = \int_z q_\phi(z|x)\log\left(\frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\right)dz$$
  $$= \underbrace{\int_z q_\phi(z|x)\log\left(\frac{p(z)}{q_\phi(z|x)}\right)dz}_{-KL(q_\phi(z|x)\,||\,p(z))} + \underbrace{\int_z q_\phi(z|x)\log p_\theta(x|z)dz}_{\mathbb{E}_{q_\phi(z|x)}\log p_\theta(x|z)}$$

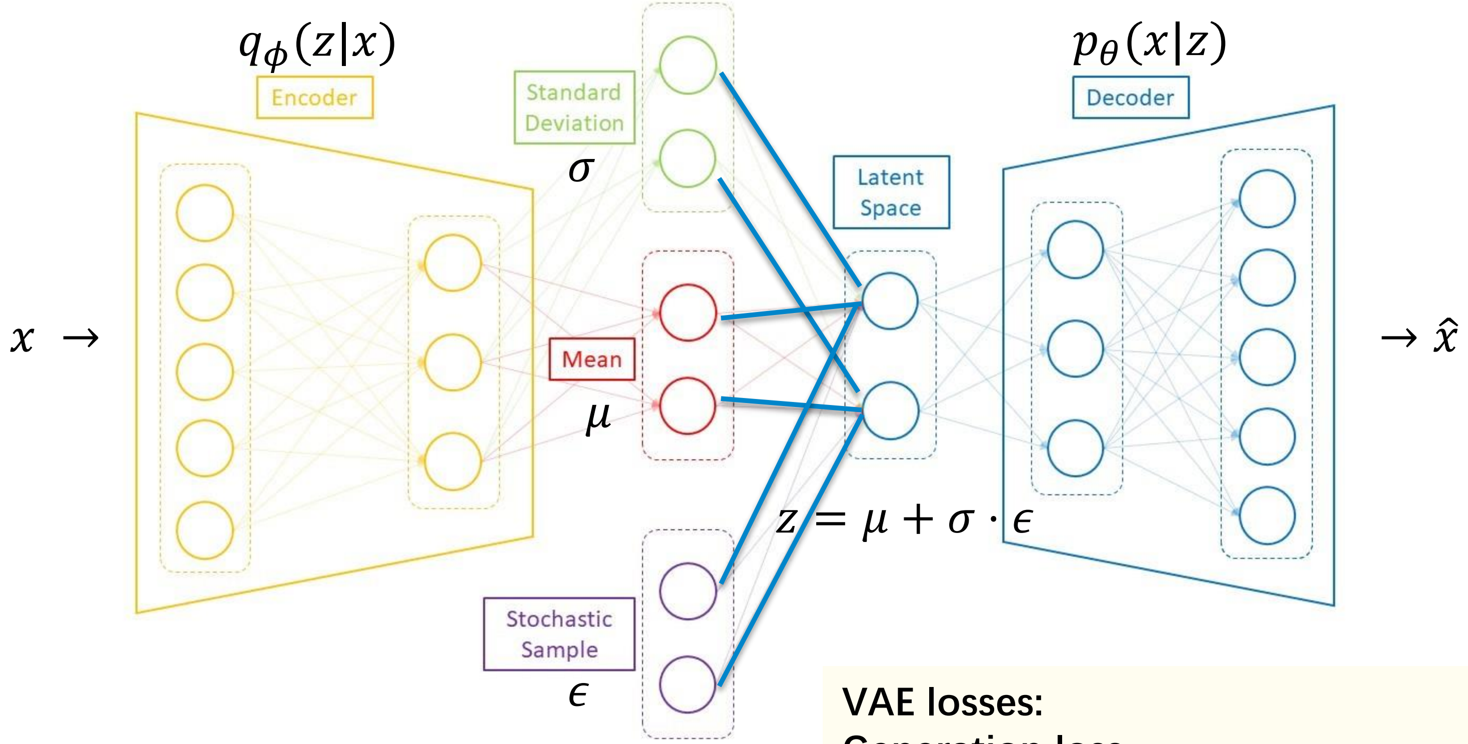
- Maximize L_b → Minimize $KL(q_\phi(z|x)\,||\,p(z))$ and maximize $\mathbb{E}_{q_\phi(z|x)}\log p_\theta(x|z)$

- Recall:
  - $p(z) = N(0, I)$, so this is essentially pushing $q_\phi(z|x) \sim N(0, I)$
  - x → ***Encoder network with parameters*** ϕ → $\mu(x), \sigma(x)$ → z

- As a result: mimimize $KL(q_\phi(z|x)\,||\,p(z))$ → minimize $KL(N(u(x), \sigma(x)^2 * I)||\,N(0, I))$
  - Which is the **VAE latent space loss**: $KL\ (latent\ variable\ ||\ unit\ Gaussian)$

- Maximize $\mathbb{E}_{q_\phi(z|x)}\log p_\theta(x|z)$ → given x_i, encode into z_i, then decode into $\hat{x}_i$ → $\hat{x}_i$ to be close to x_i
  - Which is the **VAE generative loss**: $diff(x_i, \hat{x}_i)$

Architecture of VAEs



VAE losses:

Generation loss

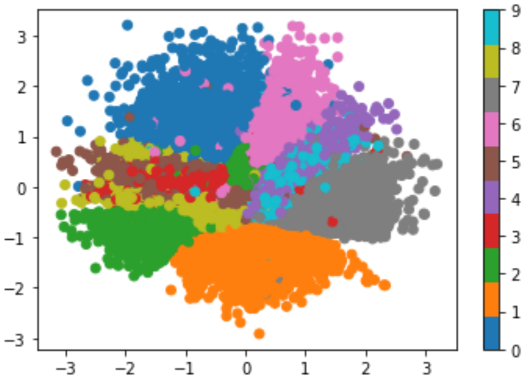
$$diff(x_i, \hat{x}_i)$$

Latent space loss

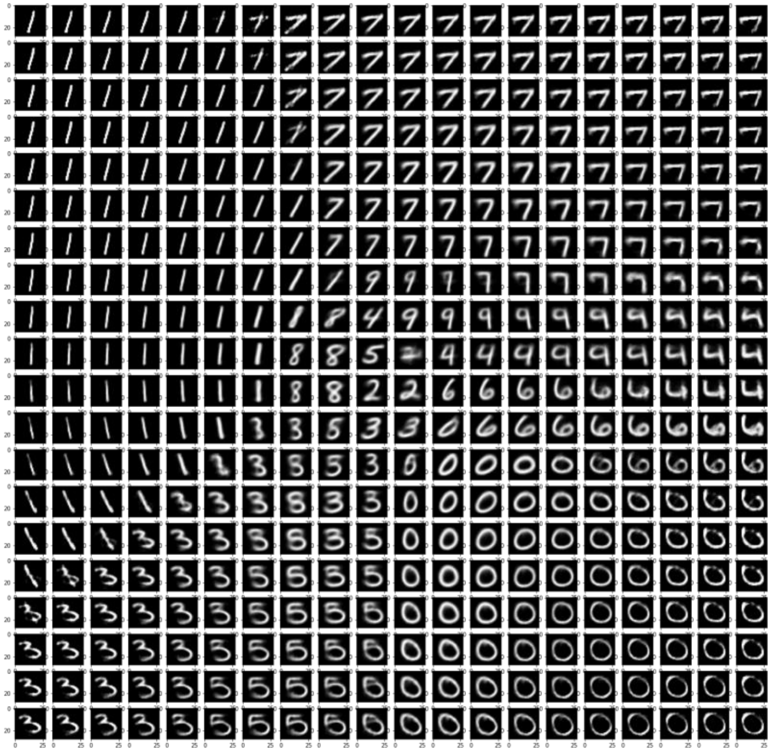
$$KL\ (latent\ variable\ ||\ unit\ Gaussian)$$

Variational Auto-encoder (VAE)

$p(z)$



$Decoder$



Conclusions

- What are generative models? Understand the role of **observed variables** and **latent variables** (Terms are not always rigorous)
- Understand Autoencoder architecture and how to train one
- Discover Variational Autoencoders, which constrain the latent space of Autoencoders (to be Unit Gaussian)
- Understand (the intuitions of) the mathematics of Variational Autoencoders
  - Likelihood
  - ELBO
  - The two loss terms of VAEs