CS 4803 / 7643: Deep Learning

Topics:

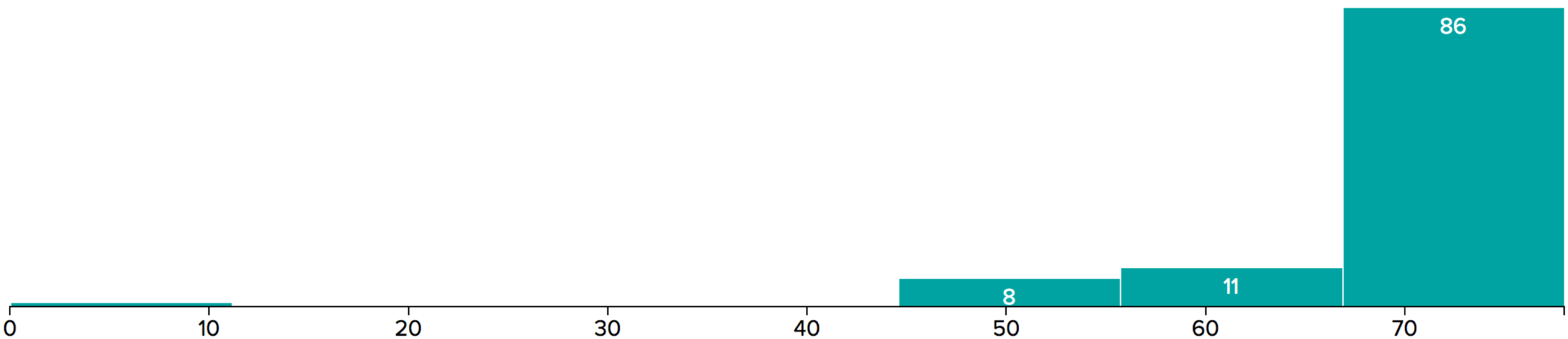
- Variational Auto-Encoders (VAEs)
- Key Ideas
  - AEs, Variational Inference, ELBO, Reparameterization

Dhruv Batra

Georgia Tech

Administrativia

- HW2 Grades Released
  - Max regular points: 54 (4803), 78 (7643)
  - Regrade requests close: 11/20, 11:55pm



MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
0.0	71.75	78.0	69.96	9.49

Recap from last time

Overview

- Unsupervised Learning
- Generative Models
  - PixelRNN and PixelCNN
  - Variational Autoencoders (VAE)
  - Generative Adversarial Networks (GAN)

Variational Autoencoders (VAE)

So far...

PixelCNNs define tractable density function, optimize likelihood of training data:

$$p_\theta(x) = \prod_{i=1}^{n} p_\theta(x_i | x_1, ..., x_{i-1})$$

$p(\vec{x})$

$p(\vec{x}, z) = p(\vec{x} | z)\, p(z)$

VAEs define intractable density function with latent **z**:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

$\sum p(x|z) p(z)$

Cannot optimize directly, derive and optimize lower bound on likelihood instead

$\boxed{\text{GMM}}$

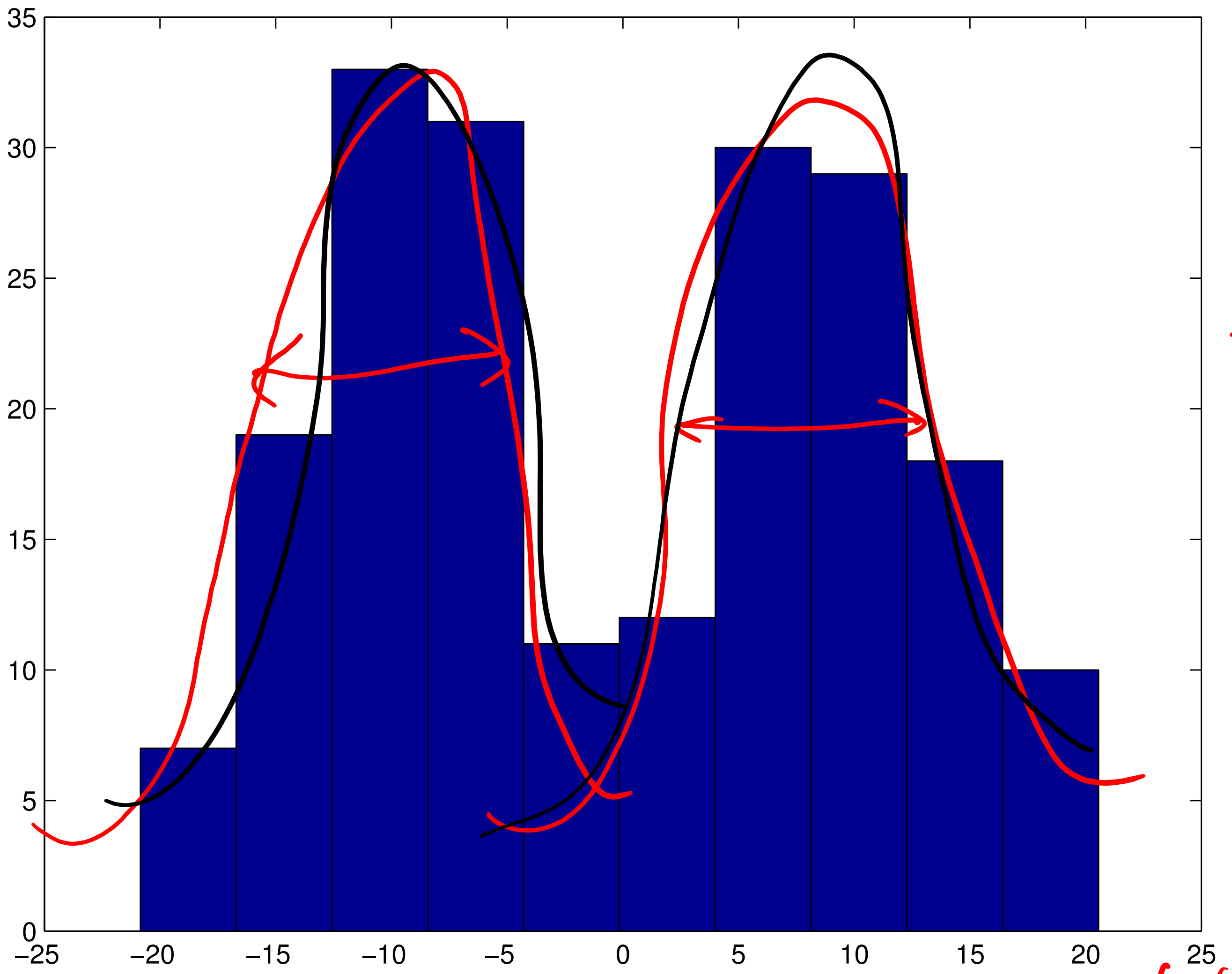
$p(x)$

$p(x)$

$p(x|z)$

$z \in \{1, 2\}$

$p(z) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$



$p(x|z=1) \sim N(\underline{\mu_1}, \underline{\sigma_1^2})$

$p(x|z=2) \sim N(\underline{\mu_2}, \underline{\sigma_2^2})$

Gaussian Mixture Model

$\rightarrow \underline{z} \sim \underline{\mathrm{Cat}}(\boxed{\vec{\pi}})$ $\begin{bmatrix} \pi_1 \\ \vdots \\ \pi_K \end{bmatrix}$ $\left\{ \begin{array}{l} \pi_c \geq 0 \\ \sum_{c=1}^{K} \pi_c = 1 \end{array} \right.$ $\pi_c = P(z=c)$

$\underline{X \mid \boxed{z=c}} \sim \underline{N(\boxed{\mu_c}, \boxed{\sigma_c^2})} = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(x-\mu_c)^2}{2\sigma_c^2}}$

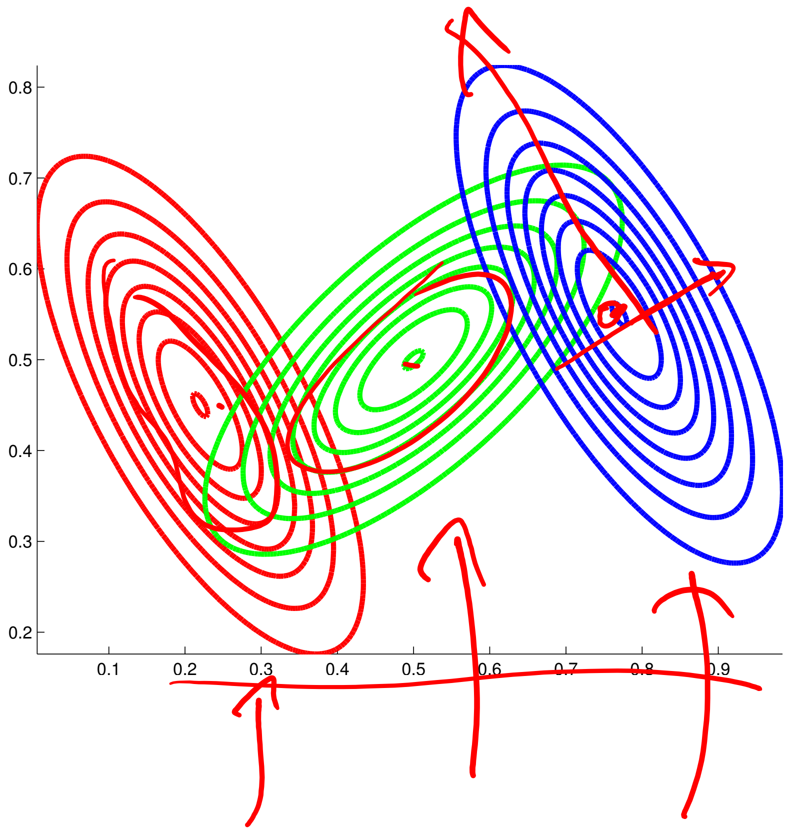
$\underbrace{p(z) \quad p(x \mid z)}_{\pi_c \cdot N(\mu_c, \sigma_c^2)} = p(x, z)$

GMM

$\vec{\mu} \in \mathbb{R}^d$

$\Sigma \in \mathbb{R}^{d \times d}$

$$N(\vec{\mu}, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} e^{-\frac{1}{2}(\vec{x}-\vec{\mu})^T \Sigma^{-1} (\vec{x}-\mu)}$$



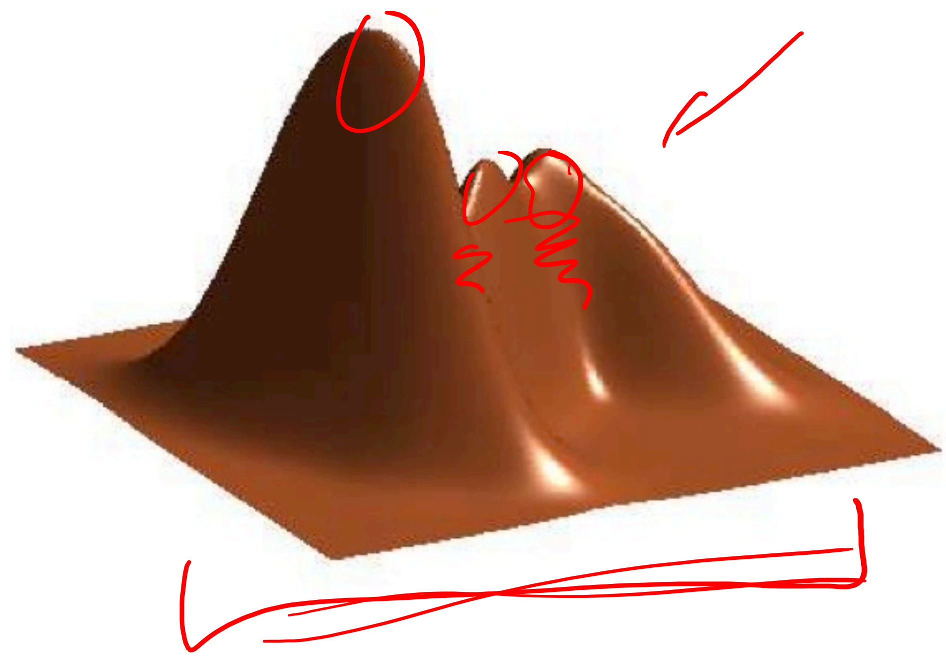


Figure Credit: Kevin Murphy

$p(z)$

Gaussian Mixture Model

$p(x, z)$

$p(x) = \sum_{c} p(z)\, p(x \mid z) \leftarrow$ Marginization

$p(z \mid x) = \dfrac{p(z, x)}{p(x)} = \dfrac{p(x \mid z)\, p(z)}{\sum_{z} (\;)(\;)}$ "Inference" Posterior

$\boxed{\sim p(x \mid z)}$

$\Rightarrow p($

Variational Auto Encoders

VAEs are a combination of the following ideas:

1. Auto Encoders

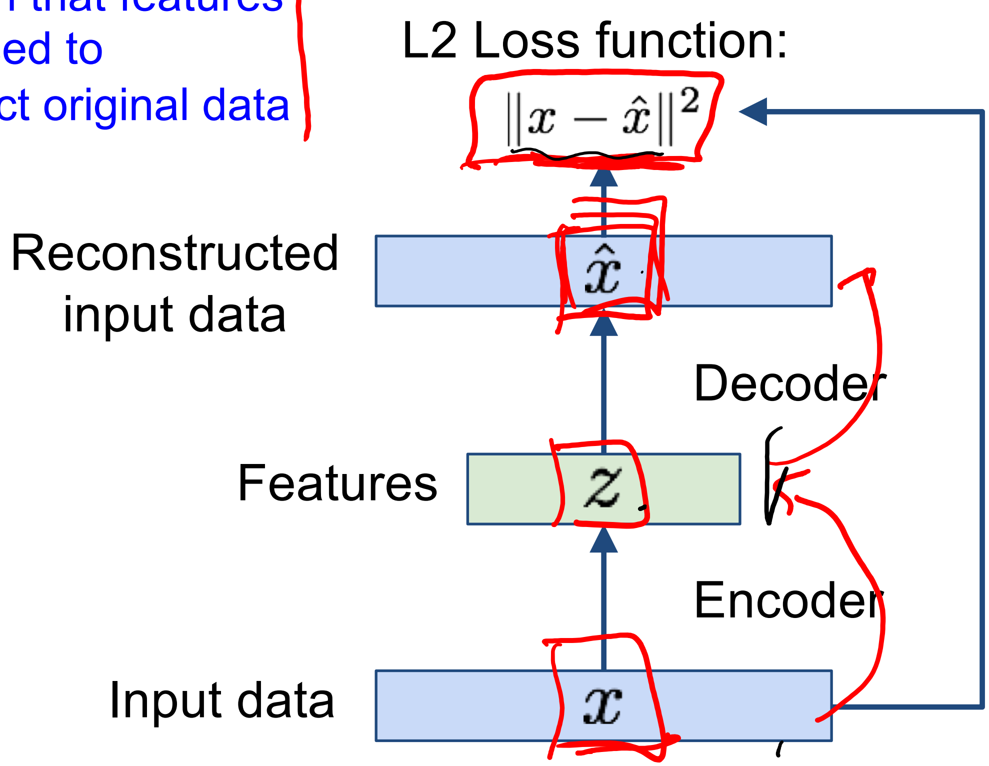
2. Variational Approximation
    - Variational Lower Bound / ELBO

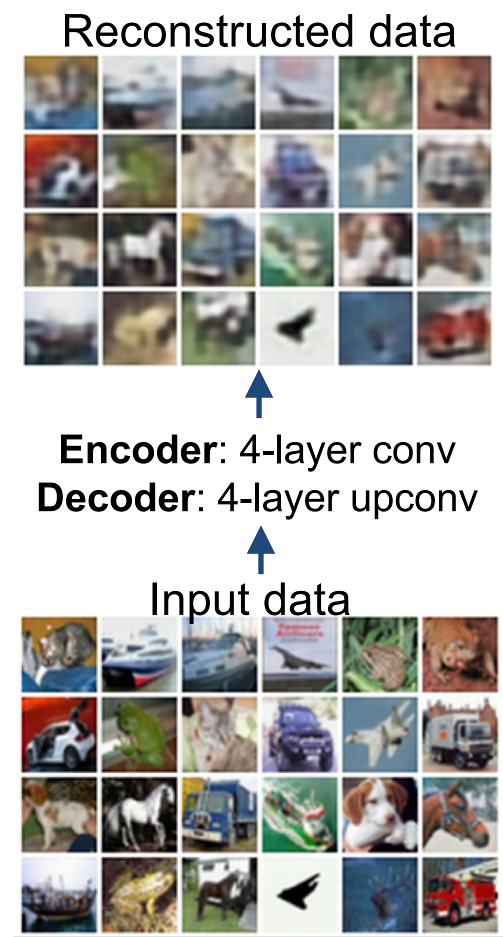
3. Amortized Inference Neural Networks

4. “Reparameterization” Trick

Autoencoders

Train such that features can be used to reconstruct original data





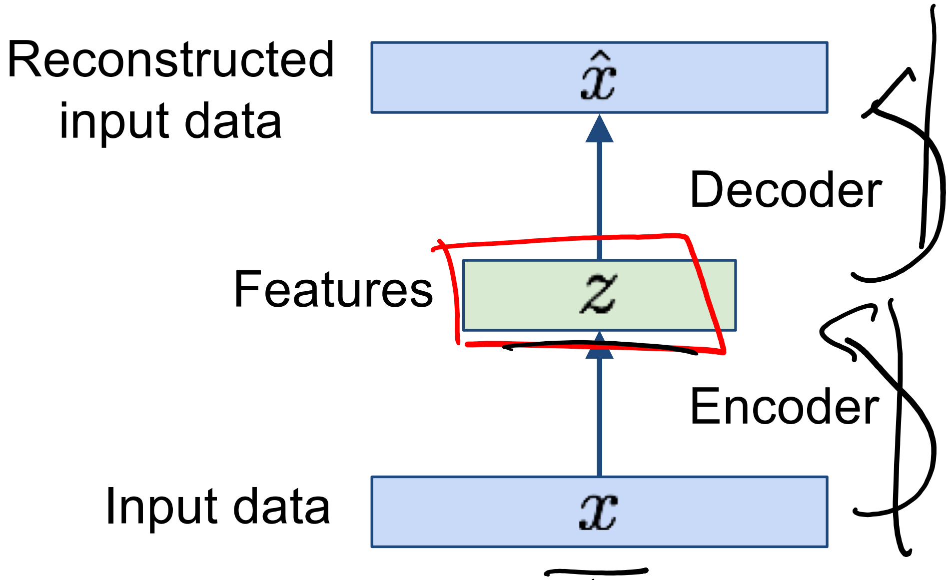
Autoencoders

$z = f_\phi(x)$

$\hat{x} = g_\theta(z)$

$p(z)$

$p(\hat{x} | z)$



Autoencoders can reconstruct data, and can learn features to initialize a supervised model

Features capture factors of variation in training data. Can we generate new images from an autoencoder?

Variational Autoencoders

Probabilistic spin on autoencoders - will let us sample from the model to generate data!

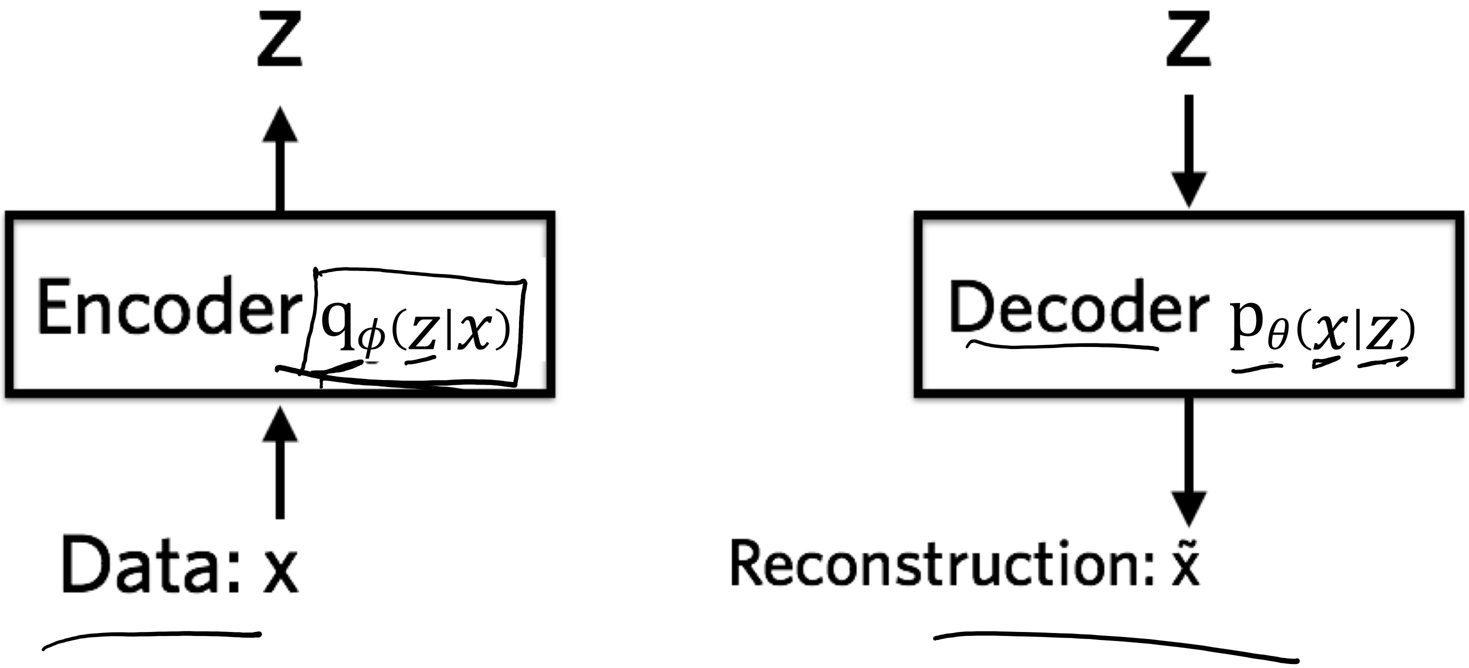


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

Variational Auto Encoders

VAEs are a combination of the following ideas:

1. Auto Encoders

2. Variational Approximation
    - Variational Lower Bound / ELBO

3. Amortized Inference Neural Networks

4. “Reparameterization” Trick

Key problem

- $\boxed{P(z|x)} = \frac{P(z,x)}{\boxed{P(x)}} = \frac{P(x|z)\,P(z)}{\boxed{\sum_z} P(x|z)P(z)}$

$\boxed{q(z)}$

What is Variational Inference?

- Key idea
  - Reality is complex
  - Can we approximate it with something "simple"?
  - Just need to make sure the simple thing is "close" to the complex thing.

Intuition

Model

$p(z)$

$\min_{q \in \mathcal{Q}}$

$\{q_\phi(z)\}_\phi$

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

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

$p(z) = 0$

error

$$KL(p(z) \| q(z)) = \sum_z p(z) \log \frac{p(z)}{q(z)}$$

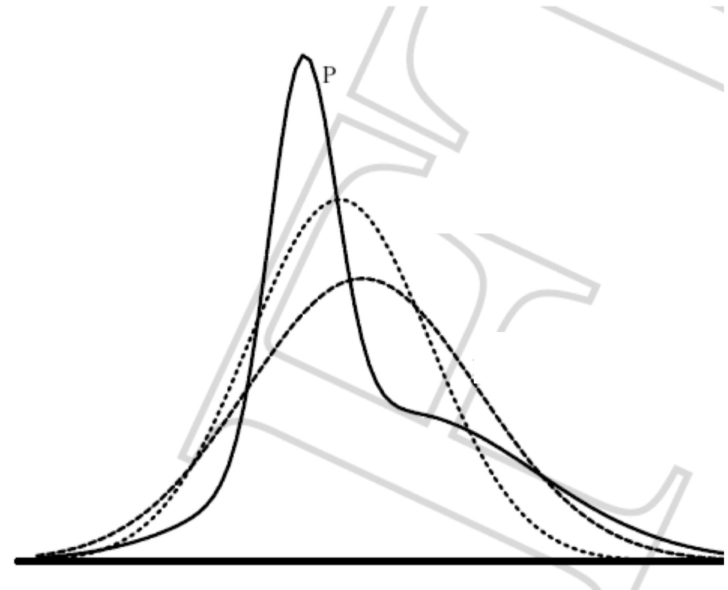
$$-\sum_{\text{sample}} \log q(z)$$

Find simple approximate distribution

- Suppose *p* is intractable posterior
- Want to find simple *q* that approximates *p*
- KL divergence not symmetric

- D(p||q)
  - true distribution p defines support of diff.
  - the "correct" direction
  - will be intractable to compute

- D(q||p)
  - approximate distribution defines support
  - tends to give overconfident results
  - will be tractable



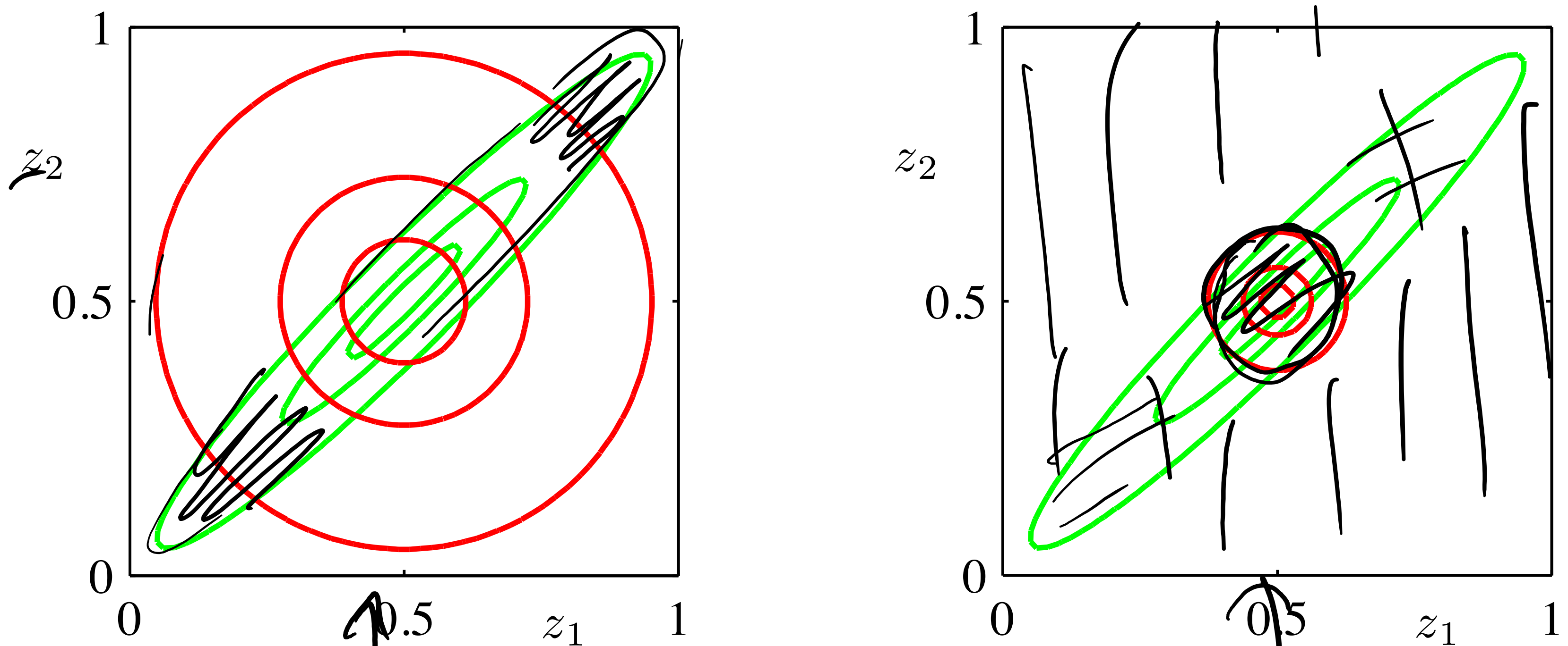
Example 1

$\Sigma_p = \begin{bmatrix} & \end{bmatrix}_{2\times 2}$

- p = 2D Gaussian with arbitrary co-variance
- q = 2D Gaussian with diagonal co-variance

$\Sigma_q = \begin{bmatrix} & 0 \\ 0 & \end{bmatrix}$

argmin_q KL (p || q) argmin_q KL (q || p)



(b) (a)

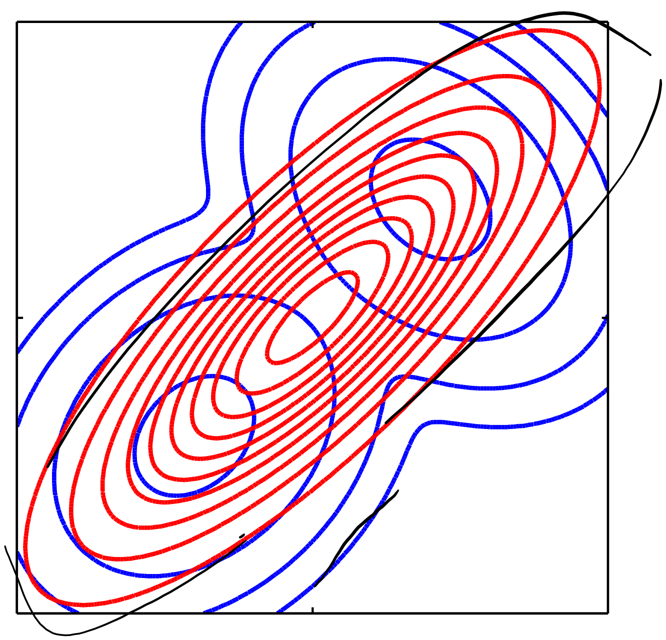
p = Green; q = Red

Example 2

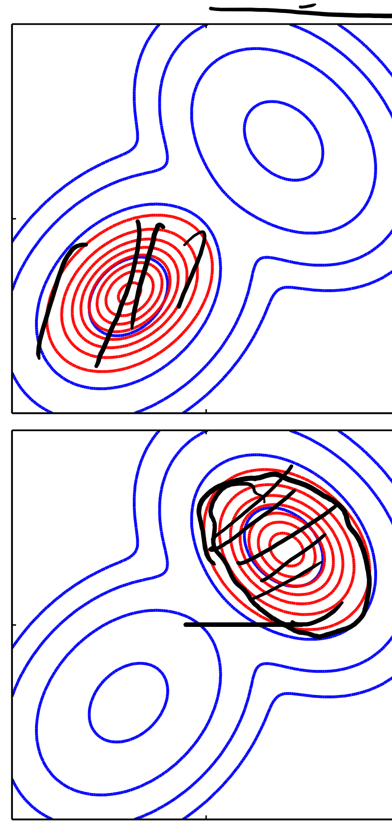
- p = Mixture of Two Gaussians
- q = Single Gaussian

$\sum q(z)$

argmin_q KL (p || q)



argmin_q KL (q || p)



p = Blue; q = Red

Plan for Today

- VAEs
  - Variational Inference → Evidence Based Lower Bound
  - Reparameterization trick
  - Putting it all together

- Generative Adversarial Networks

The general learning problem with missing data

- Marginal likelihood – **x** is observed, **z** is missing:

$$\boxed{ll(\theta : \mathcal{D})} = \log \boxed{\prod_{i=1}^{N}} P(\mathbf{x}_i \mid \theta)$$

$$= \sum_{i=1}^{N} \log P(\mathbf{x}_i \mid \theta)$$

$P(\vec{x}_i, z)$

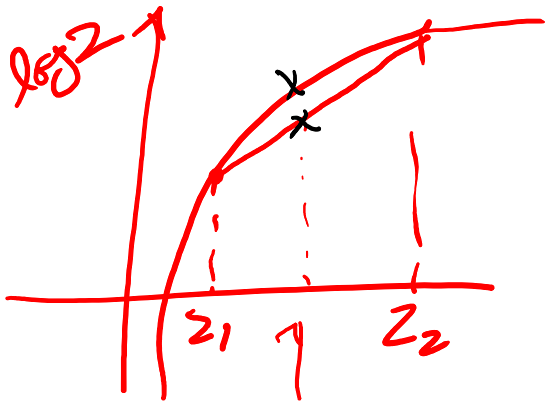
$$= \cancel{\sum_{i=1}^{N}} \log \boxed{\sum_{\mathbf{z}}} P(\mathbf{x}_i, \mathbf{z} \mid \theta)$$

$$= \log \left[\sum_{z} P(\vec{x}_i \mid \theta) \boxed{P(z \mid \vec{x}_i, \theta)} \right]$$

$$= \log \left[\mathbb{E}_{P(z \mid x_i, \theta)} \left[P(\vec{x}_i \mid \theta) \right] \right]$$

Applying Jensen's inequality

- Use: $\log \sum_{\mathbf{z}} P(\mathbf{z})\, g(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log g(\mathbf{z})$



$\lambda z_1 + \lambda_2 z_2$

$\lambda_1, \lambda_2 \geq 0$

$\lambda_1 + \lambda_2 = 1$

$$f(\lambda_1 z_1 + \lambda_2 z_2) \geq \lambda_1 f(z_1) + \lambda_2 f(z_2)$$

$$f\left(\sum_{i=1}^{K} \boxed{\lambda_i} z_i\right) \geq \sum_{i=1}^{K} \boxed{\lambda_i} f(z_i)$$

$$f(E[z]) \geq E[f(z)]$$

$z \rightarrow g(z)$

$$f(E[g(z)]) \geq E[f(g(z))]$$

Applying Jensen's inequality

- Use: $\log \sum_{\mathbf{z}} P(\mathbf{z})\, g(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log g(\mathbf{z})$

$$\rightarrow \log \sum_{z} P(\vec{x}_i, z \mid \theta) \cdot \frac{Q_i(z)}{Q_i(z)}$$

$$\geq \sum_{z} Q_i(z) \log \frac{P(\vec{x}_i, z \mid \theta)}{Q_i(z)}$$

$F(\theta, Q_i)$

"Free Energy"

Variational

VLB / ELBO

Applying Jensen's inequality

- Use: $\log \sum_{\mathbf{z}} P(\mathbf{z})\, g(\mathbf{z}) \geq \sum_{\mathbf{z}} P(\mathbf{z}) \log g(\mathbf{z})$

$P(z|\theta)P(x_i|z,\theta)$

$$\underline{ll(\theta : \mathcal{D})} = \cancel{\sum_{i=1}^{N}} \log \sum_{\mathbf{z}} Q_i(\mathbf{z}) \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}$$

$P(x_i|\theta)\, P(z|x_i,\theta)$

Evidence Lower Bound

- Define potential function F(θ,Q):

$$\underline{ll(\theta : \mathcal{D})} \geq \underline{F(\theta, Q_i)} = \underline{\sum_{i=1}^{N} \sum_{\mathbf{z}} Q_i(\mathbf{z}) \log \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}}$$

ELBO: Factorization #1

$$\to P(\vec{x}_i \mid \theta)\, P(z \mid \vec{x}_i, \theta)$$

$$ll(\theta : \mathcal{D}) \geq F(\theta, Q_i) = \cancel{\sum_{i=1}^{N}} \sum_{\mathbf{z}} Q_i(\mathbf{z}) \log \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}$$

$$\sum_z Q_i(z) \log \left(\frac{\boxed{P(\vec{x}_i \mid \theta)}\, P(z \mid \vec{x}_i, \theta)}{Q_i(z)} \right)$$

$$= \underbrace{\sum_z Q_i(z)}_{1} \underbrace{\log P(\vec{x}_i \mid \theta)} + \left[\sum_z Q_i(z) \log \frac{P(z \mid \vec{x}_i, \theta)}{Q_i(z)} \right]$$

$$F(\theta, Q_i) = \log P(\vec{x}_i \mid \theta) - \left[\underbrace{KL\left(Q_i(z) \,\|\, P(z \mid x_i, \theta) \right)}_{\geq 0} \right]$$

$$\max_{\theta, Q_i} F = ll - KL$$

ELBO: Factorization #~~1~~ 2

$$\max_{\theta, Q_i} F(\theta, Q_i)$$

$$\boxed{ll(\theta : \mathcal{D})} \geq F(\theta, Q_i) = \cancel{\sum_{i=1}^{N}} \sum_{\mathbf{z}} Q_i(\mathbf{z}) \log \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}$$

$P(z \mid \theta) P(x_i \mid z, \theta)$

$$= \sum_z Q_i(z) \log \frac{P(z \mid \theta) P(\vec{x}_i \mid z, \theta)}{Q_i(z)}$$

$$= \sum_z Q_i(z) \log P(\vec{x}_i \mid z, \theta) + \left[\sum_z Q_i(z) \log \frac{P(z \mid \theta)}{Q_i(z)} \right]$$

decoder

$$= E_{Q_i(z)} \left[\log P(\vec{x}_i \mid z, \theta) \right] - KL\left(Q_i(z) \,||\, P(z \mid \theta) \right)$$

encoder

Explain the data

Regularizer

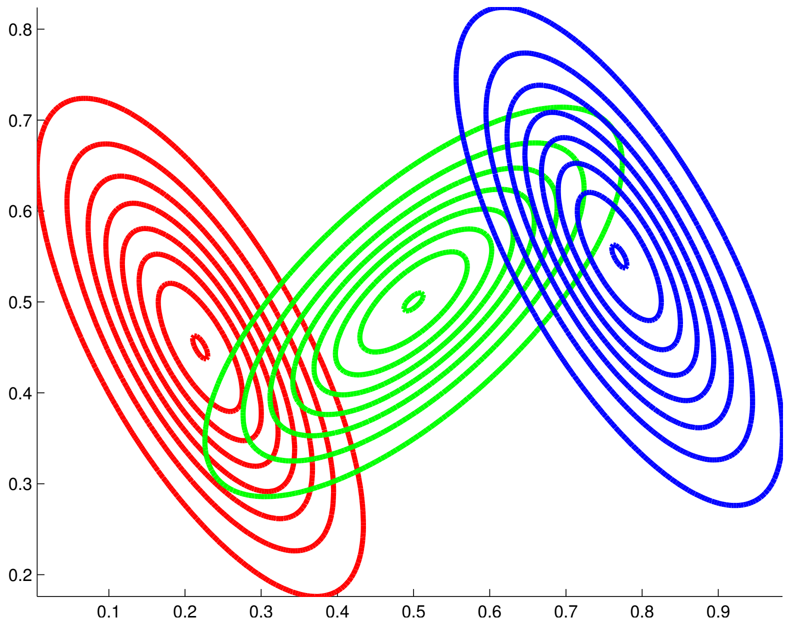
Evidence Lower Bound

- Define potential function F(θ,Q):

$$ll(\theta : \mathcal{D}) \geq \boxed{F(\theta, Q_i) = \sum_{i=1}^{N} \sum_{\mathbf{z}} Q_i(\mathbf{z}) \log \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}}$$

- EM corresponds to coordinate ascent on F
  - Thus, maximizes lower bound on marginal log likelihood

GMM



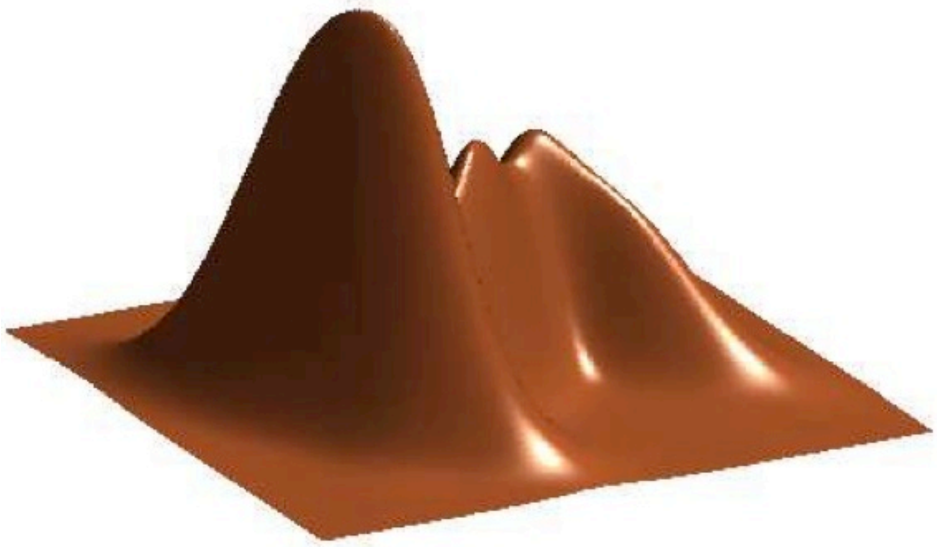


Figure Credit: Kevin Murphy

EM for Learning GMMs

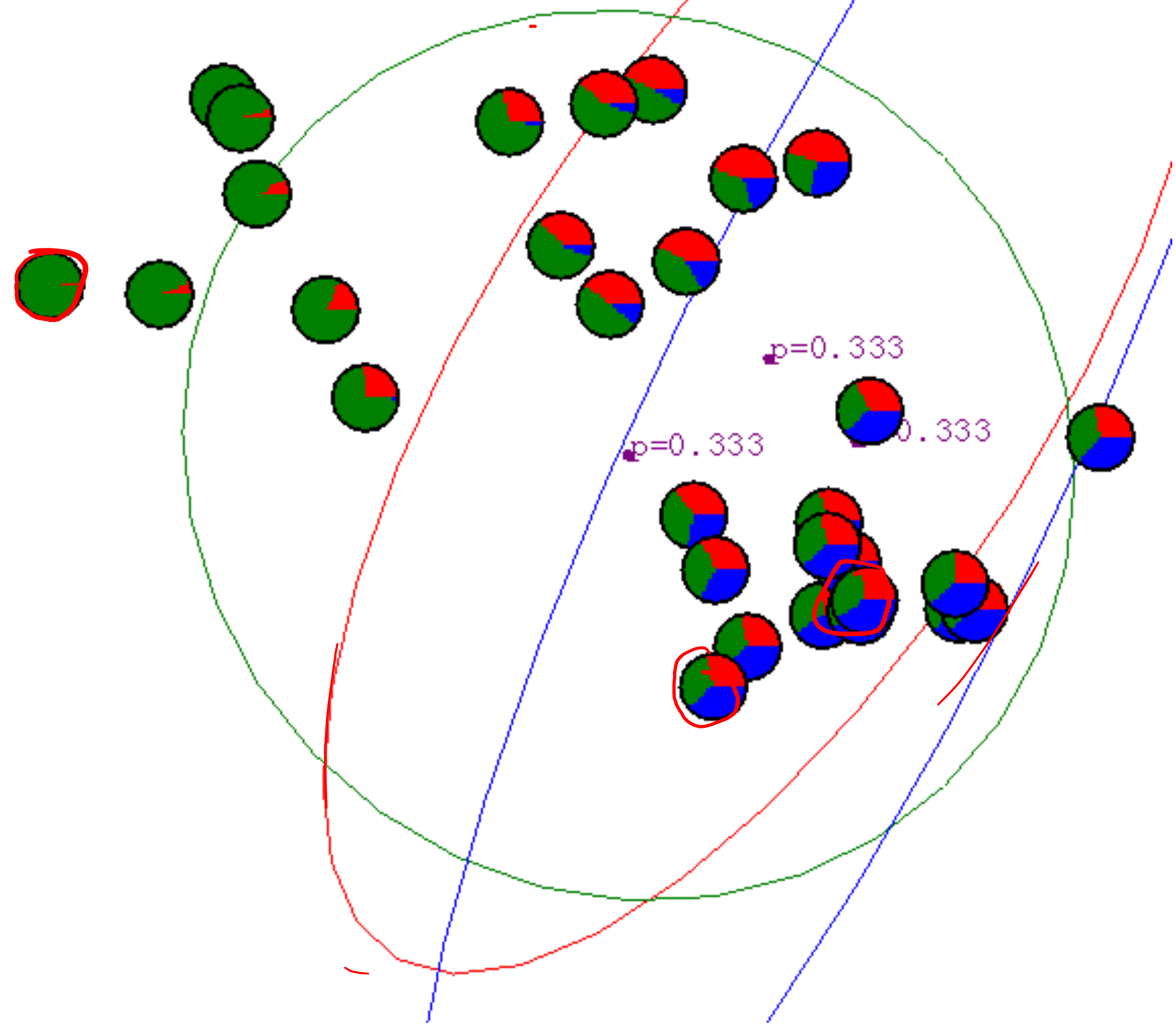
- Simple Update Rules
  - E-Step: estimate $Q_i(z) = \Pr(z = j \mid x_i)$
  - M-Step: maximize expected likelihood under $Q_i(z)$

Gaussian Mixture Example: Start

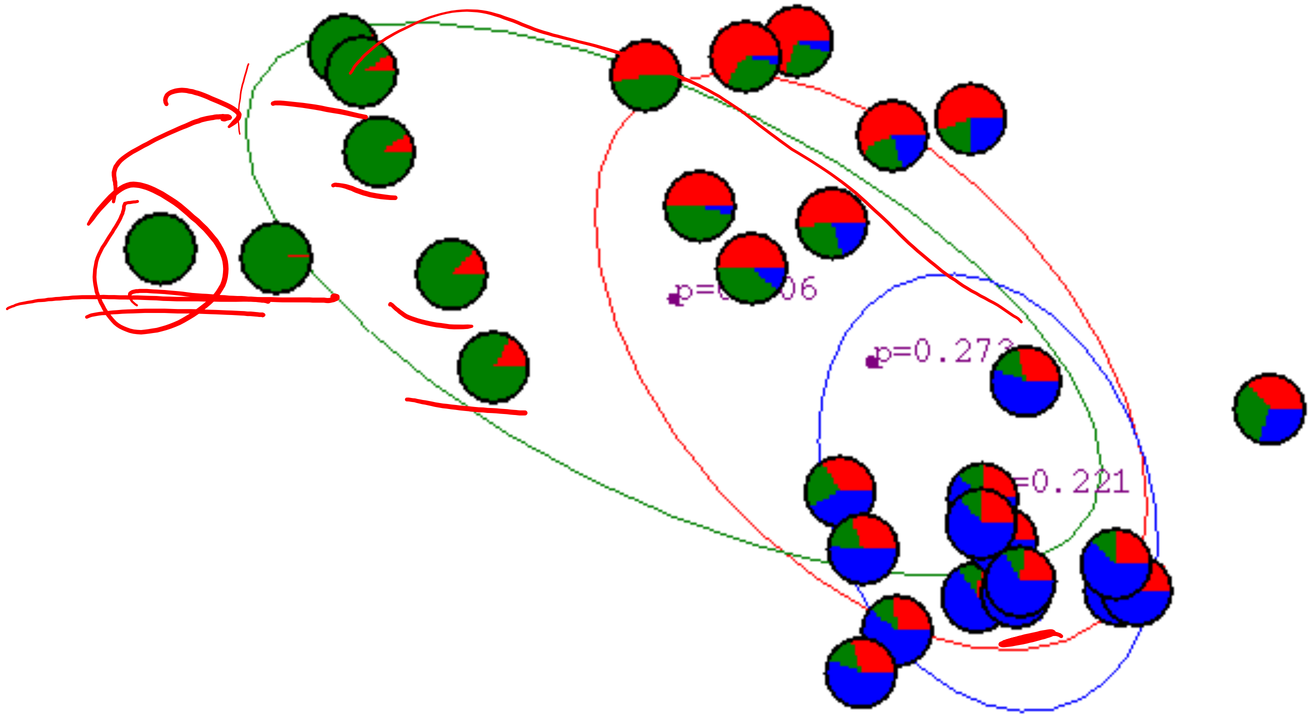
$\boxed{\theta = \{\mu_c, \Sigma_c, \pi_c\}}$

$\boxed{Q_i(z)} = P(z \mid x_i)$

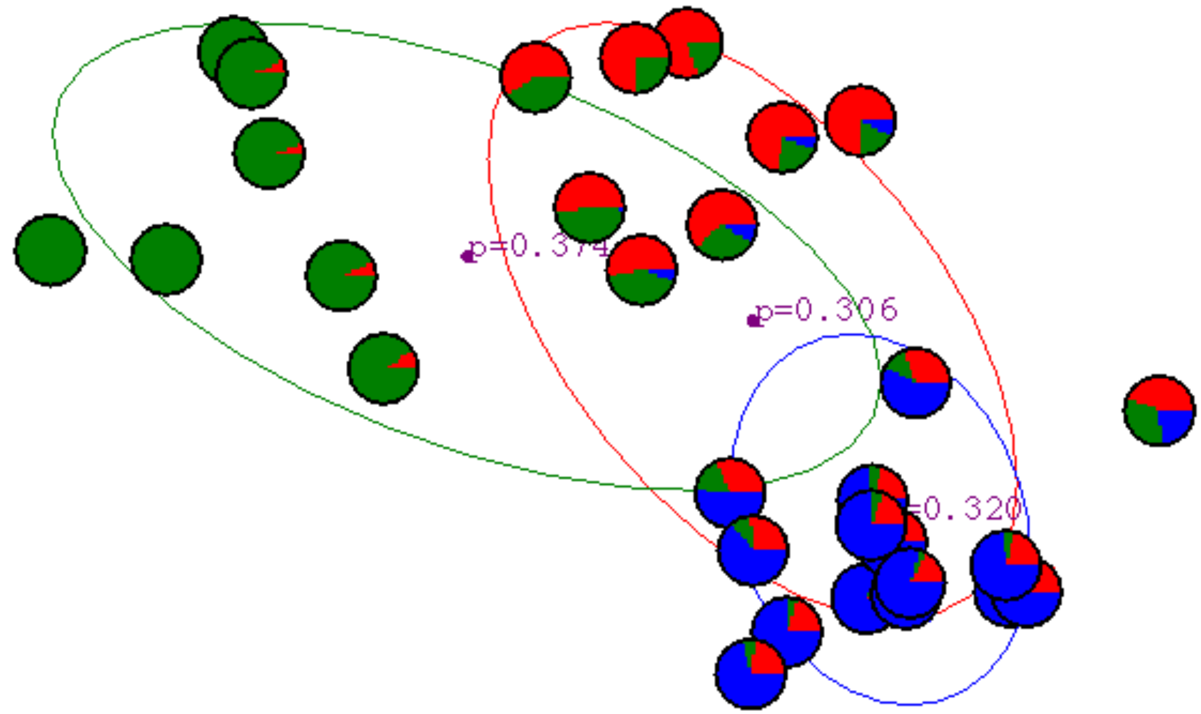
$\begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}$



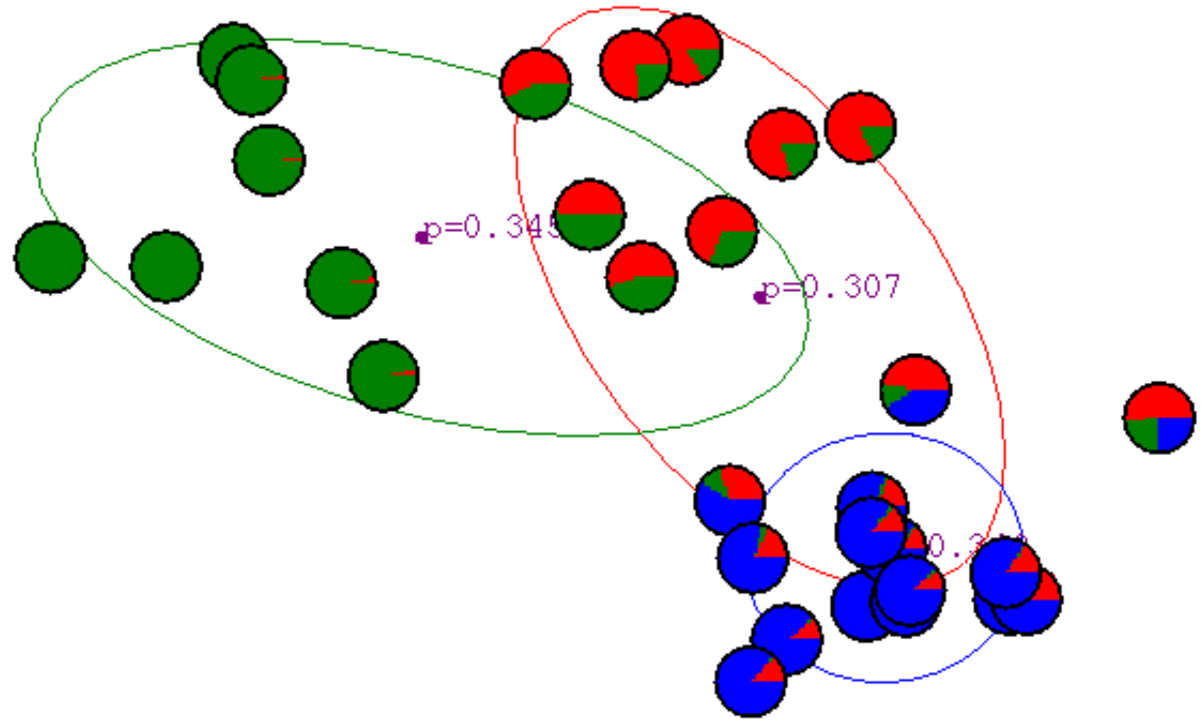
After 1st iteration



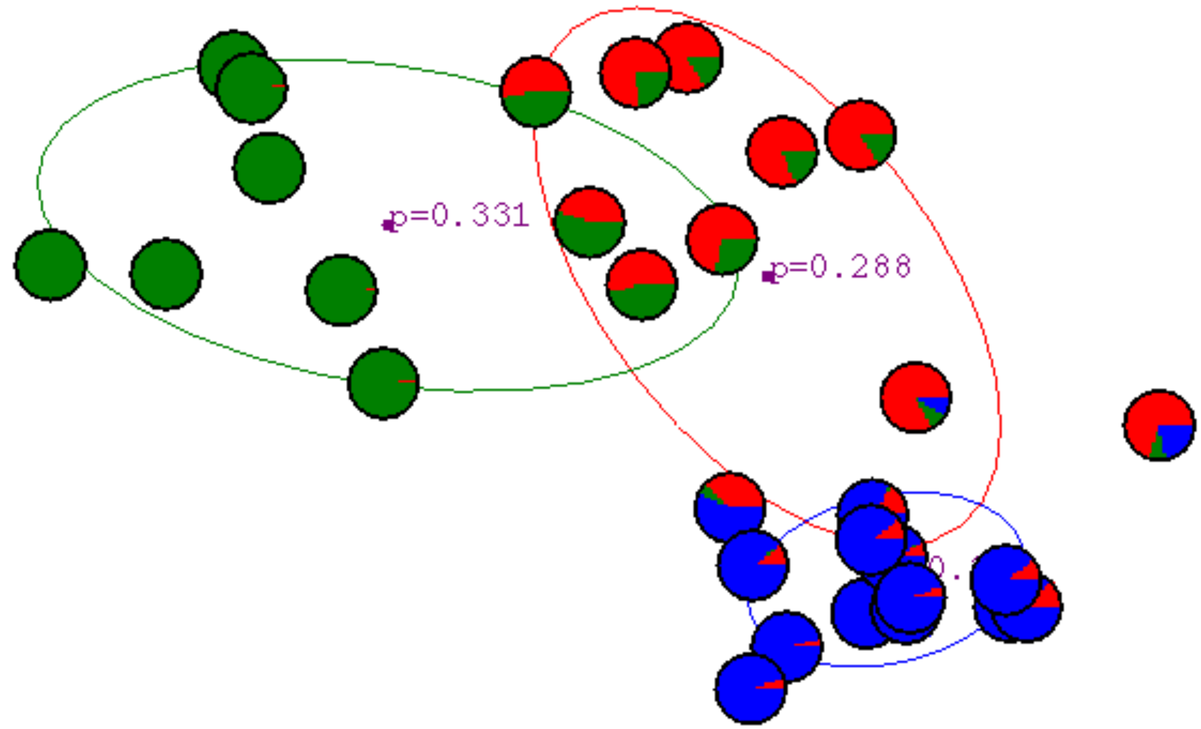
After 2nd iteration



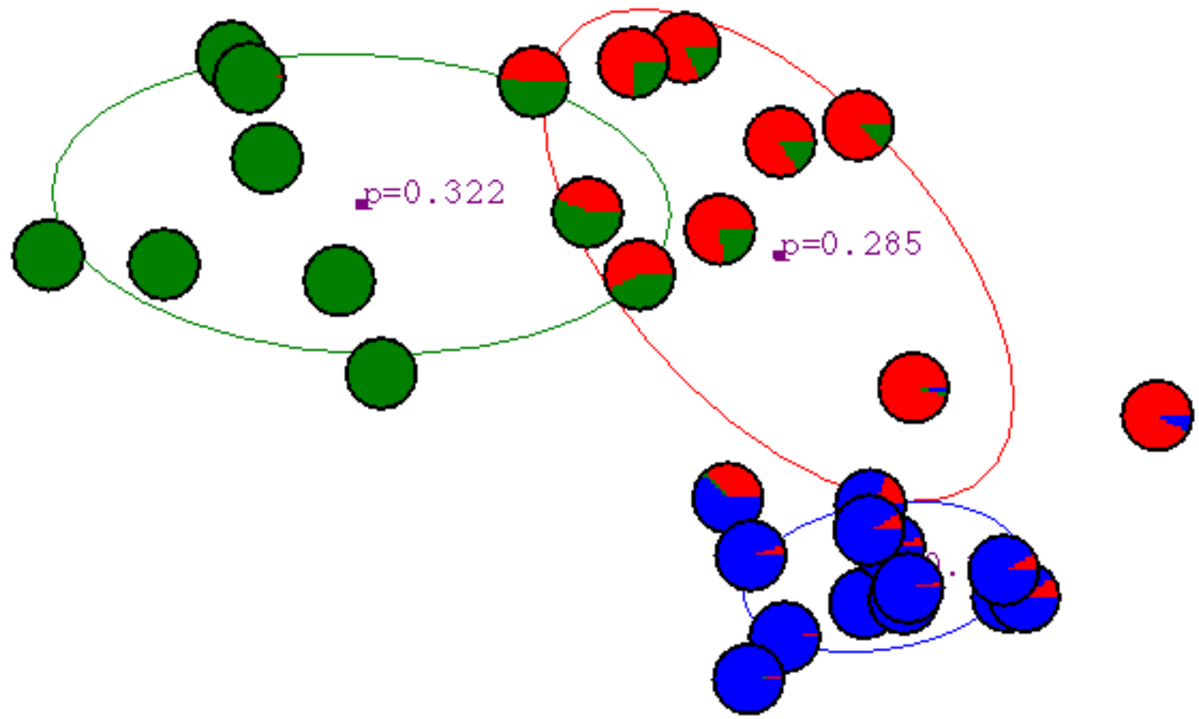
After 3rd iteration



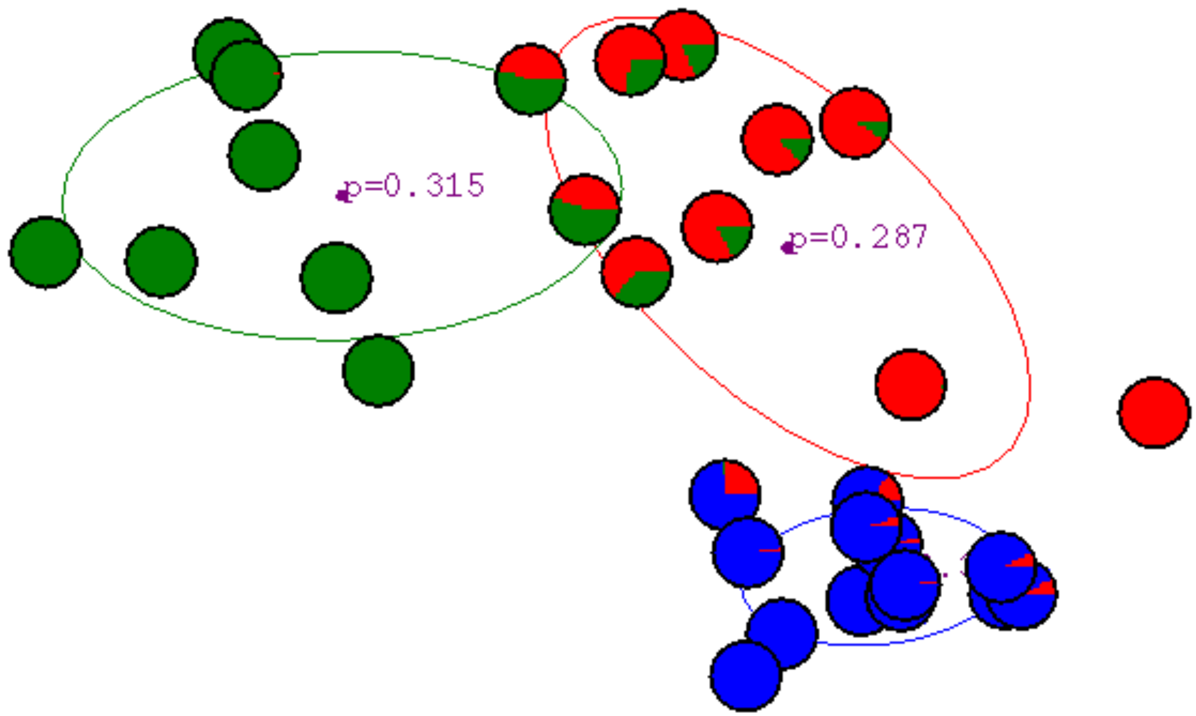
After 4th iteration



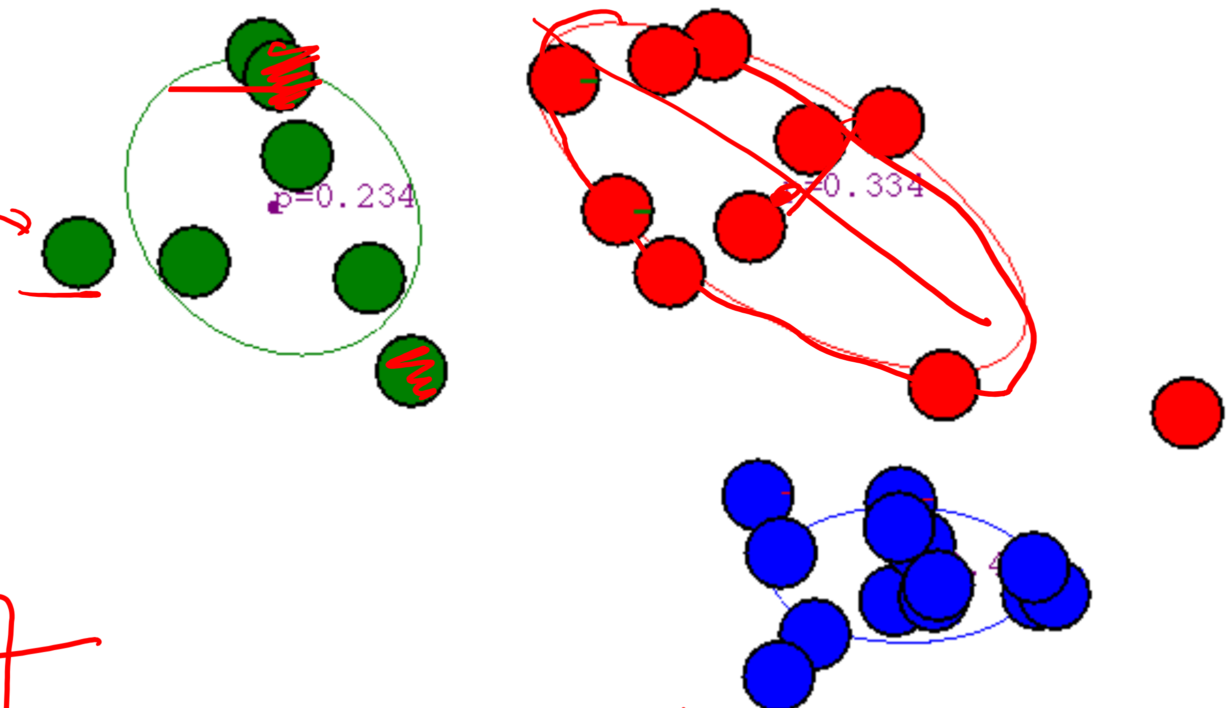
After 5th iteration



After 6th iteration



After 20th iteration



$i \rightarrow$

$$Q_i = \begin{bmatrix} \cancel{0} \\ \cancel{1} \\ 0 \end{bmatrix} \begin{matrix} R \\ G \\ B \end{matrix}$$

$Q_i(z)$

NN

softmax

$x_i \rightarrow$

$Q_i(z)$

3×1

Variational Autoencoders

Probabilistic spin on autoencoders - will let us sample from the model to generate data!

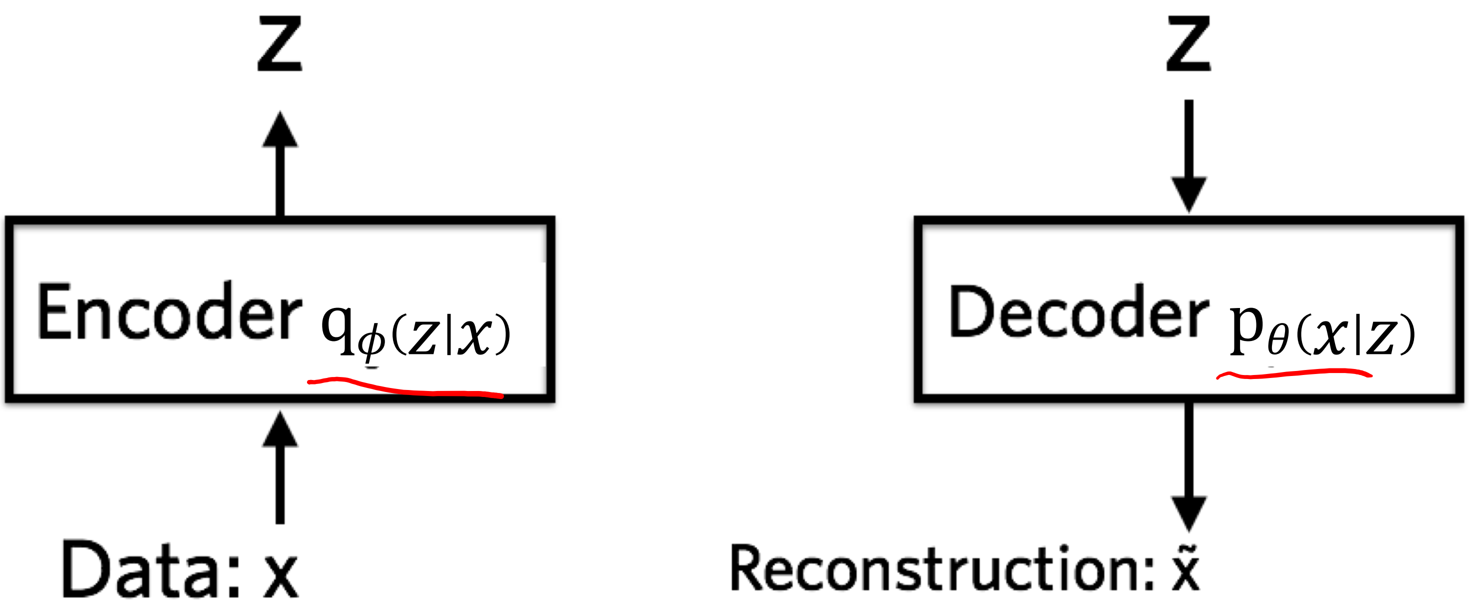


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

Evidence Lower Bound

- Define potential function F(θ,Q):

$$ll(\theta : \mathcal{D}) \geq F(\theta, Q_i) = \sum_{i=1}^{N} \sum_{\mathbf{z}} Q_i(\mathbf{z}) \log \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}$$

Variational Auto Encoders

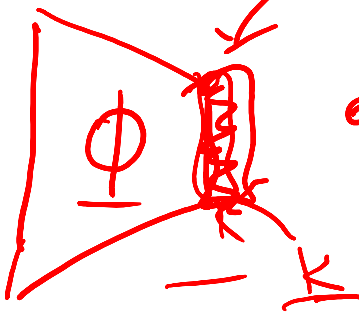
VAEs are a combination of the following ideas:

1. Auto Encoders
2. Variational Approximation
   - Variational Lower Bound / ELBO
3. Amortized Inference Neural Networks
4. "Reparameterization" Trick

Amortized Inference Neural Networks

softmax

x_i



$q_\phi(z|x_i)$ Encoder

k

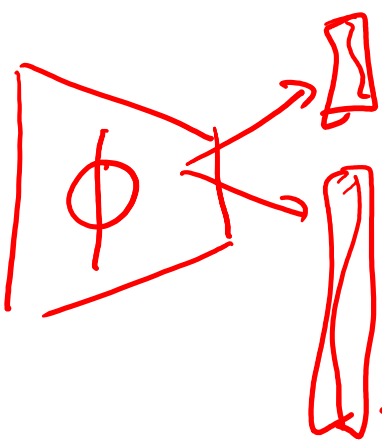
$z \sim Cat(\_)$

$z \in \mathbb{R}^k$

$z|x_i$ $\quad \cancel{z} \sim \mathcal{N}(\mu_{z|x_i}, \Sigma_{z|x_i})$

mean

$x|z \sim \mathcal{N}(.)$ $\quad x_i$



$\mu_{z|x_i}$

covariance

$\Sigma(:)\ z|x_i$

Amortized Inference Neural Networks

Variational Autoencoders

Probabilistic spin on autoencoders - will let us sample from the model to generate data!

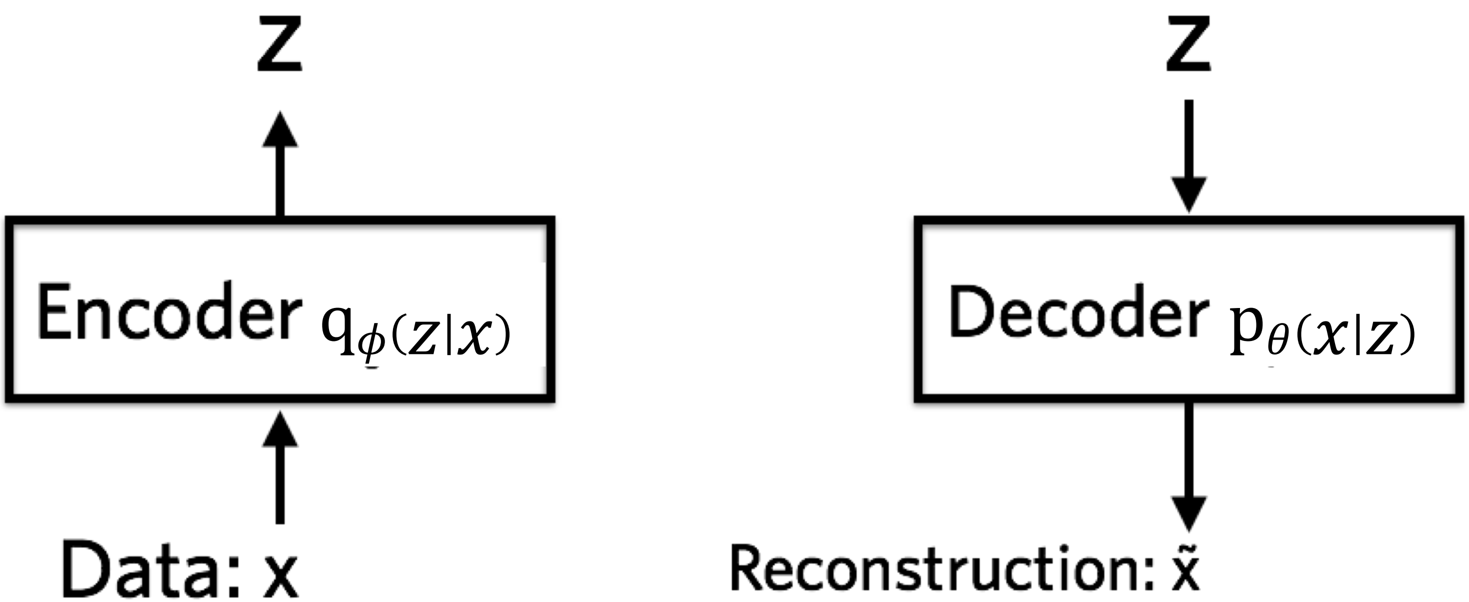


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

Variational Auto Encoders

VAEs are a combination of the following ideas:

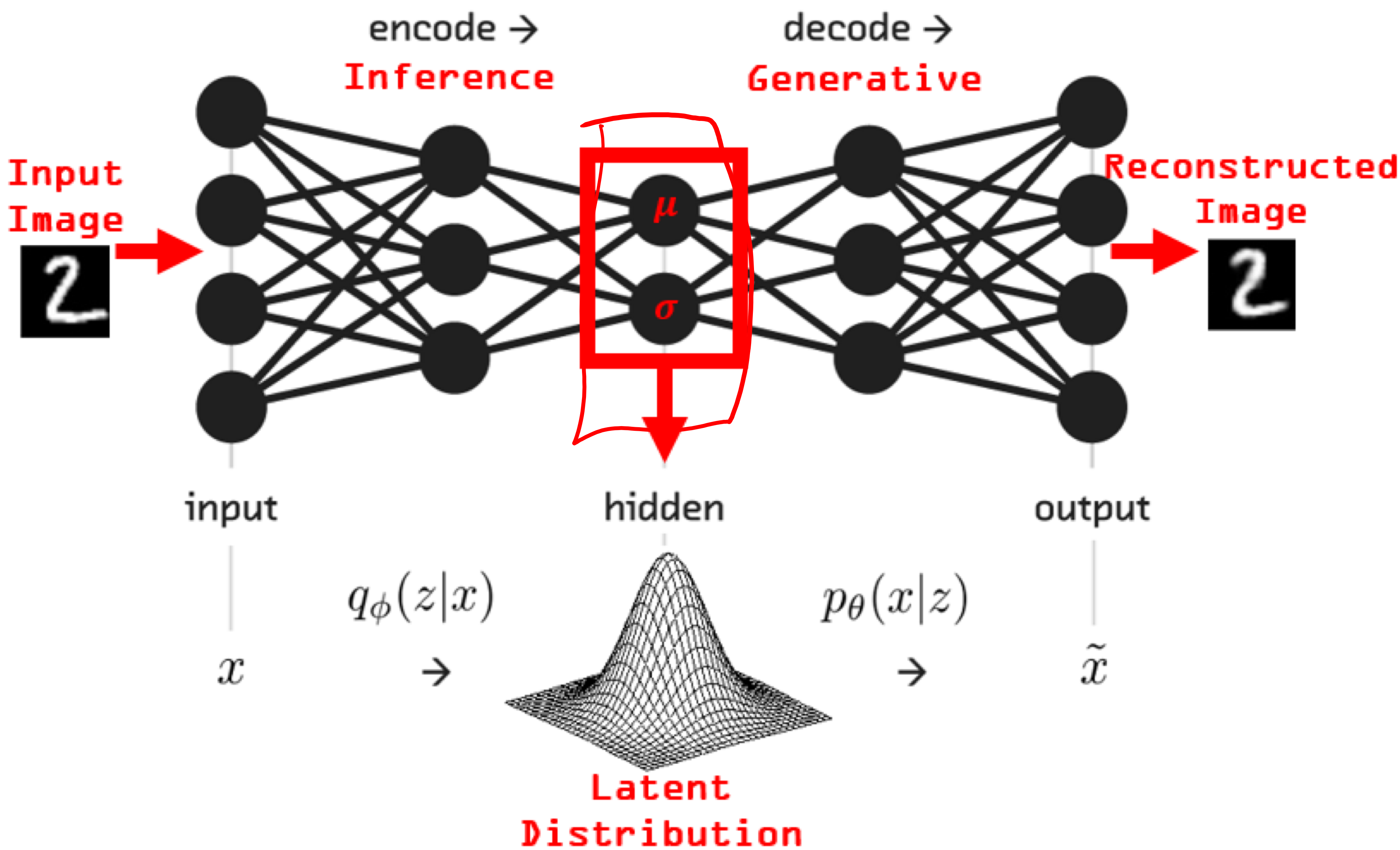
1. Auto Encoders

2. Variational Approximation
   - Variational Lower Bound / ELBO

3. Amortized Inference Neural Networks

4. "Reparameterization" Trick

VAEs



encode →
Inference
decode →
Generative
Input Image
Reconstructed Image
μ
σ
input
hidden
output
$q_\phi(z|x)$
$p_\theta(x|z)$
x
→
→
$\tilde{x}$
Latent Distribution

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

Let's look at computing the bound (forward pass) for a given minibatch of input data

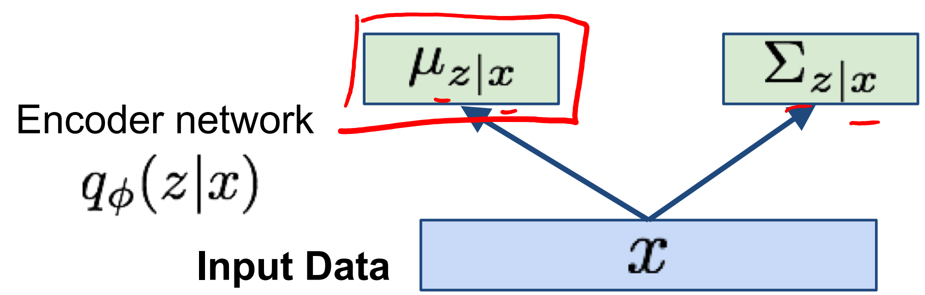
Input Data x

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

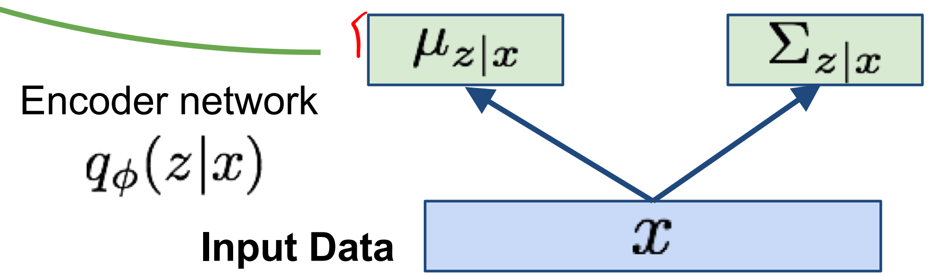


Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \boxed{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Make approximate posterior distribution close to prior

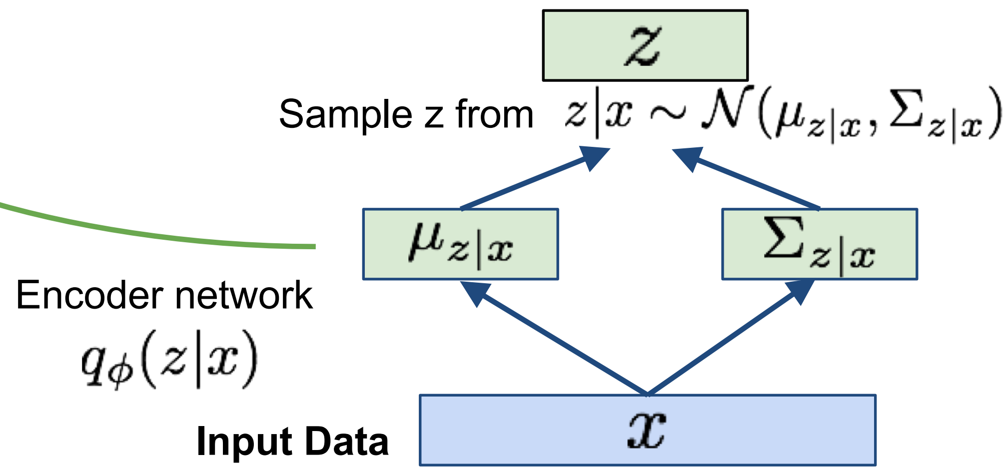


Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

Make approximate posterior distribution close to prior

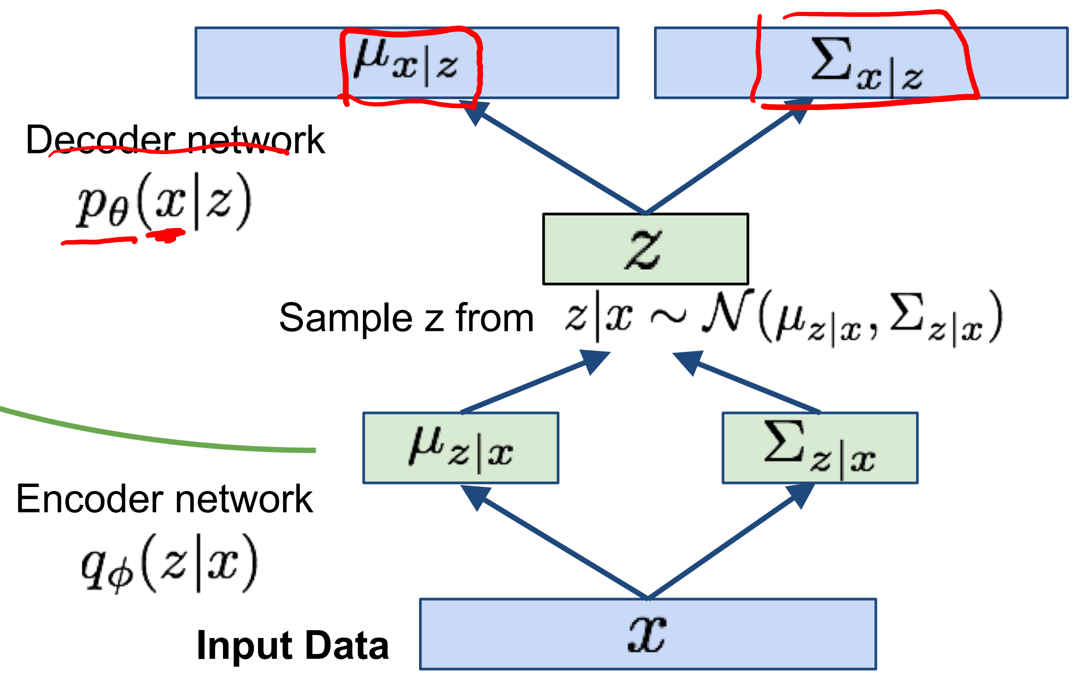


Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z \left[\log p_\theta(x^{(i)} \mid z) \right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Make approximate posterior distribution close to prior



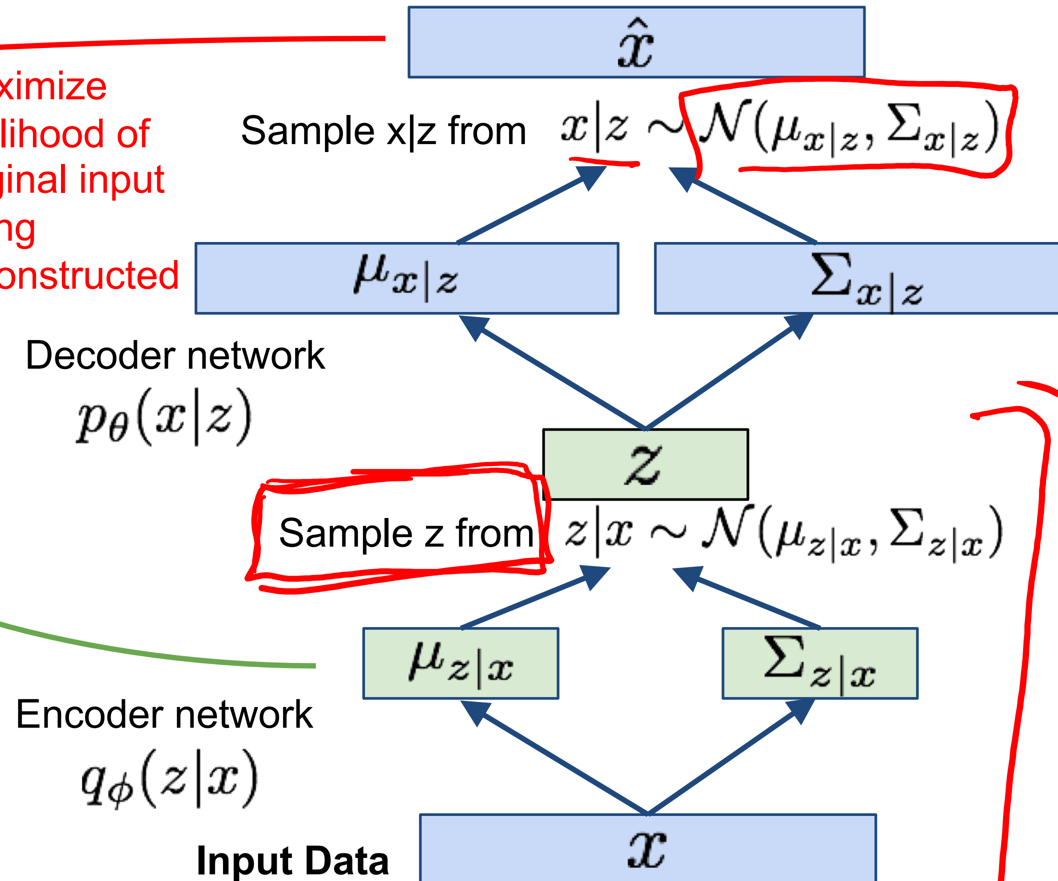
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\boxed{\log p_\theta(x^{(i)} \mid z)}\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)}$$

Maximize likelihood of original input being reconstructed

Make approximate posterior distribution close to prior



Variational Auto Encoders

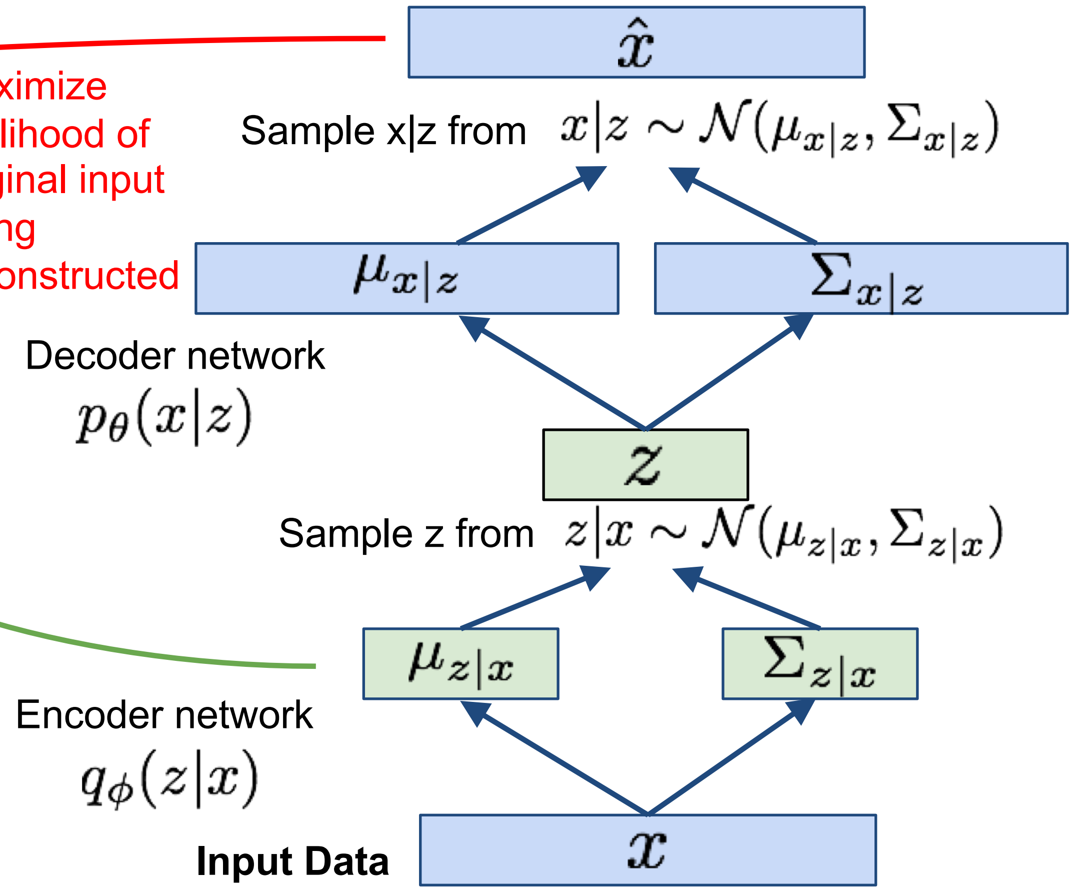
Putting it all together: maximizing the likelihood lower bound

$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Maximize likelihood of original input being reconstructed

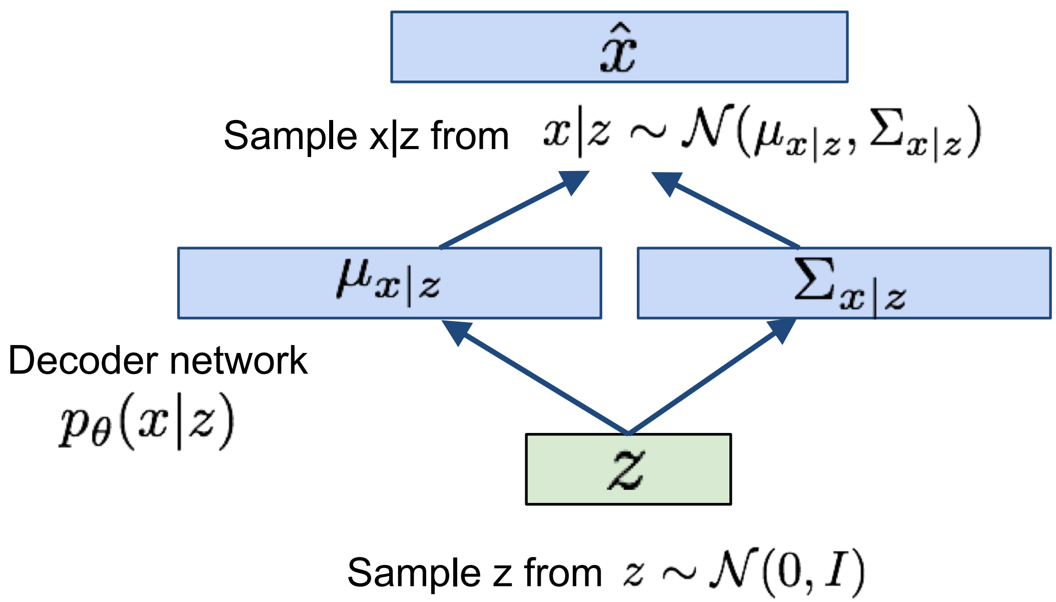
Make approximate posterior distribution close to prior

For every minibatch of input data: compute this forward pass, and then backprop!



Variational Auto Encoders: Generating Data

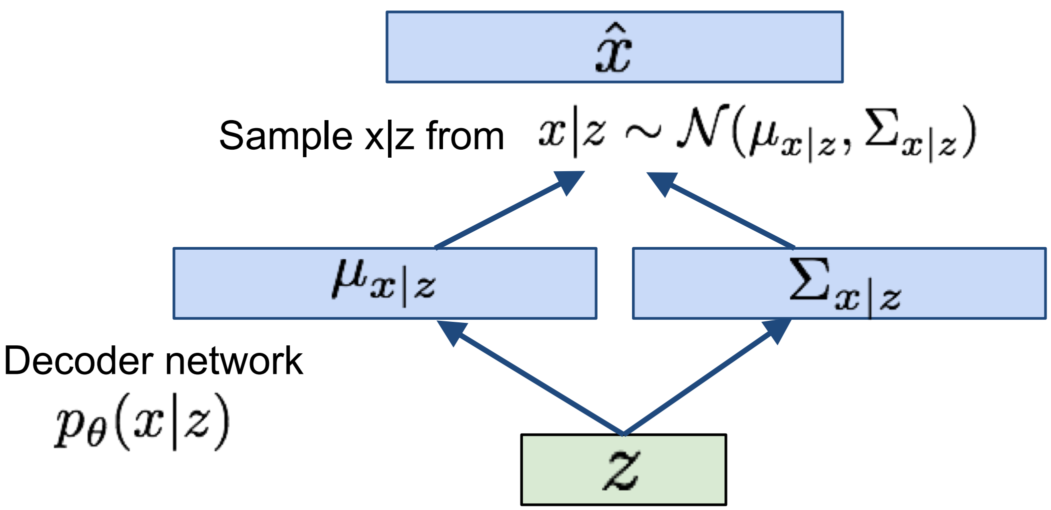
Use decoder network. Now sample z from prior!



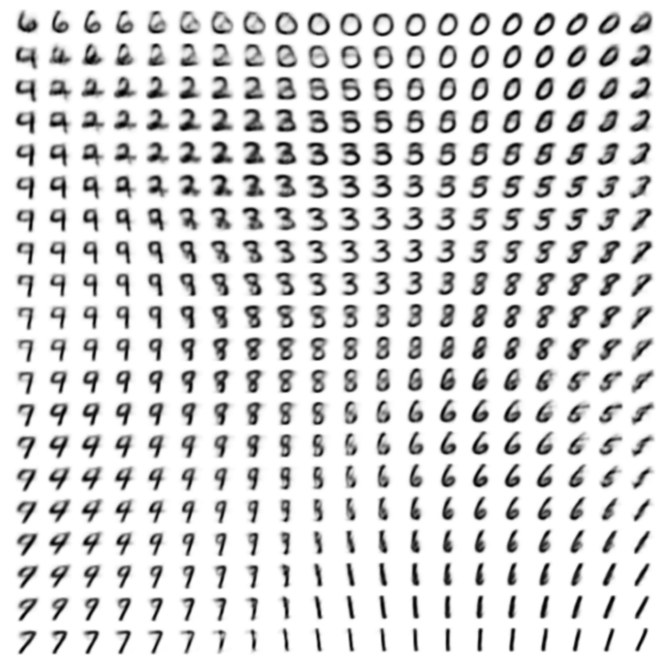
Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Variational Auto Encoders: Generating Data

Use decoder network. Now sample z from prior!



Sample z from $z \sim \mathcal{N}(0, I)$



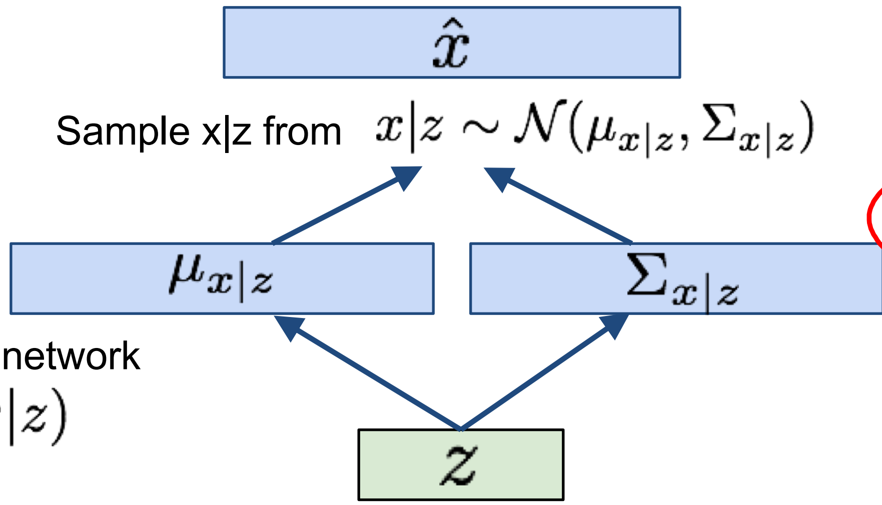
Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Variational Auto Encoders: Generating Data

Use decoder network. Now sample z from prior!

Data manifold for 2-d **z**

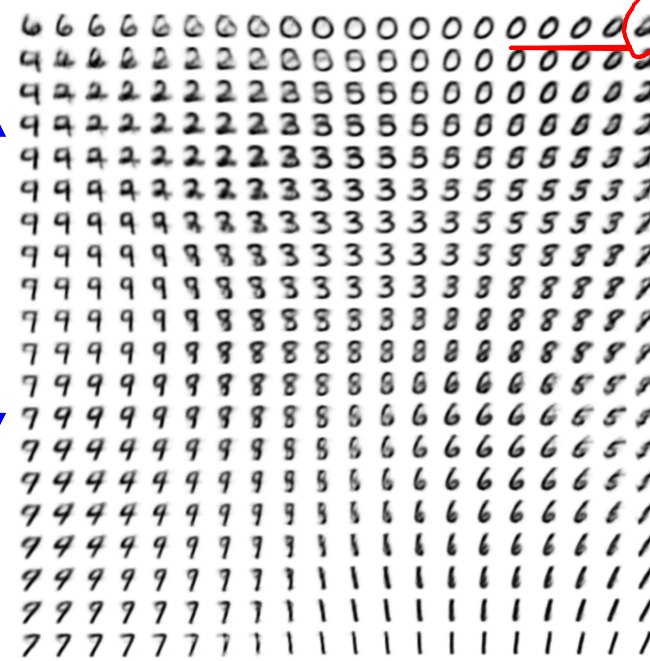


Sample x|z from $x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$

Decoder network
$p_\theta(x|z)$

Sample z from $z \sim \mathcal{N}(0, I)$

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014



Vary **z_1**

Vary **z_2**

Variational Auto Encoders: Generating Data

Diagonal prior on **z**
=> independent
latent variables

Different
dimensions of **z**
encode
interpretable factors
of variation

Degree of smile

Vary
$\mathbf{z}_1$

Vary
$\mathbf{z}_2$

Head pose

Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Variational Auto Encoders: Generating Data

Diagonal prior on **z** => independent latent variables

Different dimensions of **z** encode interpretable factors of variation

Also good feature representation that can be computed using $q_\phi(z|x)$!

Degree of smile

Vary $\mathbf{z_1}$

Vary $\mathbf{z_2}$

Head pose

Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Variational Auto Encoders: Generating Data

32x32 CIFAR-10

Labeled Faces in the Wild

Figures copyright (L) Dirk Kingma et al. 2016; (R) Anders Larsen et al. 2017. Reproduced with permission.

Variational Autoencoders

Probabilistic spin to traditional autoencoders => allows generating data
Defines an intractable density => derive and optimize a (variational) lower bound

Pros:
- Principled approach to generative models
- Allows inference of q(z|x), can be useful feature representation for other tasks

Cons:
- Maximizes lower bound of likelihood: okay, but not as good evaluation as PixelRNN/PixelCNN
- Samples blurrier and lower quality compared to state-of-the-art (GANs)

Active areas of research:
- More flexible approximations, e.g. richer approximate posterior instead of diagonal Gaussian
- Incorporating structure in latent variables

Variational Auto Encoders

VAEs are a combination of the following ideas:

1. Auto Encoders

2. Variational Approximation
   - Variational Lower Bound / ELBO

3. Amortized Inference Neural Networks

4. “Reparameterization” Trick