Imperial College London

Auto-encoders and VAEs (Latent variable models)

Oscar Bates

Outline

- Generative models
- Autoencoders
- Latent spaces and latent variables
- Training autoencoders
- Variational-autoencoders

Generative models

Generative models

- Machine learning is grouped by objectives:

  1. Discriminative models try to “discriminate” between data/images

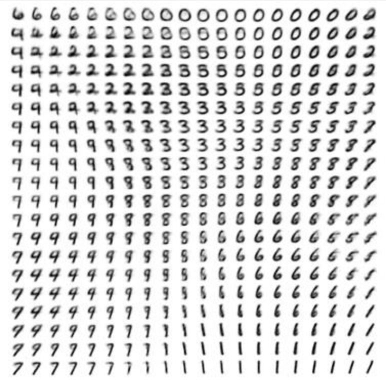
  2. Generative models try to “generate” new examples data/images

  3. Reinforcement models try to learn how to “act” in an environment

- These concepts are older than you might expect; you’ve already done some! (e.g. solving differential equations)

- Optional reading https://openai.com/research/generative-models

Generative models - Examples (deep) generative models



(b) Learned MNIST manifold

Kingma & Welling (2013) *Autoencoding variational Bayes*

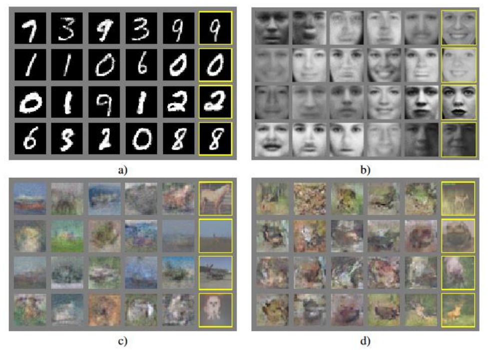
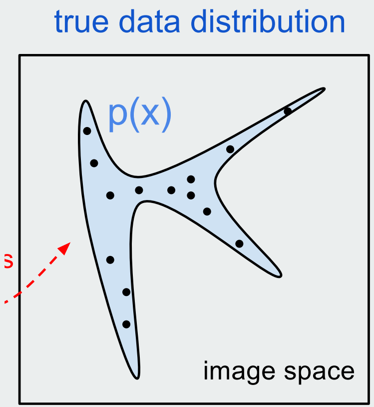


Figure 2: Visualization of samples from the model. Rightmost column shows the nearest training example of the neighboring sample, in order to demonstrate that the model has not memorized the training set. Samples are fair random draws, not cherry-picked. Unlike most other visualizations of deep generative models, these images show actual samples from the model distributions, not conditional means given samples of hidden units. Moreover, these samples are uncorrelated because the sampling process does not depend on Markov chain mixing. a) MNIST b) TFD c) CIFAR-10 (fully connected model) d) CIFAR-10 (convolutional discriminator and "deconvolutional" generator)

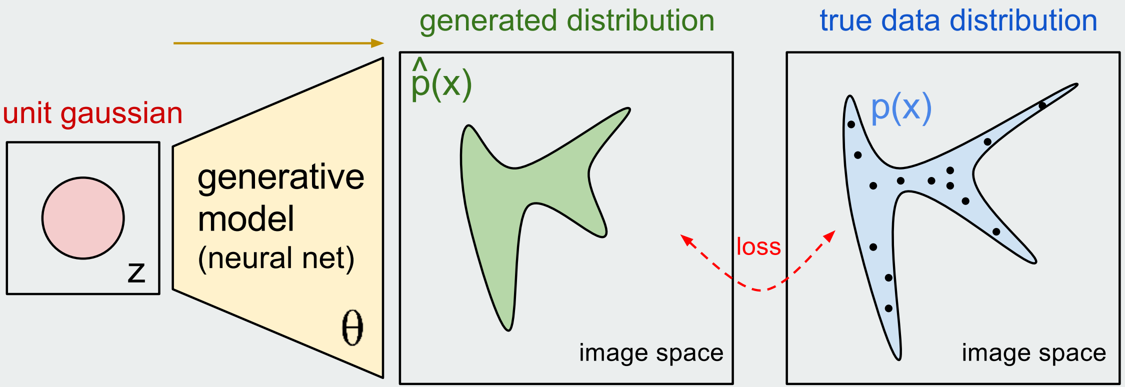
Goodfellow et al (2014) *Generative adversarial networks*

- Autoencoders
  https://doi.org/10.1002/aic.690370209
- Variational autoencoders
  https://arxiv.org/abs/1312.6114
- Generative adversarial networks
  https://arxiv.org/abs/1406.2661
- Autoregressive models, diffusion models

Generative models - Concept from probability



- Imagine you have a data-set of images **x**.
- The probability of getting an image is **p(x)**
- We can model this distribution with a generative model **p*(x)**



Generative models - Autoencoder architecture

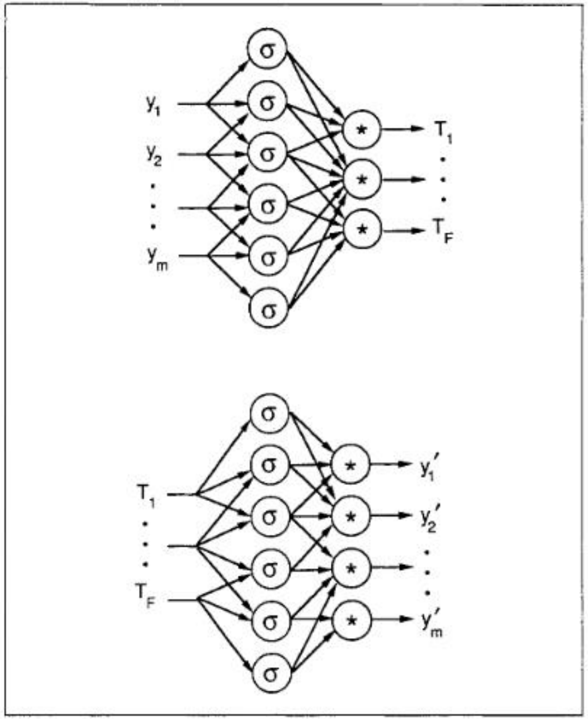


Figure 1. Networks implementing mapping and demapping functions.
Top, network for G; bottom, network for H
σ indicates sigmoidal nodes; * indicates linear or sigmoidal nodes.

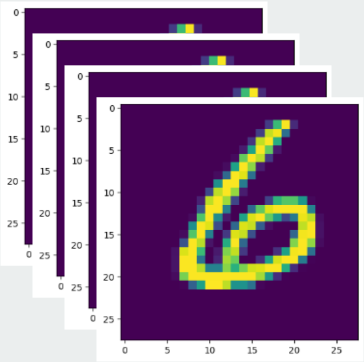
Figure from https://doi.org/10.1002/aic.690370209

- Several independent early papers https://proceedings.neurips.cc/paper/1993/file/9e3cfc48eccf81a0d57663e129aef3cb-Paper.pdf
- T_F are the *latent variables*
- y_m are the *data / images*
- σ is nonlinear node
  - In this figure, a node a weight and an activation function (not true in general)
- * are the *linear or non-linear nodes*

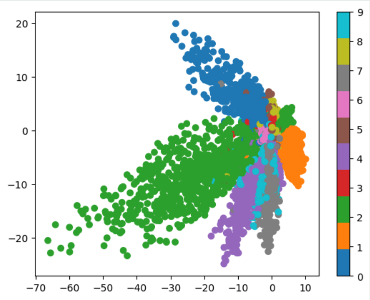
Autoencoders
Concept & naming conventions

Autoencoder architecture

Input images
or data



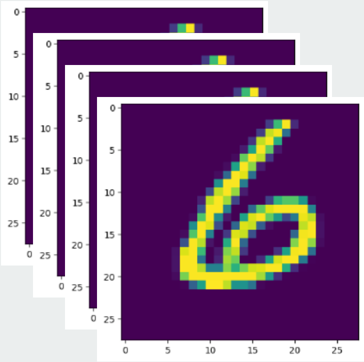
Neural
network



Low
dimensional
space

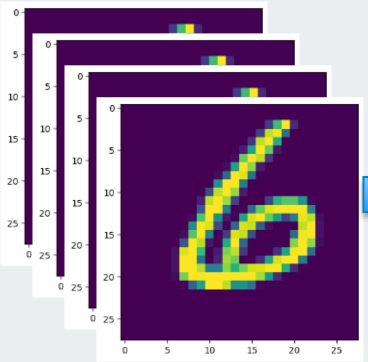
Neural
network

Output images
or data

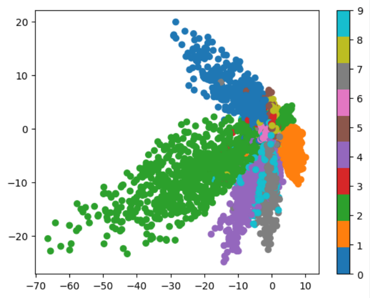


Autoencoder architecture

Input images
or data



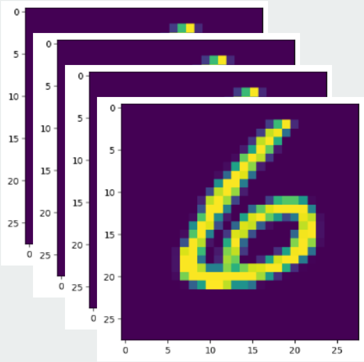
Neural
network



Low
dimensional
space

Neural
network

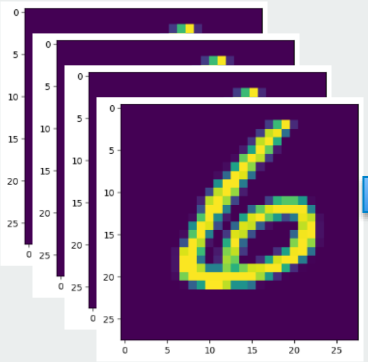
Output images
or data



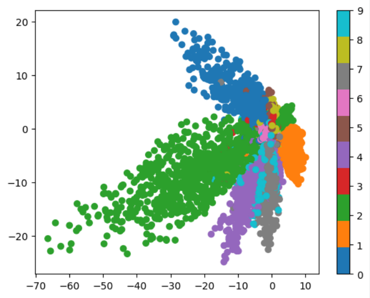
Encoder

Autoencoder architecture

Input images
or data



Neural
network

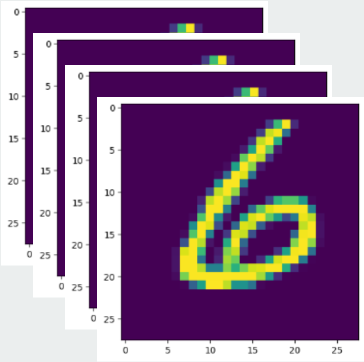


Low
dimensional
space

Latent
space

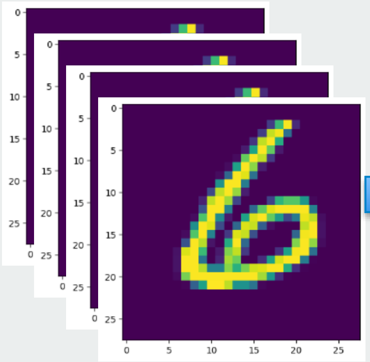
Neural
network

Output images
or data

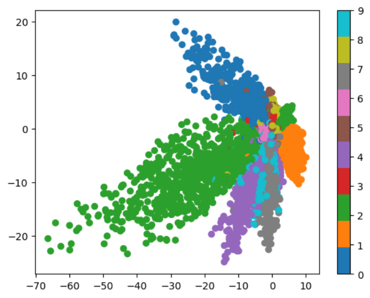


Autoencoder architecture

Input images
or data



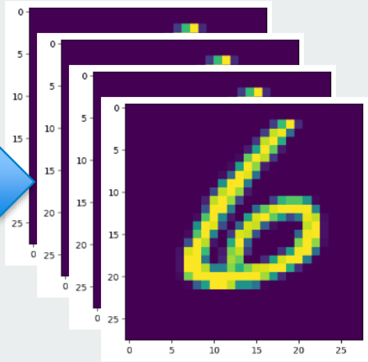
Neural
network



Low
dimensional
space

Neural
network

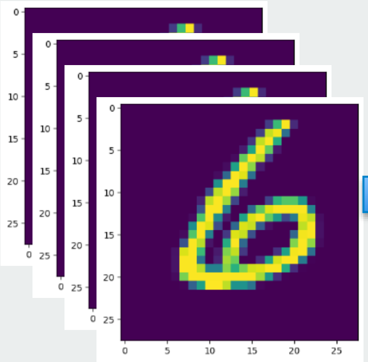
Output images
or data



Decoder

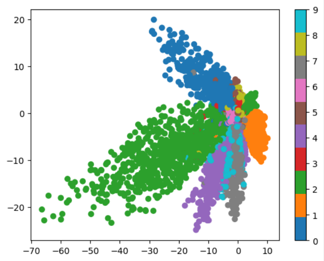
Autoencoder architecture

Input images
or data



Neural
network

Encoder



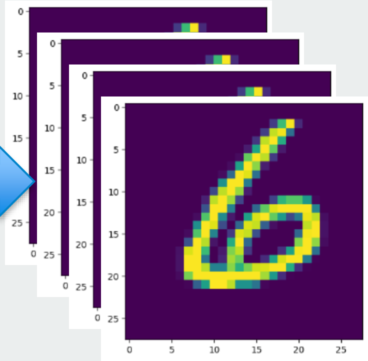
Low
dimensional
space

Latent
space

Neural
network

Decoder

Output images
or data



Vector size
784

Vector size
varies (2-20)

Vector size
784

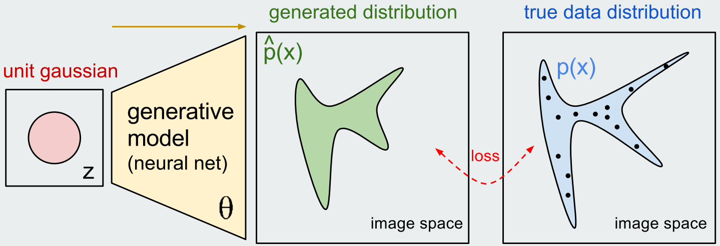
What are latent spaces?

Latent spaces

- Imagine you have a basket full of apples.
- You pick out an apple and say what variety it is

p(x) = probability of the variety of apple

x = variety of apple



Latent spaces

- Imagine you have a basket full of apples.
- You pick out an apple and say what variety it is
- **Granny smith!**

Latent spaces

- Imagine you have a basket full of apples.

- You pick out an apple and say what variety it is

- **Golden delicious!**

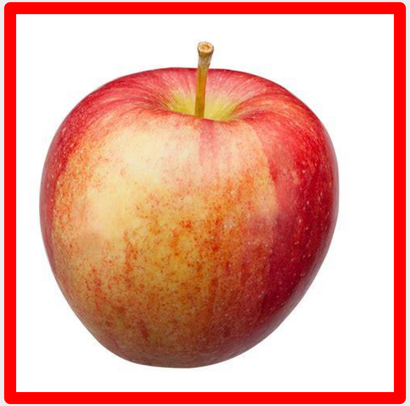
Latent spaces

- Imagine you have a basket full of apples.
- You pick out an apple and say what variety it is
- **Pink Lady!**

Latent spaces

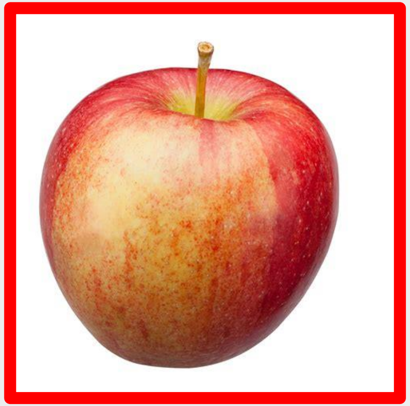
- The basket is the generative model
- The ***type*** *of apple* is the data
- **The Colour is a latent variable**

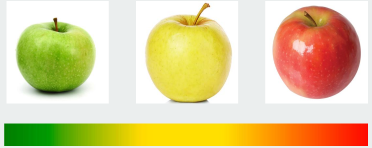
Latent spaces



- But colour **is not the only** latent variable
- **Breaburn!**

Latent spaces





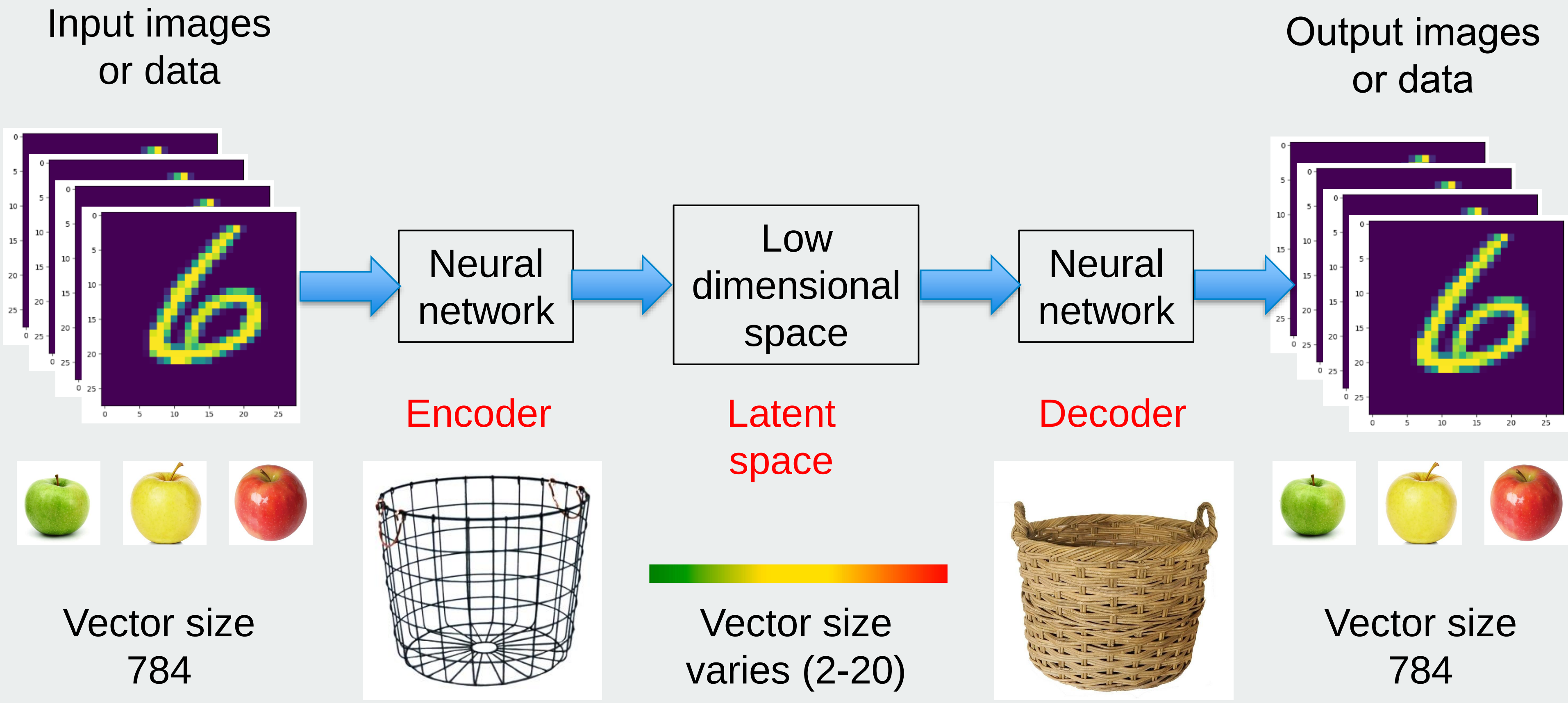
- Other latent variables might be
  - Size
  - Taste
  - Genetic code?

- Latent spaces can be any length, size, shape.

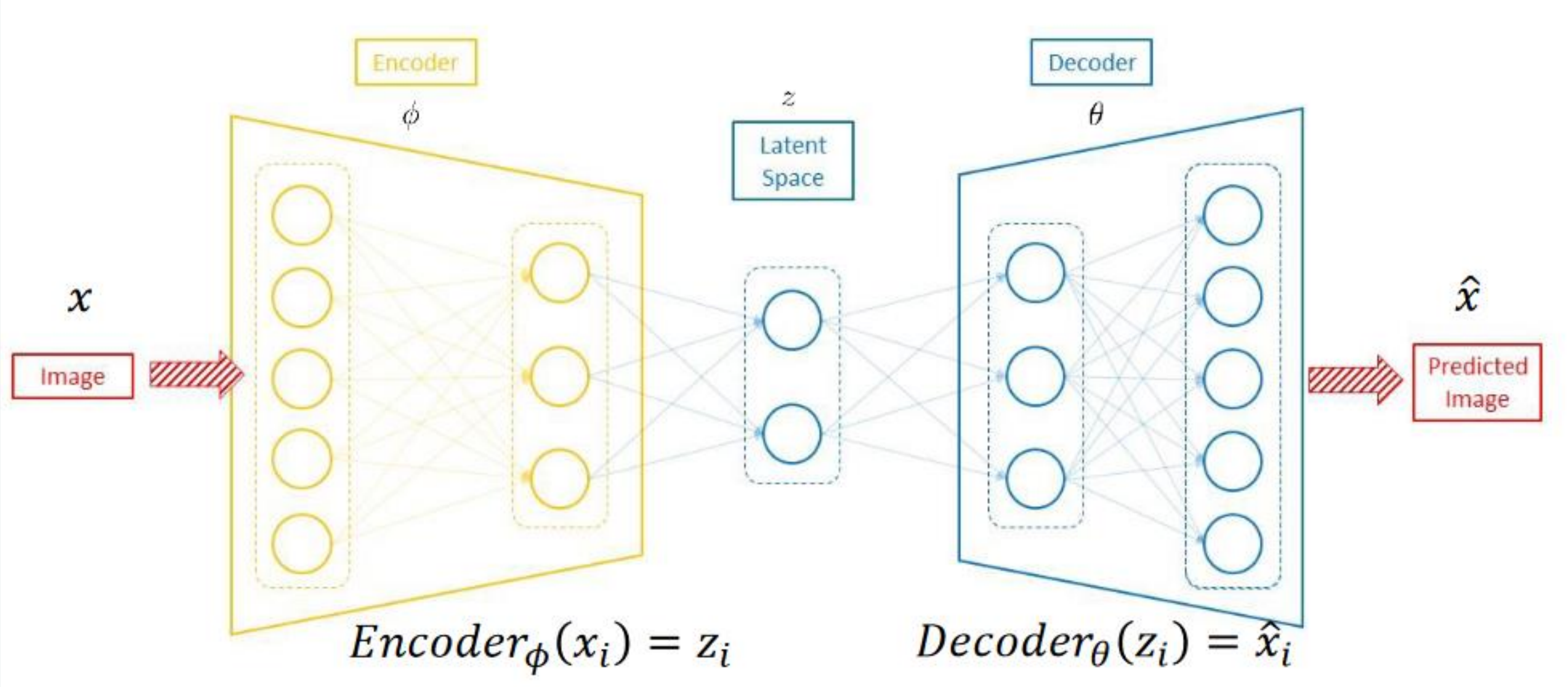
- The point is that they describe a ***feature*** of the data

Training

Autoencoder architecture

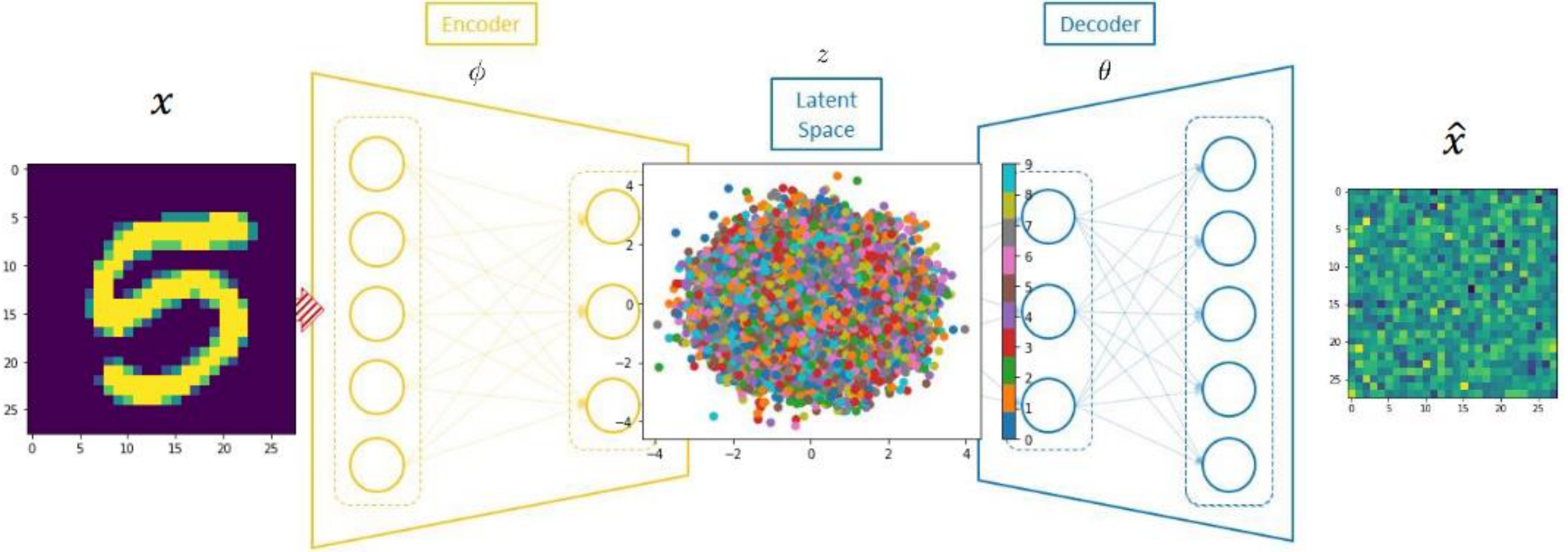


Training Autoencoders



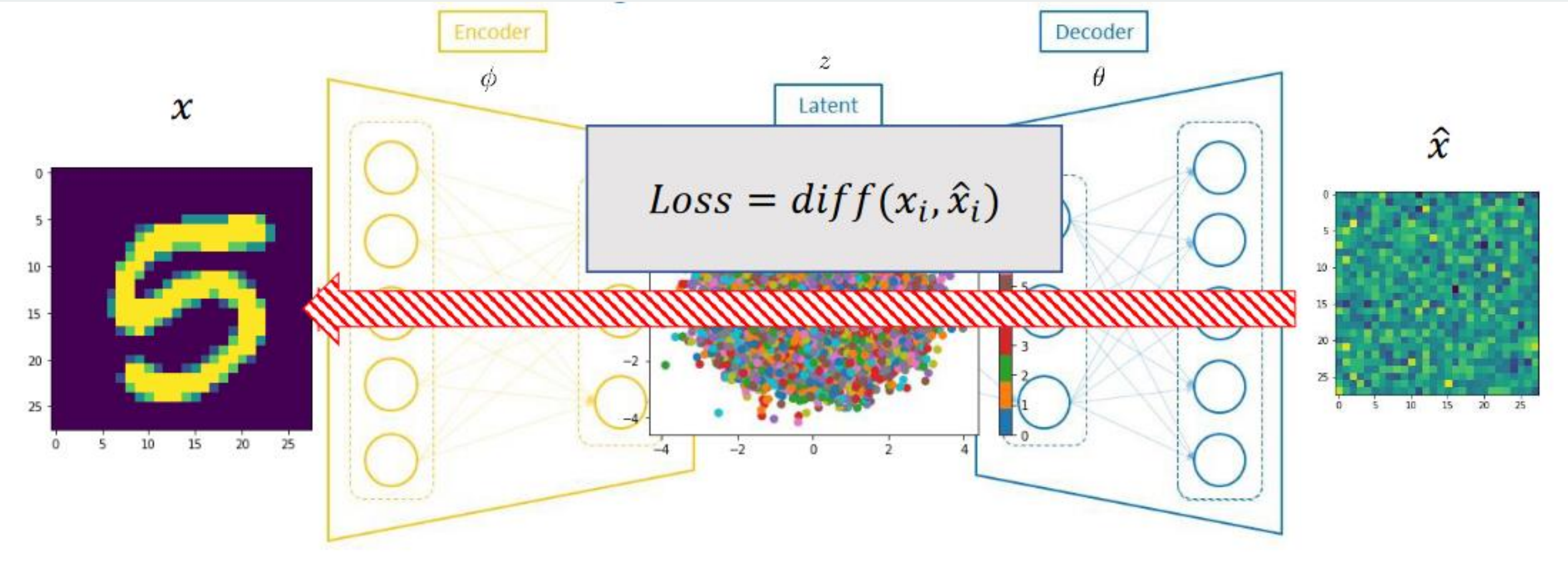
- We're trying to *compress* the images -> we have to balance **predictive accuracy** against **latent vector length**

Training Autoencoders



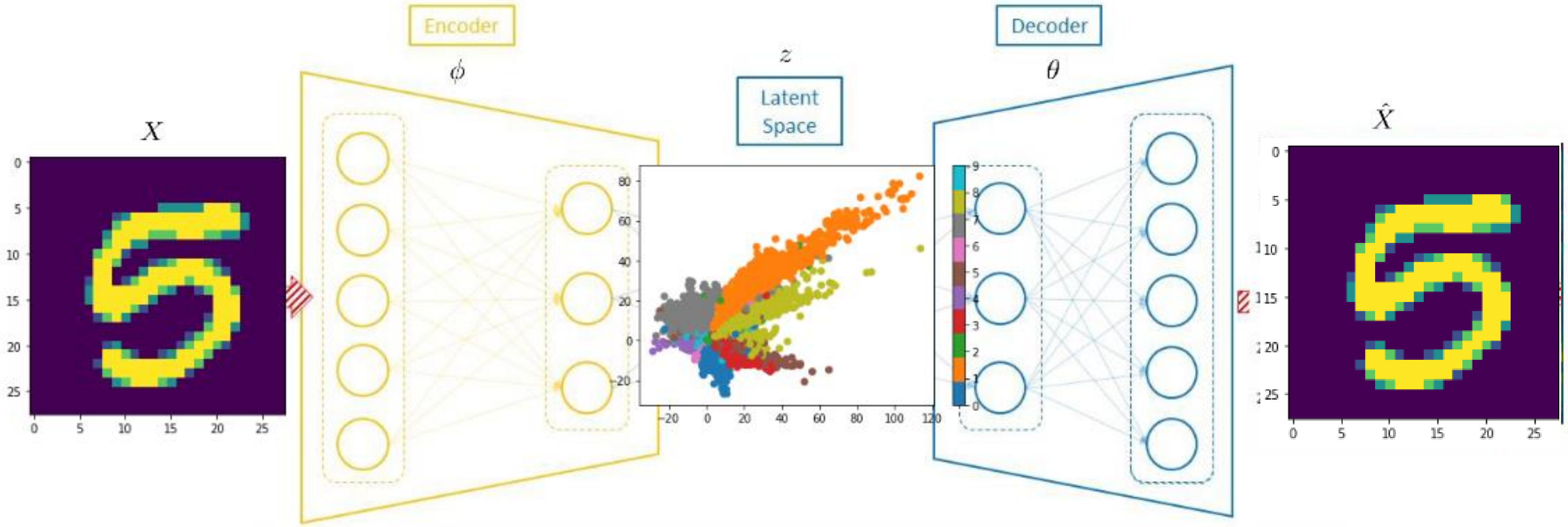
- Before training, an input image **x** has a random latent vector, $\mathbf{z_i}$, within the latent space **z**
- The decoder generates a predicted image $\hat{\mathbf{x}}$, which is random noise

Training Autoencoders



- Training is unsupervised:
  - the loss function compares the input and output directly
  - there are no labels on the data
- We are training the weights of the Encoder Φ(x) and Decoder Θ(z)

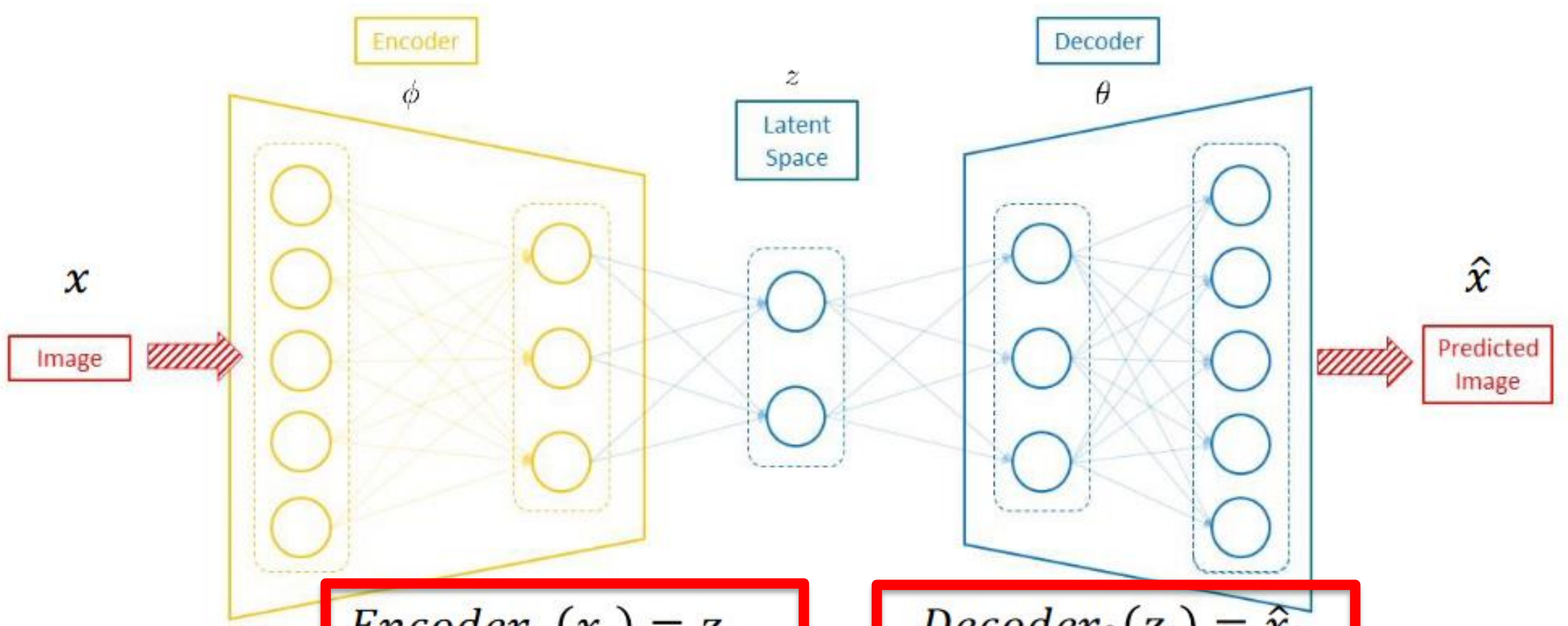
Training Autoencoders



- We are training the weights of the Encoder Φ(x) and Decoder Θ(z)
- The latent space has structured data

Training Autoencoders

- NOTICE THE EQUATIONS! We are training the weights of the Encoder Φ(x) and Decoder Θ(z)



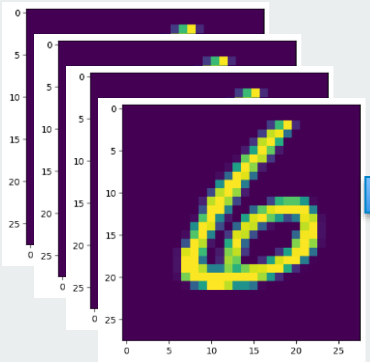
$$Encoder_{\phi}(x_i) = z_i$$

$$Decoder_{\theta}(z_i) = \hat{x}_i$$

Variational autoencoders

Variational Autoencoder - Concept

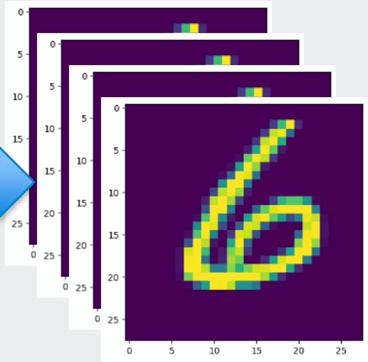
Input images
or data



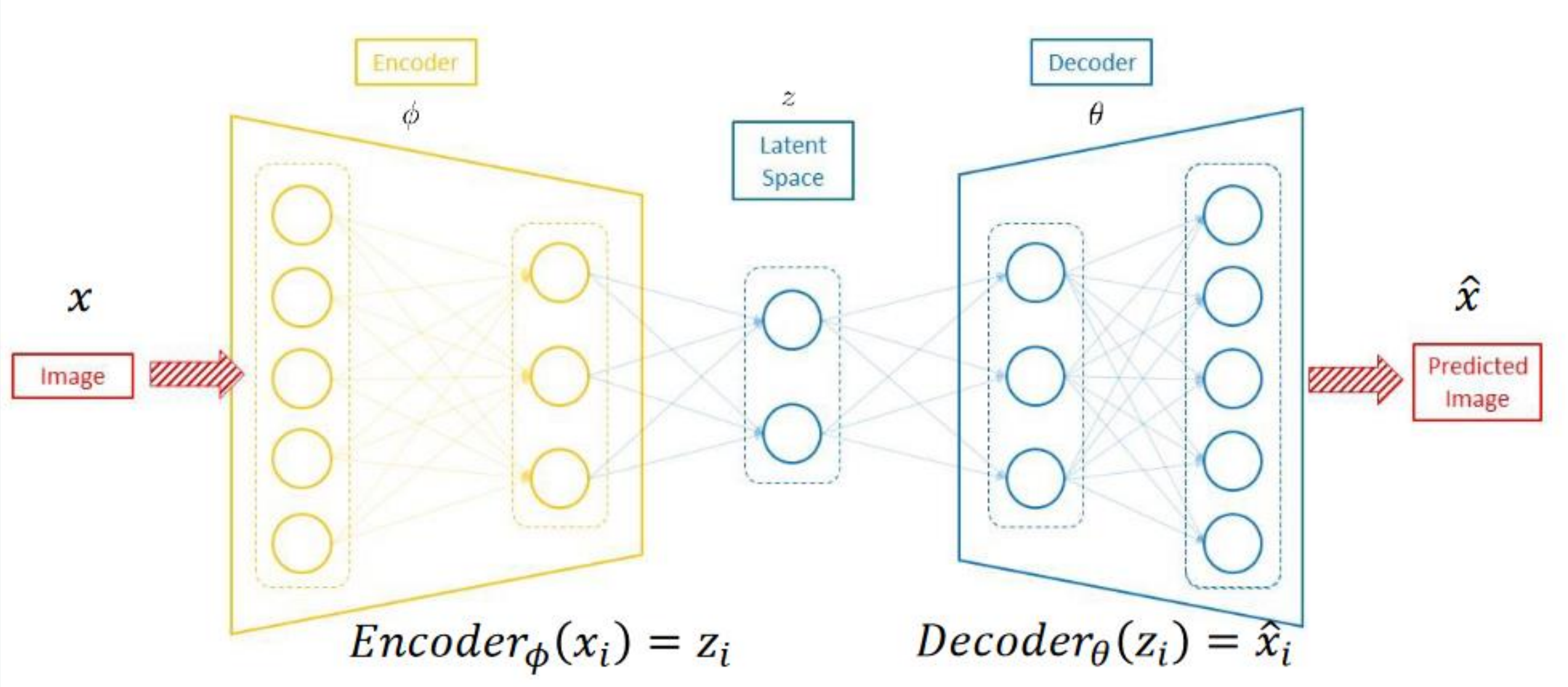
Encoder → Latent space → Decoder

has a Gaussian
distribution

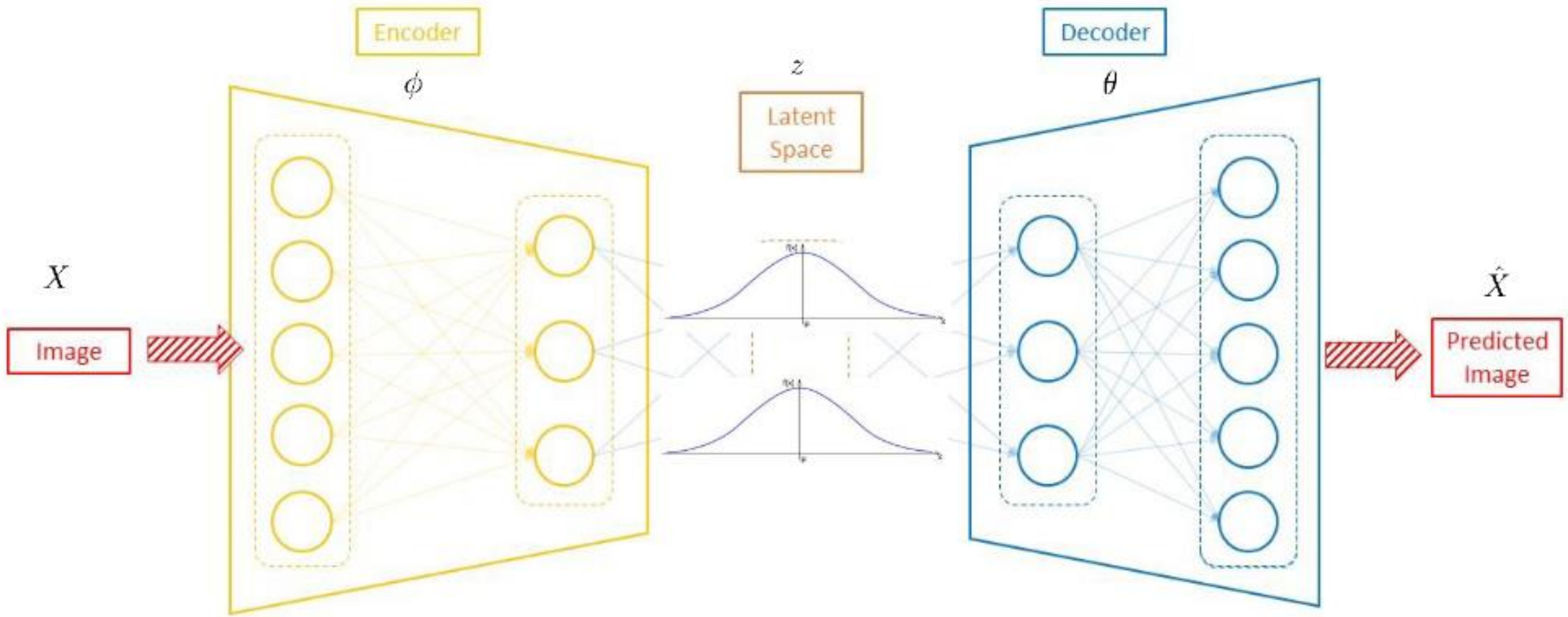
Output images
or data



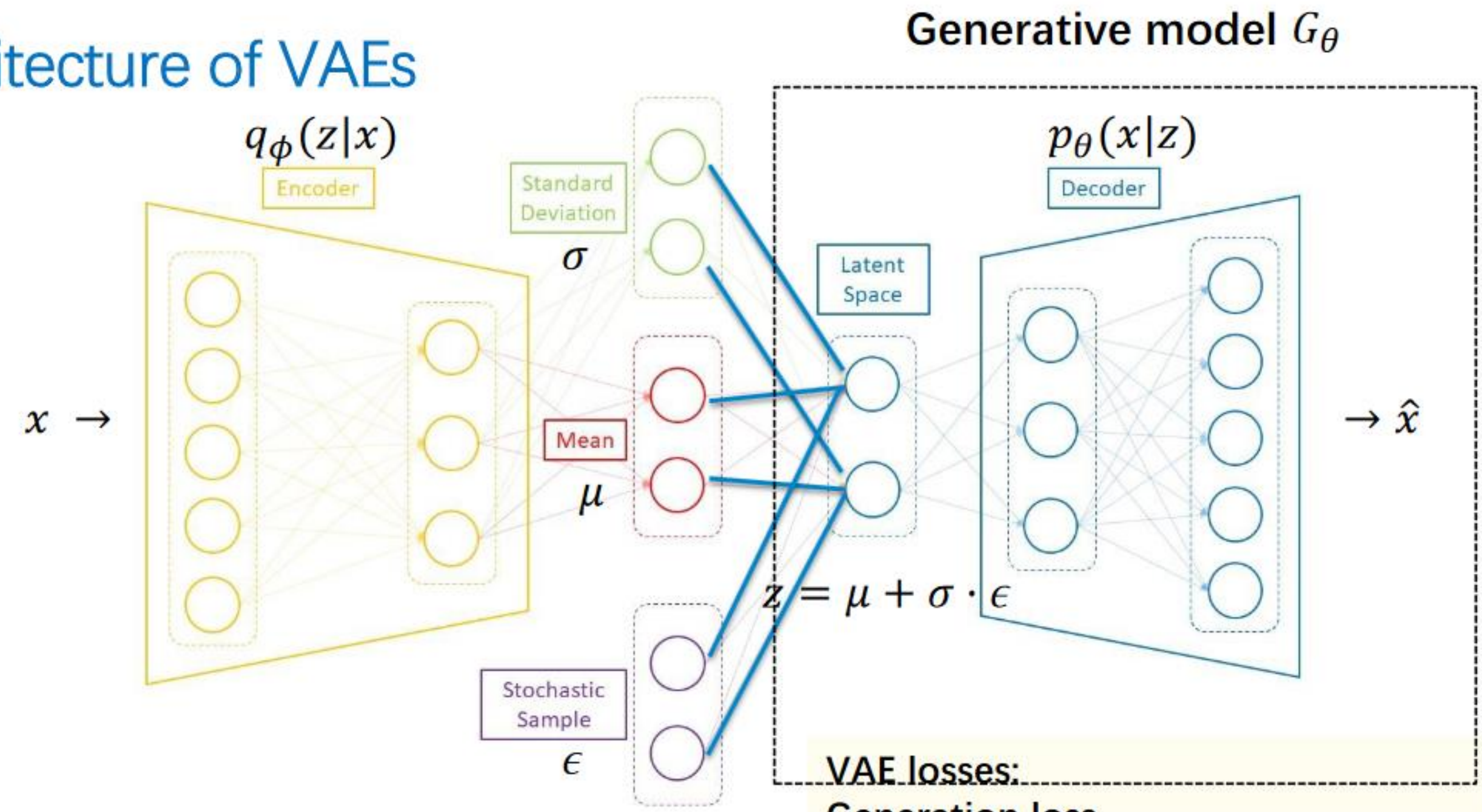
Autoencoder concept



Variational Autoencoder concept



Architecture of VAEs



VAE losses:

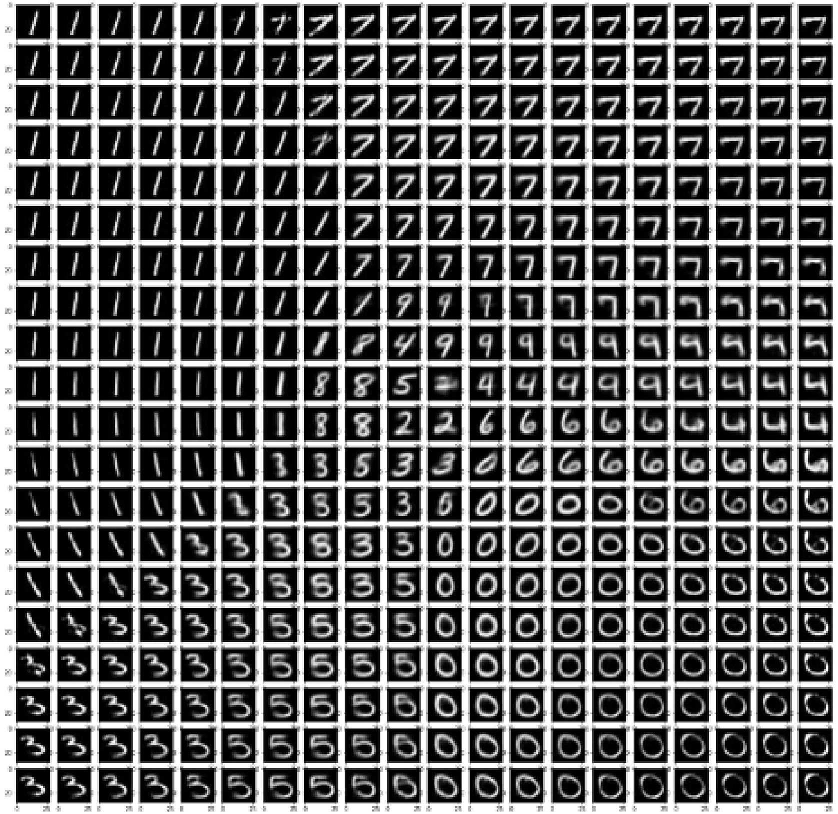
Generation loss

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

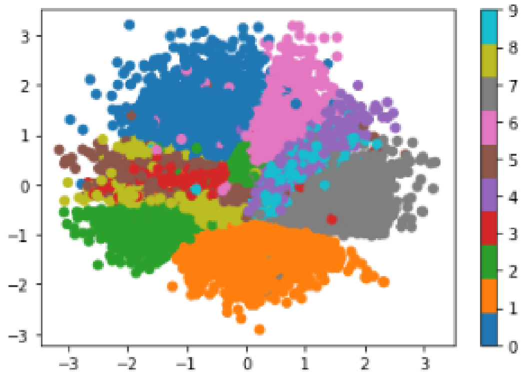
Latent space loss

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

Variational Autoencoder - Why bother?

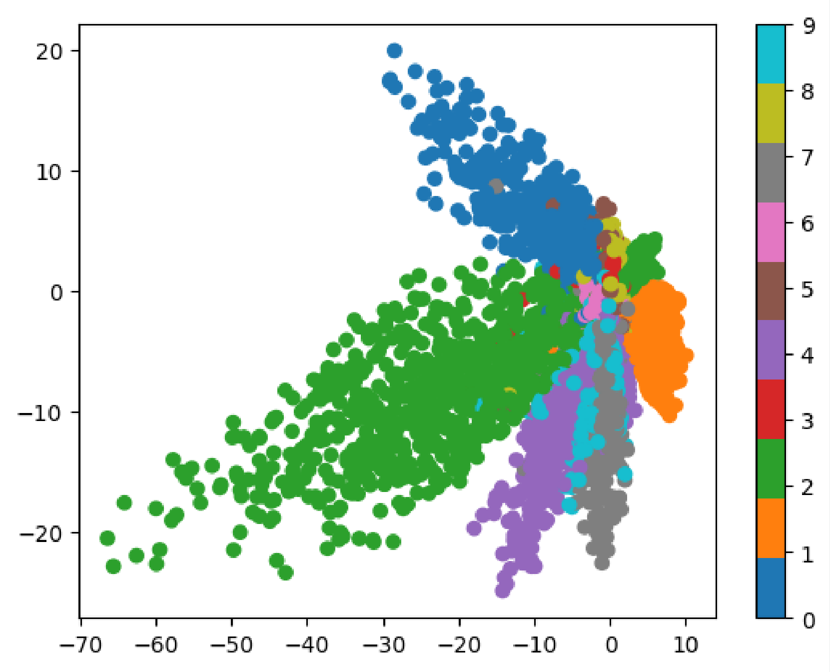


- Intuitively, by doing this, now our valid latent space representations are following a distribution.

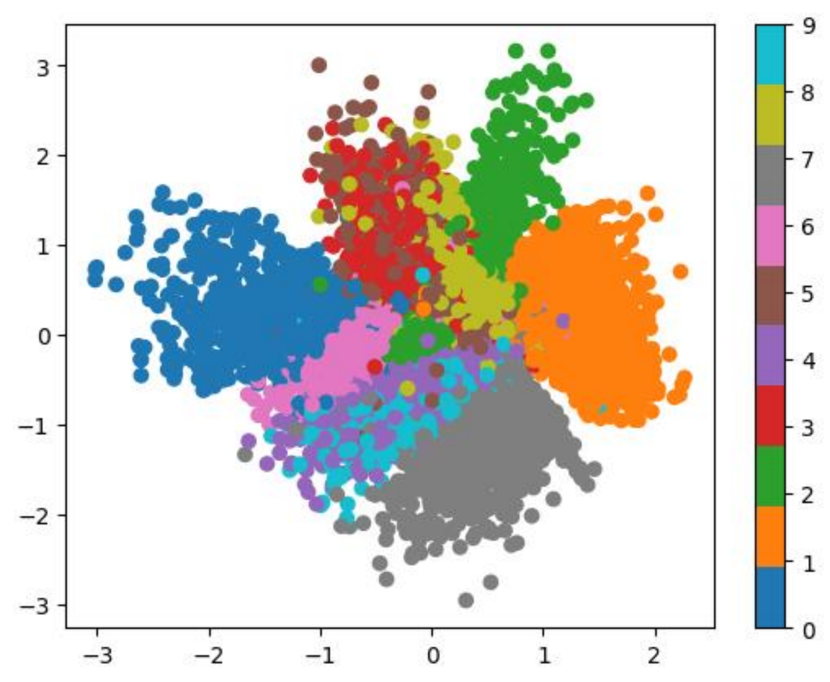


- As long as we sample z_i from this distribution, we no longer generate invalid samples.

Variational Autoencoder – why bother?

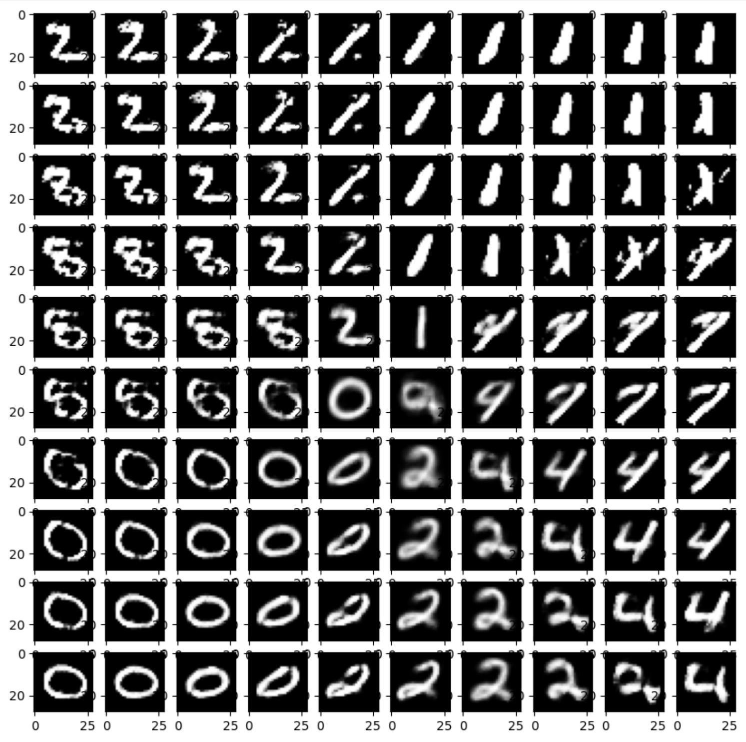


Autoencoder latent space unconstrained

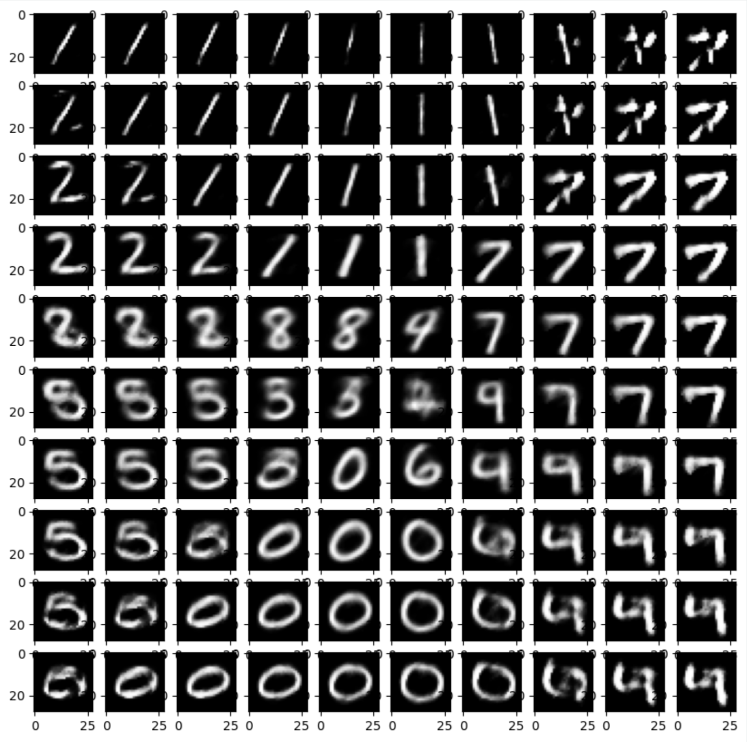


Variational autoencoder latent space has to be gaussian

Variational Autoencoder – why bother?



Some samples are very strange



Better representation?

Variational autoencoder latent space

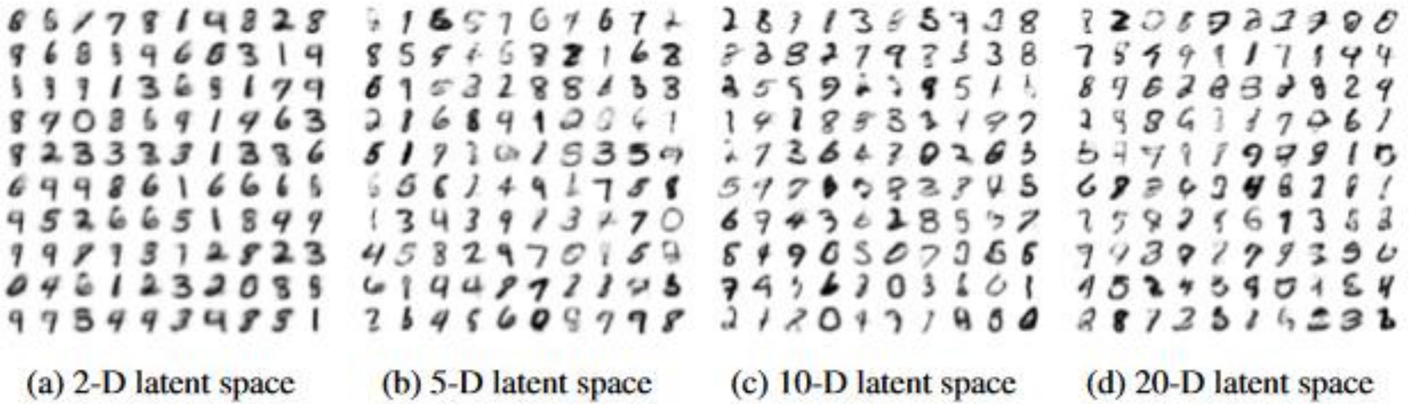
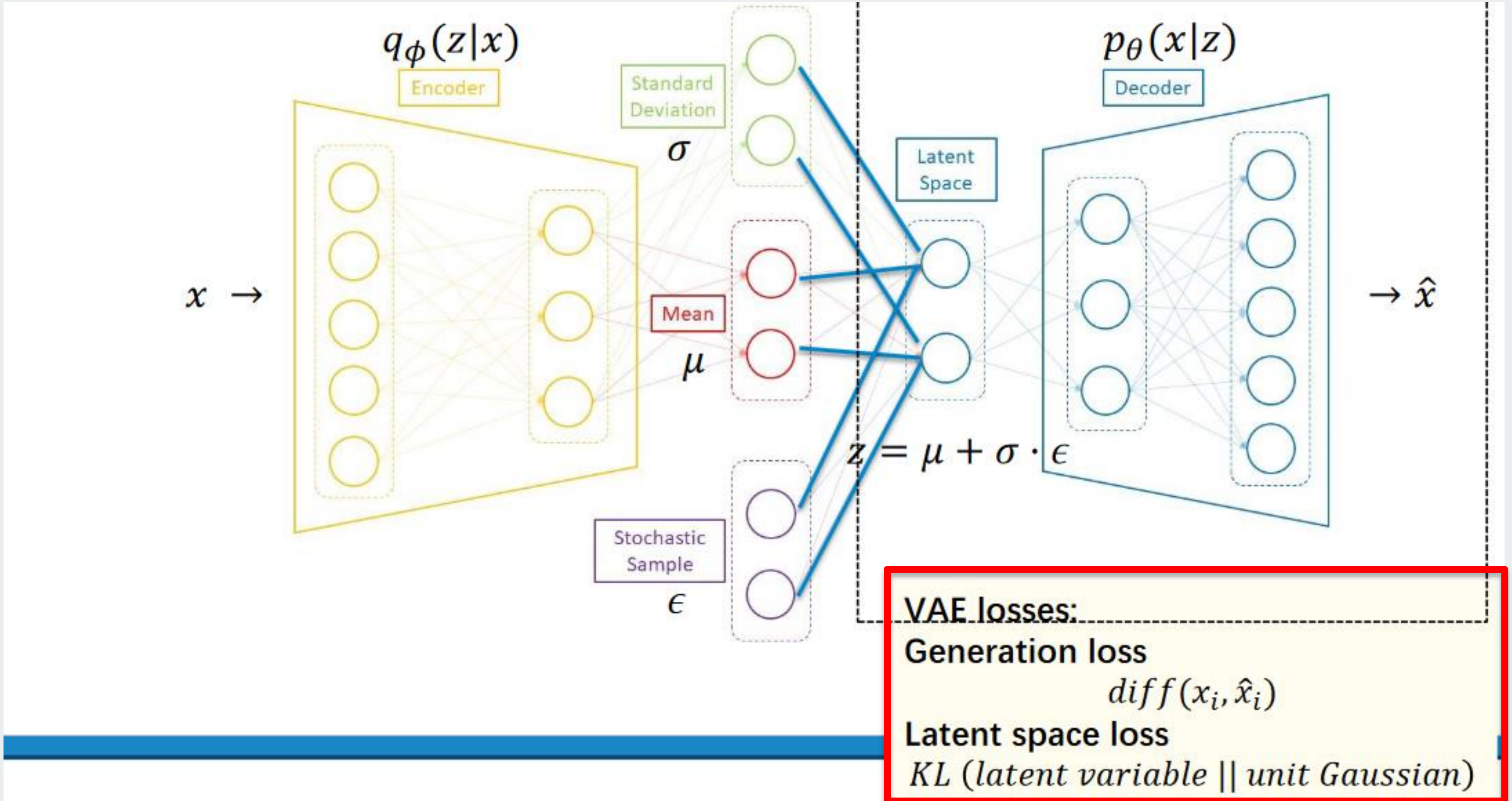


Figure 5: Random samples from learned generative models of MNIST for different dimensionalities of latent space.

Kingma & Welling used Variational Autoencoders to generate handwritten digits

Variational Autoencoder Training



KL-divergence and Evidence Lower Bound (ELBO)

$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)p(m)}{p(\mathcal{D})},$$

Bayes equation

$$\mathbb{KL}[q(m)||p(m|\mathcal{D}))] = \int_m q(m) \ln \left[\frac{q(m)}{p(m|\mathcal{D})} \right] \mathrm{d}m.$$

KL-div

$$q^*(m) = \min_{q(m)} \mathbb{KL}[q(m)||p(m|\mathcal{D}))].$$

The optimisation problem

$$\begin{aligned} q^*(m) &= \max_{q(m)} \mathbb{KL}[q(m)||p(m,\mathcal{D})] \\ &= \max_{q(m)} \int_m q(m) \ln \left[\frac{q(m)}{p(m,\mathcal{D})} \right] \mathrm{d}m. \end{aligned}$$

The Evidence Lower Bound (ELBO)

KL-divergence and Evidence Lower Bound (ELBO)

$$\mathbb{KL}[q(m)\|p(m|\mathcal{D})] = \int_m q(m)\,\ln\left[\frac{q(m)}{p(m|\mathcal{D})}\right] \mathrm{d}m.$$

KL-div

$$p(m|\mathcal{D}) = \frac{p(\mathcal{D}|m)p(m)}{p(\mathcal{D})},$$

Bayes equation

$$q^*(m) = \min_{q(m)} \mathbb{KL}[q(m)\|p(m|\mathcal{D})].$$

The optimisation problem

$$\mathbb{KL}[q(m)\|p(m|\mathcal{D})] = \ln\ p(\mathcal{D}) + \int_m q(m)\ln\left[\frac{q(m)}{p(m,\mathcal{D})}\right] \mathrm{d}m.$$

$$\mathbb{KL}[q(m)\|p(m|\mathcal{D})] \geqslant 0.$$

$$\ln\ p(\mathcal{D}) \geqslant -\mathbb{KL}[q(m)\|\ p(m,\mathcal{D})]$$

Derivation of ELBO

$$q^*(m) = \max_{q(m)} \mathbb{KL}[q(m)\|p(m,\mathcal{D})]$$

$$= \max_{q(m)} \int_m q(m)\ \ln\left[\frac{q(m)}{p(m,\mathcal{D})}\right] \mathrm{d}m.$$

The Evidence Lower Bound (ELBO)

Conclusions

- What are generative models?
  - A statistical model which tries to generate new examples of data
  - A statistical model which tries to capture prescient information about the data
  - A model which uses latent variables to describe the data

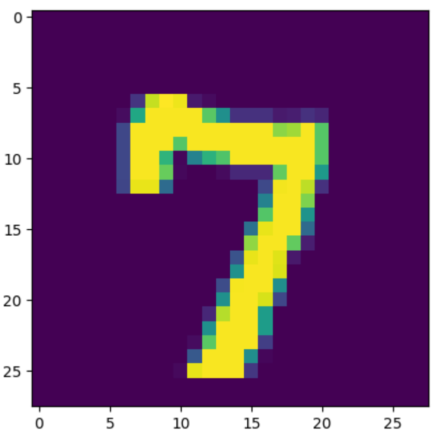
- Understand the Autoencoder architecture and how to train one.

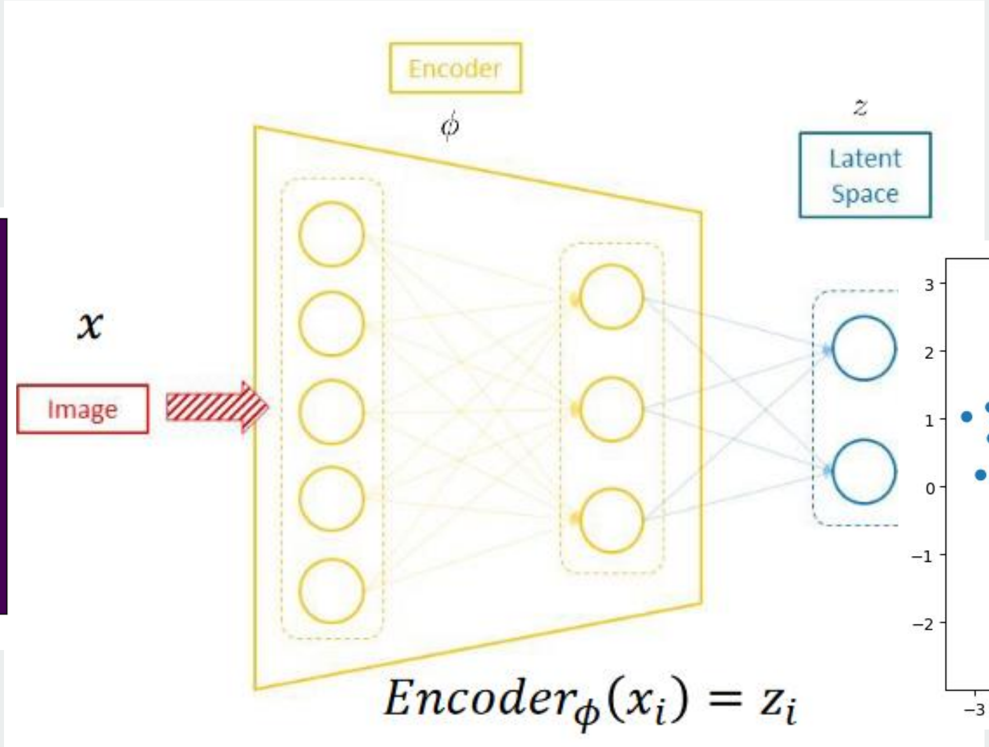
- Understand Variational Autoencoders and how they are different to Autoencoders.

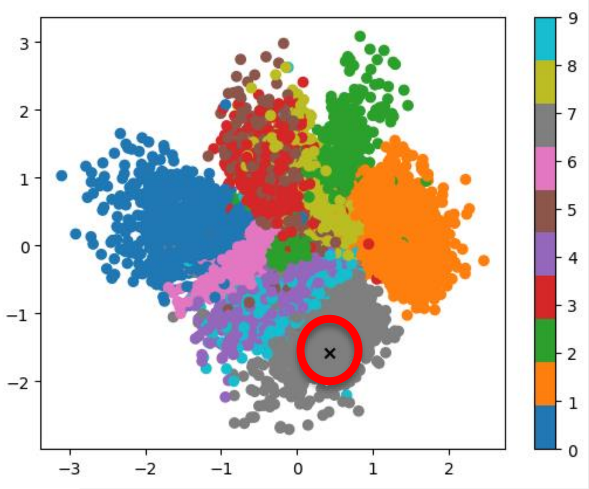
- Introduction to Likelihood, ELBO and the two loss terms in VAE’s

Further comments

Conditioning the latent space







Conditioning the VAE – *The latent vector is stochastic*