

Machine Learning

A. G. Schwing & M. Telgarsky

University of Illinois at Urbana-Champaign, 2018

L22: Variational Auto-Encoders

Goals of this lecture

Goals of this lecture

- Getting to know Variational Auto-Encoders

Goals of this lecture

- Getting to know Variational Auto-Encoders
- Understanding generative methods

Goals of this lecture

- Getting to know Variational Auto-Encoders
- Understanding generative methods
- Differentiating between discriminative and generative methods

Goals of this lecture

- Getting to know Variational Auto-Encoders
- Understanding generative methods
- Differentiating between discriminative and generative methods

Reading material

Goals of this lecture

- Getting to know Variational Auto-Encoders
- Understanding generative methods
- Differentiating between discriminative and generative methods

Reading material

- D.P. Kingma and M. Welling; Auto-Encoding Variational Bayes;
arxiv.org/abs/1312.6114

Goals of this lecture

- Getting to know Variational Auto-Encoders
- Understanding generative methods
- Differentiating between discriminative and generative methods

Reading material

- D.P. Kingma and M. Welling; Auto-Encoding Variational Bayes;
arxiv.org/abs/1312.6114
- C. Doersch; Tutorial on Variational Autoencoders;
arxiv.org/abs/1606.05908

Recap: Maximum likelihood so far?

Recap: Maximum likelihood so far?

Model:

$$p(\mathbf{y}|x) = \frac{\exp F(\mathbf{y}, x, \mathbf{w})/\epsilon}{\sum_{\hat{\mathbf{y}}} \exp F(\hat{\mathbf{y}}, x, \mathbf{w})/\epsilon}$$

Recap: Maximum likelihood so far?

Model:

$$p(\mathbf{y}|x) = \frac{\exp F(\mathbf{y}, x, \mathbf{w})/\epsilon}{\sum_{\hat{\mathbf{y}}} \exp F(\hat{\mathbf{y}}, x, \mathbf{w})/\epsilon}$$

- \mathbf{y} :

Recap: Maximum likelihood so far?

Model:

$$p(\mathbf{y}|x) = \frac{\exp F(\mathbf{y}, x, \mathbf{w})/\epsilon}{\sum_{\hat{\mathbf{y}}} \exp F(\hat{\mathbf{y}}, x, \mathbf{w})/\epsilon}$$

- \mathbf{y} : discrete output space

Recap: Maximum likelihood so far?

Model:

$$p(\mathbf{y}|x) = \frac{\exp F(\mathbf{y}, x, \mathbf{w})/\epsilon}{\sum_{\hat{\mathbf{y}}} \exp F(\hat{\mathbf{y}}, x, \mathbf{w})/\epsilon}$$

- \mathbf{y} : discrete output space
- x :

Recap: Maximum likelihood so far?

Model:

$$p(\mathbf{y}|x) = \frac{\exp F(\mathbf{y}, x, \mathbf{w})/\epsilon}{\sum_{\hat{\mathbf{y}}} \exp F(\hat{\mathbf{y}}, x, \mathbf{w})/\epsilon}$$

- \mathbf{y} : discrete output space
- x : input data

Recap: Maximum likelihood so far?

Model:

$$p(\mathbf{y}|x) = \frac{\exp F(\mathbf{y}, x, \mathbf{w})/\epsilon}{\sum_{\hat{\mathbf{y}}} \exp F(\hat{\mathbf{y}}, x, \mathbf{w})/\epsilon}$$

- \mathbf{y} : discrete output space
- x : input data

Now and the past few lectures:

Recap: Maximum likelihood so far?

Model:

$$p(\mathbf{y}|x) = \frac{\exp F(\mathbf{y}, x, \mathbf{w})/\epsilon}{\sum_{\hat{\mathbf{y}}} \exp F(\hat{\mathbf{y}}, x, \mathbf{w})/\epsilon}$$

- \mathbf{y} : discrete output space
- x : input data

Now and the past few lectures:

How about modeling a distribution $p(x)$ for the data?

Given data points x , how can we model $p(x)$?

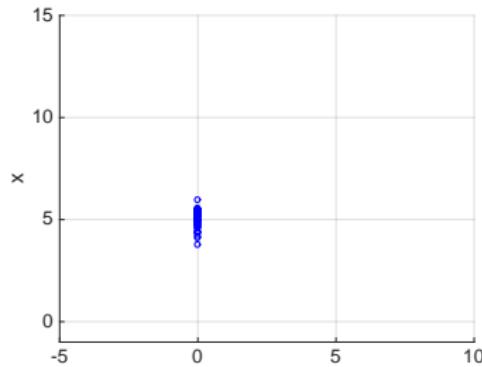
Given data points x , how can we model $p(x)$?

- Fit mean and variance (= parameters θ) of a distribution (e.g., Gaussian)

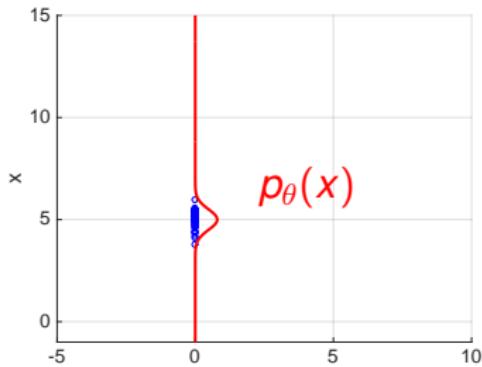
Given data points x , how can we model $p(x)$?

- Fit mean and variance (= parameters θ) of a distribution (e.g., Gaussian)
- Fit parameters θ of a mixture distribution (e.g., mixture of Gaussian, k-means)

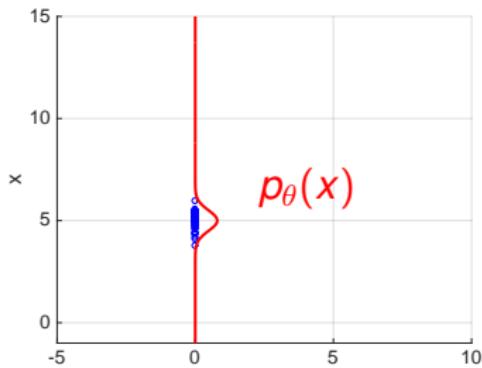
Example:



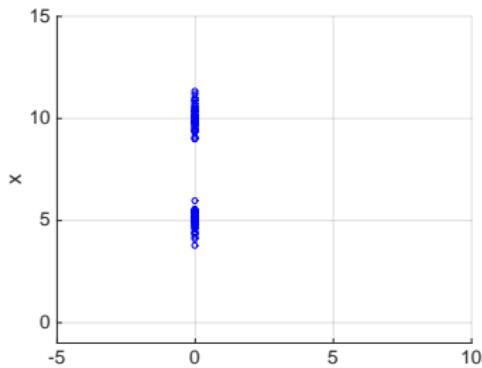
Example:



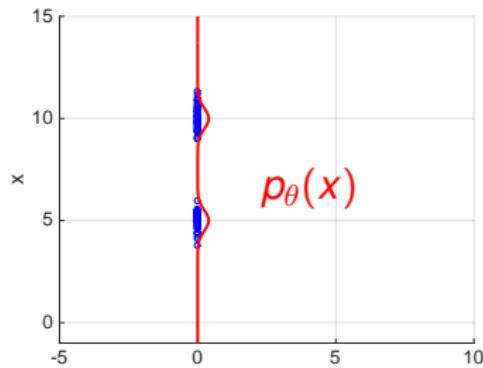
Example:



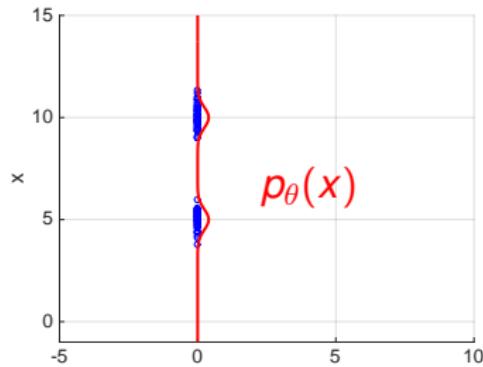
Example:



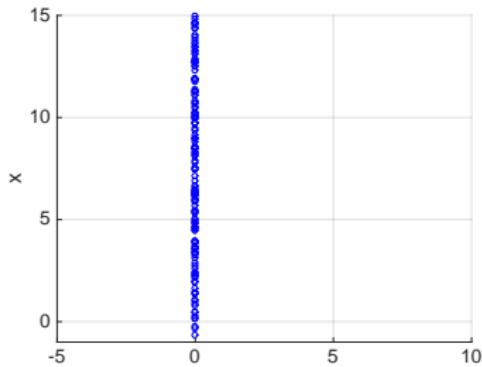
Example:



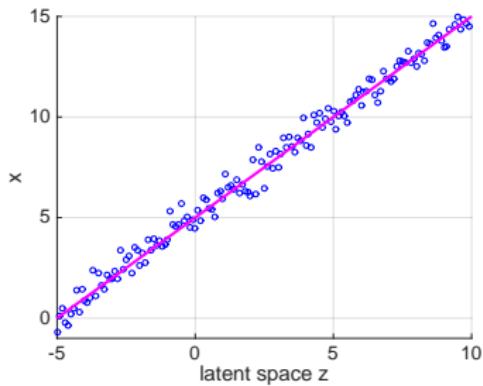
Example:



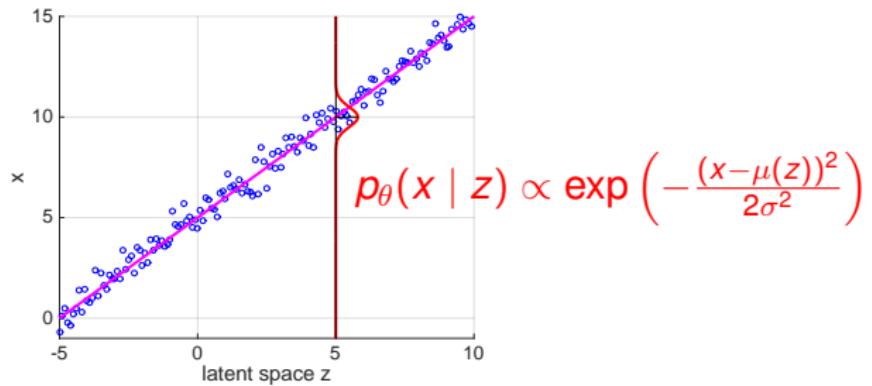
Example:



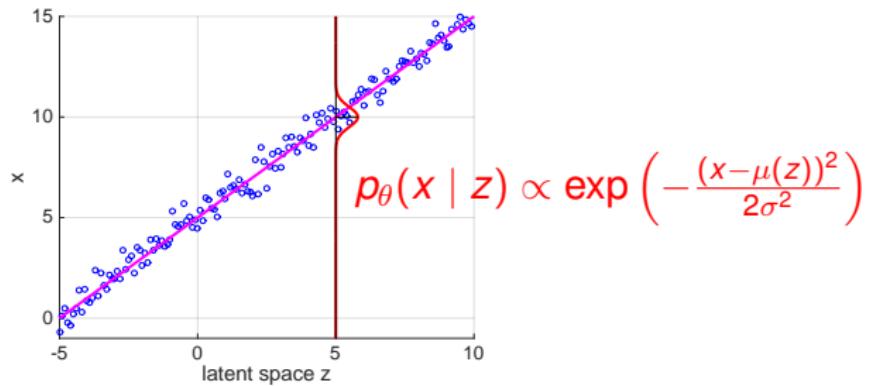
Example:



Example:

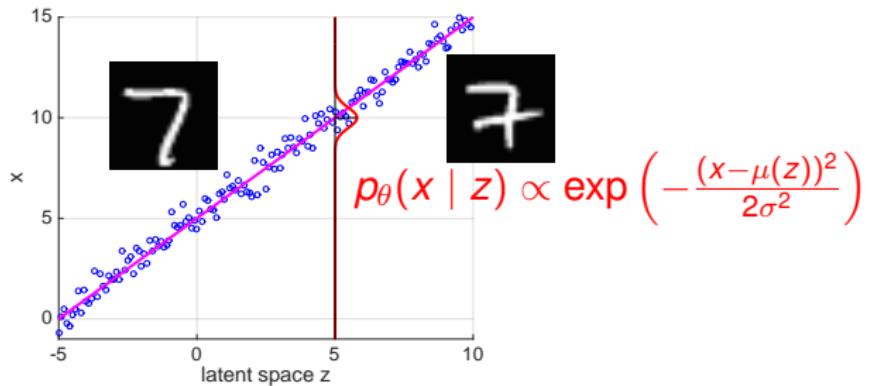


Example:



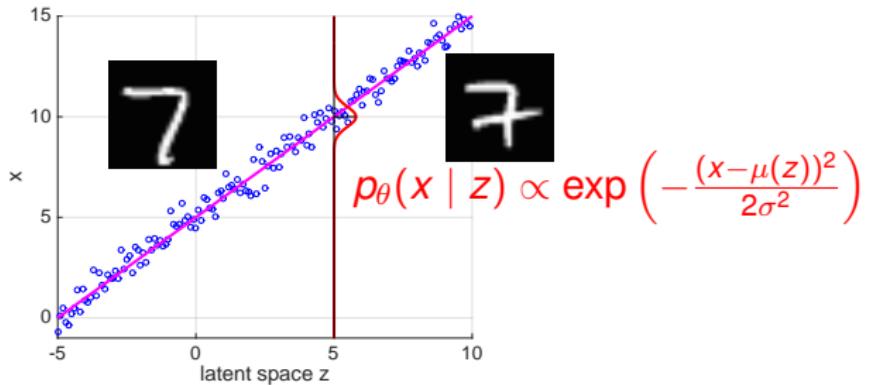
Increasing dimensions:

$$x \in \mathbb{R}^{784}$$



Increasing dimensions:

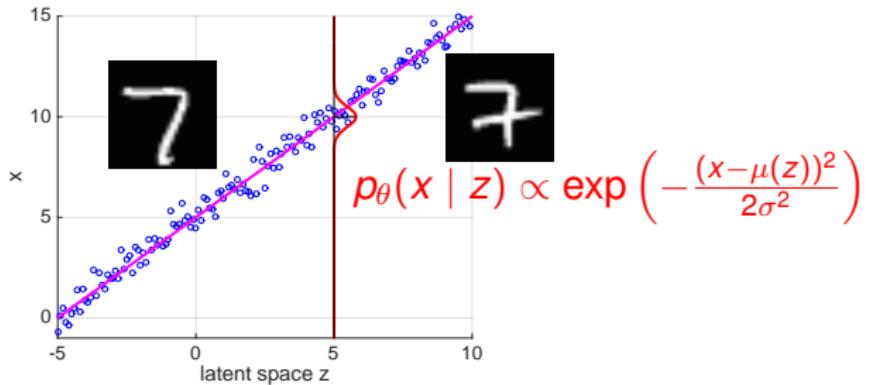
$$x \in \mathbb{R}^{784}$$



Manifold hypothesis:

Increasing dimensions:

$$x \in \mathbb{R}^{784}$$

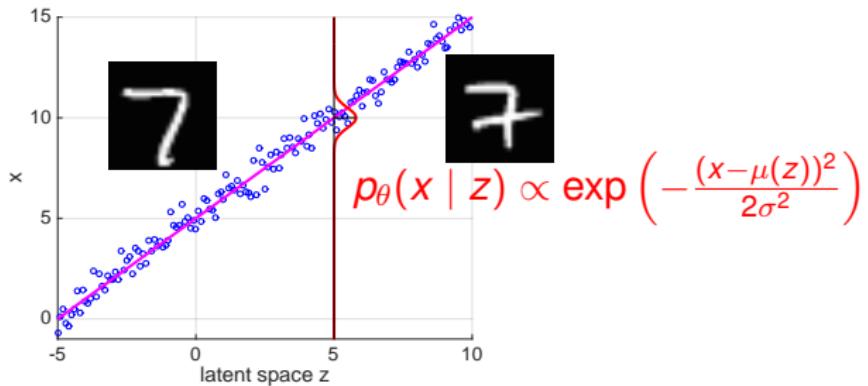


Manifold hypothesis:

- x is a high dimensional vector

Increasing dimensions:

$$x \in \mathbb{R}^{784}$$



Manifold hypothesis:

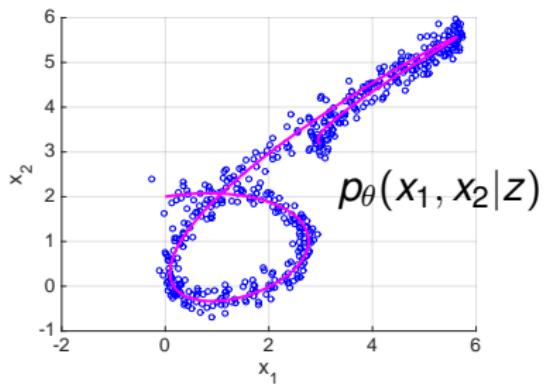
- x is a high dimensional vector
- data is concentrated around a low dimensional manifold

Examples of low-dimensional manifolds:

$$z \in [0, 1] \rightarrow$$

Examples of low-dimensional manifolds:

$$z \in [0, 1] \rightarrow$$



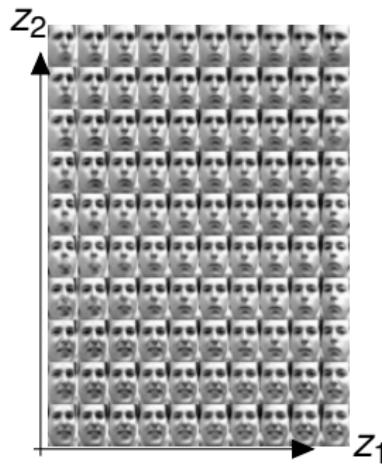
Examples of low-dimensional manifolds: ($z \in \mathbb{R}^2$)

Examples of low-dimensional manifolds: ($z \in \mathbb{R}^2$)

Kingma and Welling: “Auto-Encoding Variational Bayes” (ICLR 2014)

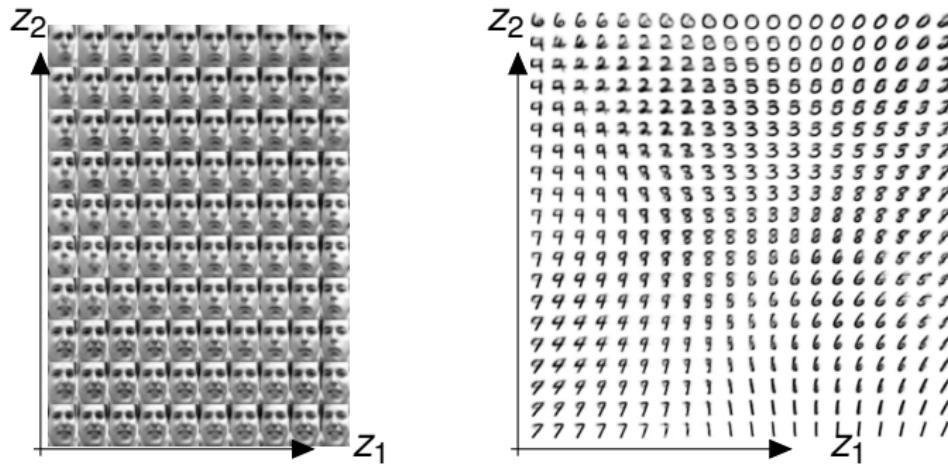
Examples of low-dimensional manifolds: ($z \in \mathbb{R}^2$)

Kingma and Welling: "Auto-Encoding Variational Bayes" (ICLR 2014)



Examples of low-dimensional manifolds: ($z \in \mathbb{R}^2$)

Kingma and Welling: “Auto-Encoding Variational Bayes” (ICLR 2014)



What is $p_\theta(x|z)$?

What is $p_{\theta}(x|z)$?

- Given z , we can generate, compute or sample data x

What is $p_\theta(x|z)$?

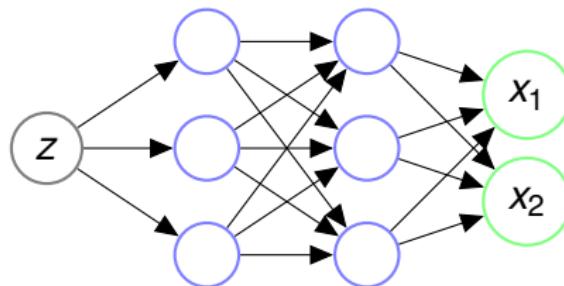
- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net

What is $p_\theta(x|z)$?

- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net
- Decoder

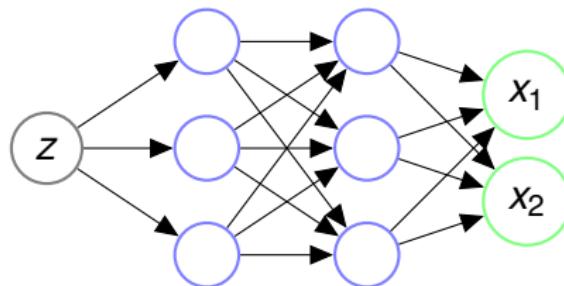
What is $p_\theta(x|z)$?

- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net
- Decoder



What is $p_\theta(x|z)$?

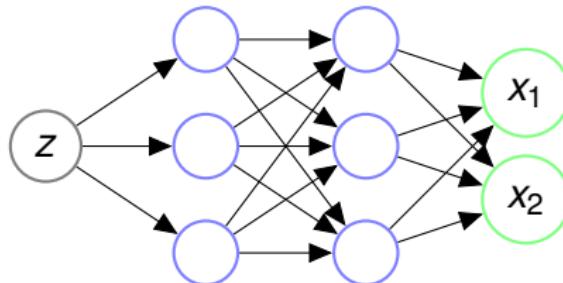
- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net
- Decoder



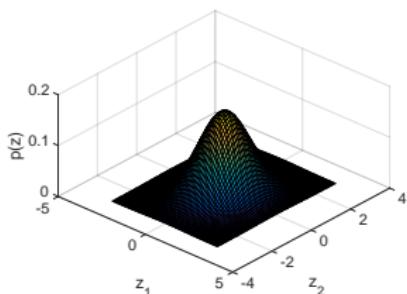
More generally:

What is $p_\theta(x|z)$?

- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net
- Decoder

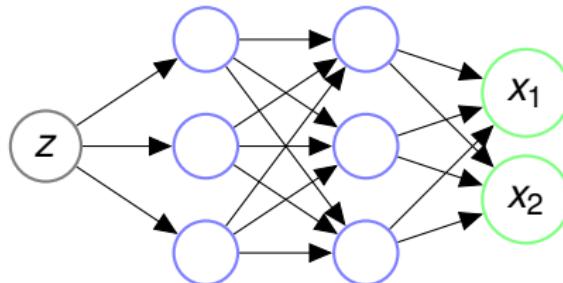


More generally:

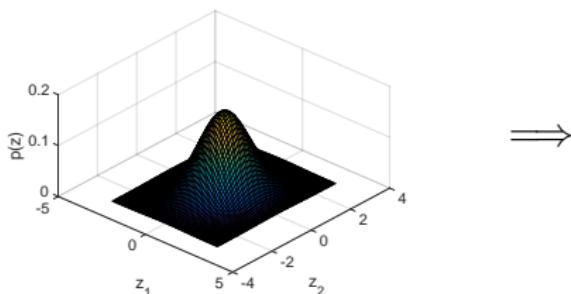


What is $p_\theta(x|z)$?

- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net
- Decoder

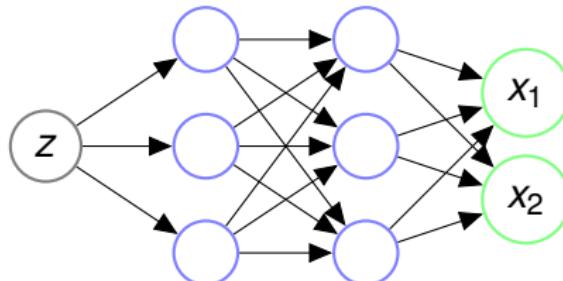


More generally:

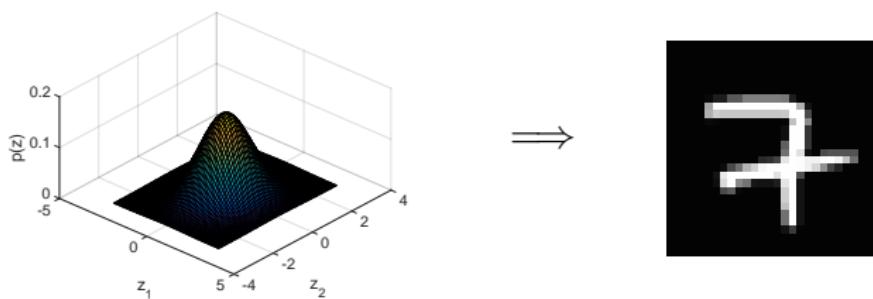


What is $p_\theta(x|z)$?

- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net
- Decoder

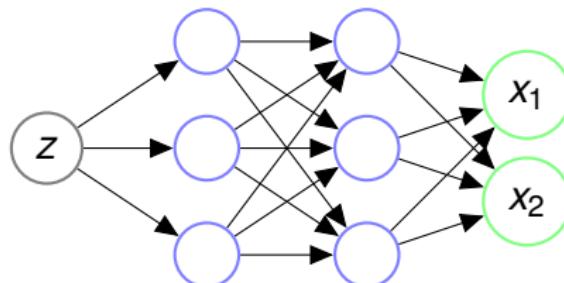


More generally:

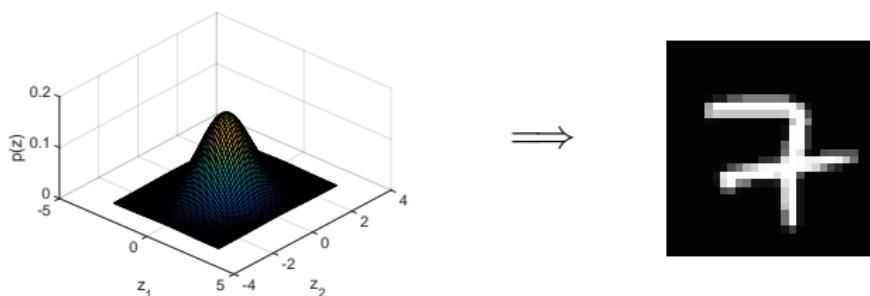


What is $p_\theta(x|z)$?

- Given z , we can generate, compute or sample data x
- Idea: z from simple Gaussian & transformation via deep net
- Decoder

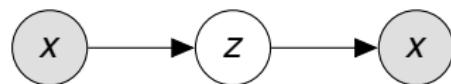


More generally:



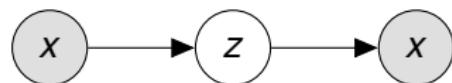
But how to optimize for θ ? How do we know which z maps to which x ?

Training as an auto-encoder: How about

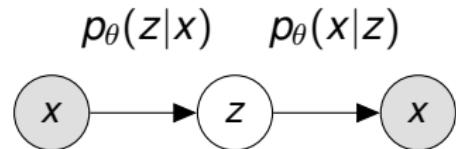


Training as an auto-encoder: How about

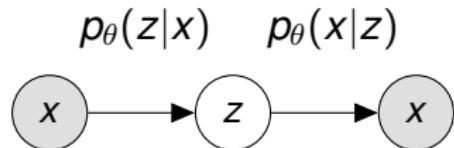
$$p_{\theta}(x|z)$$



Training as an auto-encoder: How about

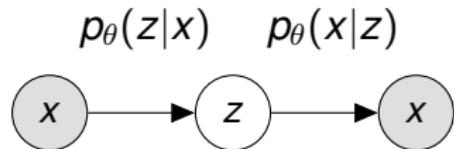


Training as an auto-encoder: How about



What is $p_{\theta}(z|x)$?

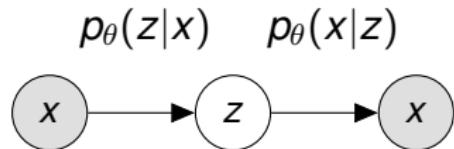
Training as an auto-encoder: How about



What is $p_{\theta}(z|x)$?

$$p_{\theta}(z|x) = \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(x)}$$

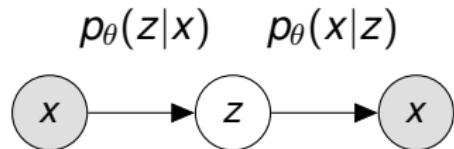
Training as an auto-encoder: How about



What is $p_{\theta}(z|x)$?

$$p_{\theta}(z|x) = \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(x)} = \frac{p_{\theta}(x|z)p(z)}{\int_{\hat{z}} p_{\theta}(x|\hat{z})p(\hat{z})d\hat{z}}$$

Training as an auto-encoder: How about

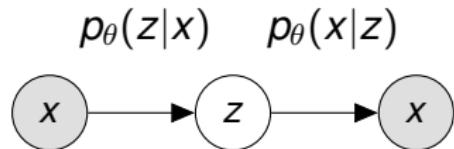


What is $p_{\theta}(z|x)$?

$$p_{\theta}(z|x) = \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(x)} = \frac{p_{\theta}(x|z)p(z)}{\int_{\hat{z}} p_{\theta}(x|\hat{z})p(\hat{z})d\hat{z}}$$

How to compute normalization constant?

Training as an auto-encoder: How about



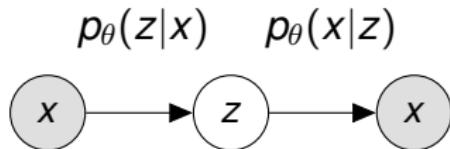
What is $p_{\theta}(z|x)$?

$$p_{\theta}(z|x) = \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(x)} = \frac{p_{\theta}(x|z)p(z)}{\int_{\hat{z}} p_{\theta}(x|\hat{z})p(\hat{z})d\hat{z}}$$

How to compute normalization constant?

- Sampling techniques (generally very costly)

Training as an auto-encoder: How about



What is $p_{\theta}(z|x)$?

$$p_{\theta}(z|x) = \frac{p_{\theta}(x|z)p(z)}{p_{\theta}(x)} = \frac{p_{\theta}(x|z)p(z)}{\int_{\hat{z}} p_{\theta}(x|\hat{z})p(\hat{z})d\hat{z}}$$

How to compute normalization constant?

- Sampling techniques (generally very costly)
- Approximate $p_{\theta}(z|x)$ with $q_{\phi}(z|x)$ (encoder)

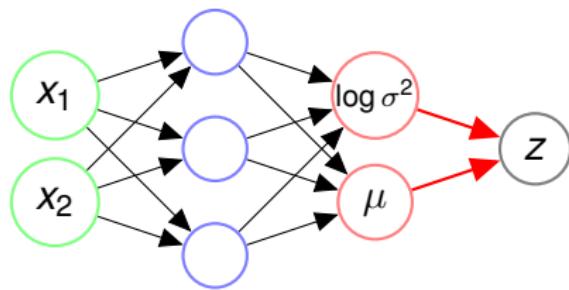
Encoder $q_\phi(z|x)$ as a deep net which computes mean and variance:

Encoder $q_\phi(z|x)$ as a deep net which computes mean and variance:

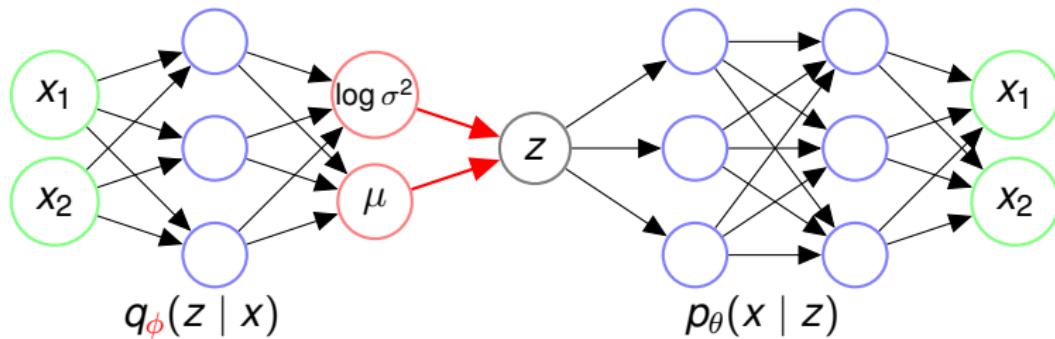
$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x))$$

Encoder $q_\phi(z|x)$ as a deep net which computes mean and variance:

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x))$$



Variational auto-encoder architecture:



Learning parameters θ, ϕ via back-propagation

What do we want to optimize? What is the loss function?

What do we want to optimize? What is the loss function?

$$\log p_{\theta}(x)$$

What do we want to optimize? What is the loss function?

$$\log p_{\theta}(x) = \int_Z q_{\phi}(z|x) \log p_{\theta}(x)$$

What do we want to optimize? What is the loss function?

$$\begin{aligned}\log p_\theta(x) &= \int_Z q_\phi(z|x) \log p_\theta(x) \\ &= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)}\end{aligned}$$

What do we want to optimize? What is the loss function?

$$\begin{aligned}\log p_\theta(x) &= \int_Z q_\phi(z|x) \log p_\theta(x) \\ &= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)} \\ &= \int_Z q_\phi(z|x) \log \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p_\theta(z|x)} \right)\end{aligned}$$

What do we want to optimize? What is the loss function?

$$\begin{aligned}\log p_\theta(x) &= \int_Z q_\phi(z|x) \log p_\theta(x) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)} \\&= \int_Z q_\phi(z|x) \log \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p_\theta(z|x)} \right) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} + \int_Z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)}\end{aligned}$$

What do we want to optimize? What is the loss function?

$$\begin{aligned}\log p_\theta(x) &= \int_Z q_\phi(z|x) \log p_\theta(x) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)} \\&= \int_Z q_\phi(z|x) \log \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p_\theta(z|x)} \right) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} + \int_Z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} \\&= \mathcal{L}(p_\theta, q_\phi) + D_{KL}(q_\phi, p_\theta)\end{aligned}$$

What do we want to optimize? What is the loss function?

$$\begin{aligned}\log p_\theta(x) &= \int_Z q_\phi(z|x) \log p_\theta(x) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)} \\&= \int_Z q_\phi(z|x) \log \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p_\theta(z|x)} \right) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} + \int_Z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} \\&= \mathcal{L}(p_\theta, q_\phi) + D_{KL}(q_\phi, p_\theta) \\&\geq \mathcal{L}(p_\theta, q_\phi)\end{aligned}$$

What do we want to optimize? What is the loss function?

$$\begin{aligned}\log p_\theta(x) &= \int_Z q_\phi(z|x) \log p_\theta(x) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)} \\&= \int_Z q_\phi(z|x) \log \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p_\theta(z|x)} \right) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} + \int_Z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} \\&= \mathcal{L}(p_\theta, q_\phi) + D_{KL}(q_\phi, p_\theta) \\&\geq \mathcal{L}(p_\theta, q_\phi)\end{aligned}$$

- Assume $q_\phi(z|x)$ can approximate $p_\theta(z|x)$ well

What do we want to optimize? What is the loss function?

$$\begin{aligned}\log p_\theta(x) &= \int_Z q_\phi(z|x) \log p_\theta(x) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{p_\theta(z|x)} \\&= \int_Z q_\phi(z|x) \log \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \cdot \frac{q_\phi(z|x)}{p_\theta(z|x)} \right) \\&= \int_Z q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)} + \int_Z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p_\theta(z|x)} \\&= \mathcal{L}(p_\theta, q_\phi) + D_{KL}(q_\phi, p_\theta) \\&\geq \mathcal{L}(p_\theta, q_\phi)\end{aligned}$$

- Assume $q_\phi(z|x)$ can approximate $p_\theta(z|x)$ well
- \mathcal{L} is often referred to as empirical lower bound (ELBO)

Approximate Inference:

$$\mathcal{L}(p_\theta, q_\phi) = \int_z q_\phi(z|x) \log \frac{p_\theta(x, z)}{q_\phi(z|x)}$$

Approximate Inference:

$$\begin{aligned}\mathcal{L}(p_\theta, q_\phi) &= \int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \\ &= \int_z q_\phi(z|x) \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)}\end{aligned}$$

Approximate Inference:

$$\begin{aligned}\mathcal{L}(p_\theta, q_\phi) &= \int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \\ &= \int_z q_\phi(z|x) \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \\ &= \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} + \int_z q_\phi(z|x) \log p_\theta(x|z)\end{aligned}$$

Approximate Inference:

$$\begin{aligned}\mathcal{L}(p_\theta, q_\phi) &= \int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \\&= \int_z q_\phi(z|x) \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \\&= \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} + \int_z q_\phi(z|x) \log p_\theta(x|z) \\&= -D_{KL}(q_\phi, p) + \mathbb{E}_{q_\phi}[\log p_\theta(x|z)]\end{aligned}$$

Approximate Inference:

$$\begin{aligned}\mathcal{L}(p_\theta, q_\phi) &= \int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \\&= \int_z q_\phi(z|x) \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \\&= \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} + \int_z q_\phi(z|x) \log p_\theta(x|z) \\&= -D_{KL}(q_\phi, p) + \mathbb{E}_{q_\phi}[\log p_\theta(x|z)]\end{aligned}$$

- Regularization $-D_{KL}(q_\phi, p)$ with prior $p(z)$ (Gaussian)

Approximate Inference:

$$\begin{aligned}\mathcal{L}(p_\theta, q_\phi) &= \int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} \\&= \int_z q_\phi(z|x) \log \frac{p_\theta(x|z)p(z)}{q_\phi(z|x)} \\&= \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)} + \int_z q_\phi(z|x) \log p_\theta(x|z) \\&= -D_{KL}(q_\phi, p) + \mathbb{E}_{q_\phi} [\log p_\theta(x|z)]\end{aligned}$$

- Regularization $-D_{KL}(q_\phi, p)$ with prior $p(z)$ (Gaussian)
- Reconstruction $\mathbb{E}_{q_\phi} [\log p_\theta(x|z)]$

Regularization:

$$-D_{KL}(q_\phi, p) = \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)}$$

Regularization:

$$-D_{KL}(q_\phi, p) = \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)}$$

- $p(z) = \mathcal{N}(z; 0, 1)$

Regularization:

$$-D_{KL}(q_\phi, p) = \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)}$$

- $p(z) = \mathcal{N}(z; 0, 1)$
- $q_\phi(z|x)$ is Gaussian with mean $\mu_\phi(x)$, variance $\sigma_\phi^2(x)$

Regularization:

$$-D_{KL}(q_\phi, p) = \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)}$$

- $p(z) = \mathcal{N}(z; 0, 1)$
- $q_\phi(z|x)$ is Gaussian with mean $\mu_\phi(x)$, variance $\sigma_\phi^2(x)$

$$-D_{KL}(q_\phi, p) =$$

Regularization:

$$-D_{KL}(q_\phi, p) = \int_z q_\phi(z|x) \log \frac{p(z)}{q_\phi(z|x)}$$

- $p(z) = \mathcal{N}(z; 0, 1)$
- $q_\phi(z|x)$ is Gaussian with mean $\mu_\phi(x)$, variance $\sigma_\phi^2(x)$

$$-D_{KL}(q_\phi, p) = \textbf{Homework}$$

Reconstruction:

$$\mathbb{E}_{q_\phi} [\log p_\theta(x|z)] = \int_z q_\phi(z|x) \log p_\theta(x|z)$$

Reconstruction:

$$\mathbb{E}_{q_\phi} [\log p_\theta(x|z)] = \int_z q_\phi(z|x) \log p_\theta(x|z)$$

Approximate $\mathbb{E}_{q_\phi} [\log p_\theta(x|z)]$ by sampling from $q_\phi(z|x)$

Reconstruction:

$$\mathbb{E}_{q_\phi} [\log p_\theta(x|z)] = \int_z q_\phi(z|x) \log p_\theta(x|z)$$

Approximate $\mathbb{E}_{q_\phi} [\log p_\theta(x|z)]$ by sampling from $q_\phi(z|x)$

$$\mathbb{E}_{q_\phi} [\log p_\theta(x|z)] \approx \frac{1}{N} \sum_{i=1}^N \log p_\theta(x|z^i) \quad \text{where} \quad z^i \sim \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x))$$

Variational auto-encoder loss function:

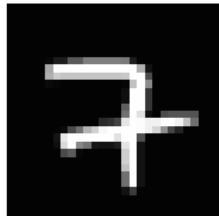
$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N} \sum_{i=1}^N \log p_\theta(x|z^i)$$

Intuitively:

Variational auto-encoder loss function:

$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N} \sum_{i=1}^N \log p_\theta(x|z^i)$$

Intuitively:



Variational auto-encoder loss function:

$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N} \sum_{i=1}^N \log p_\theta(x|z^i)$$

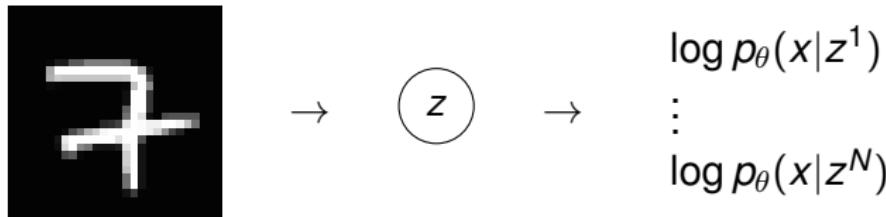
Intuitively:



Variational auto-encoder loss function:

$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N} \sum_{i=1}^N \log p_\theta(x|z^i)$$

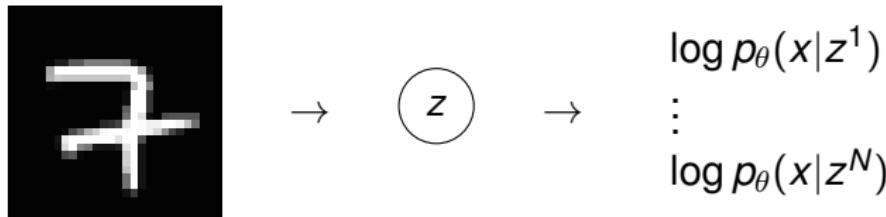
Intuitively:



Variational auto-encoder loss function:

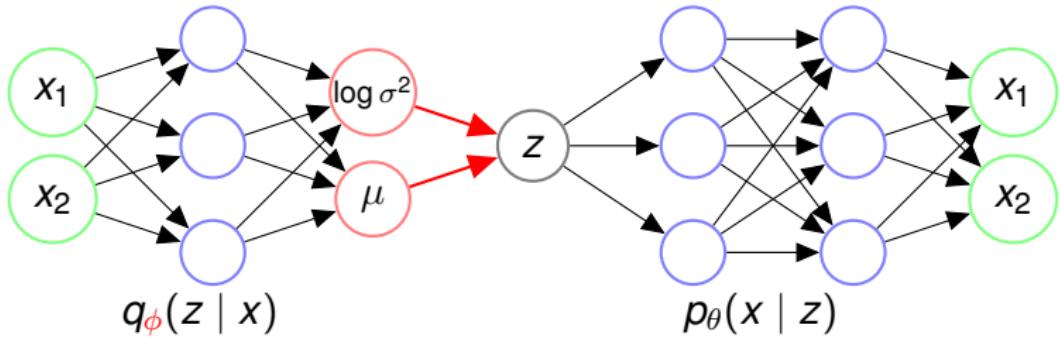
$$\mathcal{L}(p_\theta, q_\phi) \approx -D_{KL}(q_\phi, p) + \frac{1}{N} \sum_{i=1}^N \log p_\theta(x|z^i)$$

Intuitively:

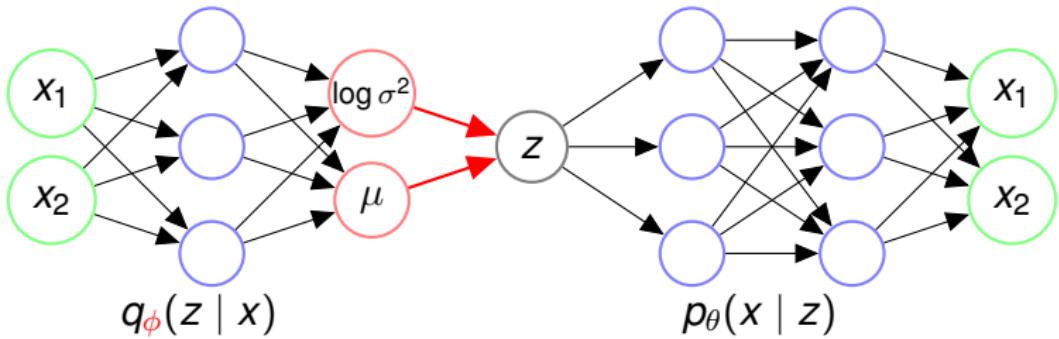


Typically N small, e.g., N = 1

Where is an issue?



Where is an issue?



How to backpropagate through the sampling step?

$$z \sim q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma_\phi(x))$$

Reparameterization trick:

$$z \sim q_{\phi}(z|x) = \mathcal{N}(z; \mu_{\phi}(x), \sigma_{\phi}(x))$$

is equivalent to:

Reparameterization trick:

$$z \sim q_{\phi}(z|x) = \mathcal{N}(z; \mu_{\phi}(x), \sigma_{\phi}(x))$$

is equivalent to:

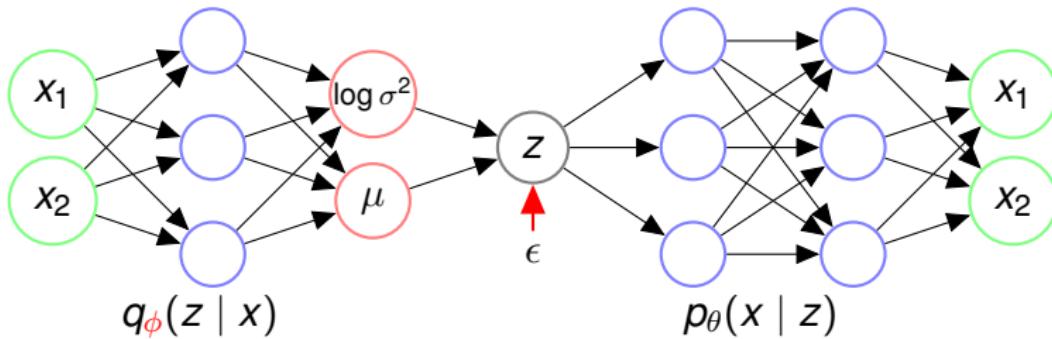
$$z = \mu_{\phi}(x) + \sigma_{\phi}(x) \cdot \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, 1)$$

Reparameterization trick:

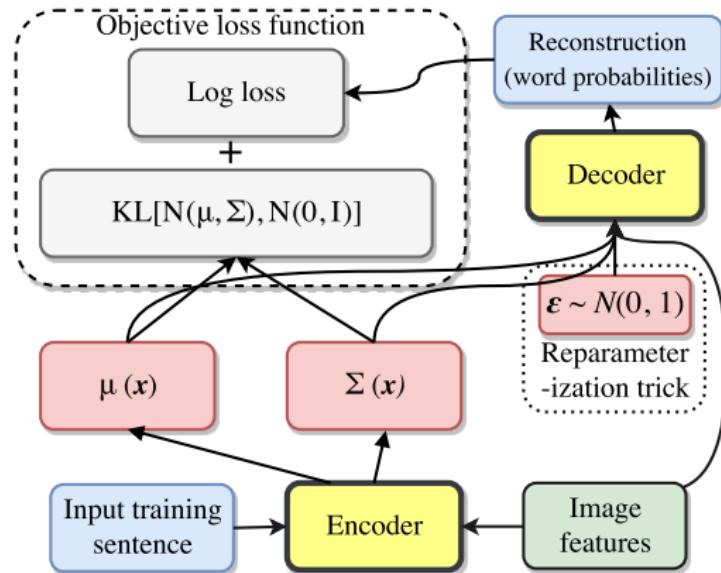
$$z \sim q_{\phi}(z|x) = \mathcal{N}(z; \mu_{\phi}(x), \sigma_{\phi}(x))$$

is equivalent to:

$$z = \mu_{\phi}(x) + \sigma_{\phi}(x) \cdot \epsilon \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, 1)$$



Variational auto-encoder flowchart:



Variational auto-encoder for MNIST digits:

6 5 / 7 8 / 4 8 2 8	5 1 6 5 7 0 7 6 7 +	2 8 3 8 3 8 5 9 3 8	8 2 0 8 0 0 2 3 2 0 0
9 6 0 3 9 6 0 3 1 9	8 5 9 4 6 8 2 1 6 2	8 3 8 2 7 9 3 3 3 8	7 5 1 9 1 1 7 1 4 4
3 3 7 1 3 6 8 1 7 9	6 1 5 3 2 8 8 1 3 3	8 5 9 9 6 7 8 5 1 1	8 9 6 2 0 3 2 8 2 9
8 9 0 8 6 9 1 9 6 3	2 1 6 8 9 1 0 0 4 1	1 9 8 8 9 8 3 1 9 7	2 4 8 6 3 1 7 0 6 1
9 2 3 3 3 1 3 8 6	5 1 9 1 0 1 5 3 5 9	2 7 3 6 4 3 0 2 0 3	5 4 7 9 1 9 9 9 1 5
6 9 9 8 6 1 6 6 6 6	6 5 6 1 4 9 1 7 5 8	5 9 7 0 5 8 2 2 4 5	6 9 8 4 9 0 8 8 2 8 1
9 5 2 6 6 5 1 8 9 9	1 3 4 3 9 1 3 4 7 0	6 9 4 3 6 2 8 5 5 2	7 5 8 2 9 6 1 3 8 8
9 9 8 1 3 1 2 8 2 3	4 5 8 2 9 7 0 1 5 9	5 4 9 0 5 0 7 0 5 5	7 9 3 9 2 2 9 3 5 6
0 4 6 1 2 3 2 0 8 5	6 9 4 4 2 7 2 3 9 5	7 4 5 6 2 0 3 6 0 1	4 5 2 4 3 9 0 1 8 4
9 7 5 4 9 3 4 8 5 1	2 6 4 5 6 0 9 7 9 8	2 1 2 0 1 7 1 8 5 0	2 8 7 2 5 1 6 2 3 6

(a) 2-D latent space

(b) 5-D latent space

(c) 10-D latent space

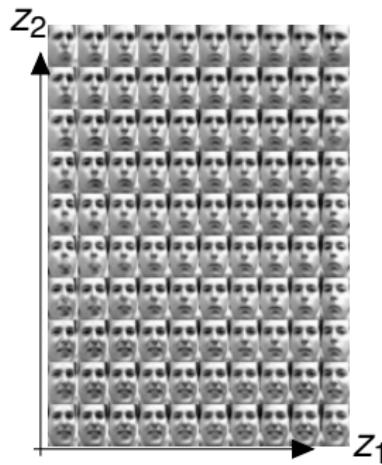
(d) 20-D latent space

Examples of low-dimensional manifolds: ($z \in \mathbb{R}^2$)

Kingma and Welling: “Auto-Encoding Variational Bayes” (ICLR 2014)

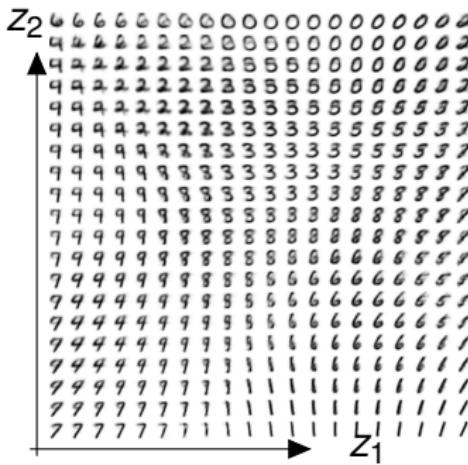
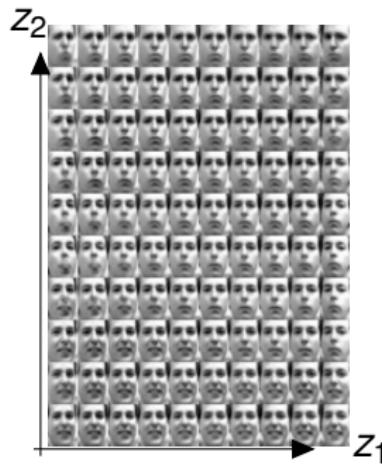
Examples of low-dimensional manifolds: ($z \in \mathbb{R}^2$)

Kingma and Welling: "Auto-Encoding Variational Bayes" (ICLR 2014)

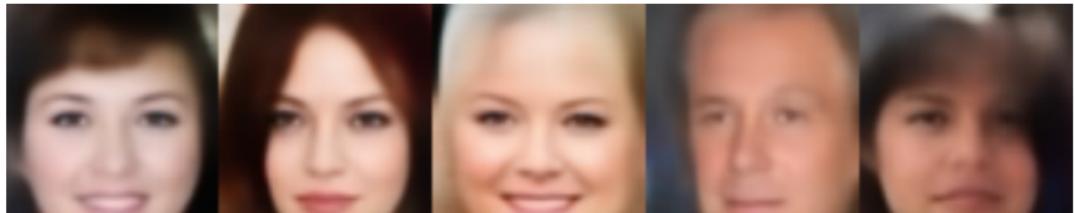


Examples of low-dimensional manifolds: ($z \in \mathbb{R}^2$)

Kingma and Welling: "Auto-Encoding Variational Bayes" (ICLR 2014)



Variational auto-encoder for face generation:



Variational auto-encoder for semi-supervised learning:

Maaloe et al. “Improving Semi-Supervised Learning with Auxiliary Deep Generative Models” (2015)

	100 labels
AtlasRBF (Pitelis et al., 2014)	8.10% (± 0.95)
Deep Generative Model (M1+M2) (Kingma et al., 2014)	3.33% (± 0.14)
Virtual Adversarial (Miyato et al., 2015)	2.12%
Ladder (Rasmus et al., 2015)	1.06% (± 0.37)
Auxiliary Deep Generative Model (1 MC)	2.25% (± 0.08)
Auxiliary Deep Generative Model (10 MC)	0.96% (± 0.02)

Quiz:

Quiz:

- What is the difference between generative and discriminative modeling?

Quiz:

- What is the difference between generative and discriminative modeling?
- What generative modeling techniques do you know about?

Quiz:

- What is the difference between generative and discriminative modeling?
- What generative modeling techniques do you know about?
- What are the approximations used in variational auto-encoders?

Quiz:

- What is the difference between generative and discriminative modeling?
- What generative modeling techniques do you know about?
- What are the approximations used in variational auto-encoders?
- Why are variational auto-encoder results smooth?

Important topics of this lecture

Important topics of this lecture

- Getting to know variational auto-encoders, a generative modeling technique

Important topics of this lecture

- Getting to know variational auto-encoders, a generative modeling technique
- Understanding the reasons for approximations

Important topics of this lecture

- Getting to know variational auto-encoders, a generative modeling technique
- Understanding the reasons for approximations

Next up:

Autoregressive Methods