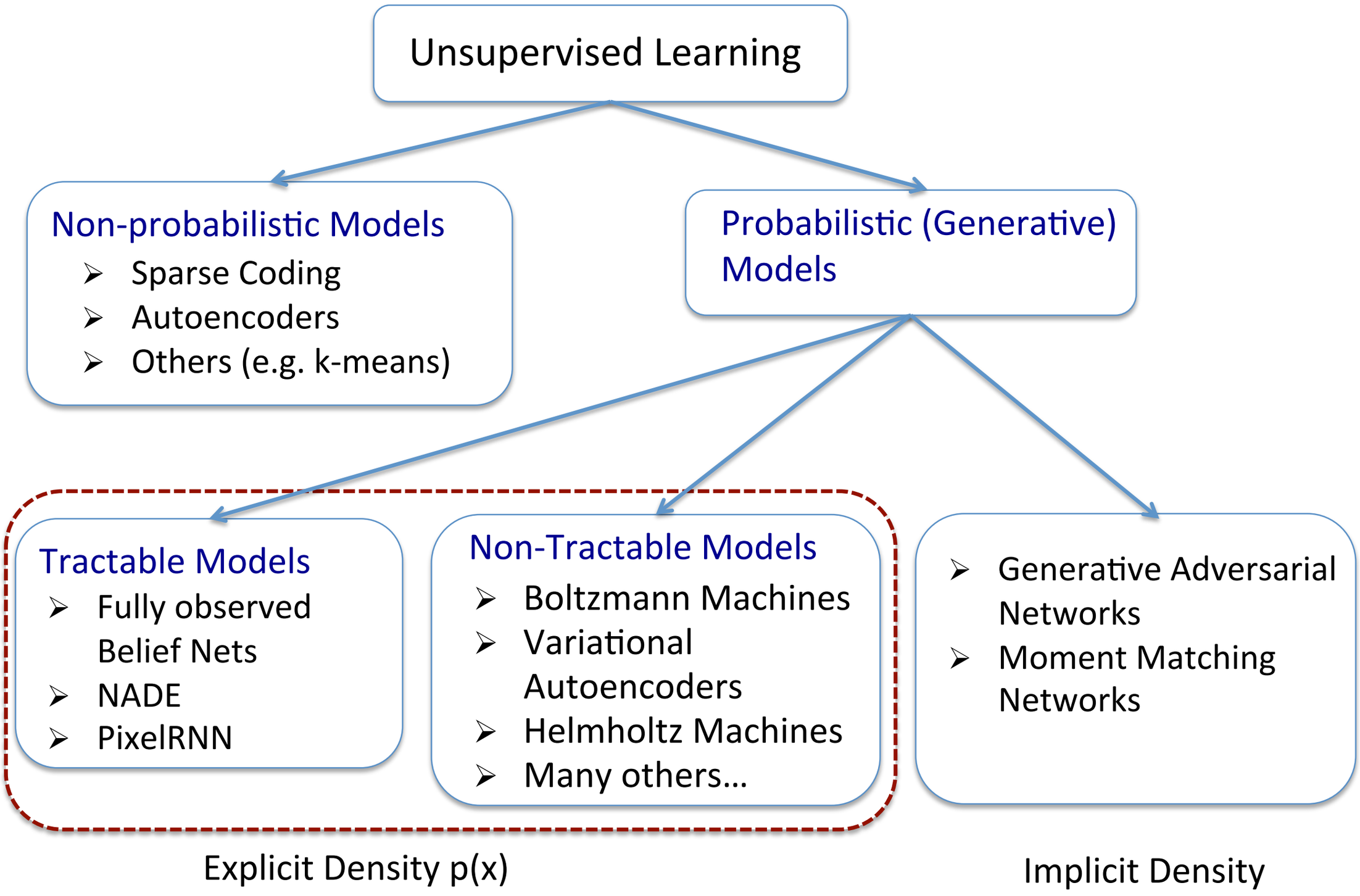
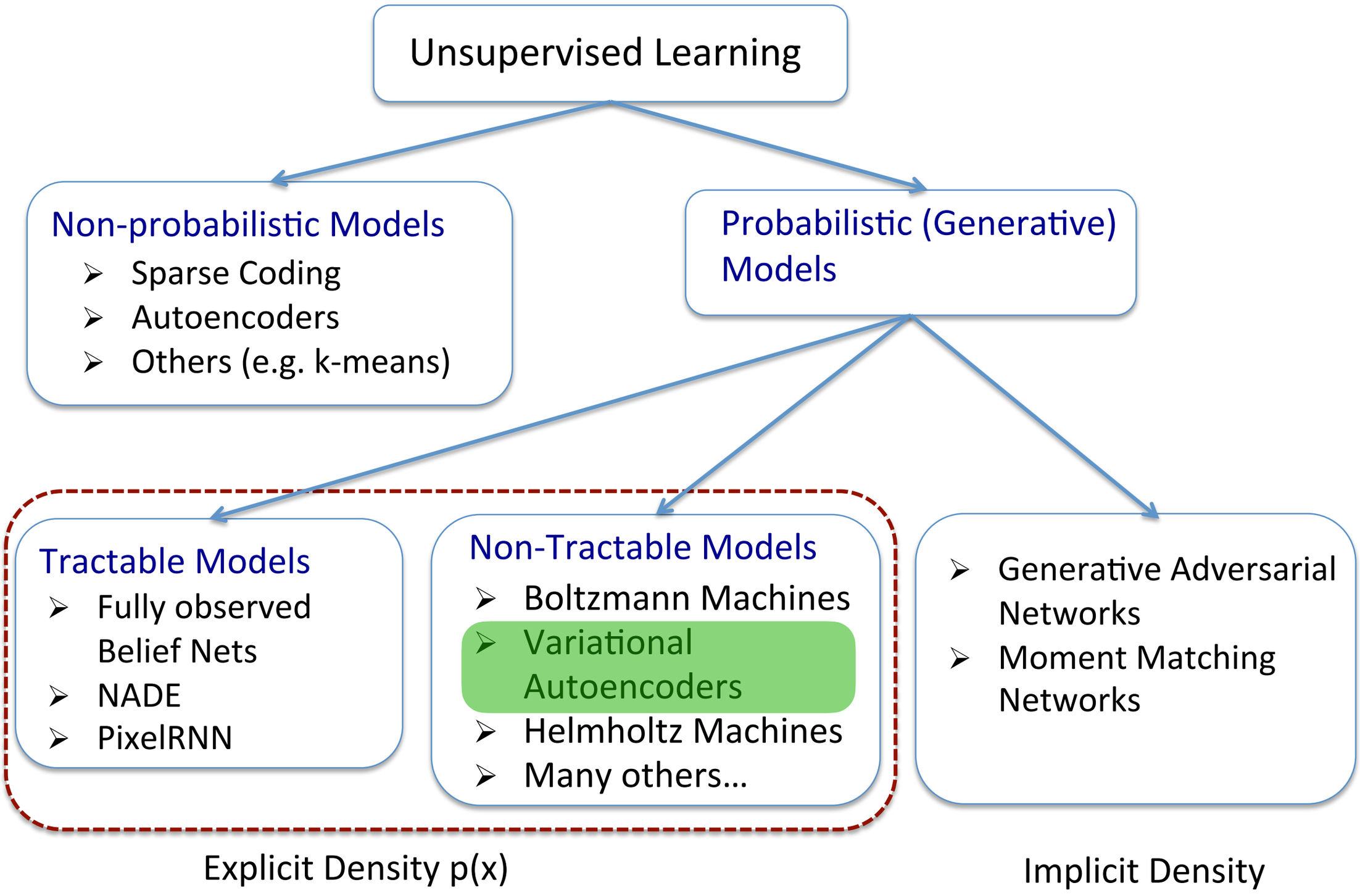
Deep Learning with Neural Networks

Deep Probabilistic Modelling: VAEs

Alejandro Lancho Serrano, alancho@ing.uc3m.es

Material original por: Pablo Martínez Olmos


Unsupervised Learning
Non-probabilistic Models
Sparse Coding
Autoencoders
Others (e.g. k-means)
Probabilistic (Generative) Models
Tractable Models
Fully observed Belief Nets
NADE
PixelRNN
Non-Tractable Models
Boltzmann Machines
Variational Autoencoders
Helmholtz Machines
Many others…
Generative Adversarial Networks
Moment Matching Networks
Explicit Density p(x)
Implicit Density


Unsupervised Learning
Non-probabilistic Models
Sparse Coding
Autoencoders
Others (e.g. k-means)
Probabilistic (Generative) Models
Tractable Models
Fully observed Belief Nets
NADE
PixelRNN
Non-Tractable Models
Boltzmann Machines
Variational Autoencoders
Helmholtz Machines
Many others…
Generative Adversarial Networks
Moment Matching Networks
Explicit Density p(x)
Implicit Density

uc3m

Quick recap of autoencoders

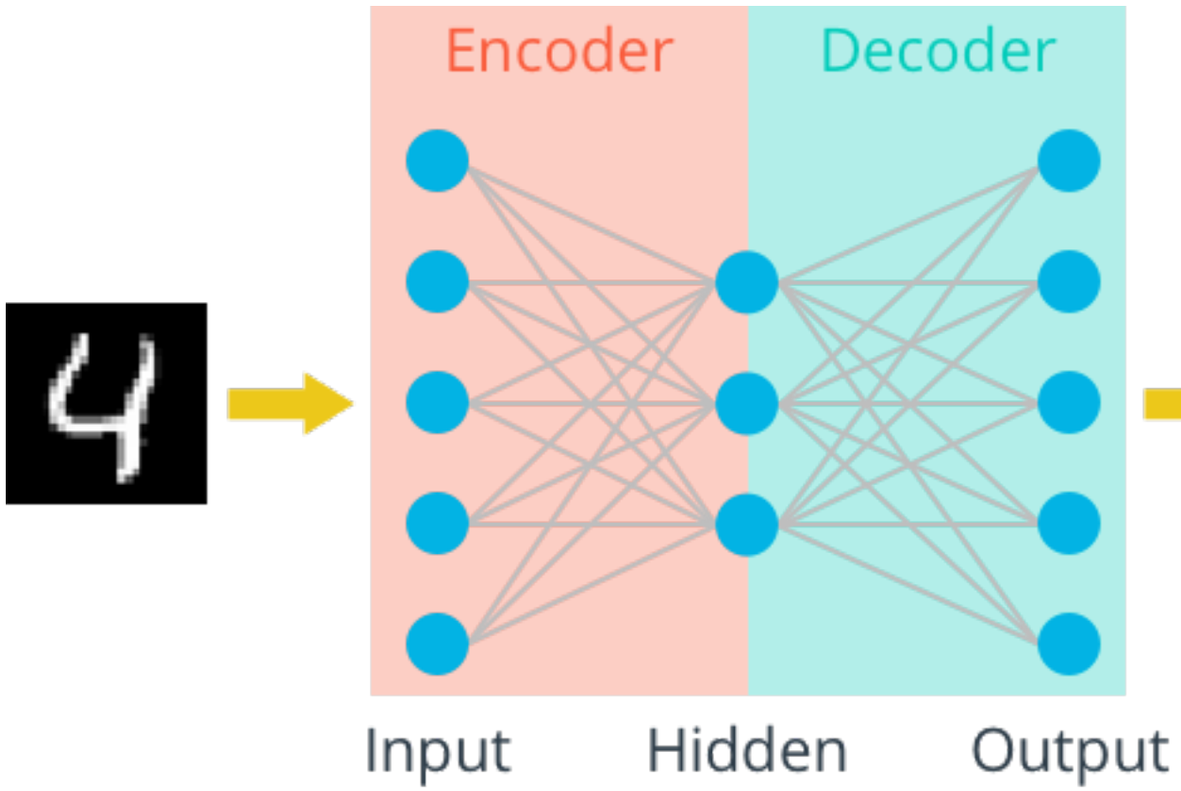
- The “encoder” learns a mapping from the data to a lower-dimensional latent space

$$\mathbf{z} = f(\mathbf{x}) = E_\eta(\mathbf{x}_n)$$

- The “decoder” learns a mapping from the latent space back to reconstruct the observation

$$\mathbf{z} = g(\mathbf{z}) = g(f(\mathbf{x})) = D_\theta(E_\eta(\mathbf{x}_n))$$

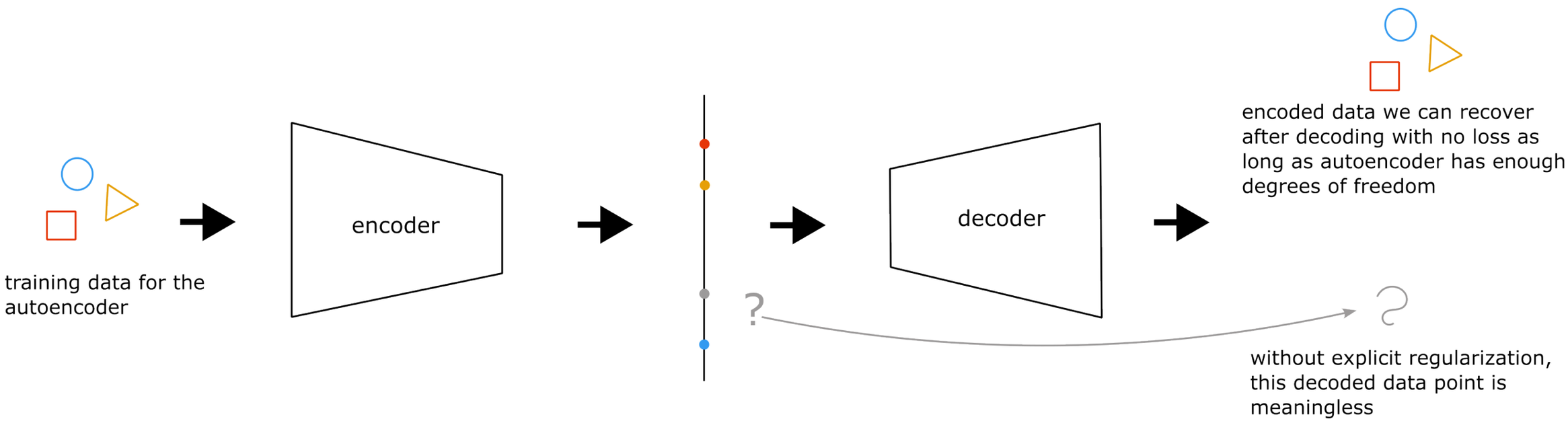
- The loss function doesn’t use any labels



$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

Quick recap of autoencoders

- The latent space is sometimes hard to regularize
- Struggle handling uncertainty
- One alternative are variational autoencoders (VAEs)
  - Regularize the latent space with variational inference principles
  - The latent space becomes rich enough for generation purposes

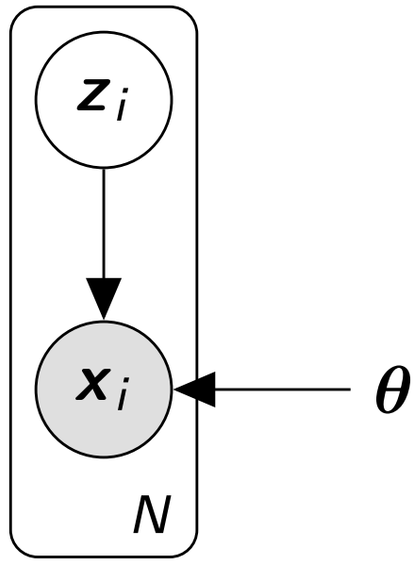


Variational Autoencoder (Kigma & Welling 2014)

Consider a set of i.i.d observations $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$, $i = 1, \ldots, N$. Let $\mathcal{D} = \{\boldsymbol{x}^{(i)}\}_{i=1}^N$.

Generative Model using a Latent Space

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}$$



- $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_k)$ $(k \leq d)$ (low-dimensional embedding)
- $\boldsymbol{x}|\boldsymbol{z} \sim \mathcal{N}(\mu_{\boldsymbol{\theta}}(\boldsymbol{z}), \text{diag}(\sigma_{\boldsymbol{\theta}}(\boldsymbol{z})))$
- $\mu_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $\log(\sigma_{\boldsymbol{\theta}}(\boldsymbol{z}))$ are the outputs of a NN $\mathbb{R}^k \to \mathbb{R}^d$ with parameter vector $\boldsymbol{\theta}$ (Decoding network).

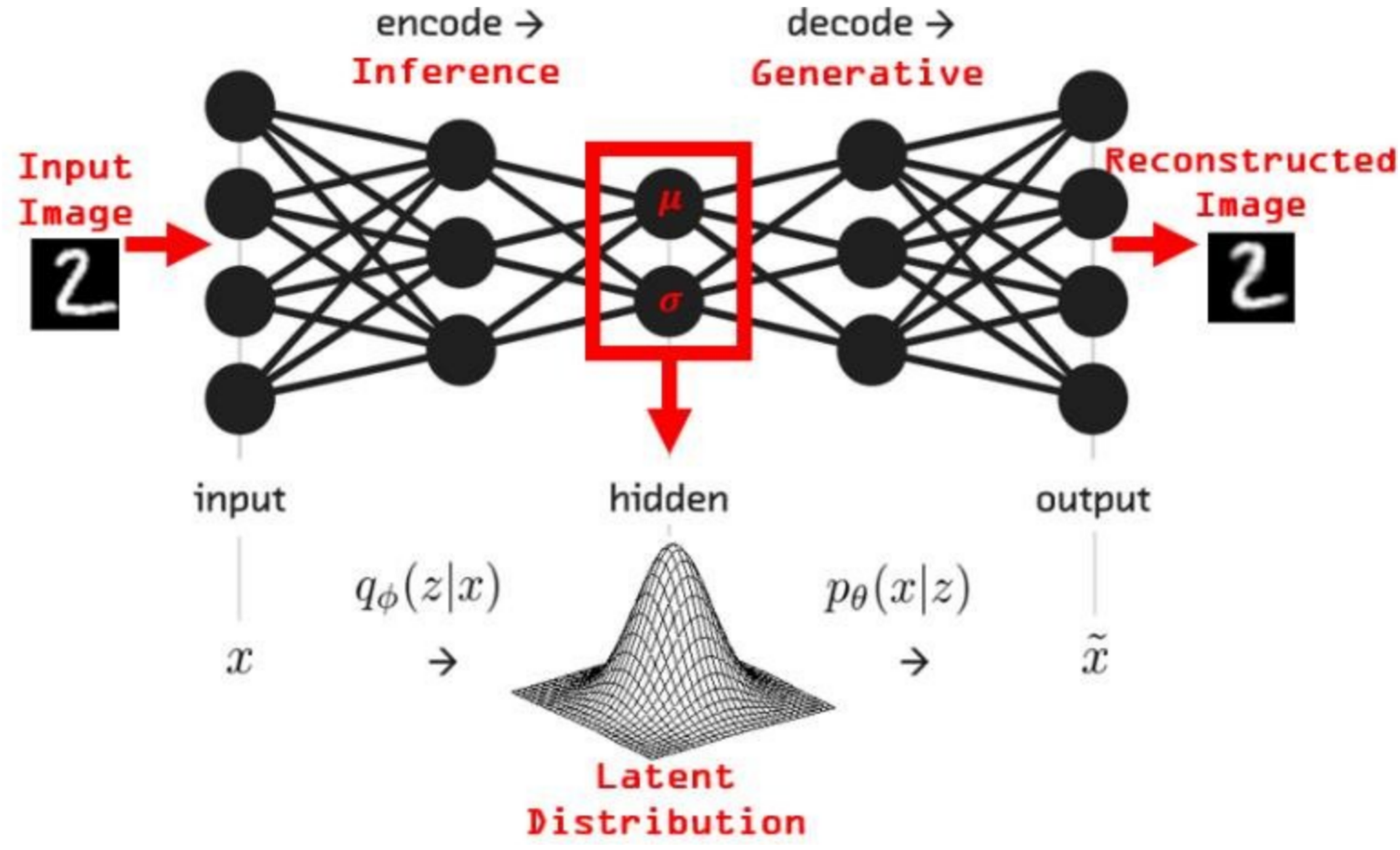
Variational Autoencoder. Approximate posterior

The Inference Network or Encoding Network

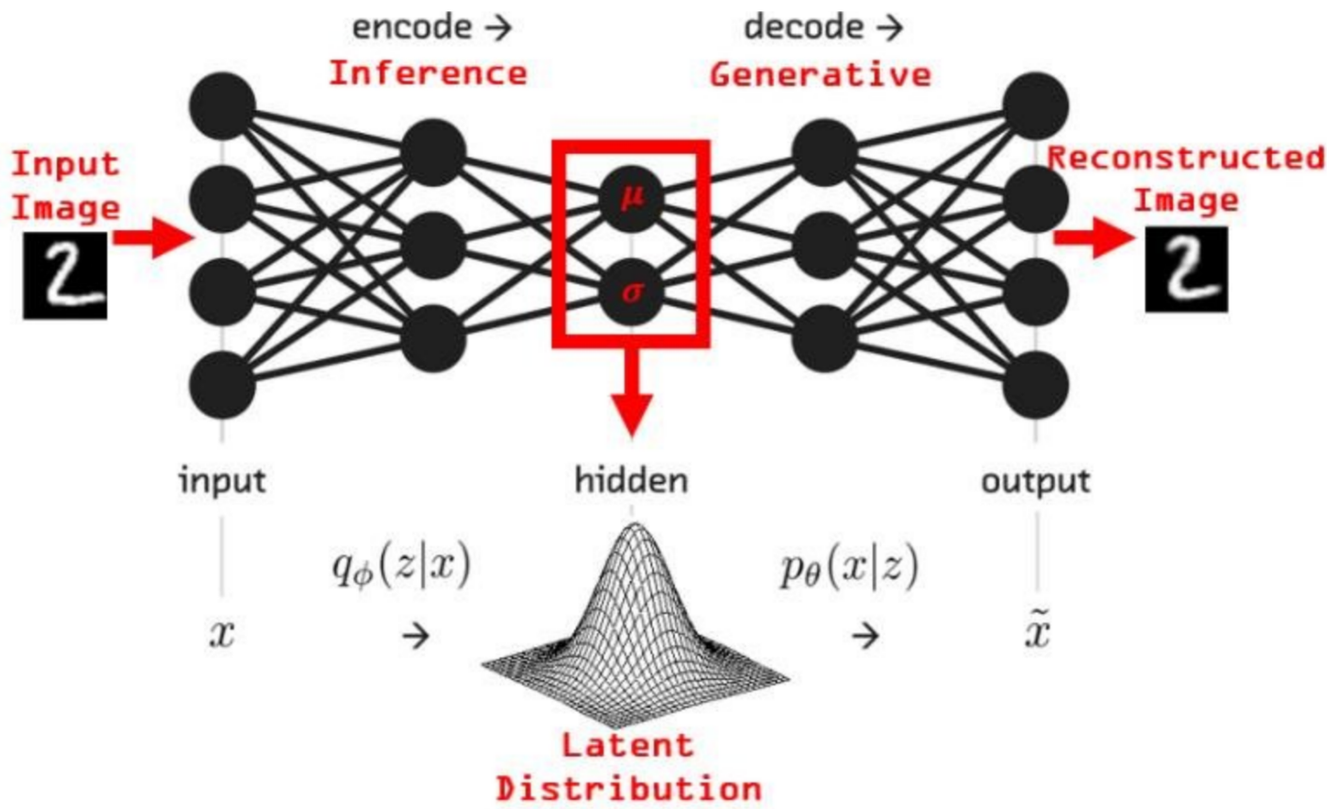
$$p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \approx q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\eta}}(\boldsymbol{x}), \text{diag}(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x})))$$

where $\mu_{\boldsymbol{\eta}}(\boldsymbol{x})$ and $\log(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x}))$ are the outputs of a NN $\mathbb{R}^d \to \mathbb{R}^k$ with parameter vector $\boldsymbol{\eta}$.

The variational autoencoder objective function



The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

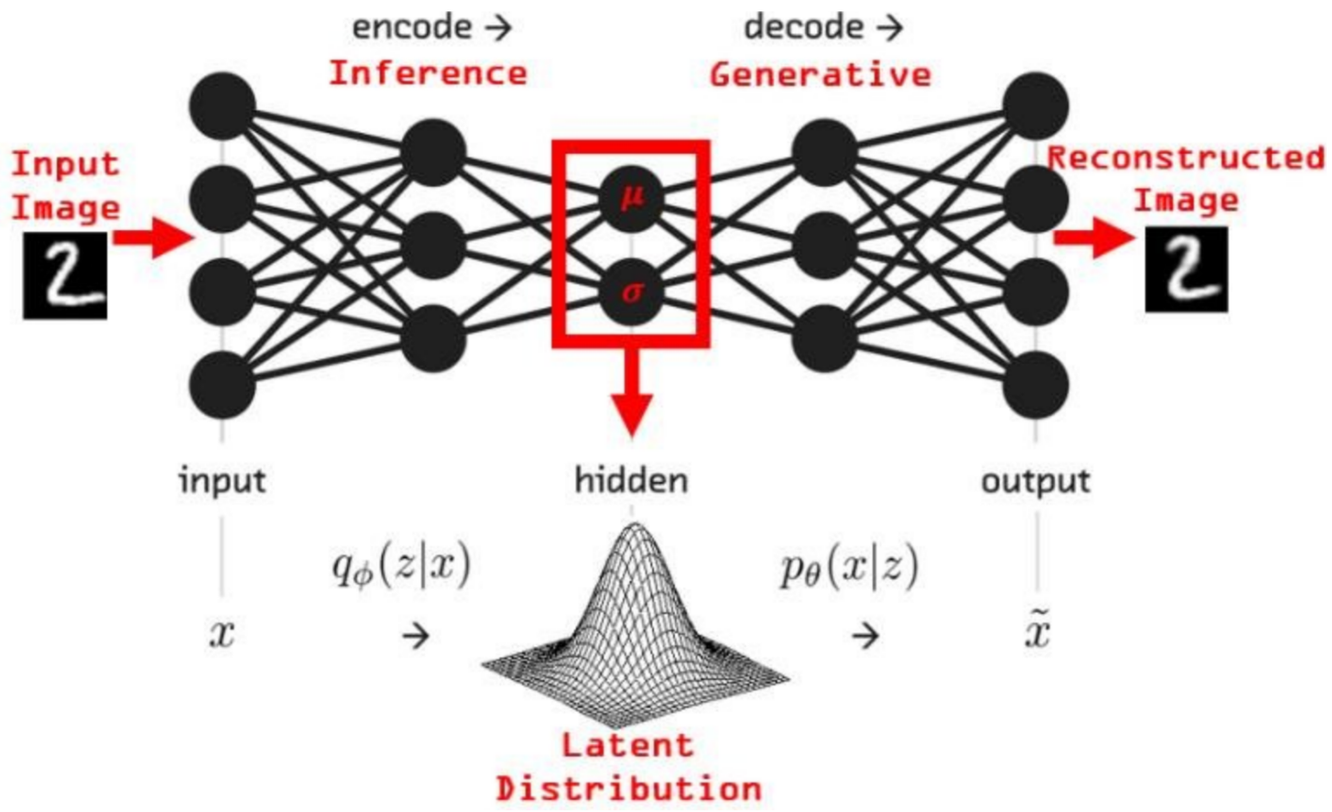
Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

Reconstruction term
(how well we explain training data)

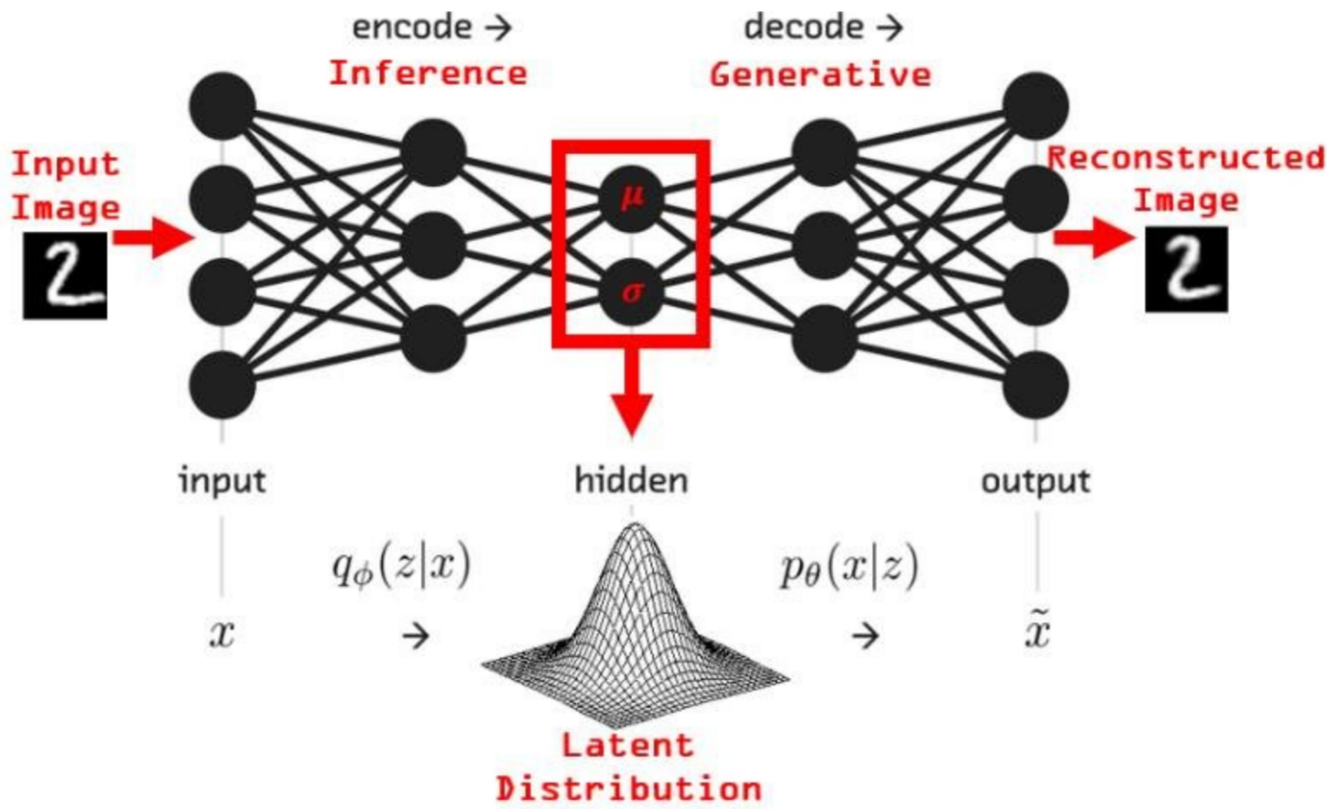
Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

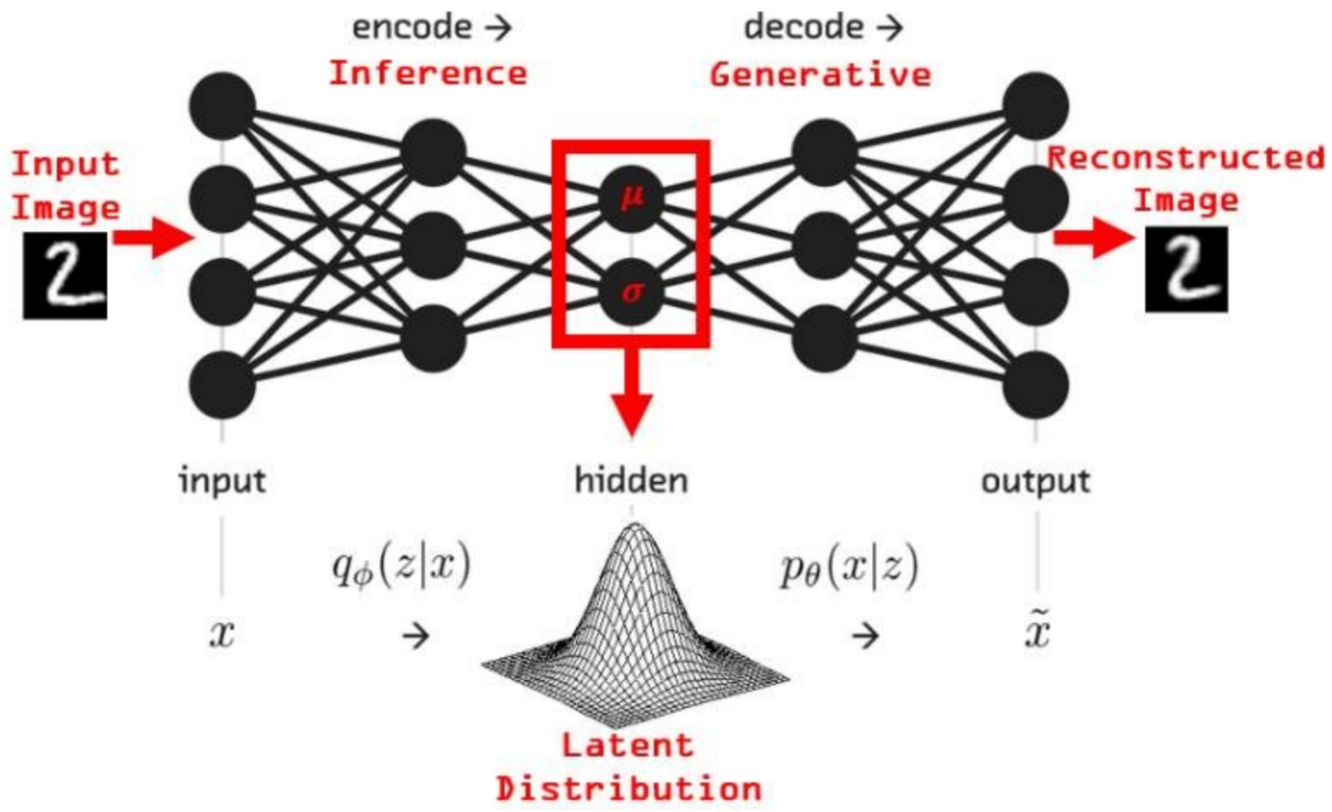
Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

Regularizer (>0)

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

Variational Autoencoder. Maximum Likelihood is intractable!

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

- We cannot compute such integral! ML training is infeasible
- We optimize a lower bound: the **variational** trick

Variational Autoencoder. Maximum Likelihood is intractable!

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

- We cannot compute such integral! ML training is infeasible
- We optimize a lower bound: the **variational** trick

An important concept: KL divergence

- Measures the discrepancy between two distributions
- 0 if the two distributions are identical
- Always positive

$$\mathrm{KL}\left(p(\boldsymbol{x})||q(\boldsymbol{x})\right) = \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x} \geq 0$$

The variational objective is a lower bound on marginal likelihood

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

Consider **any** approximation $q(\boldsymbol{z})$ to $p_\theta(\boldsymbol{z}|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$

$$\log p_\theta(\boldsymbol{x}) = \mathrm{KL}\left(q(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right) + \mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta})$$

The variational objective is a lower bound on marginal likelihood

$$p_\theta(\boldsymbol{x}) = \int p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

Consider **any** approximation $q(\boldsymbol{z})$ to $p_\theta(\boldsymbol{z}|\boldsymbol{x}) \propto p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})$

$$\log p_\theta(\boldsymbol{x}) = \mathrm{KL}\left(q(\boldsymbol{z})||p_\theta(\boldsymbol{z}|\boldsymbol{x})\right) + \mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta})$$

where

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) = \int q(\boldsymbol{z}) \log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q(\boldsymbol{z})} d\boldsymbol{z} = \mathbb{E}_{q(\boldsymbol{z})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right] - \mathrm{KL}\left(q(\boldsymbol{z})||p(\boldsymbol{z})\right)$$

is the Evidence Lower Bound (ELBO).

We will optimize $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ But first we need to select a variational family for $q(\boldsymbol{z})$.

Variational Autoencoder. The variational objective

$$p_{\theta}(\boldsymbol{x}) = \int p_{\theta}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

The Inference Network or Encoding Network

$$p_{\theta}(\boldsymbol{z}|\boldsymbol{x}) \approx q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\eta}}(\boldsymbol{x}), \text{diag}(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x})))$$

where $\mu_{\boldsymbol{\eta}}(\boldsymbol{x})$ and $\log(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x}))$ are the outputs of a NN $\mathbb{R}^d \to \mathbb{R}^k$ with parameter vector $\boldsymbol{\eta}$.

Variational Autoencoder. The variational objective

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$$

The Inference Network or Encoding Network

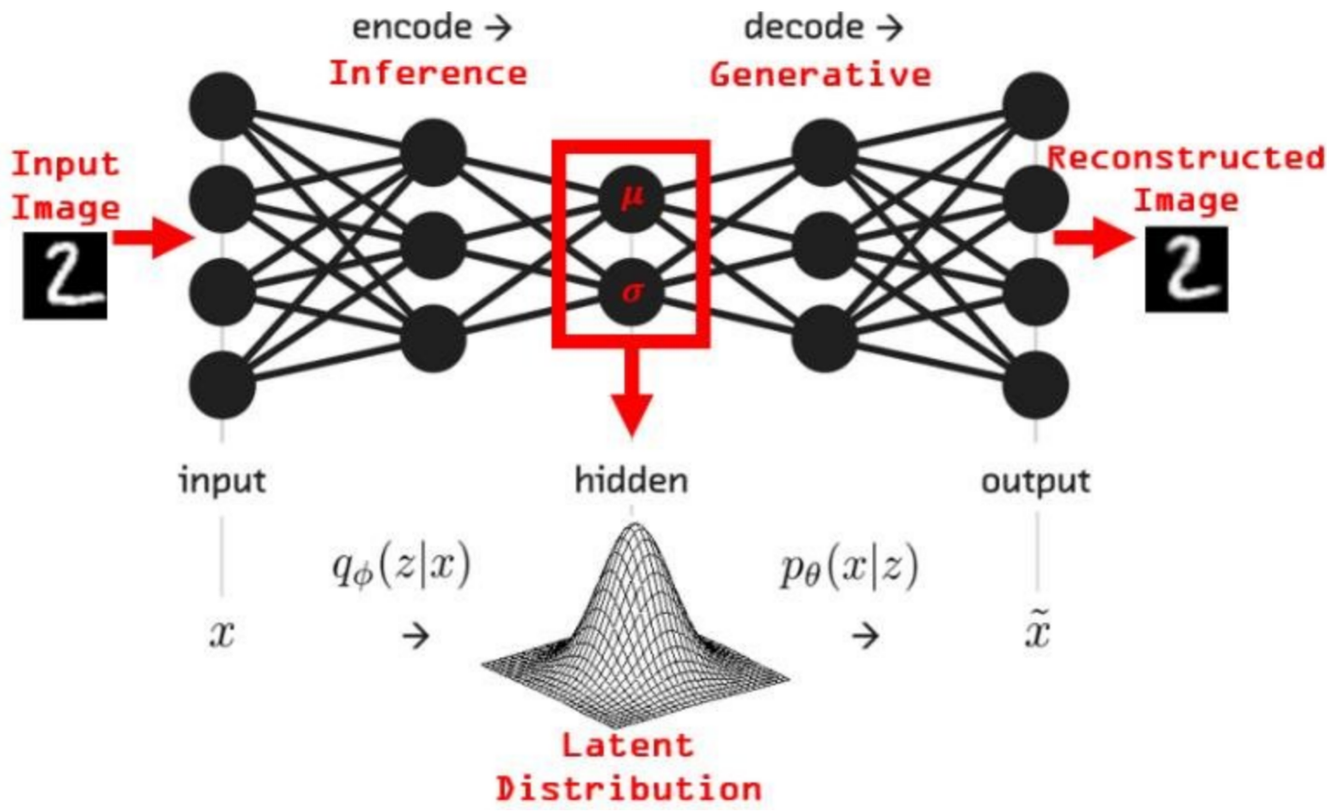
$$p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) \approx q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\eta}}(\boldsymbol{x}), \text{diag}(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x})))$$

where $\mu_{\boldsymbol{\eta}}(\boldsymbol{x})$ and $\log(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x}))$ are the outputs of a NN $\mathbb{R}^d \to \mathbb{R}^k$ with parameter vector $\boldsymbol{\eta}$.

The VAE optimization problem

$$\max_{\boldsymbol{\theta},\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta},\boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) \right] - \text{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

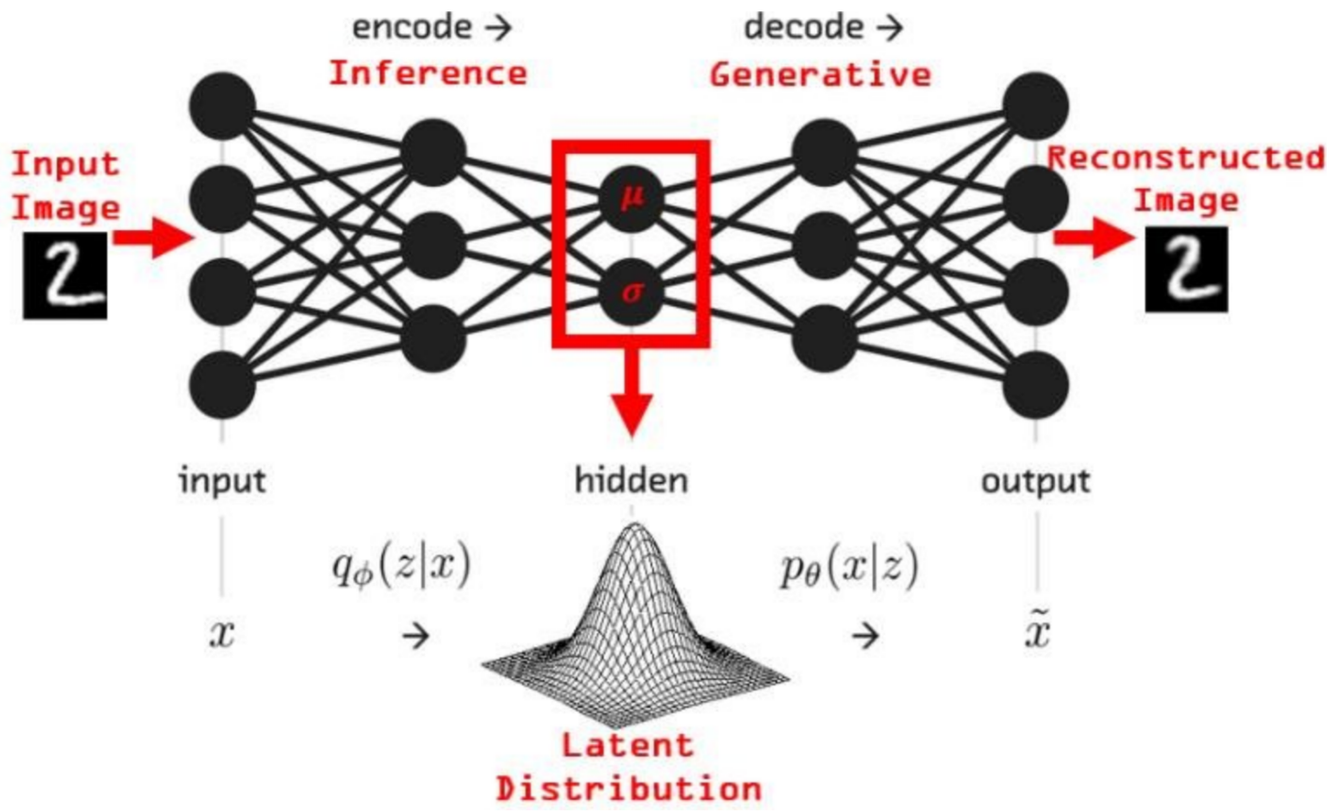
Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

Reconstruction term
(how well we explain training data)

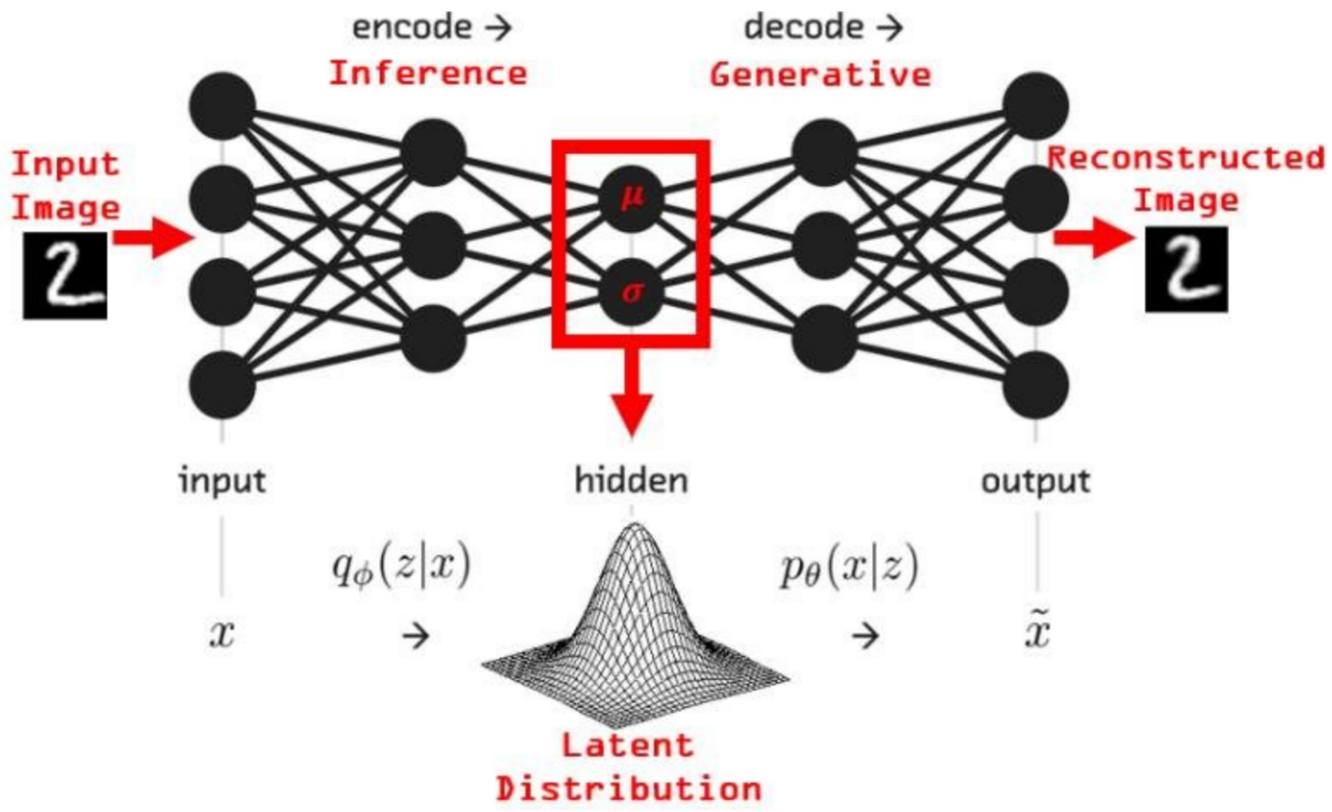
Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

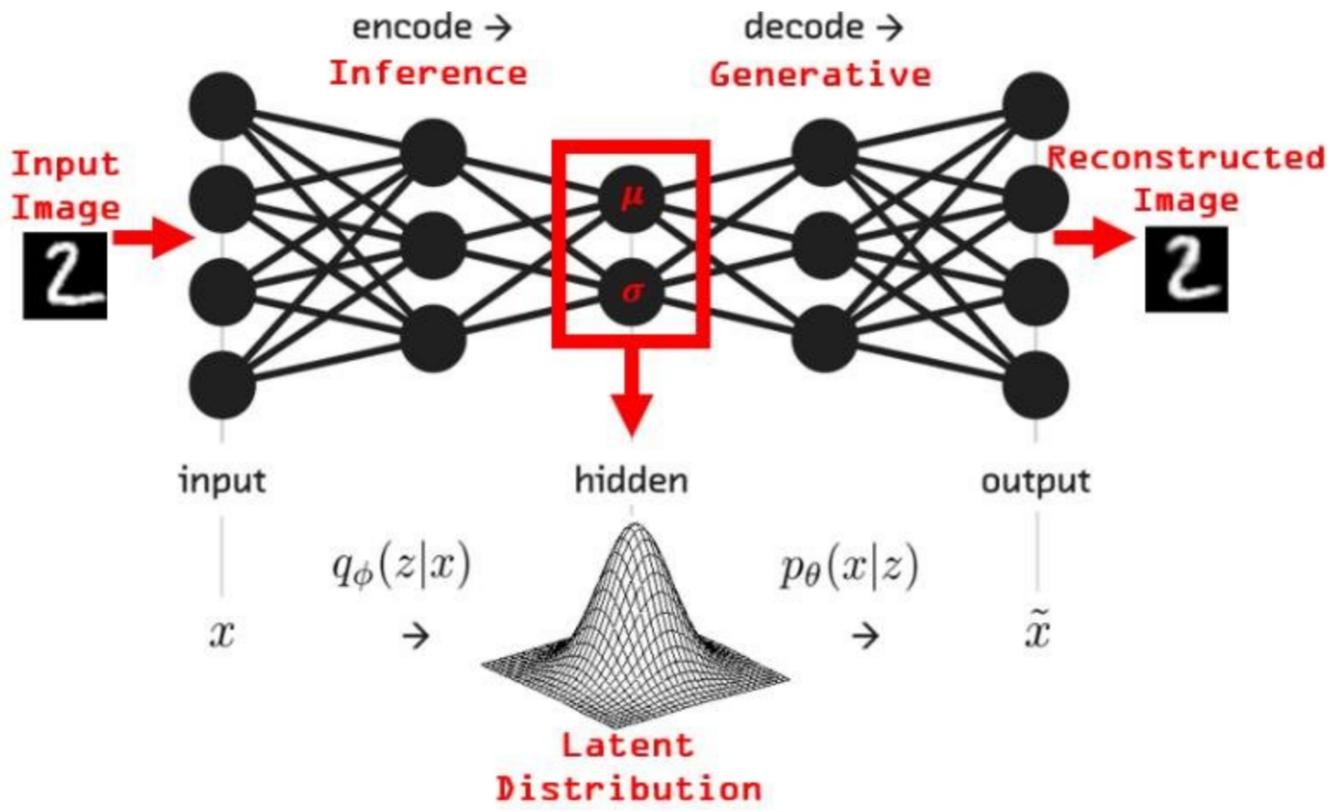
Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

The variational autoencoder objective function



- Probabilistic formulation of autoencoders
- Handle uncertainty
- They perform density estimation!

Autoencoder

$$\mathcal{L}(\eta, \theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n \left(\mathbf{x}_n - D_\theta \left(E_\eta(\mathbf{x}_n) \right) \right)$$

VAE

$$\max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\eta}) = \frac{1}{N} \max_{\boldsymbol{\theta}, \boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | \boldsymbol{z}) \right] - \mathrm{KL} \left(q_{\boldsymbol{\eta}, \boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

Regularizer (>0)

Variational Autoencoder. Stochastic Optimization over the ELBO

$$\max_{\boldsymbol{\theta},\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\eta}) = \max_{\boldsymbol{\theta},\boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) \right] - \mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

Recall that $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\eta}}(\boldsymbol{x}), \mathrm{diag}(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x})))$, thus

$$\mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) = \frac{1}{2} \left(-k + \sum_{j=1}^{k} \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) - \log \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) + \mu_{\boldsymbol{\eta},j}^2 \right)$$

$\rightarrow$ Differentiable w.r.t. $\boldsymbol{\eta}$

Variational Autoencoder. Stochastic Optimization over the ELBO

$$\max_{\boldsymbol{\theta},\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\eta}) = \max_{\boldsymbol{\theta},\boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) \right] - \mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

Recall that $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\eta}}(\boldsymbol{x}), \mathrm{diag}(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x})))$, thus

$$\mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) = \frac{1}{2} \left(-k + \sum_{j=1}^{k} \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) - \log \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) + \mu_{\boldsymbol{\eta},j}^2 \right)$$

$\rightarrow$ Differentiable w.r.t. $\boldsymbol{\eta}$

We use a Monte Carlo sampling estimator

$$\mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) \right] \approx \frac{1}{S} \sum_{s=1}^{S} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{i,s})$$

where $\boldsymbol{z}^{(s,i)}$, $s = 1, \ldots, S$ are i.i.d. samples from $q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z})$.

We typically use a **single** sample, i.e., $S = 1$ (huge estimator variance, but cheap computation).

Variational Autoencoder. Stochastic Optimization over the ELBO

$$\max_{\boldsymbol{\theta},\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\eta}) = \max_{\boldsymbol{\theta},\boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) \right] - \mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

Recall that $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\eta}}(\boldsymbol{x}), \mathrm{diag}(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x})))$, thus

$$\mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) = \frac{1}{2} \left(-k + \sum_{j=1}^{k} \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) - \log \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) + \mu_{\boldsymbol{\eta},j}^2 \right)$$

$\rightarrow$ Differentiable w.r.t. $\boldsymbol{\eta}$

Variational Autoencoder. Stochastic Optimization over the ELBO

$$\max_{\boldsymbol{\theta},\boldsymbol{\eta}} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\eta}) = \max_{\boldsymbol{\theta},\boldsymbol{\eta}} \left(\sum_{i=1}^{N} \mathbb{E}_{q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}|\boldsymbol{z}) \right] - \mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) \right)$$

Recall that $p(\boldsymbol{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) = \mathcal{N}(\mu_{\boldsymbol{\eta}}(\boldsymbol{x}), \mathrm{diag}(\sigma_{\boldsymbol{\eta}}(\boldsymbol{x})))$, thus

$$\mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}}(\boldsymbol{z}) || p(\boldsymbol{z}) \right) = \frac{1}{2} \left(-k + \sum_{j=1}^{k} \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) - \log \sigma_{\boldsymbol{\eta},j}(\boldsymbol{x}) + \mu_{\boldsymbol{\eta},j}^2 \right)$$

$\rightarrow$ Differentiable w.r.t. $\boldsymbol{\eta}$

Reparameterization Trick

Express each sample $\boldsymbol{z}^{(s,i)}$ as a deterministic function of $\boldsymbol{x}^{(i)}$ and some noise vector $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ that is independent of $\boldsymbol{\eta}$. For a Gaussian distribution we have

$$\boldsymbol{z}^{(s,i)} = f_{\boldsymbol{\eta}}(\boldsymbol{x}^{(i)}, \epsilon) = \mu_{\boldsymbol{\eta}}(\boldsymbol{x}^{(i)}) + \sqrt{\sigma_{\boldsymbol{\eta}}(\boldsymbol{x}^{(i)})} \cdot \epsilon$$

Variational Autoencoder. Algorithm

Algorithm 1 The Variational Autoencoder (S=1)

1: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_0$, $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta}_0$

2: $\ell \leftarrow 0$

3: **while** not converged **do**

4: Sample minibatch $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}\}$ from $\mathcal{D}$

5: Sample $\{\epsilon^1, \ldots, \epsilon^M\}$ from $p(\epsilon) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

6: Compute noisy gradients:

$$\boldsymbol{g}_{\boldsymbol{\theta}}, \boldsymbol{g}_{\boldsymbol{\eta}} \leftarrow \frac{1}{M} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta},\boldsymbol{\eta}} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)} | f_{\boldsymbol{\eta}}(\epsilon^i, \boldsymbol{x}^{(i)})) - \mathrm{KL}\left(q_{\boldsymbol{\eta},\boldsymbol{x}^{(i)}}(\boldsymbol{z}) || p(\boldsymbol{z})\right) \right]$$

7: Perform SGD-updates:

$$\boldsymbol{\theta}_{\ell+1} \leftarrow \boldsymbol{\theta}_{\ell+1} + h_\ell \boldsymbol{g}_{\boldsymbol{\theta}}$$
$$\boldsymbol{\eta}_{\ell+1} \leftarrow \boldsymbol{\eta}_{\ell+1} + h_\ell \boldsymbol{\eta}_{\boldsymbol{\theta}}$$

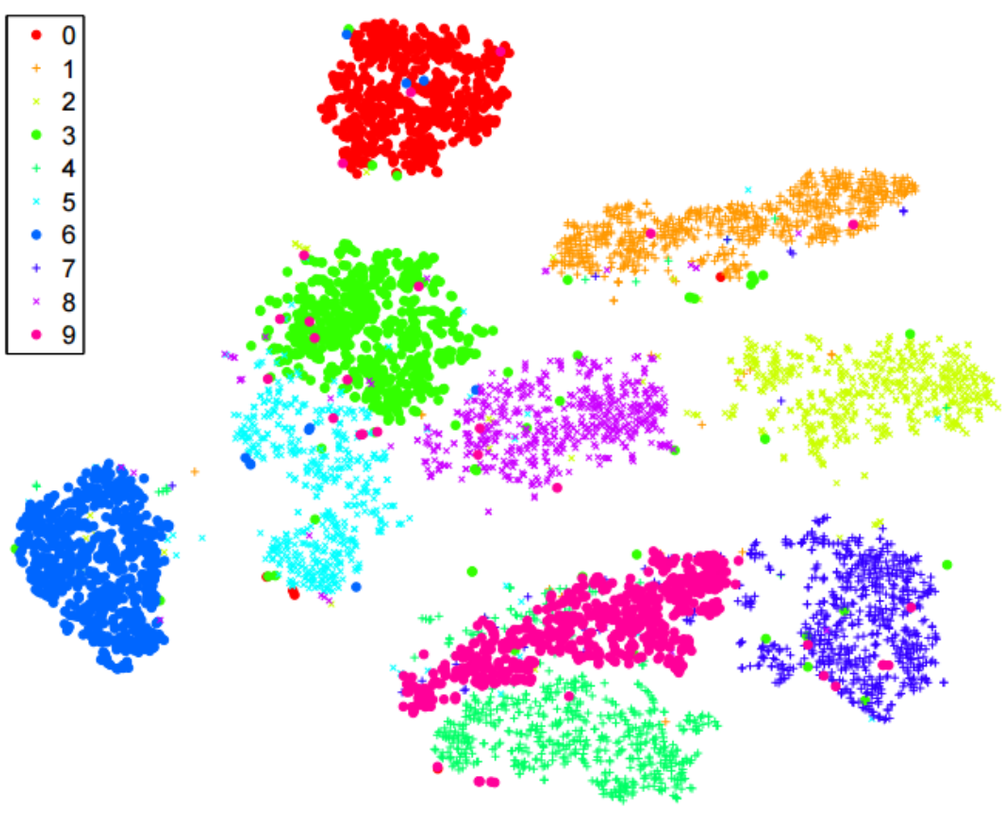
8: $\ell \leftarrow \ell + 1$

9: **end while**

Variational Autoencoder. Clustering in the latent space

Generated Samples

Projection over latent space



VQ-VAE

- Uses vector quantization (VQ) to obtain a **discrete latent representation**
- How it differs from traditional VAEs:
  - Encoder network output is discrete
  - Prior is learnt rather than static
- VQ ideas used to learn the discrete latent space
  - Help improve posterior collapse

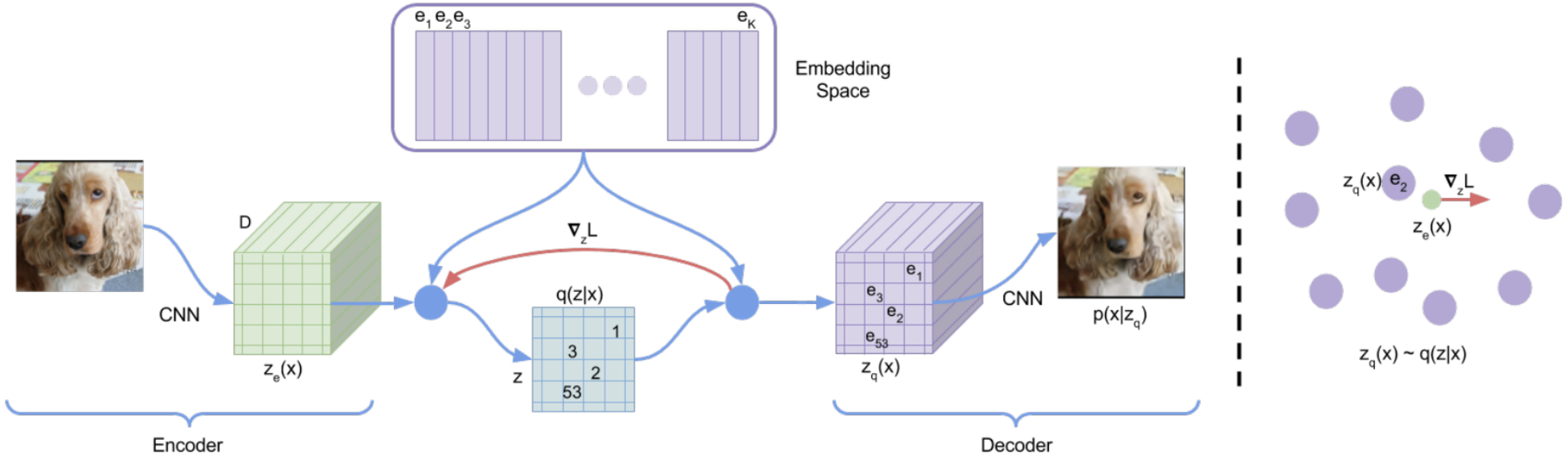


Figure 1: Left: A figure describing the VQ-VAE. Right: Visualisation of the embedding space. The output of the encoder $z(x)$ is mapped to the nearest point e_2. The gradient $\nabla_z L$ (in red) will push the encoder to change its output, which could alter the configuration in the next forward pass.

Aaron van den Oord, Oriol Vinyals, Koray Kavukcuoglu, “Neural Discrete Representation Learning” NeurIPS 2017

VQ-VAE

- Uses vector quantization (VQ) to obtain a **discrete latent representation**
- How it differs from traditional VAEs:
  - Encoder network output is discrete
  - Prior is learnt rather than static
- VQ ideas used to learn the discrete latent space
  - Help improve posterior collapse

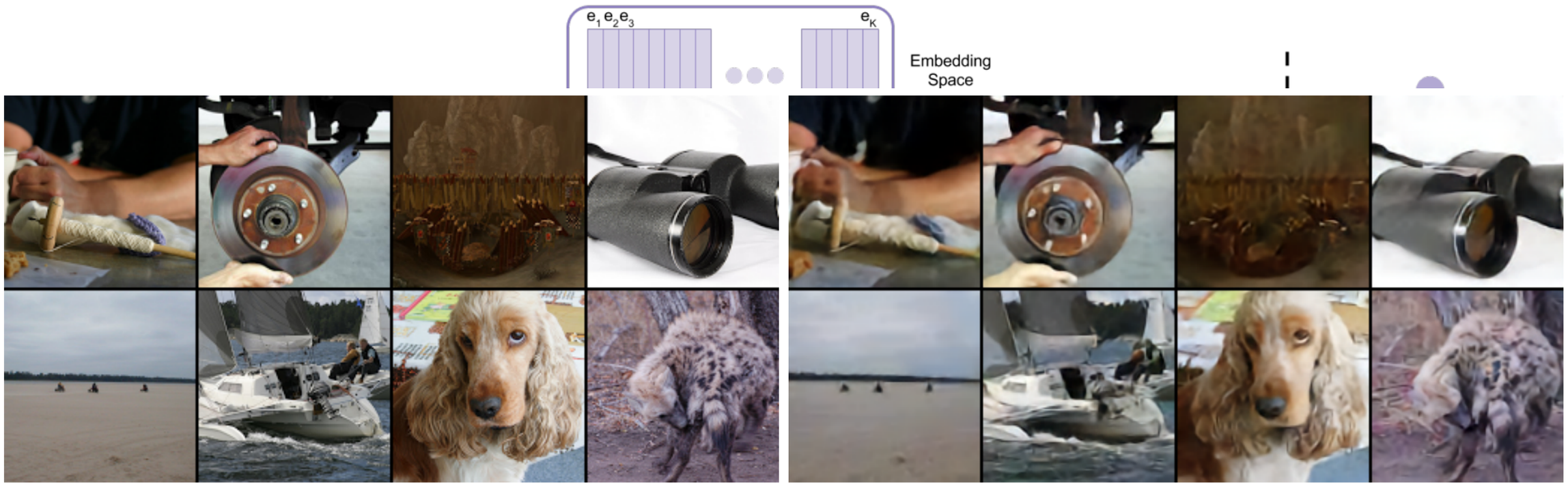
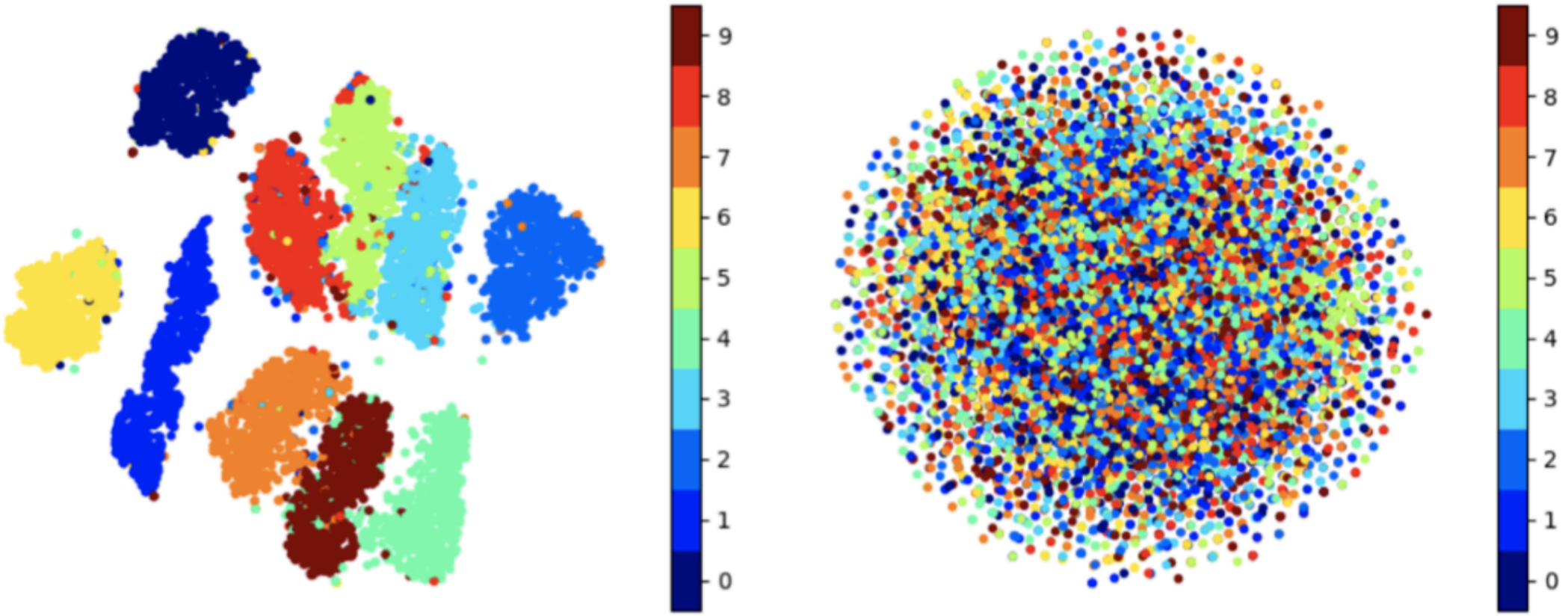


Figure 2: Left: ImageNet 128x128x3 images, right: reconstructions from a VQ-VAE with a 32x32x1 latent space, with K=512.

Aaron van den Oord, Oriol Vinyals, Koray Kavukcuoglu, “Neural Discrete Representation Learning” NeurIPS 2017

Posterior Collapse



Ideal

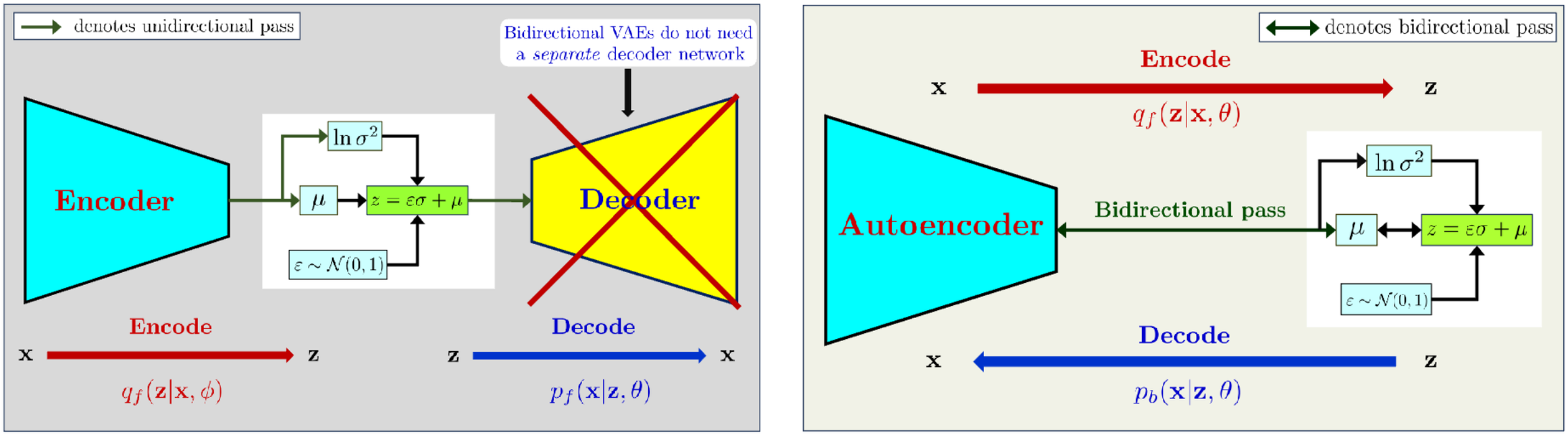
Reality

Jha et al. (CVPR, 2018)

Model fits well - good predictive likelihood and good generation

Posterior equal to prior - non informative, useless as representation

VAEs in complex datasets



(a) Unidirectional VAE architecture

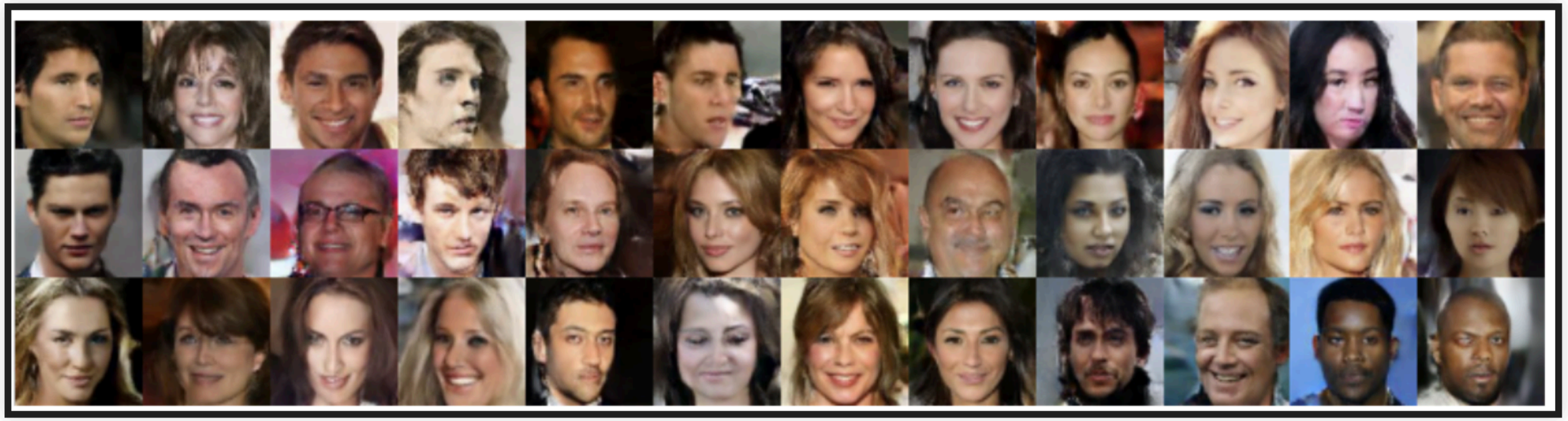
(b) Bidirectional VAE architecture

Fig. 1: Bidirectional vs. unidirectional variational autoencoders: Unidirectional VAEs use the forward passes of two separate networks for encoding and decoding. Bidirectional VAEs run their encoding on the forward pass and decoding on the backward pass with the same synaptic webs–weight matrices in both directions. This cuts the number of tunable parameters roughly in half. (a) The decoder network with parameter θ approximates $p(x|z, \theta)$ and the encoder network with parameter ϕ approximates $q(z|x, \theta)$. (b) Bidirectional VAEs use the forward pass of a network with parameter θ to approximate $q(z|x, \theta)$ and the backward pass of the network to approximate $p(x|z, \theta)$.

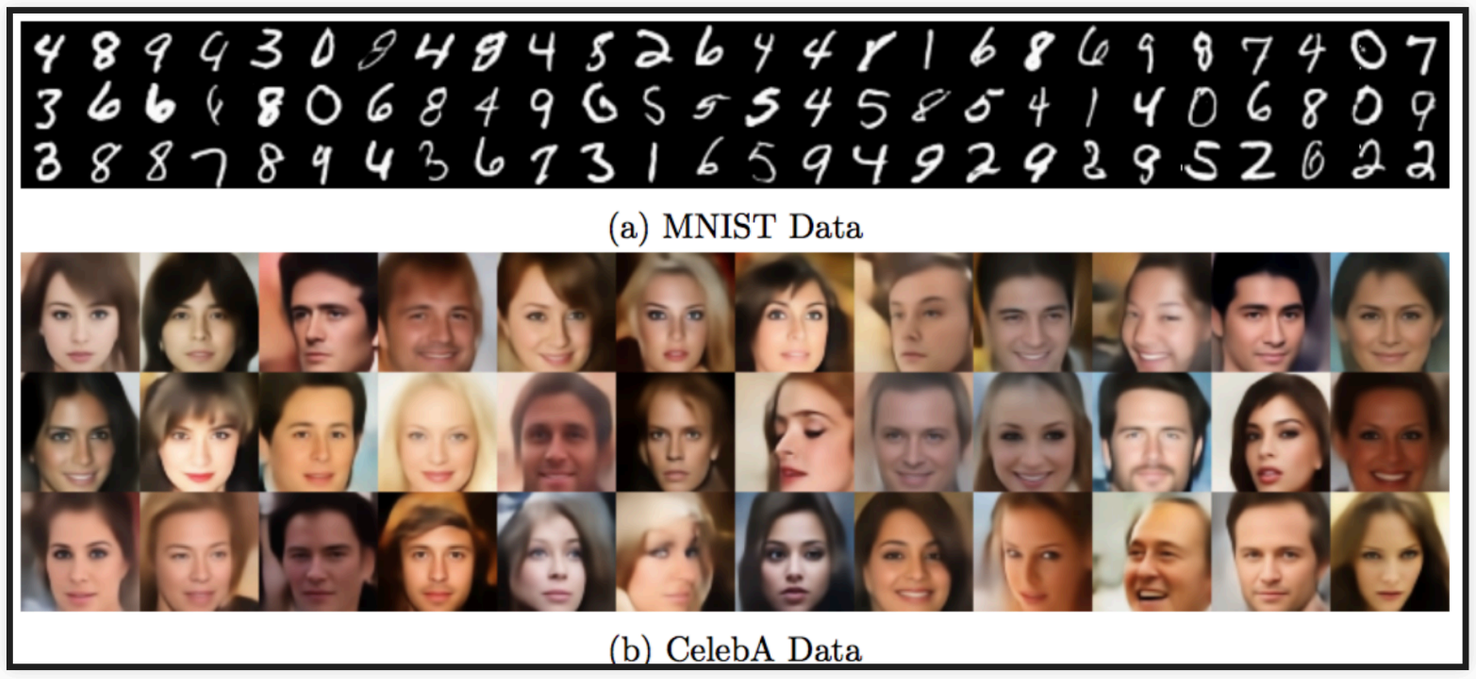
VAEs in complex datasets

Bidirectional Inference VAE (Maaloe et al. 2019)

VAEs in complex datasets



Bidirectional Inference VAE (Maaloe et al. 2019)



Two-stage VAE (Dai & Wipf 2019)