CS 4803 / 7643: Deep Learning

Topics:

- Variational Auto-Encoders (VAEs)
  - Variational Inference, ELBO

Dhruv Batra

Georgia Tech

Administrativia

- Project submission instructions released
  - Due: 12/03, 11:55pm
  - Last deliverable in the class
  - Can’t use late days
  - https://www.cc.gatech.edu/classes/AY2020/cs7643_fall/

Recap from last time

Variational Autoencoders (VAE)

So far...

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

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

$p(\vec{x})$ $D = \{\vec{x}\}$

$p(\vec{x} | \boxed{z})$ "latent" "hidden"

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

conditional prior

So far...

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

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

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

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

z continuous

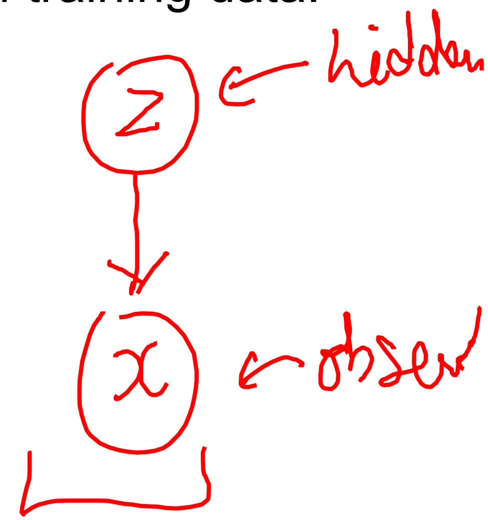
$$\sum_z p_\theta(z) \, p_\theta(x|z)$$

z discrete

So far...

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

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



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

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

$$\sum$$

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

GMM

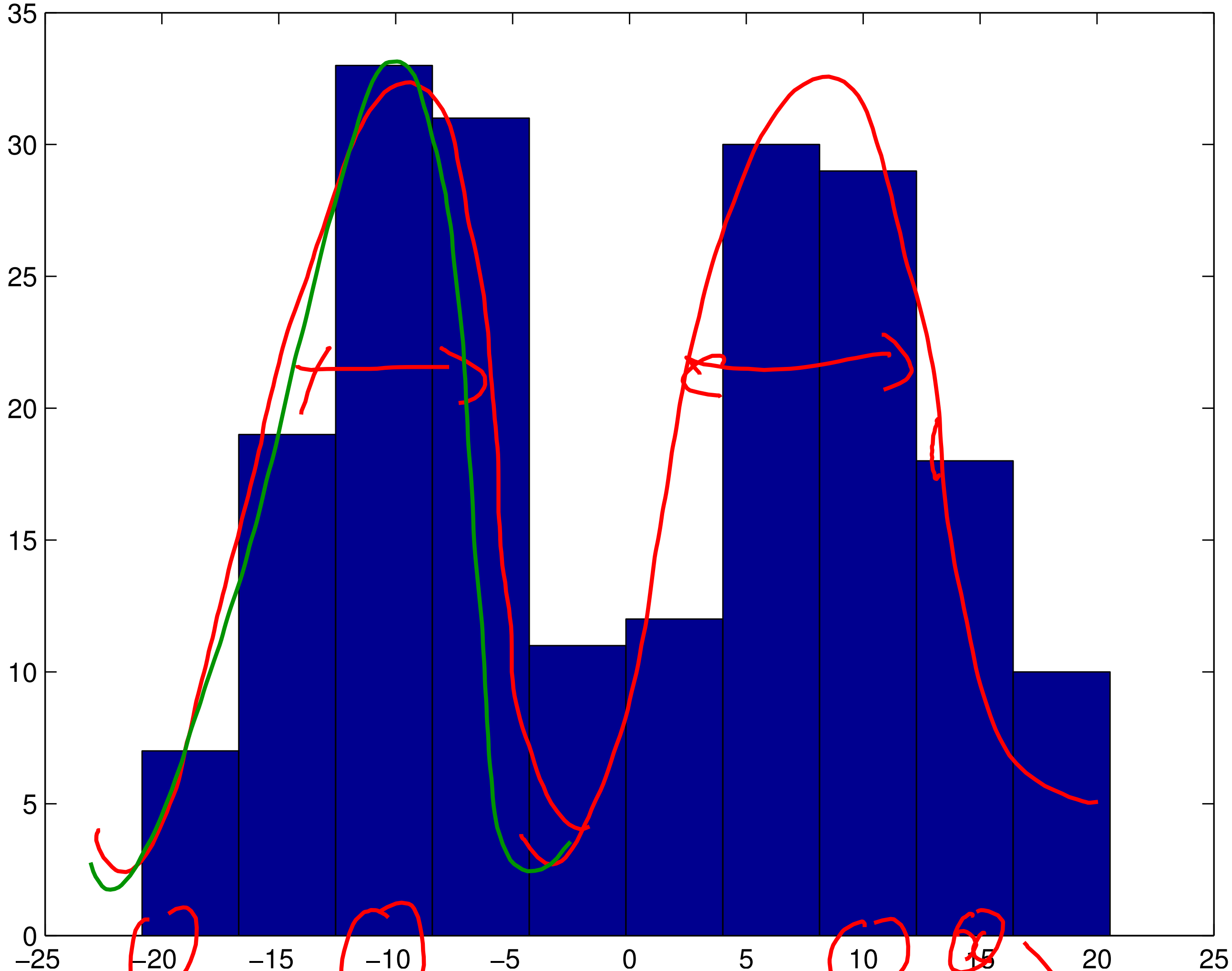
Gaussian Mixture Model

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

$z \to x$

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

$p(x)$



x

$\mathcal{N}(\mu_1, \sigma_1^2)$ $\qquad$ $\mathcal{N}(\mu_2, \sigma_2^2)$

$\mu_1 = -10$ $\qquad$ $\mu_2 = +10$

Gaussian Mixture Model

$z \to x$

$$z \sim \mathrm{Cat}(\vec{\pi}) \qquad \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_k \end{bmatrix} \qquad z \in \{1, \ldots, k\}$$

$$\pi_c = P(z = c)$$

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

$$\underbrace{p(\vec{x})}_{p(x,z)} = \sum \underbrace{p(z)}_{\boxed{\pi_z}} \underbrace{p(\vec{x} \mid z)}_{\boxed{\mathcal{N}}}$$

Gaussian Mixture Model

$P(z) = \pi_z$

$P(x|z) = N(\mu_z, \sigma_z^2)$

Available from model

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

= Marginalization

$$\boxed{P(z|\vec{x})} = \frac{P(z,\vec{x})}{P(\vec{x})} = \frac{P(\vec{x}|z)\,P(z)}{\sum(\quad)(\quad)}$$

"Inference"

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

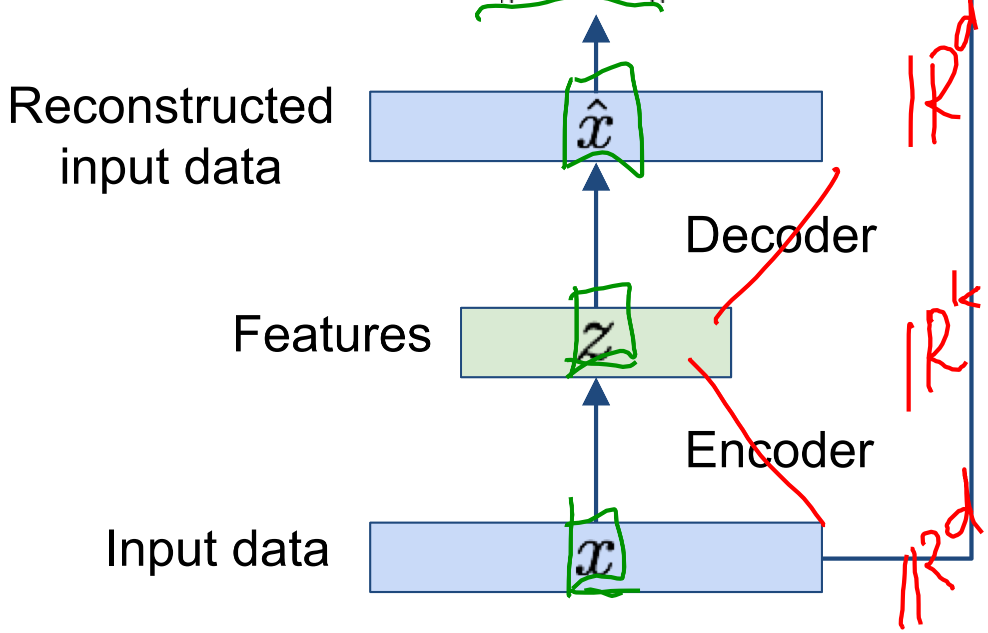
Autoencoders

Train such that features can be used to reconstruct original data

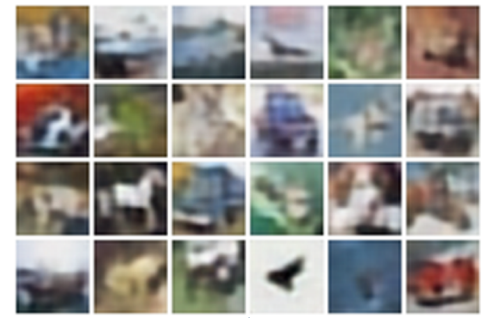
Doesn't use labels!

L2 Loss function:

$$\|x - \hat{x}\|^2$$



Reconstructed data



Encoder: 4-layer conv
Decoder: 4-layer upconv

Input data

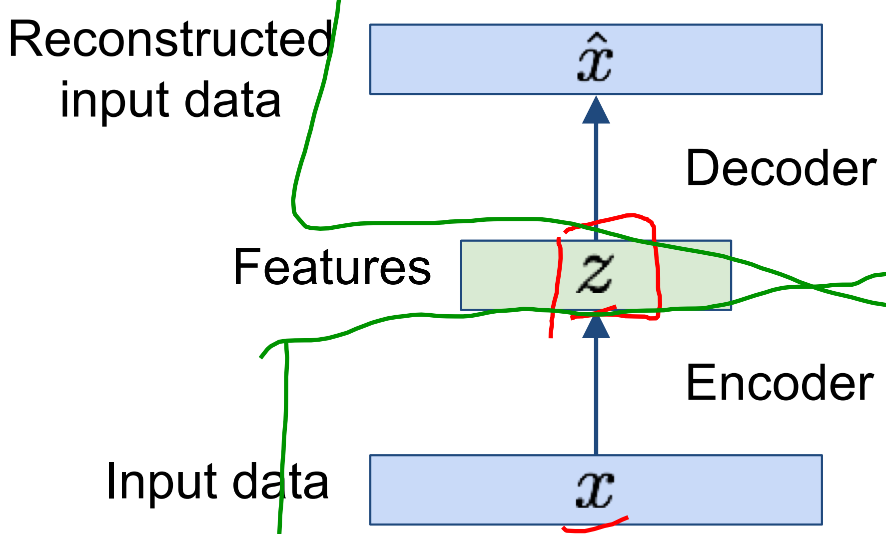
Autoencoders

$p(y|x)$

$z = f_\theta(x)$

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

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



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!

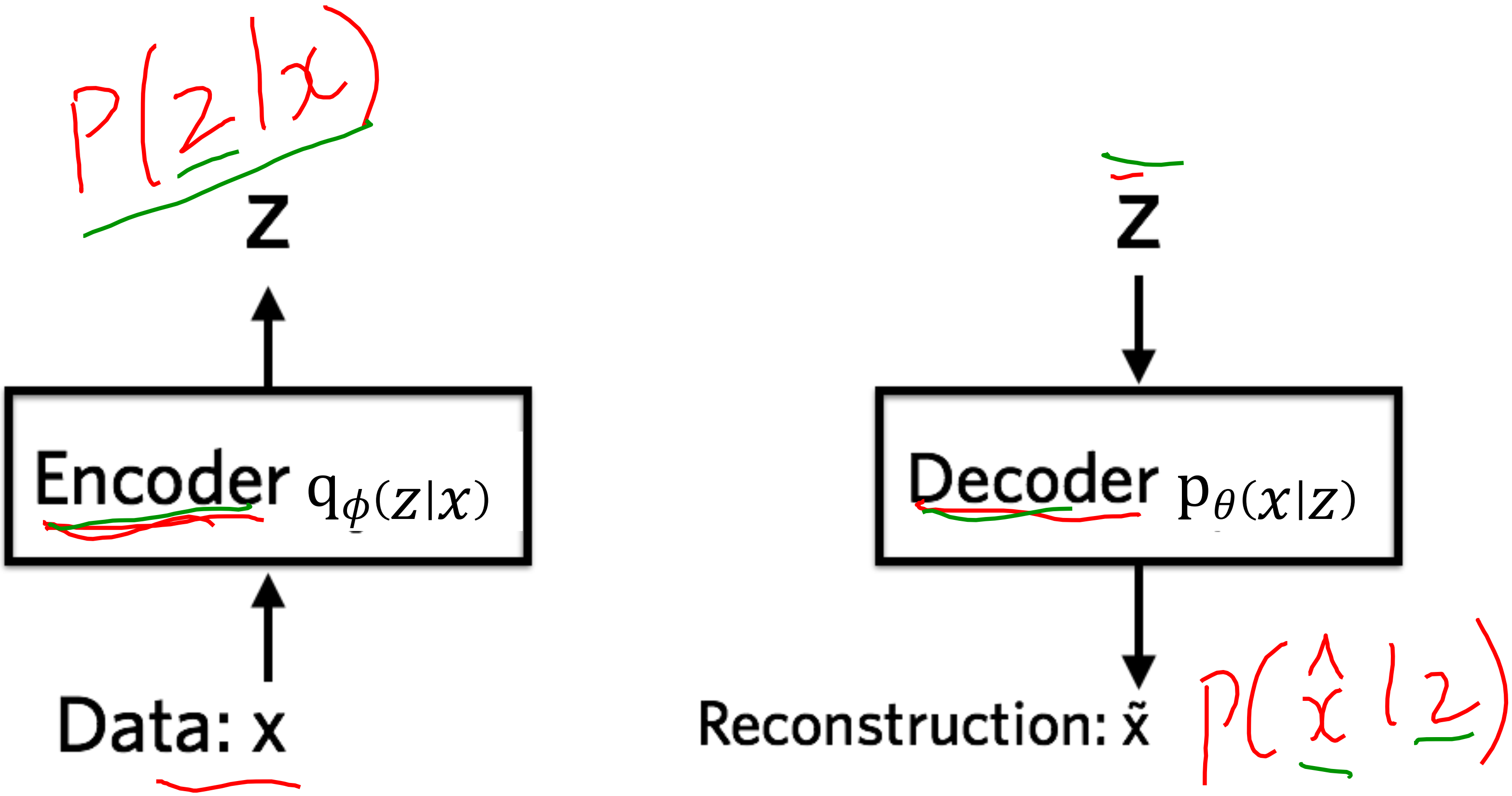


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

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

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

$\downarrow$

$q_i(z)$

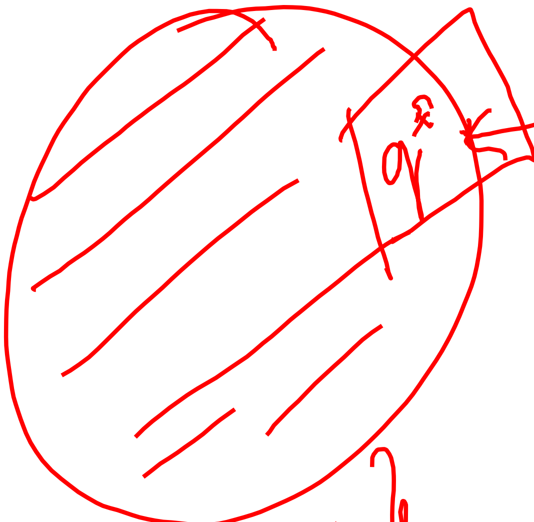
$\int$ Hard

$z \rightarrow \vec{x}$

What is Variational Inference?

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

Intuition



$p(z)$

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

$$\min_{q \in \{\ \}}$$

$$d(p, q)$$

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

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

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

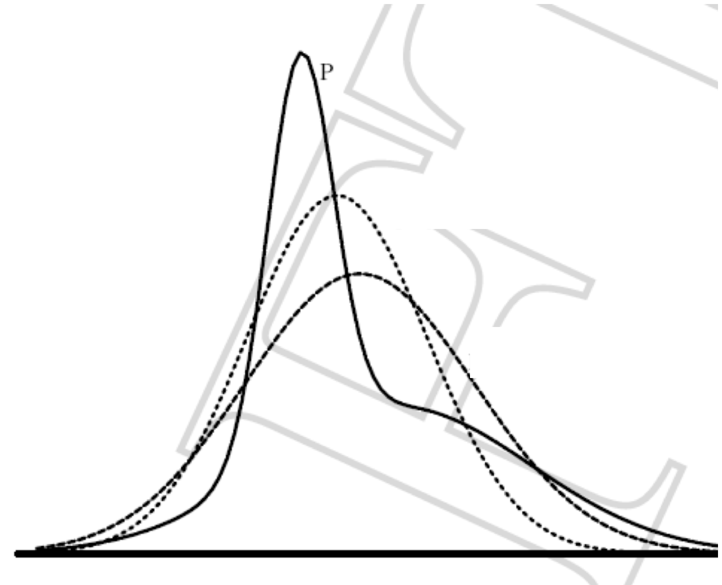
error

$$\sum q(z) \log \frac{q(z)}{p(z)}$$

support

Find simple approximate distribution

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



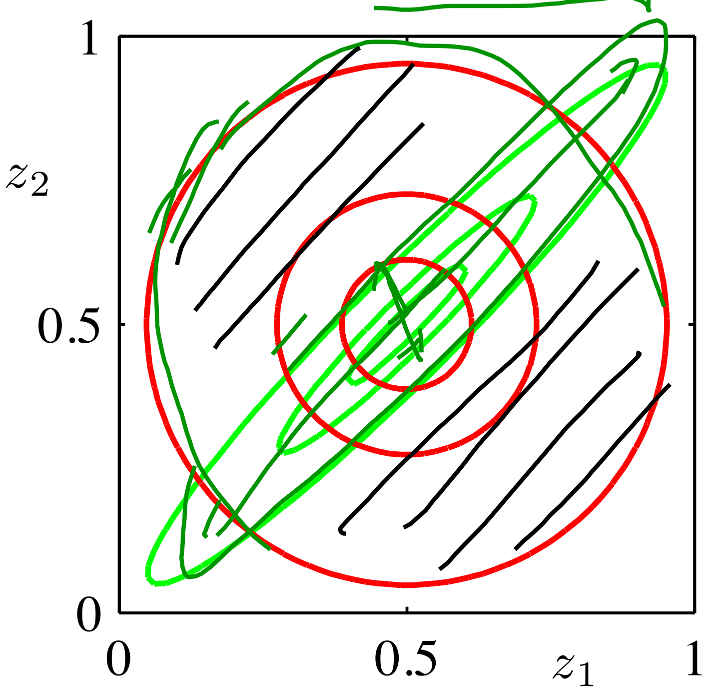
- D(p||q)
  - true distribution p defines support of diff.
  - the "correct" direction
  - will be intractable to compute

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

Example 1

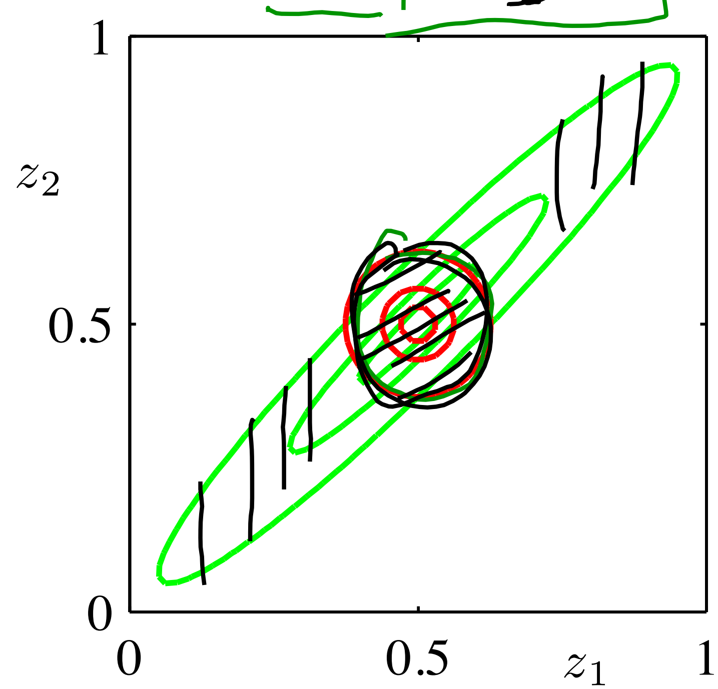
- p = 2D Gaussian with arbitrary co-variance (Σ) $\begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}_{2\times 2}$
- q = 2D Gaussian with isotropic co-variance ($\sigma^2 I$) $\begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$

argmin_q KL (p || q)



(b)

argmin_q KL (q || p)



(a)

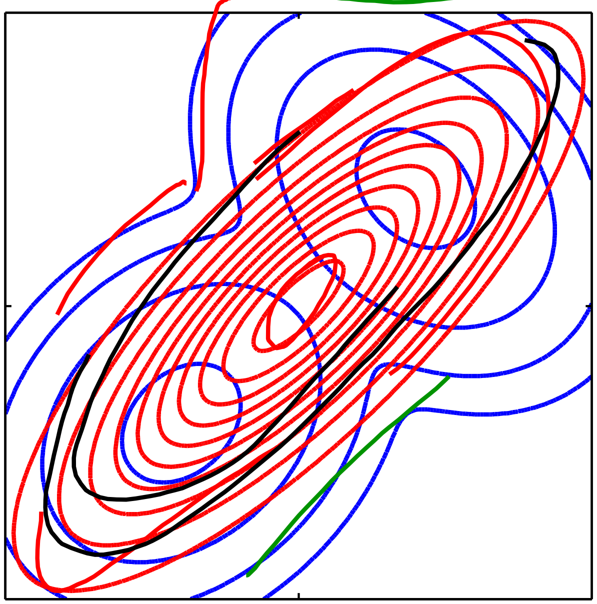
p = Green; q = Red

Example 2

- p = Mixture of Two Gaussians
- q = Single Gaussian μ, σ

SL → "right"

argmin_q KL (p || q)

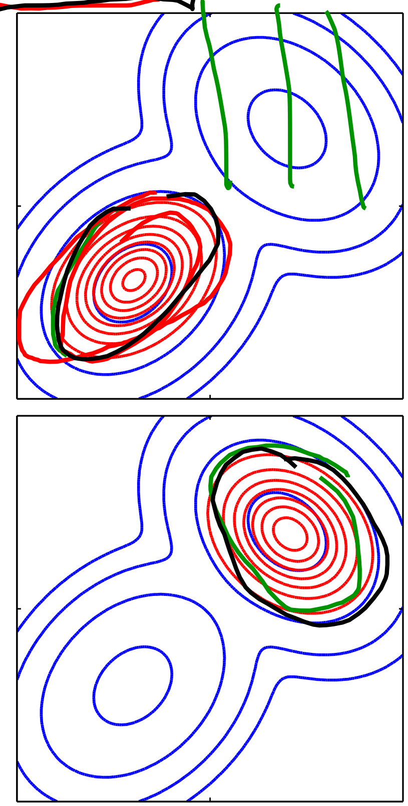


UL

"wrong"

argmin_q KL (q || p)

q(z)



p = Blue; q = Red

Plan for Today

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

- Next time:
  - Reparameterization trick for optimizing VAEs

The general learning problem with missing data

z

$\downarrow$

x

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

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

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

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

$D = \{ \vec{x}_1, \ldots, \vec{x}_N \}$

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

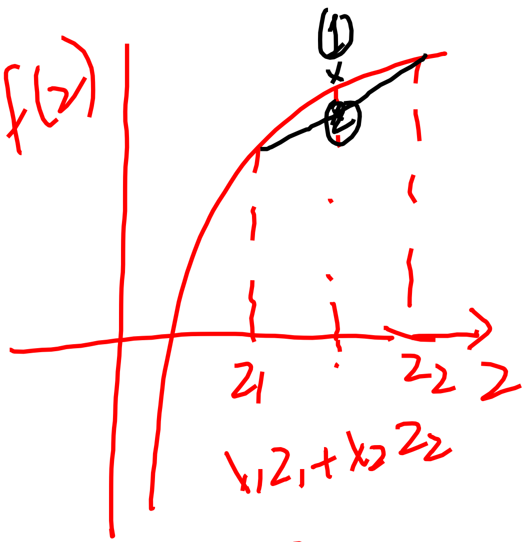
$P(x_i \mid \theta) P(z \mid x_i, \theta)$

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

$= \log \left[\sum P(z \mid x_i, \theta) \right] [P(x_i \mid \theta)]$

Applying Jensen's inequality

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



$\lambda_1, \lambda_2 \geq 0$

$\boxed{\lambda_1 + \lambda_2 = 1}$

1 ≥ 2

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

$2 \to K$

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

$$\equiv f(E[z]) \geq E_{p(z)}[f(z)]$$

$z \to g(z)$

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

Applying Jensen's inequality

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

$z \to x$

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

$$ll(\theta) \geq F(\theta, Q_i) \quad \boxed{\geq} \sum_z Q_i(z) \log \frac{P(x_i, z \mid \theta)}{Q_i(z)}$$

"Free Energy" $F(\theta, Q_i)$

Variational Lower Bound

Evidence Lower Bound (ELBO)

Evidence Lower Bound

- Define potential function F(θ,Q):

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

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

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

ELBO: Factorization #1 (GMMs)

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

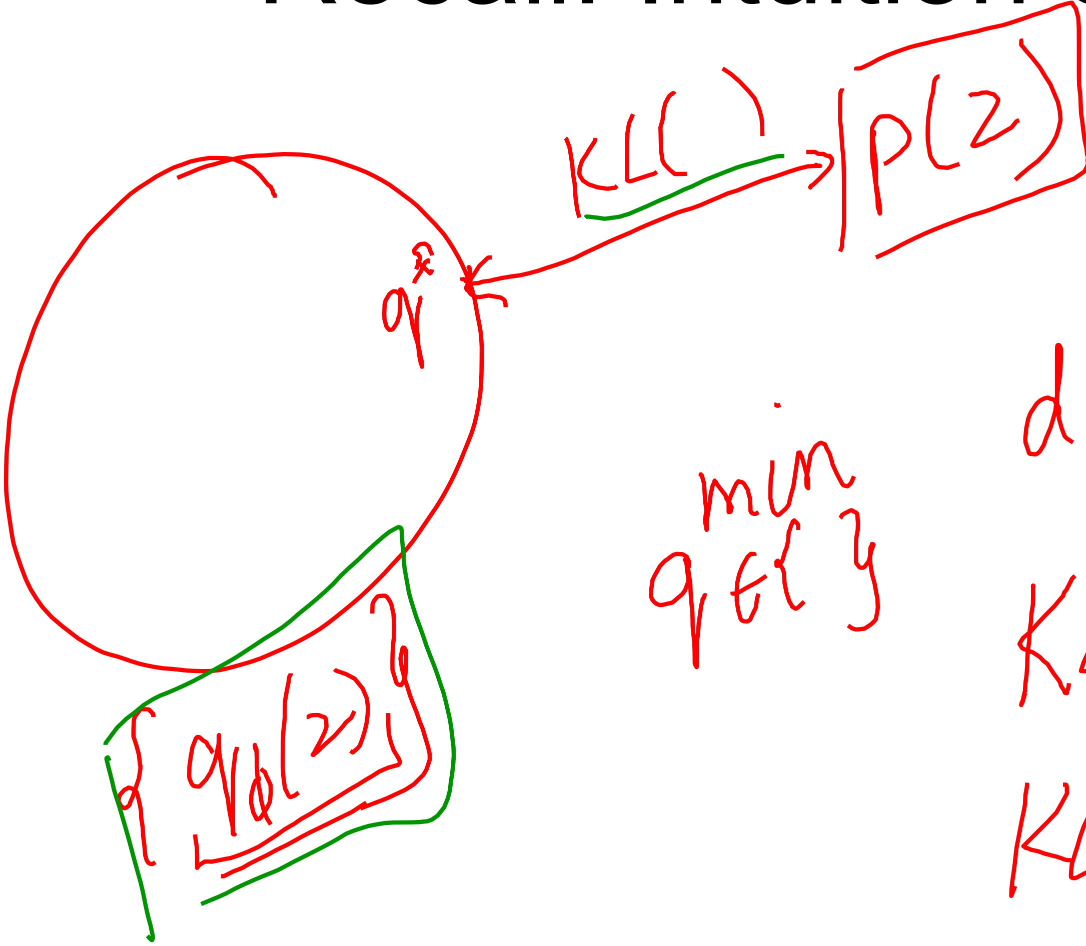
$$ll(\theta : \mathcal{D}) \geq F(\theta, Q_i) = \cancel{\sum_{i=1}^{N}} \sum_{\mathbf{z}} Q_i(\mathbf{z}) \log \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}$$

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

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

$$ll(\theta) \geq F(\theta, Q_i) \qquad ll(\theta) = F(\theta, Q_i) + KL(\quad)$$

Recall: Intuition of VI



$$\min_{q \in \{\ \}}$$

$$d(p, q)$$

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

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

ELBO: Factorization #1 (GMMs)

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

$\boxed{\max_{\theta, Q_i}}$

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

- **E-step**: Fix $\theta^{(t)}$, maximize F over Q_i
- **M-step**: Fix $Q_i^{(t)}$, maximize F over θ

EM for Learning GMMs

- Simple Update Rules

- **E-step**: Fix $\theta^{(t)}$, maximize F over Q_i

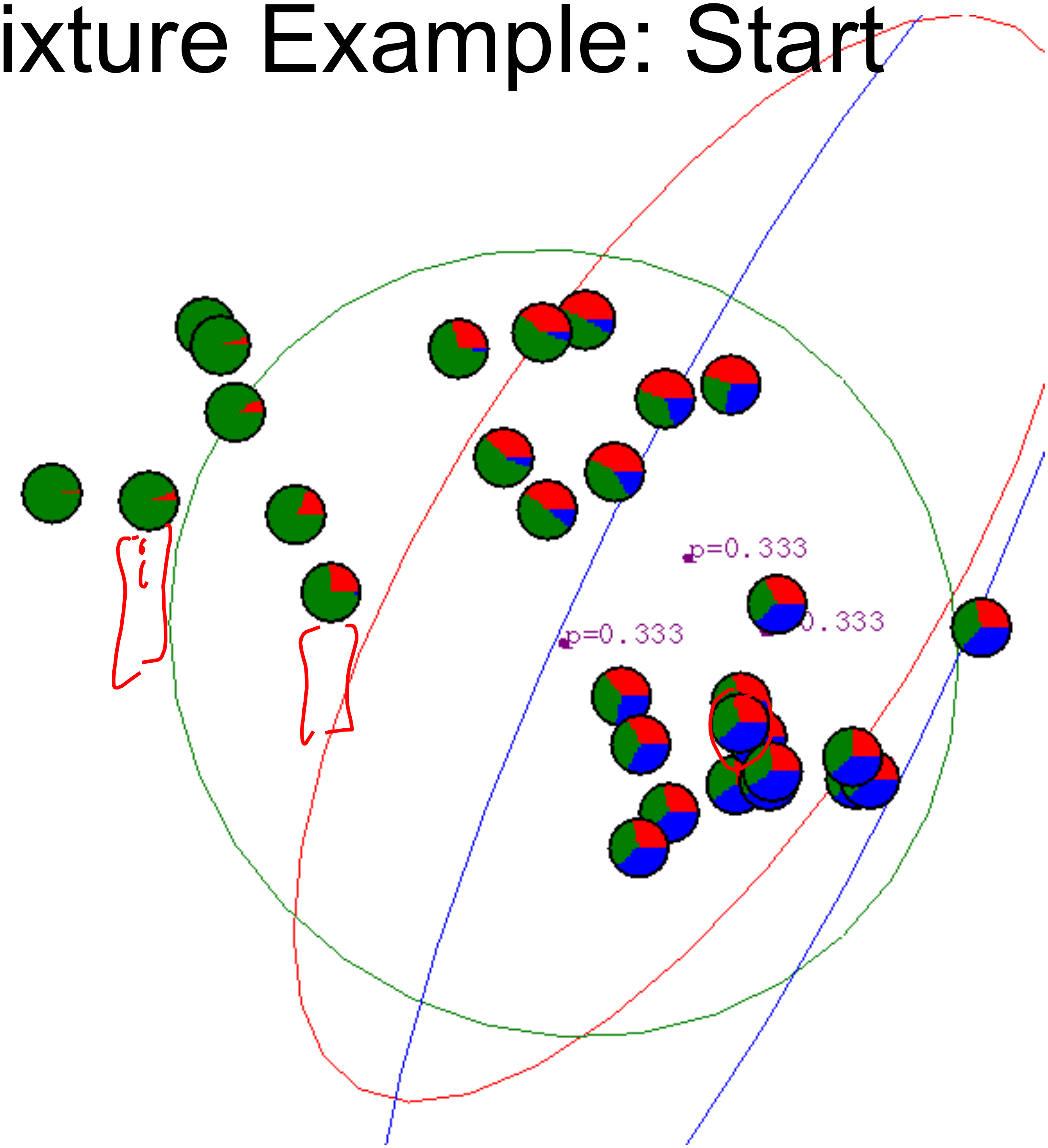
$$Q_i^{(t)}(\mathbf{z}) = P(\mathbf{z} \mid \mathbf{x}_i, \theta^{(t)})$$

- **M-step**: Fix $Q_i^{(t)}$, maximize F over θ
  - maximize expected likelihood under $Q_i(z)$
  - Corresponds to weighted dataset:
    - <$\mathbf{x}_1$,$\mathbf{z}$=1> with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}_1)$
    - <$\mathbf{x}_1$,$\mathbf{z}$=2> with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}_1)$
    - <$\mathbf{x}_1$,$\mathbf{z}$=3> with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}_1)$
    - <$\mathbf{x}_2$,$\mathbf{z}$=1> with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}_2)$
    - <$\mathbf{x}_2$,$\mathbf{z}$=2> with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}_2)$
    - <$\mathbf{x}_2$,$\mathbf{z}$=3> with weight $Q^{(t+1)}(\mathbf{z}=3|\mathbf{x}_2)$

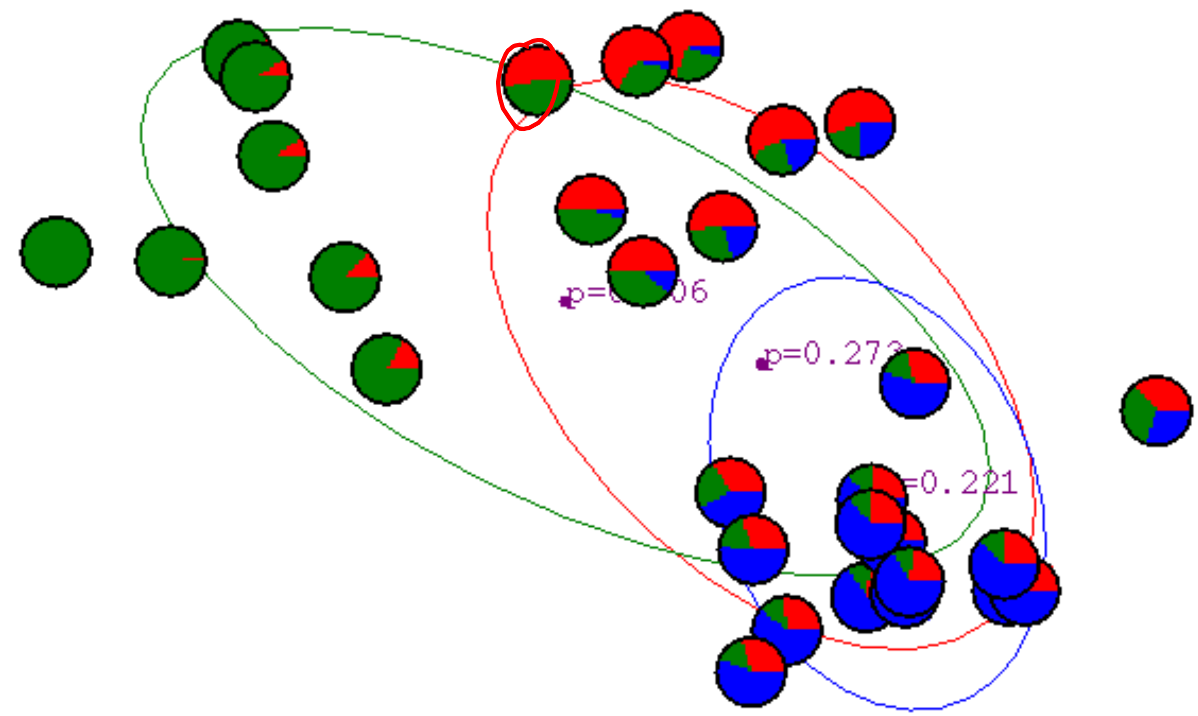
Gaussian Mixture Example: Start

$\underline{\theta} = \{\pi_i \{\mu_i, \Sigma_i\}\}$ $k=3$

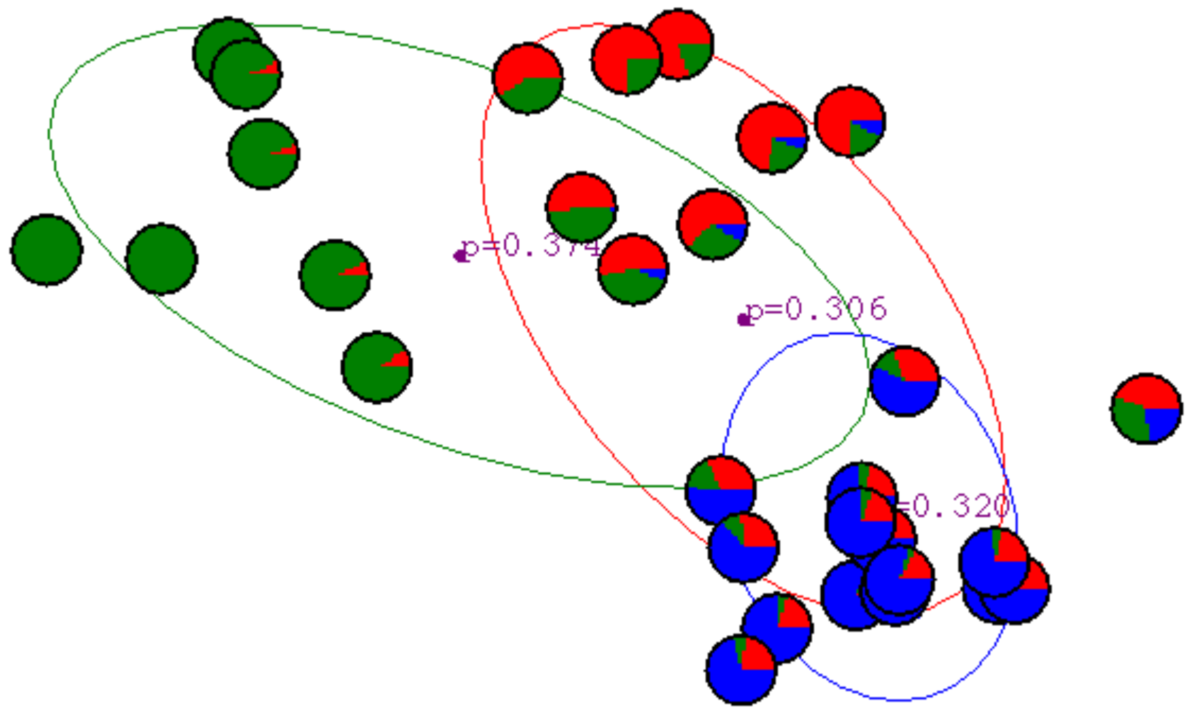
$\boxed{q(\underline{z}|\underline{x_i})} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}$



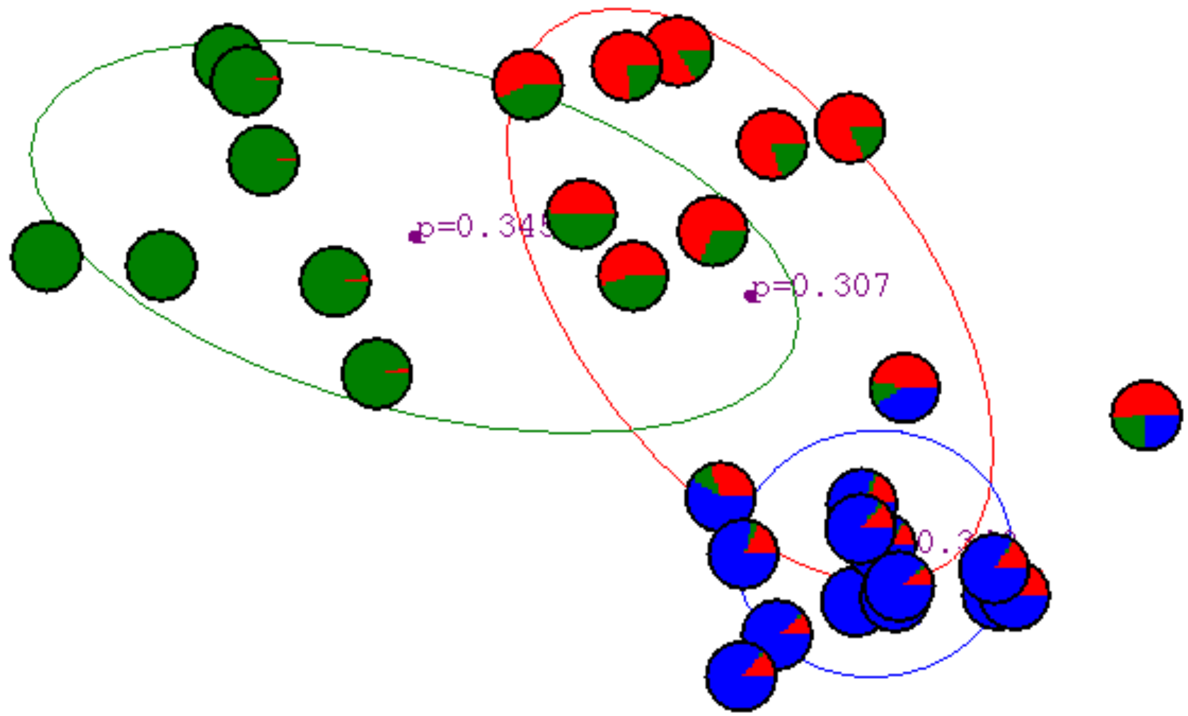
After 1st iteration



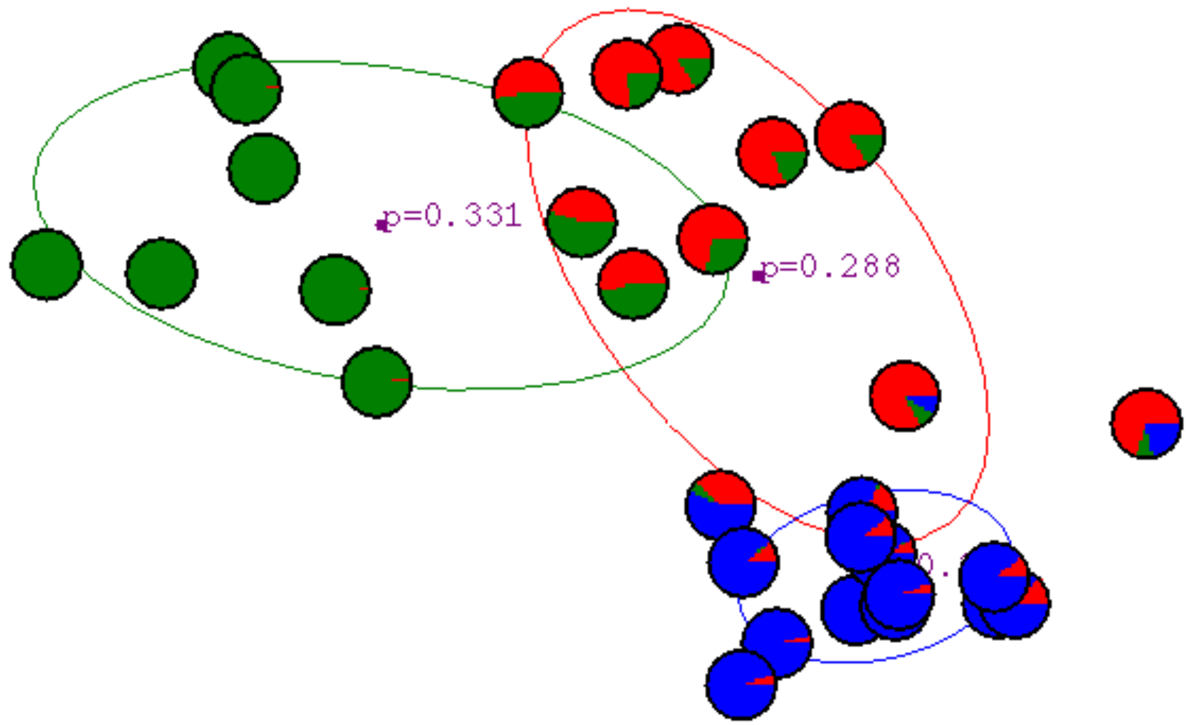
After 2nd iteration



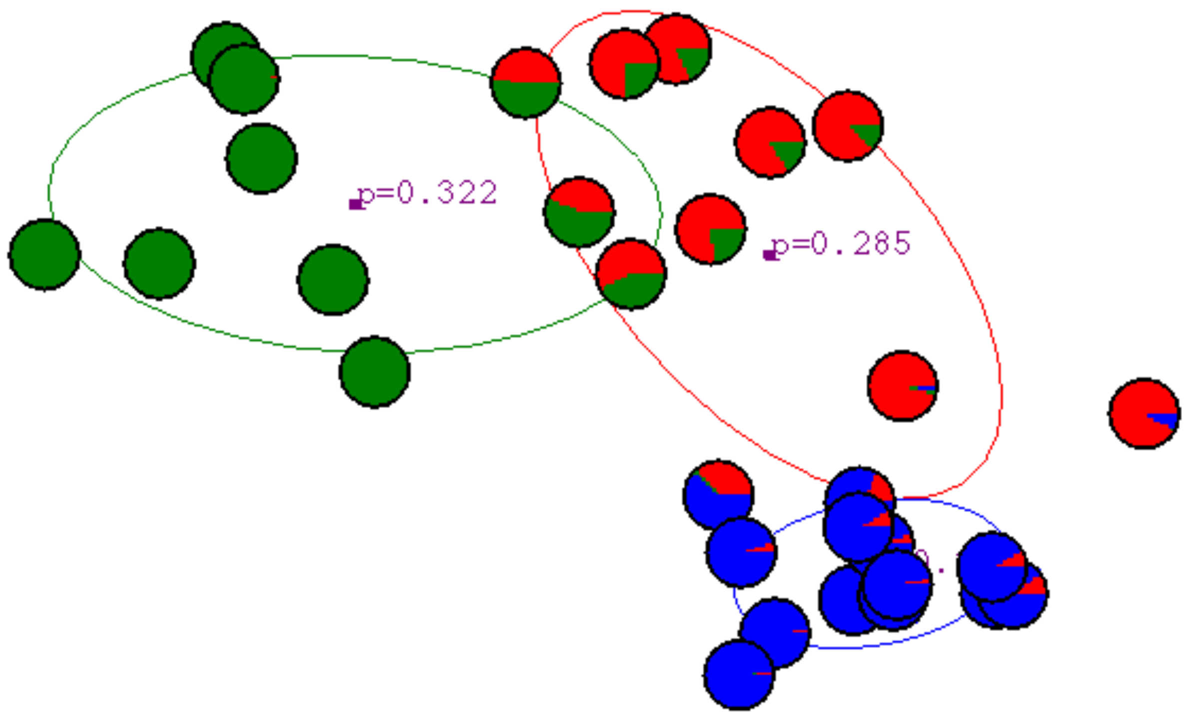
After 3rd iteration



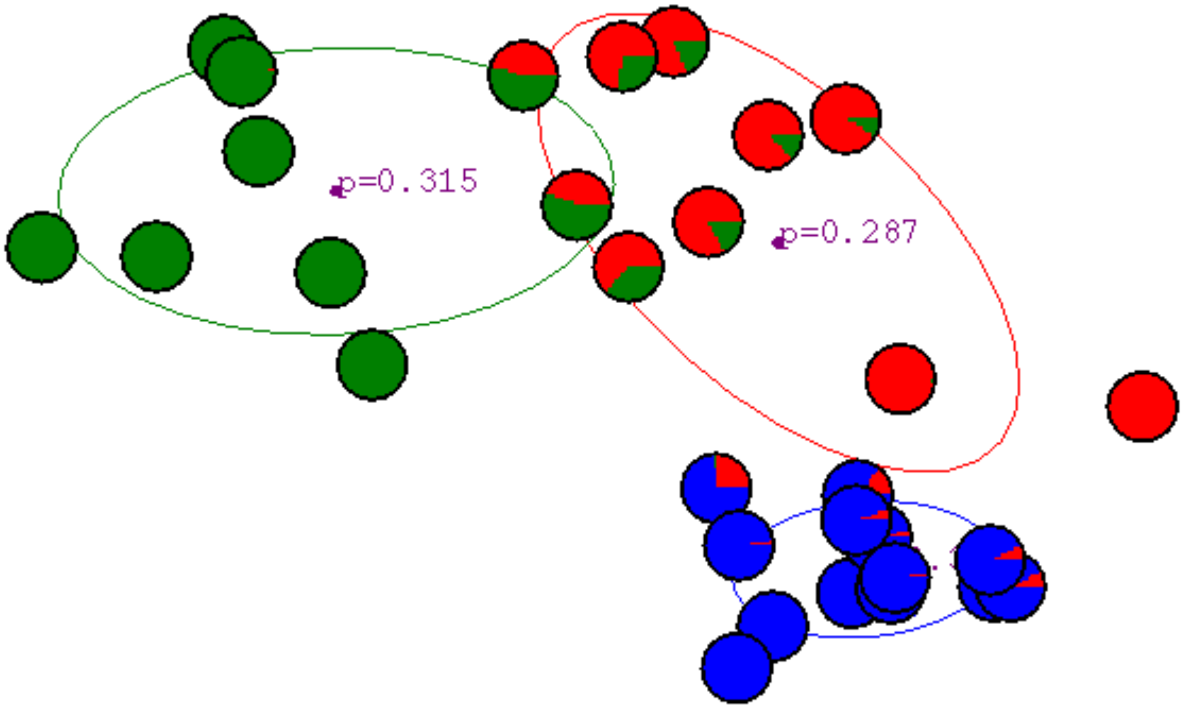
After 4th iteration



After 5th iteration



After 6th iteration



After 20th iteration

$P(x)$

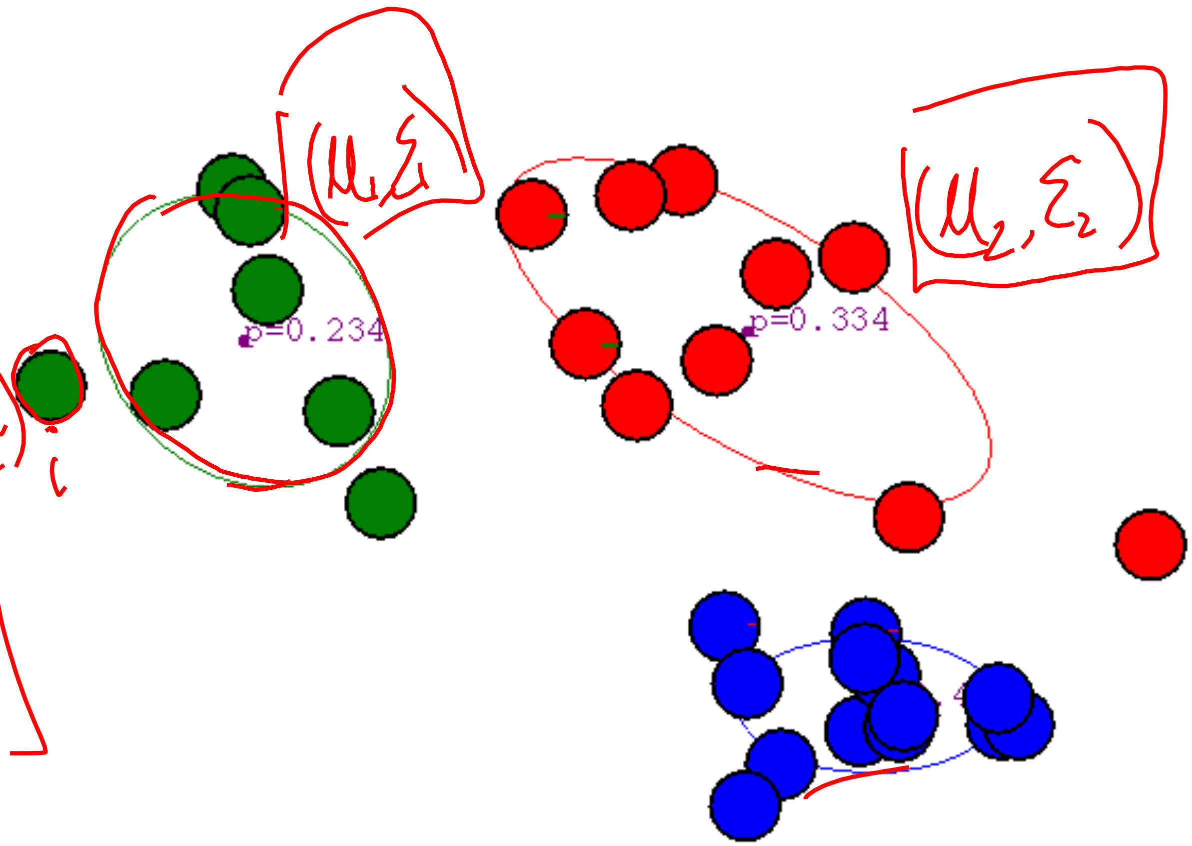
$x \sim P(\ )$

$z \sim P(z)$

$x \sim P(x|z)$

$x \sim$

$Q(z|x_i) \sim \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$



ELBO: Factorization #2 (VAEs)

$$\max_{\theta, Q_i} \quad P(z \mid \theta)\, P(x_i \mid z, \theta)$$

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

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

(VAEs)

$$= \underbrace{E_{\boxed{Q_i(z)}}\left[\log P(\vec{x}_i \mid z, \theta) \right]}_{\text{"Explain the data"}} \underbrace{- KL\left(Q_i(z) \,\|\, P(z \mid \theta) \right)}_{\text{"Regulariser"}}$$

Be simple

Variational Auto Encoders

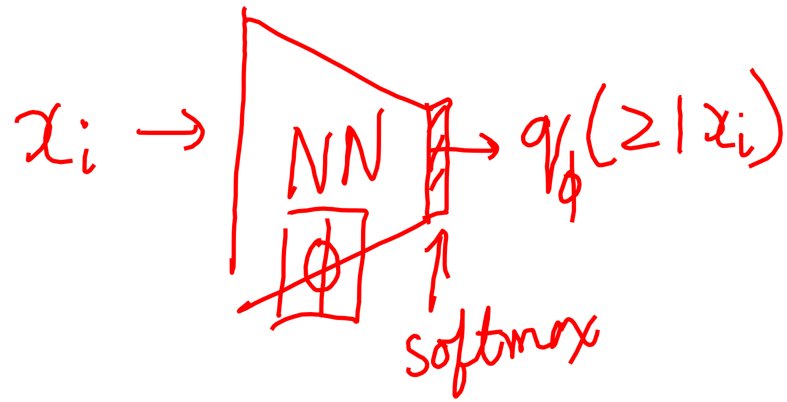
VAEs are a combination of the following ideas:

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

Amortized Inference Neural Networks

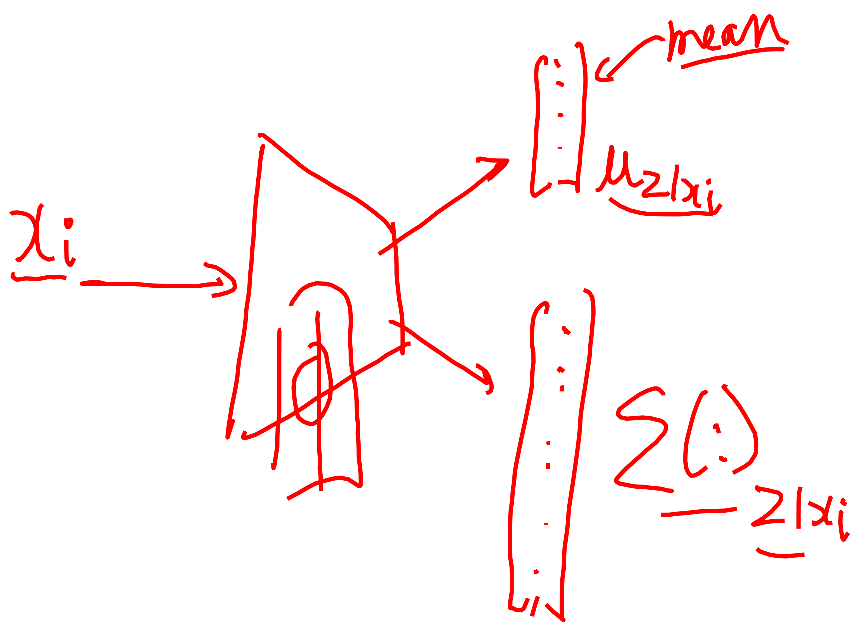
$$Q_i(z) = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}$$

$$z \sim Cat()$$



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

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



$$\sim p(z|x_i)$$

$$\mathcal{N}(\;,\;)$$

Variational Autoencoders

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

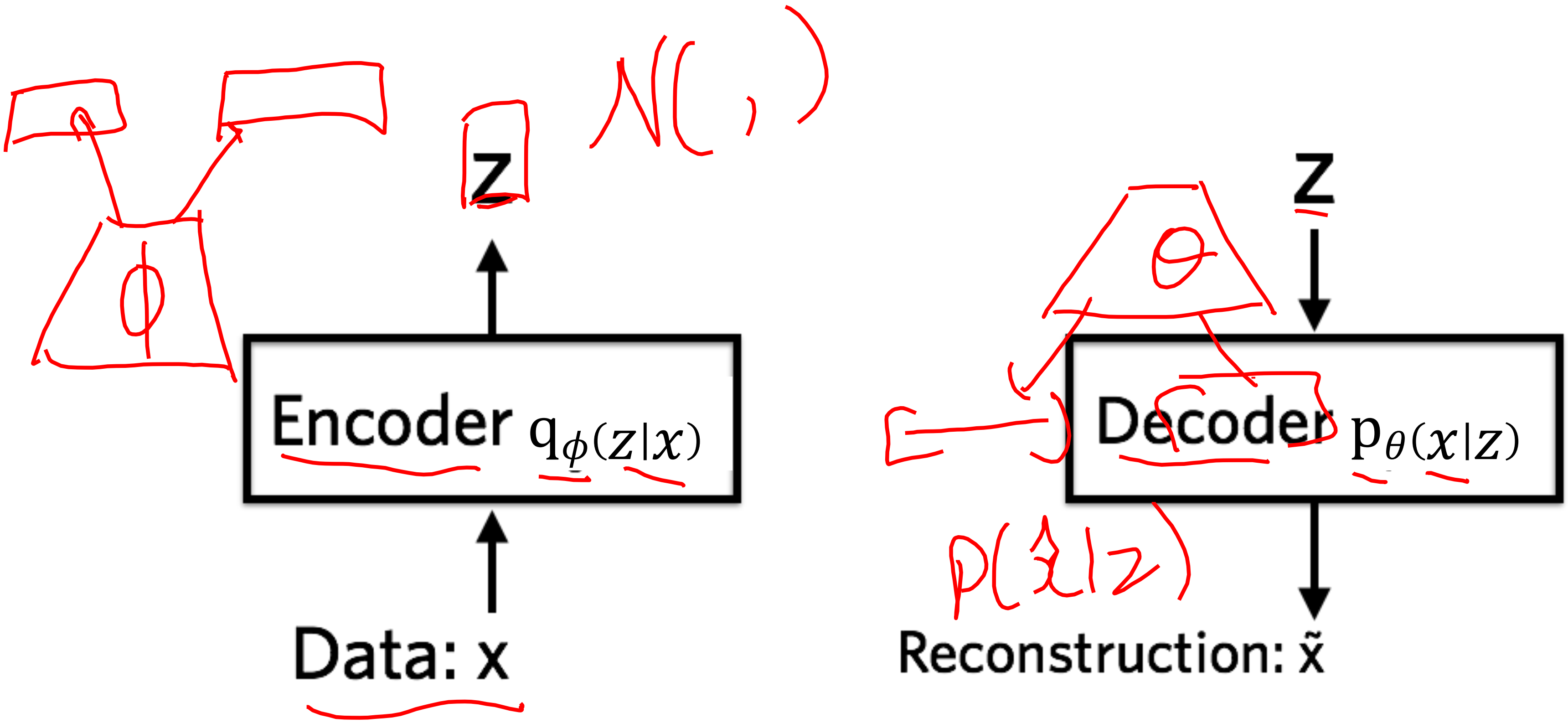
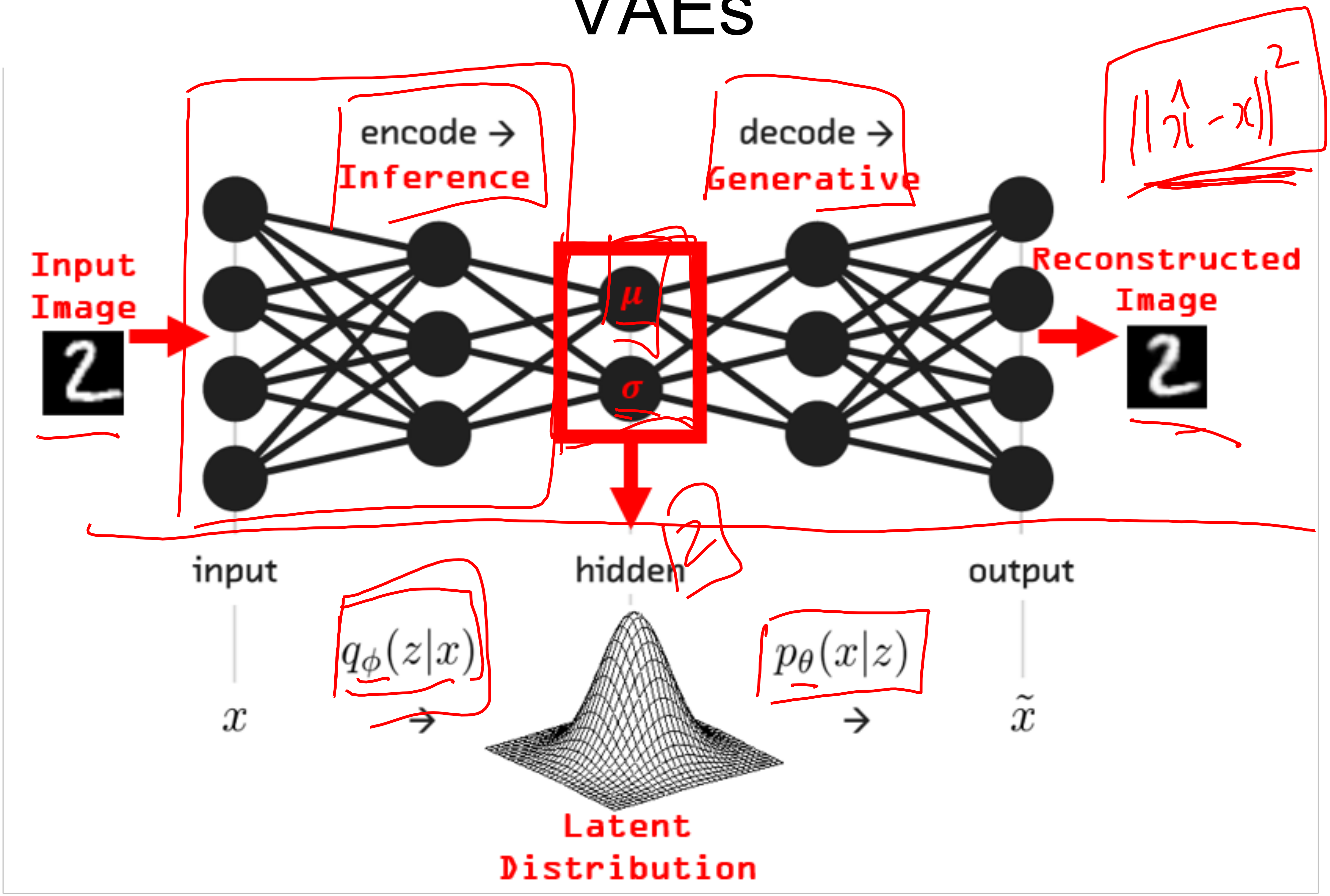


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

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
$\|\hat{x} - x\|^2$

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

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

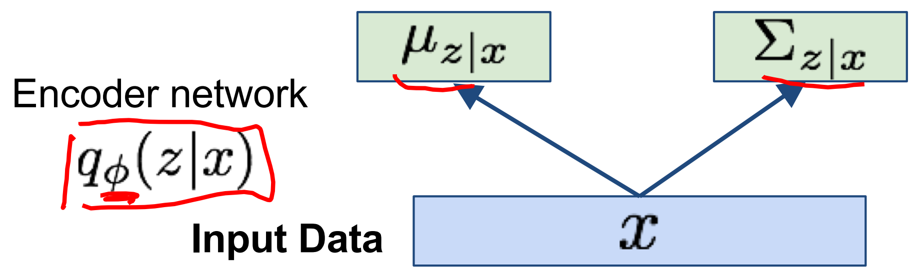
Let's look at computing the bound (forward pass) for a given minibatch of input data

Input Data x

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

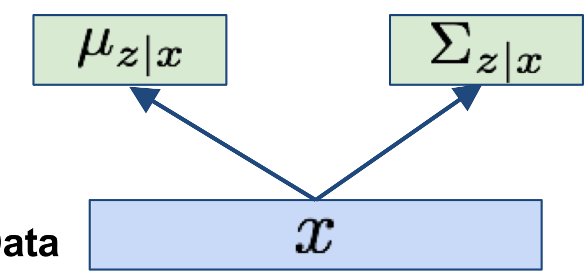
$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Make approximate posterior distribution close to prior

Encoder network

$q_\phi(z|x)$

Input Data

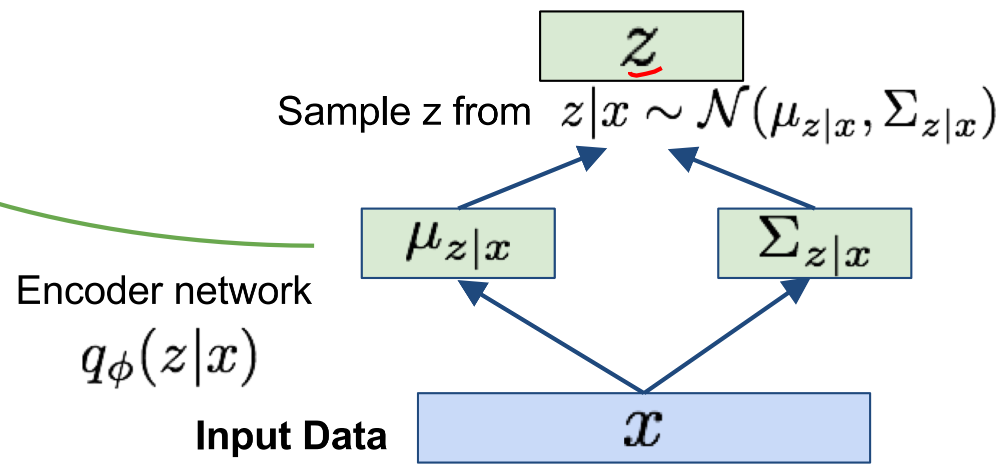


Variational Auto Encoders

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



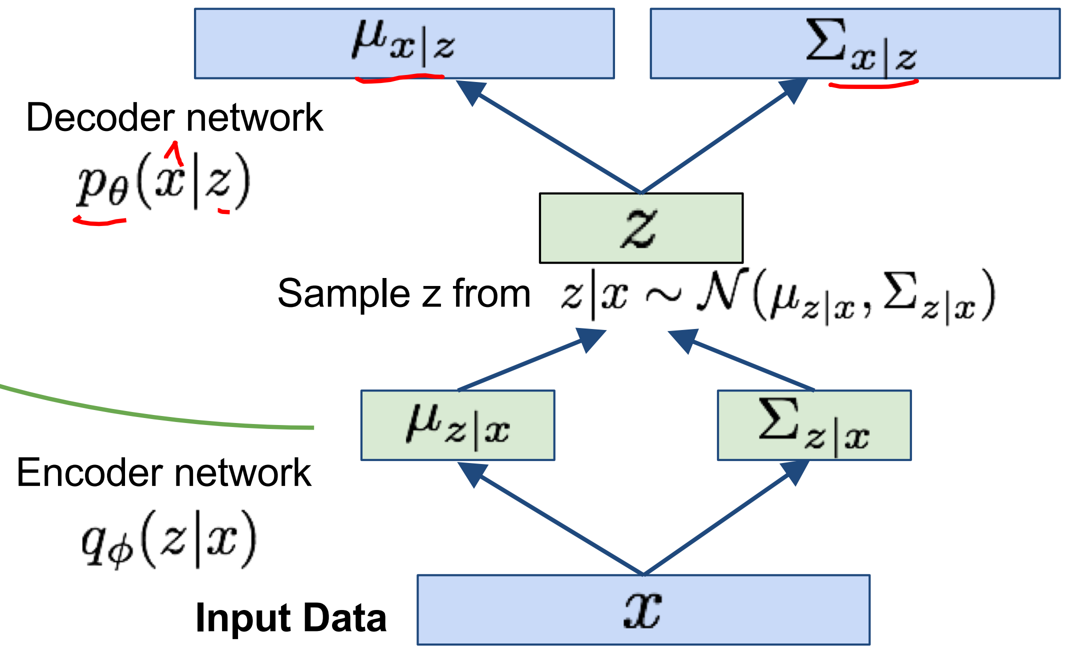
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

Make approximate posterior distribution close to prior



Variational Auto Encoders

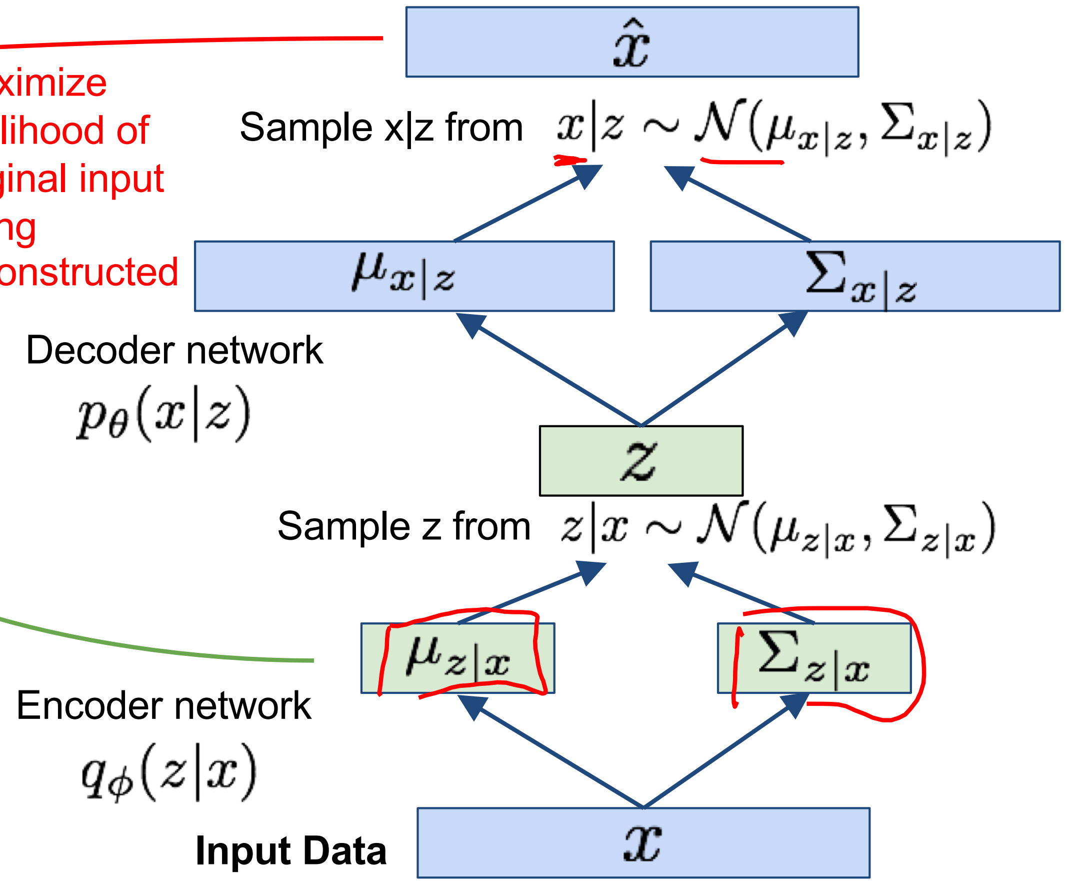
Putting it all together: maximizing the likelihood lower bound

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

Maximize likelihood of original input being reconstructed

loss(θ, ϕ)

Make approximate posterior distribution close to prior



Variational Auto Encoders

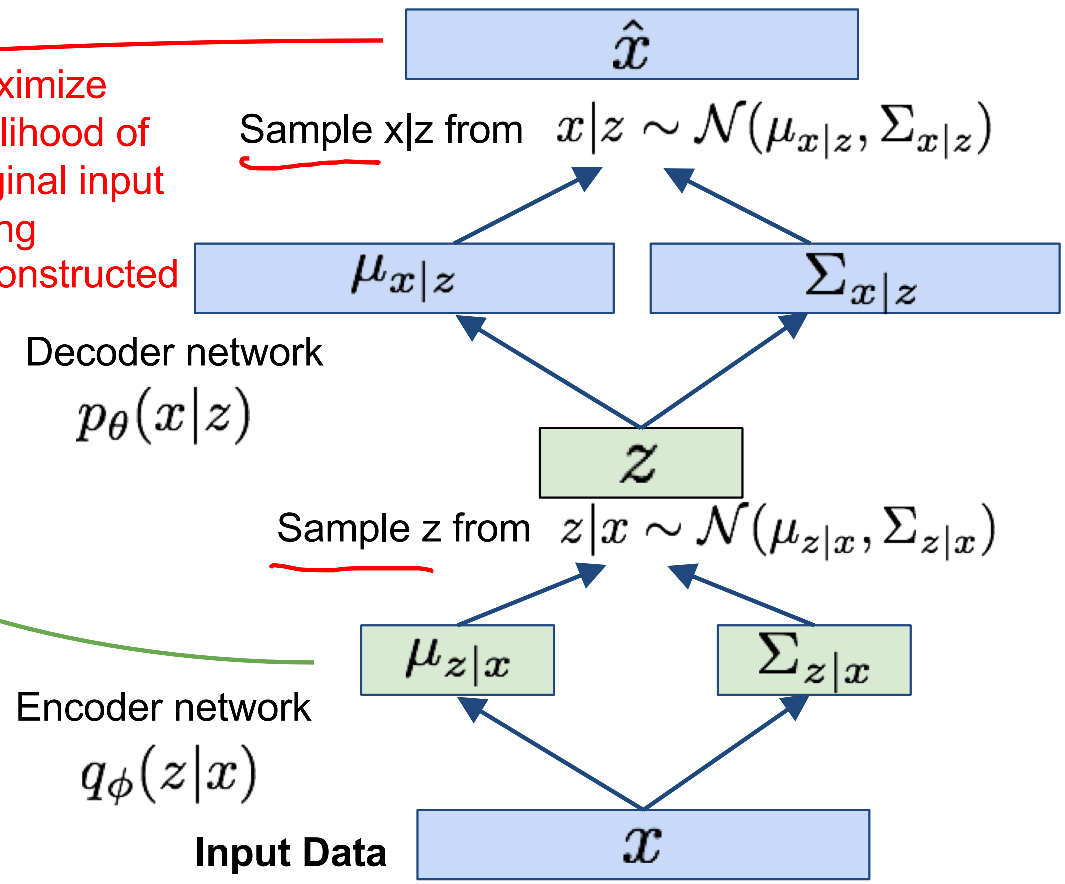
Putting it all together: maximizing the likelihood lower bound

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

Maximize likelihood of original input being reconstructed

Make approximate posterior distribution close to prior

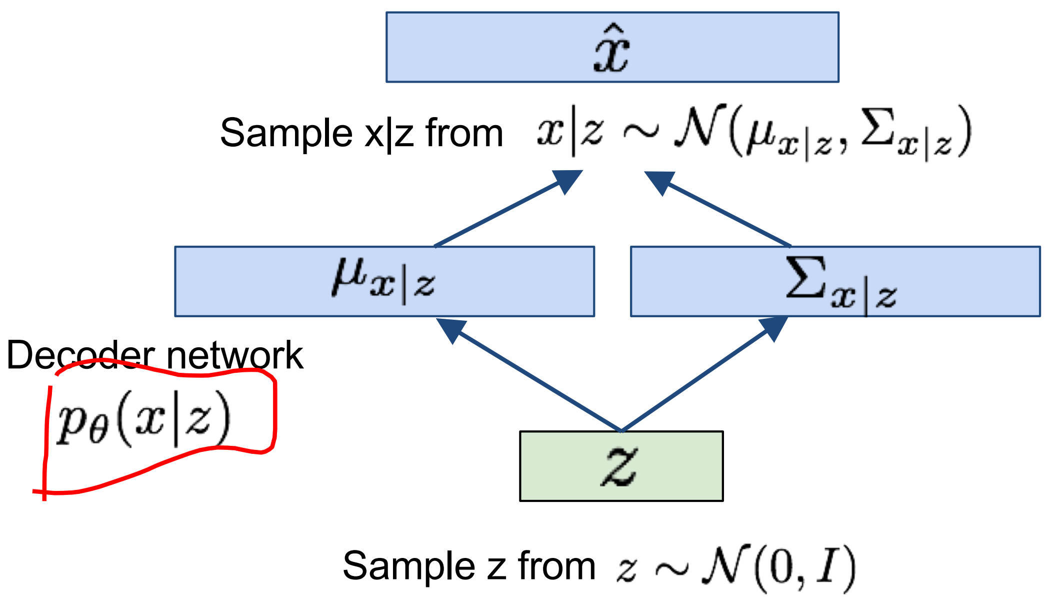
For every minibatch of input data: compute this forward pass, and then backprop!



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

Variational Auto Encoders: Generating Data

Use decoder network. Now sample z from prior!

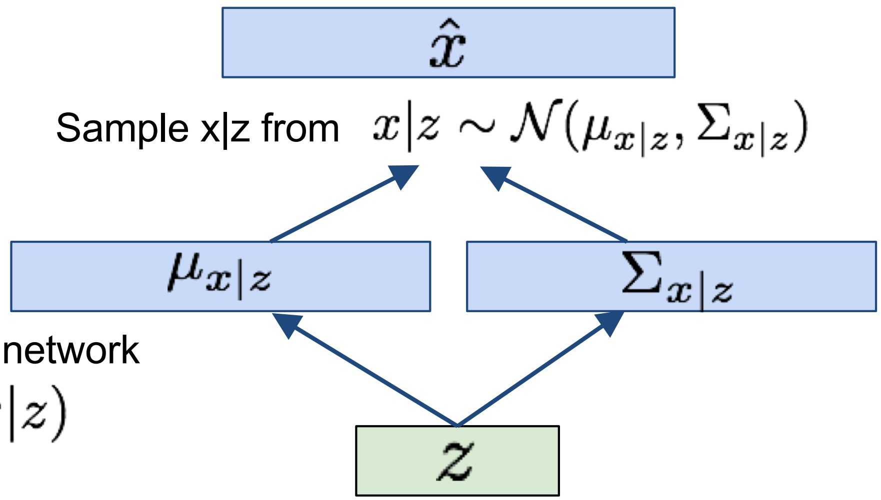


Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Variational Auto Encoders: Generating Data

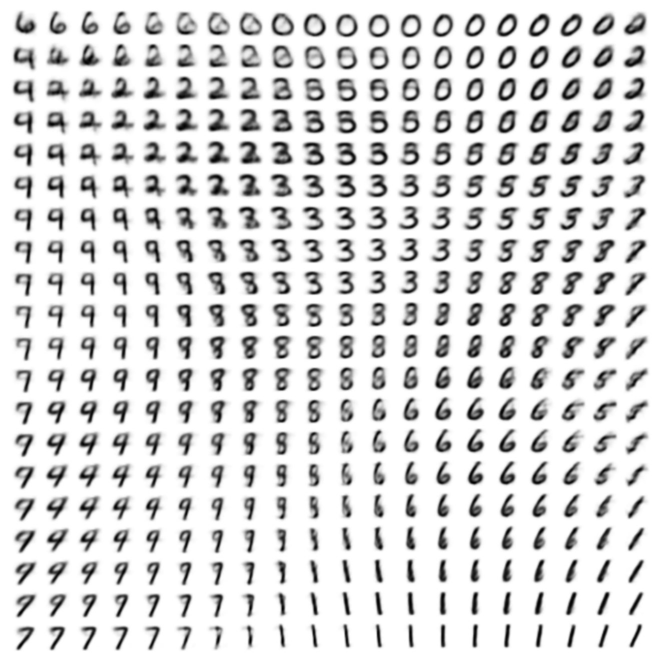
Use decoder network. Now sample z from prior!

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



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

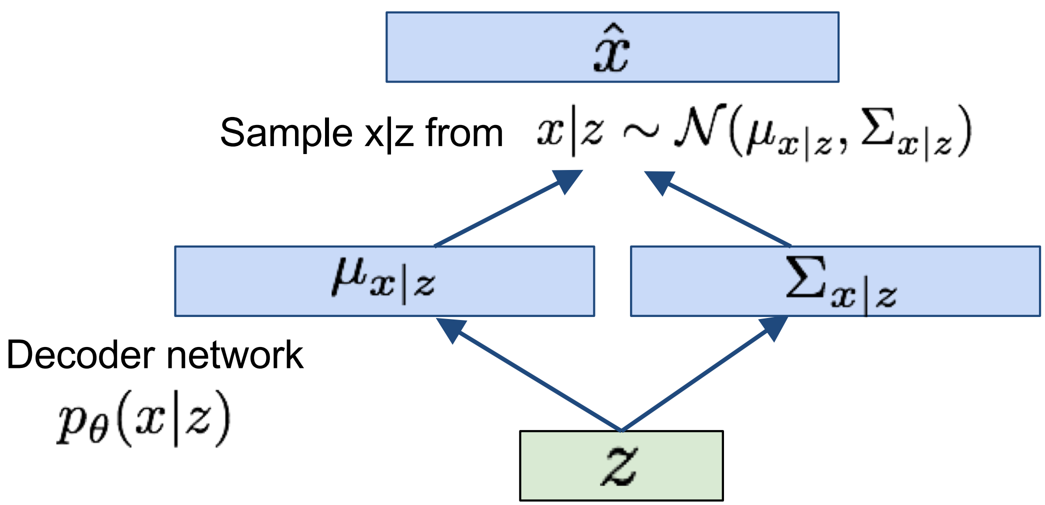
Sample z from $z \sim \mathcal{N}(0, I)$



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

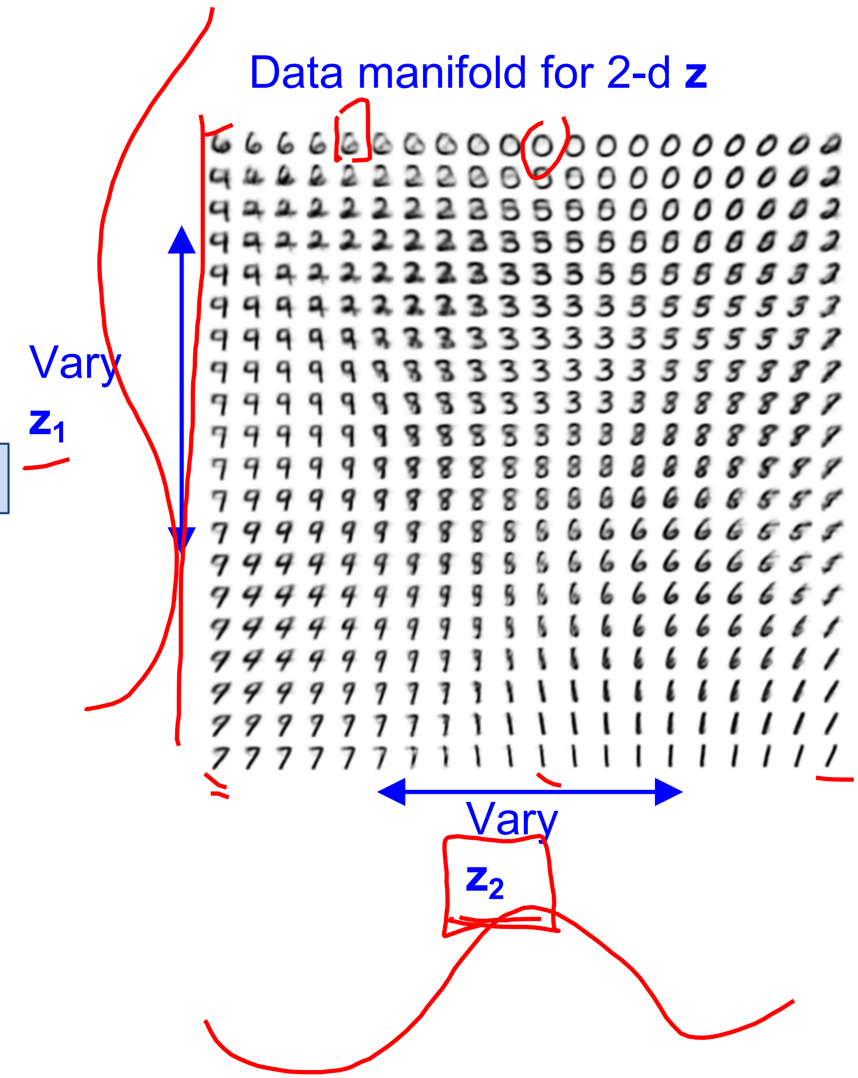
Variational Auto Encoders: Generating Data

Use decoder network. Now sample z from prior!



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

Data manifold for 2-d **z**



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

Variational Auto Encoders: Generating Data

$p(z|\theta)$

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

Degree of smile

Vary $\mathbf{z}_1$

Vary $\mathbf{z}_2$

Head pose

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

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

Variational Auto Encoders: Generating Data

32x32 CIFAR-10

Labeled Faces in the Wild

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