CS 4803 / 7643: Deep Learning

Topics:

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

Dhruv Batra

Georgia Tech

Administrativia

- Project submission instructions released
  - Due: 11/24, 11:59pm
  - Last deliverable in the class
  - ~~Can’t use late days~~ 8 free late days
  - https://www.cc.gatech.edu/classes/AY2021/cs7643_fall/

Administrativia

- HW5 Grades Released
  - Regrade requests close: 11/24, 11:59pm

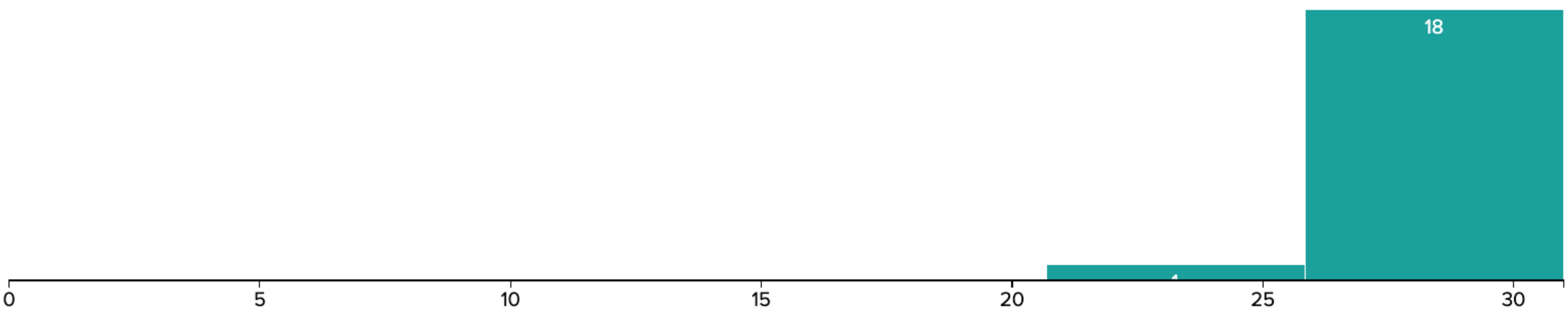
- Grade histogram: 7643
  - Max possible: 39 (regular credit) + 11 (extra credit)

MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
1.0	39.25	47.0	35.77	10.88

Administrativia

- HW5 Grades Released
  - Regrade requests close: 11/24, 11:59pm

- Grade histogram: 4803
  - Max possible: 31 (regular) + 19 (extra credit)



MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
21.0	34.0	40.75	32.64	4.93

Recap from last time

Variational Autoencoders (VAE)

So far...

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

chain

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

$\boxed{D = \{\vec{x}_i\}_{i=1}^{N}}$ $N = \#$ samples

"complex" $\longrightarrow p(\vec{x})$

R.V

"Latent Variables"

Unobserved RV

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

$= \underbrace{\boxed{p(\vec{x} | \vec{z})}}_{\text{Conditional}} \; \underbrace{p(\vec{z})}_{\boxed{\text{Prior}}}$

"simpler"

So far...

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

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

$D = \{ \vec{x}_i \}$

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

$\boxed{\text{VAEs}}$ define intractable density function with latent **z**:

$$\boxed{p_\theta(\vec{x})} = \boxed{\int} \overbrace{p_\theta(z) p_\theta(x|z)}^{p(x,z)} dz$$

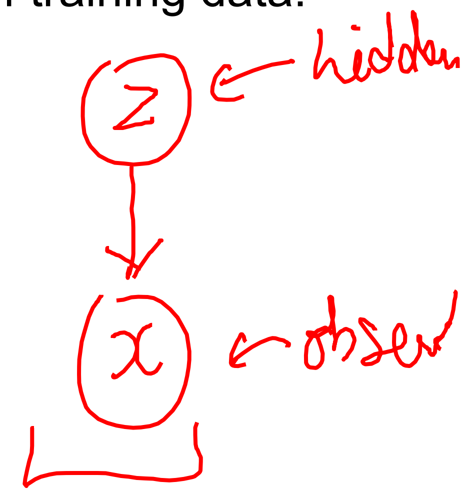
if z is continuous

$\sum_z p_\theta(z) \, p(x|z)$ if z is 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**:

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

Σ

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

GMM

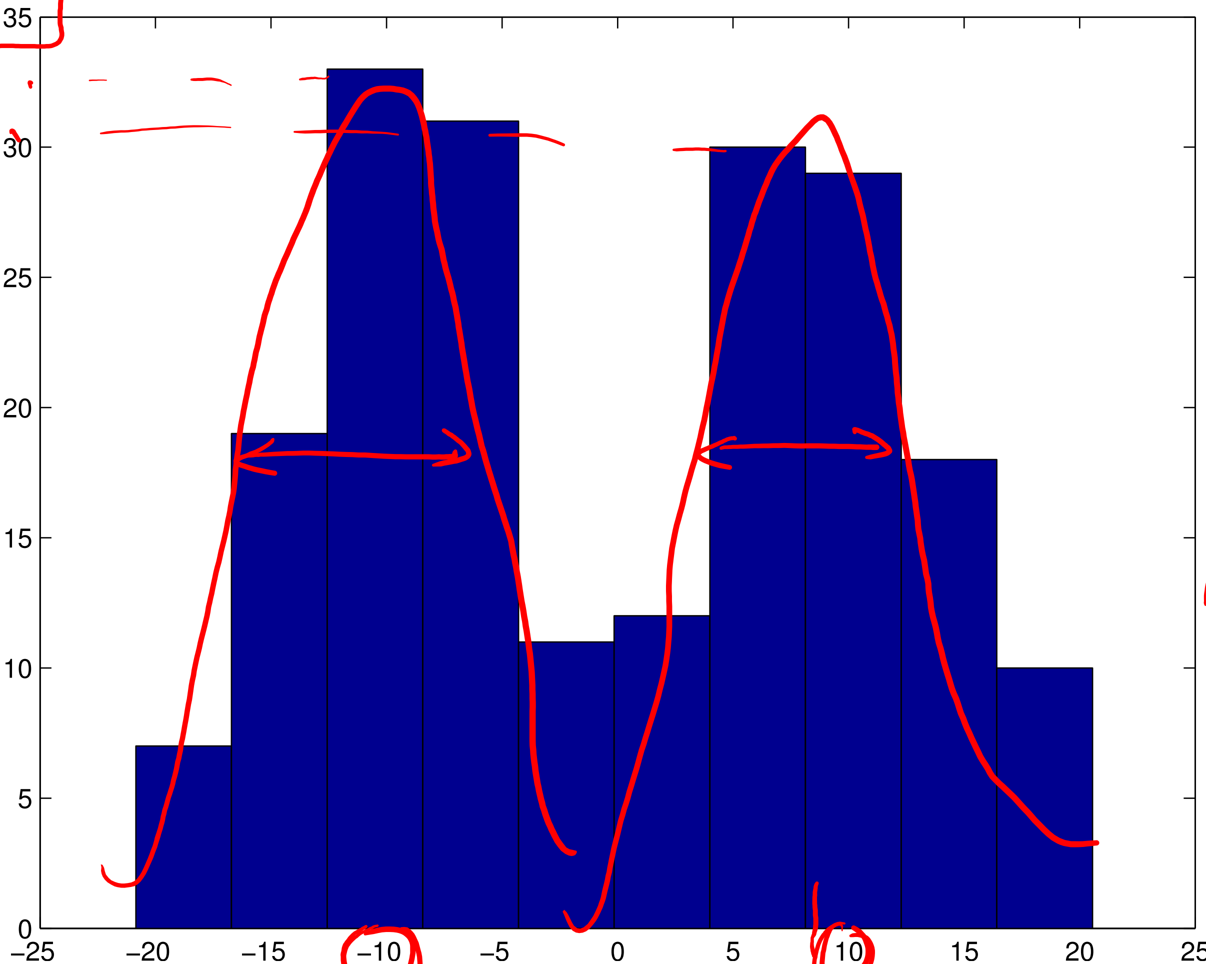
$x \in \mathbb{R}^1$

$\sim p(x)$

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

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

$p(z) = \begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix}$



$z=1$ $z=2$

$p(x \mid z) \sim \mathcal{N}(\mu_1, \sigma_1^2)$ $\mu_1 = -10$

$\mathcal{N}(\mu_2, \sigma_2^2)$ $\mu_2 = +10$

Gaussian Mixture Model

$z \in \{1, \dots, k\}$

$P(x, z)$

z Latent

$\downarrow$

x Observed

$$z \sim \mathrm{Cat}(\vec{\pi}) \qquad \begin{bmatrix} \pi_1 \\ \vdots \\ \pi_k \end{bmatrix}$$

$$\boxed{\boxed{\pi_c} = P_\pi(z=c)}$$

$$\boxed{P(X \mid z=c)} = \boxed{\mathcal{N}(\boxed{\mu_c}, \boxed{\sigma_c^2})}$$

$$= \frac{1}{\sqrt{2\pi\sigma_c^2}} \, e^{-\frac{(x-\mu_c)^2}{2\sigma_c^2}}$$

$$\boxed{P(x, z)} = \boxed{P(x \mid z)} \, P(z)$$

Gaussian Mixture Model

$P(z=c) = \pi_c$

$P(x \mid z) = N(\quad)$

Available from model

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

$= \sum_z p(x \mid z)\, p(z)$ = Marginalization

$\boxed{p(z \mid x)} = \frac{p(z, x)}{p(x)} = \frac{p(x \mid z)\, p(z)}{\boxed{\sum_z} (\quad) (\quad)}$

= (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

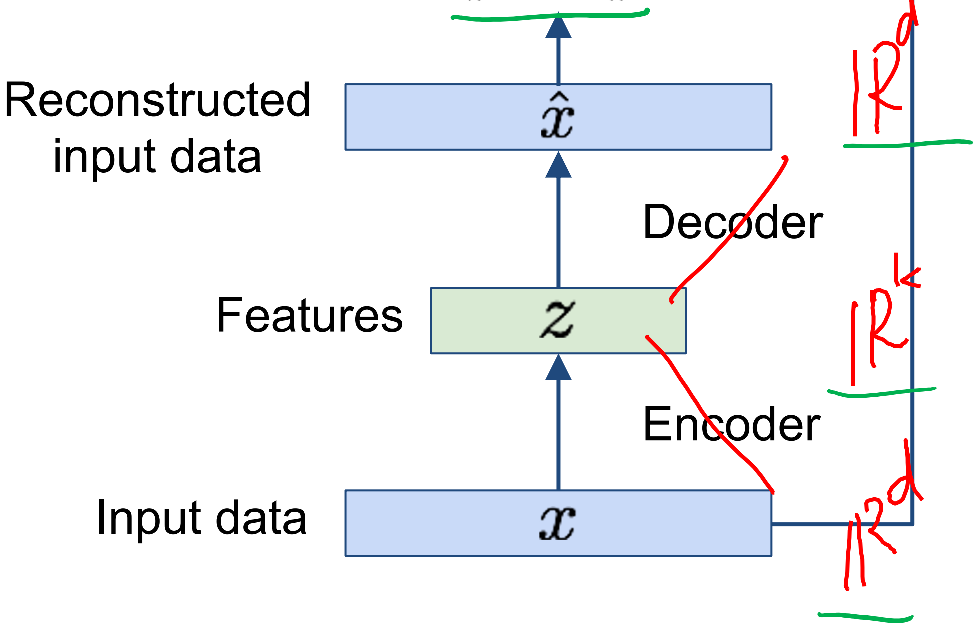
Autoencoders

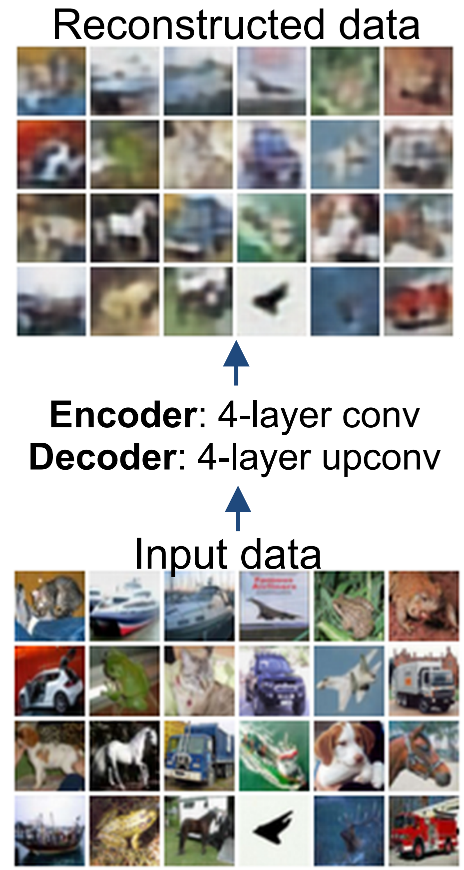
Train such that features can be used to reconstruct original data

Doesn't use labels!

L2 Loss function:

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





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

Autoencoders

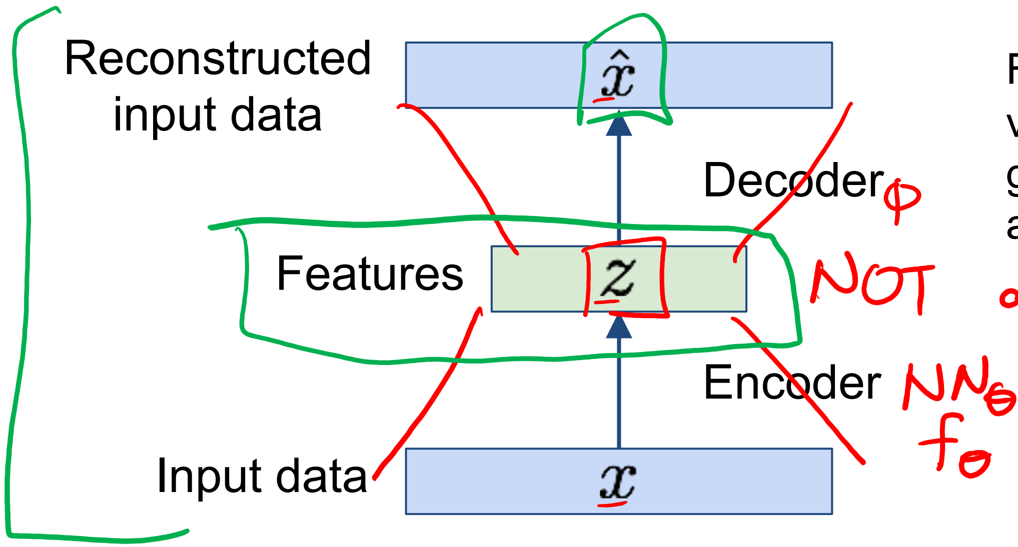
$z = f_\theta(x)$

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

$p(z|x)$

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

VAE



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!

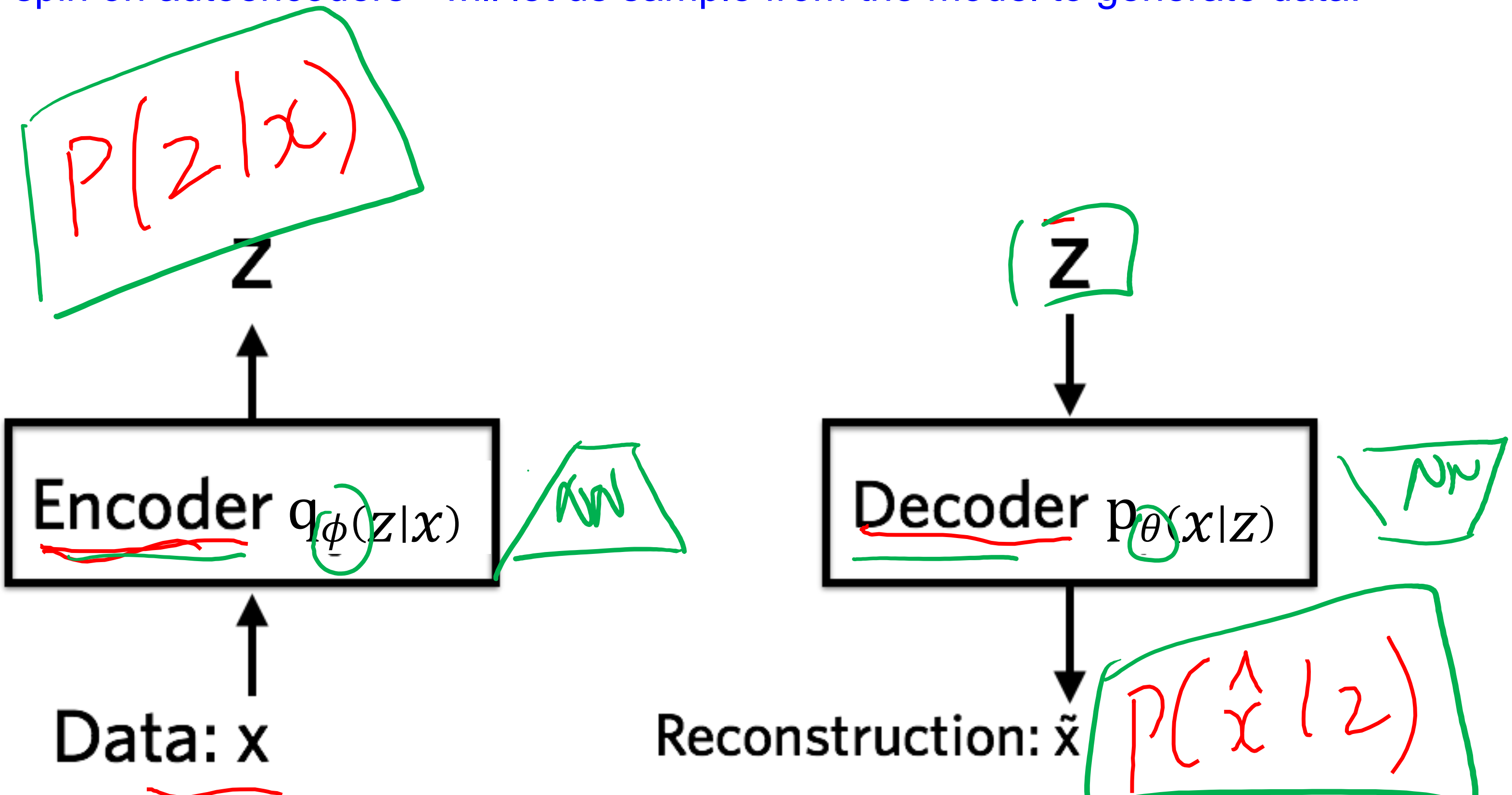


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

Plan for Today

- VAEs
  - Variational Inference
  - Evidence Based Lower Bound
  - Putting it all together
- Next time:
  - Reparameterization trick for optimizing VAEs

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.

$p(z)$

$E_{\boxed{p(z)}}[f(z)] \rightarrow E_{\boxed{q(z)}}[f(z)]$

Key problem

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

"complex" $P(z)$

$\downarrow$

"simple" $q(z)$

$\int$

$p(z)$

$\{q^*_{\phi^*}(z)\}$

$$\min_{q \in \{q_\phi(z)\}} \underbrace{d(p, q)}_{KL(\quad)}$$

$\boxed{KL(p\|q) \text{ "right" "hard"}}$

$\boxed{KL(q\|p) \text{ "wrong" "easy"}}$

"simple" distributions $\{q_\phi(z)\}$

Intuition

"error"

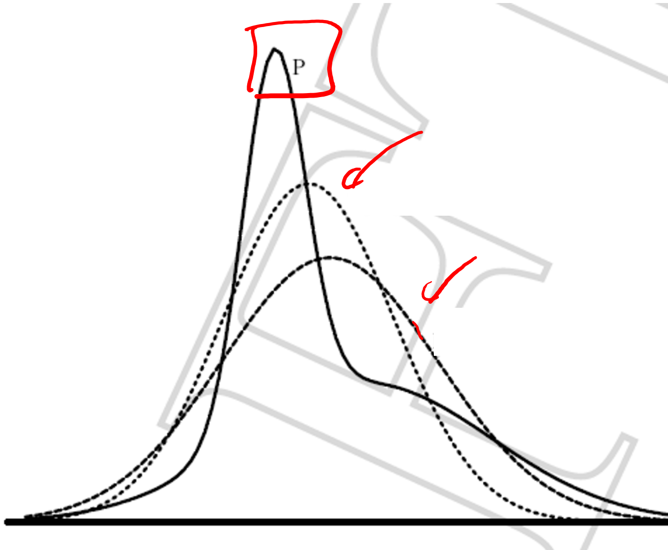
$$KL(p \| q) = \sum_z \boxed{p(z)} \log \boxed{\frac{p(z)}{q(z)}}$$

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

"Support"

Find simple approximate distribution

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



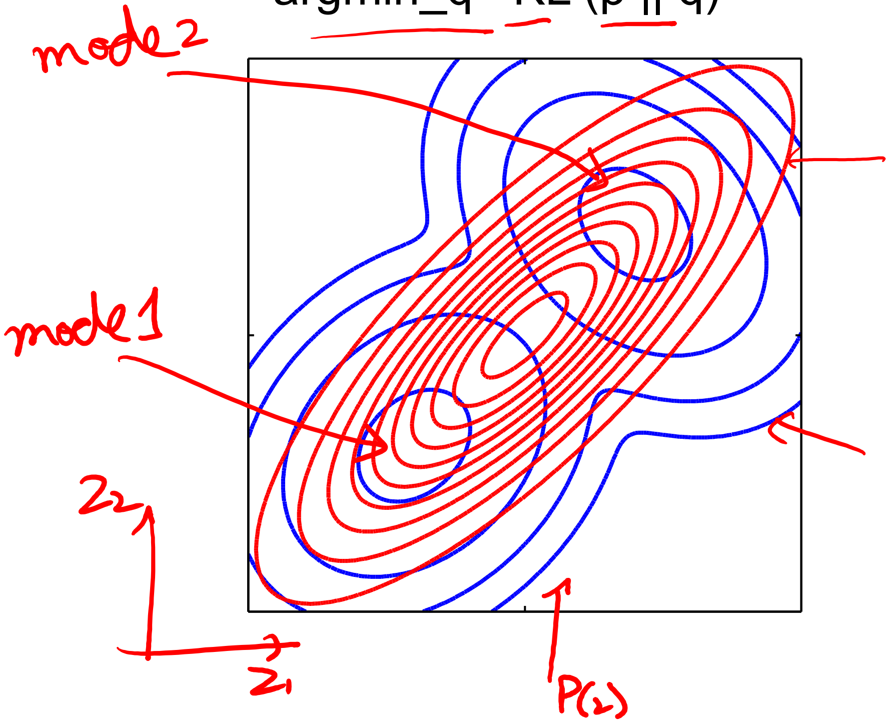
- D(p||q)
  - true distribution p defines support of diff.
  - the “correct” direction
  - will be intractable to compute

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

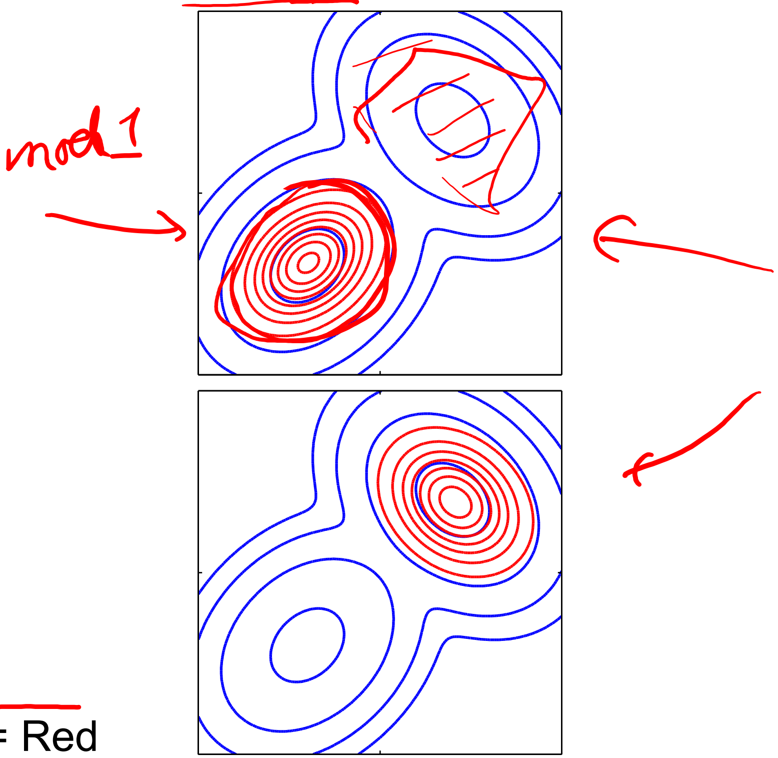
Example 2

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

argmin_q KL (p || q)



argmin $\boxed{q}$ KL (q || p)



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

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

$$D = \{\vec{x}_i\}_{i=1}^{N}$$

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

log. likelihood

$$
\begin{aligned}
ll(\theta : \mathcal{D}) &= \log \prod_{i=1}^{N} P(\mathbf{x}_i \mid \theta) \\
&= \sum_{i=1}^{N} \log P(\mathbf{x}_i \mid \theta) \\
&= \cancel{\sum_{i=1}^{N}} \log \boxed{\sum_{\mathbf{z}}} P(\mathbf{x}_i, \mathbf{z} \mid \theta)
\end{aligned}
$$

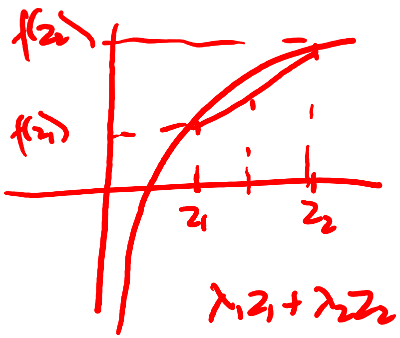
$$\log \sum_{z} P(\vec{x}_i \mid \theta) P(z \mid \vec{x}_i, \theta)$$

$$\log E_{\boxed{P(z \mid \vec{x}_i, \theta)}} \Big[\underbrace{P(\vec{x}_i \mid \theta)}_{f(.)} \Big]$$

"complex" $\rightarrow$ simple

Applying Jensen's inequality

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



$$f(\lambda_1 z_1 + \lambda_2 z_2) \geq \lambda_1 f(z_1) + \lambda_2 f(z_2) \quad [\text{concavity of } f]$$

$$f\left(\sum_{i=1}^{k} \lambda_i z_i\right) \geq \sum_{i=1}^{k} \lambda_i f(z_i) \qquad \boxed{\lambda_1, \ldots \lambda_k}$$

$\lambda_1, \lambda_2 \geq 0$

$\lambda_1 + \lambda_2 = 1$

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

$z \rightarrow g(z)$

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

Applying Jensen's inequality

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

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

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

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

$$\max_\theta \; ll(\theta) \geq F(\theta, Q_i)$$

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

Variational Lower Bound

Evidence-based LB (ELBO)

Evidence Lower Bound

- Define potential function F(θ,Q):

(VAE) objective

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

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

(GMM objective)

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

ELBO: Factorization #1 (GMMs)

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

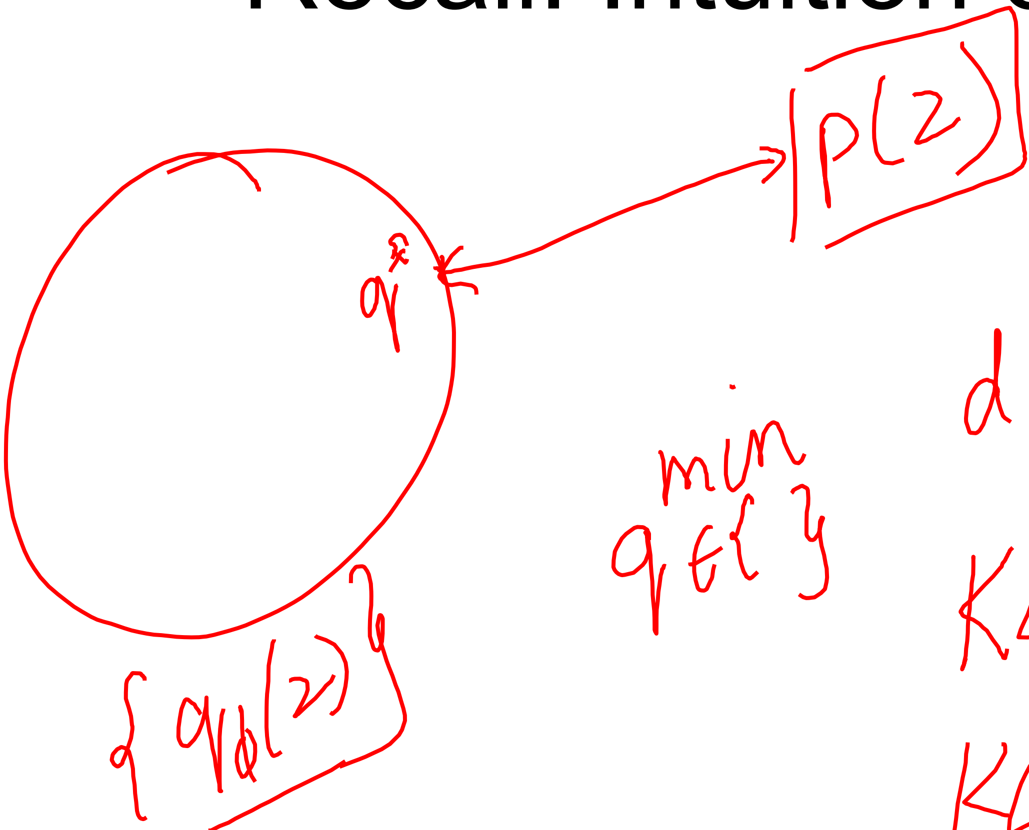
$$\boxed{ll(\theta : \mathcal{D}) \geq F(\theta, Q_i)} = \cancel{\sum_{i=1}^{N}} \sum_{\mathbf{z}} Q_i(\mathbf{z}) \log \frac{P(\mathbf{x}_i, \mathbf{z} \mid \theta)}{Q_i(\mathbf{z})}$$

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

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

$$\max_{\theta, Q_i} F(\theta, Q_i) = ll(\theta) - \boxed{KL(\quad)}$$

Recall: Intuition of VI



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

$$d(p, q)$$

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

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

ELBO: Factorization #1 (GMMs)

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

$\max_{\theta, Q_i} F$

- $\boxed{\text{EM}}$ 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

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

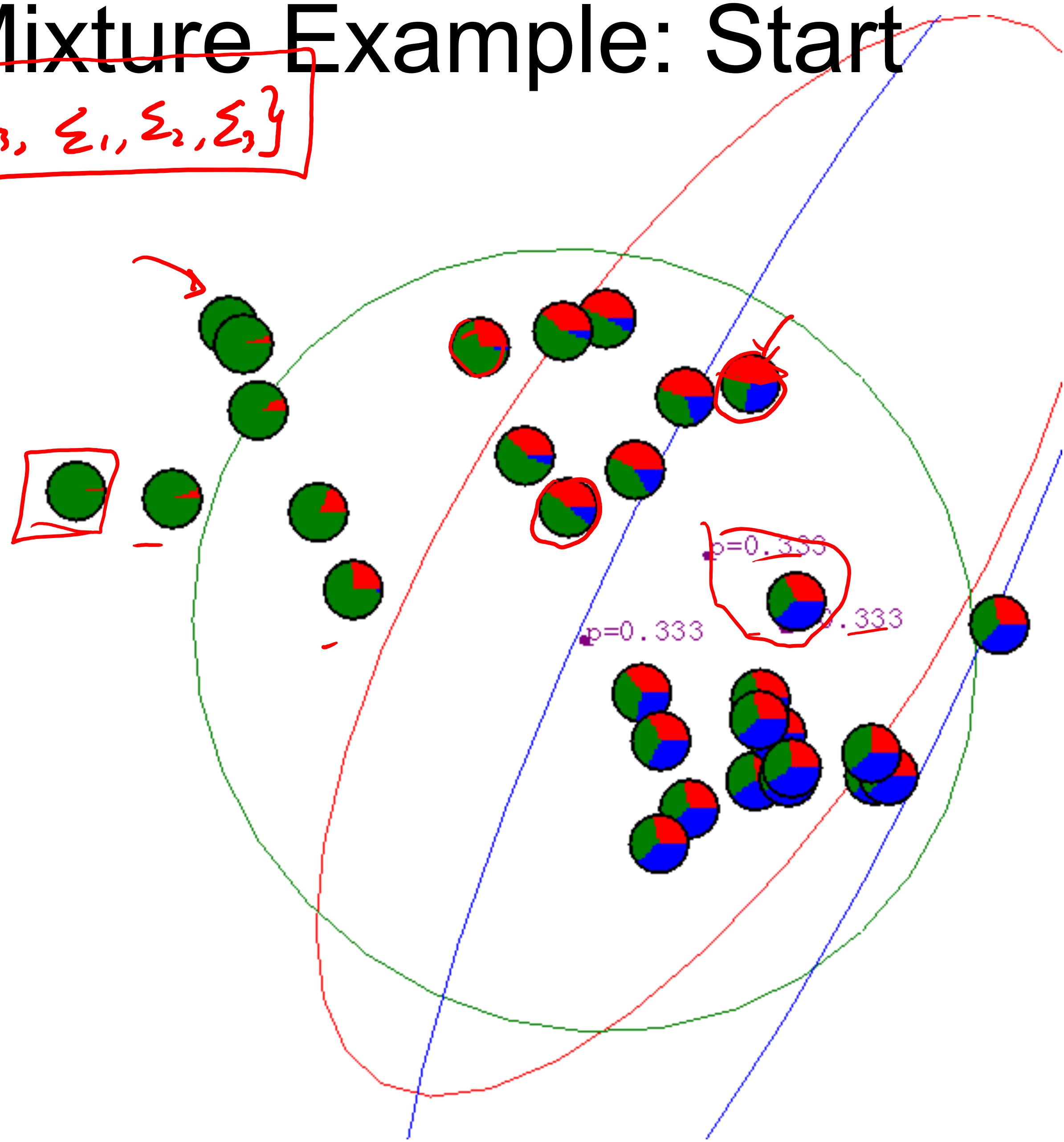
- **M-step**: Fix $Q_i^{(t)}$, maximize F over θ
  - maximize expected likelihood under $Q_i(z)$
  - Corresponds to weighted dataset:
    - <$\mathbf{x}_1$,$\mathbf{z}$=1> with weight $Q^{(t+1)}(\mathbf{z}=1|\mathbf{x}_1)$
    - <$\mathbf{x}_1$,$\mathbf{z}$=2> with weight $Q^{(t+1)}(\mathbf{z}=2|\mathbf{x}_1)$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

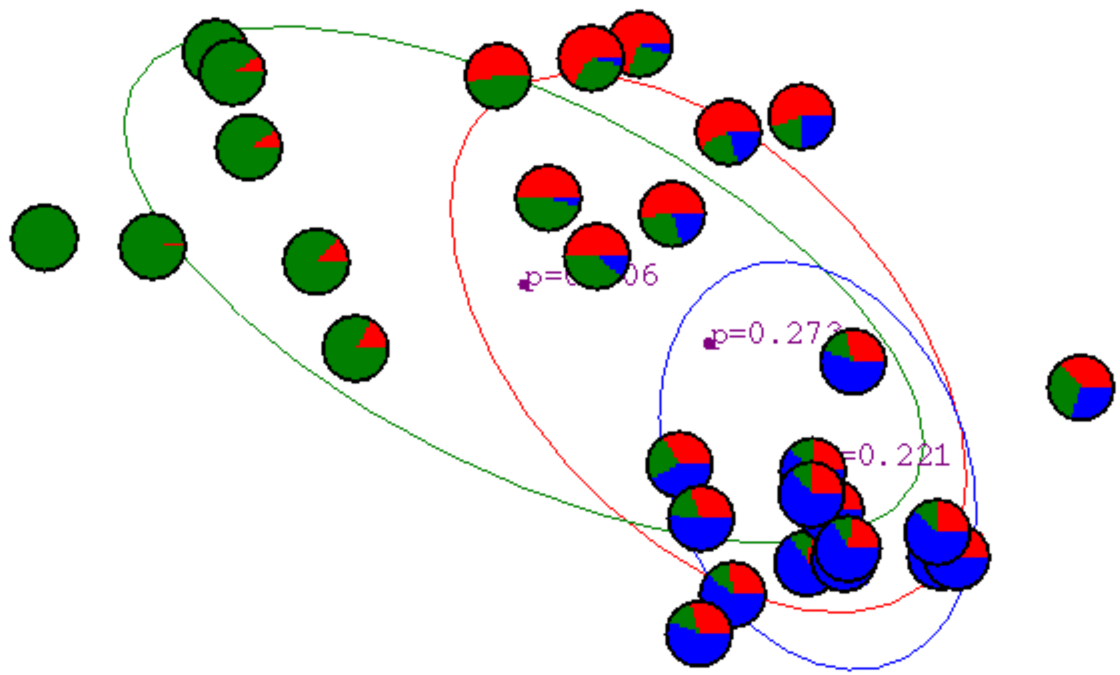
$\boxed{\theta = \{\pi_1, \pi_2, \pi_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3\}}$

$\boxed{q_i(2)} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}_K$

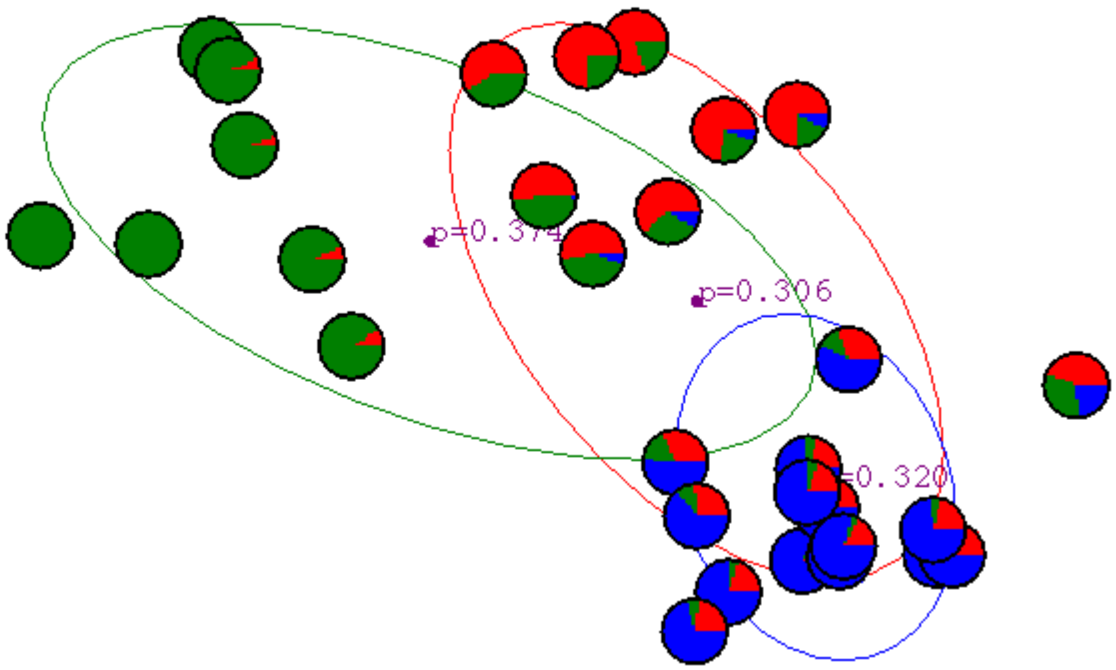
$i = 1:N$



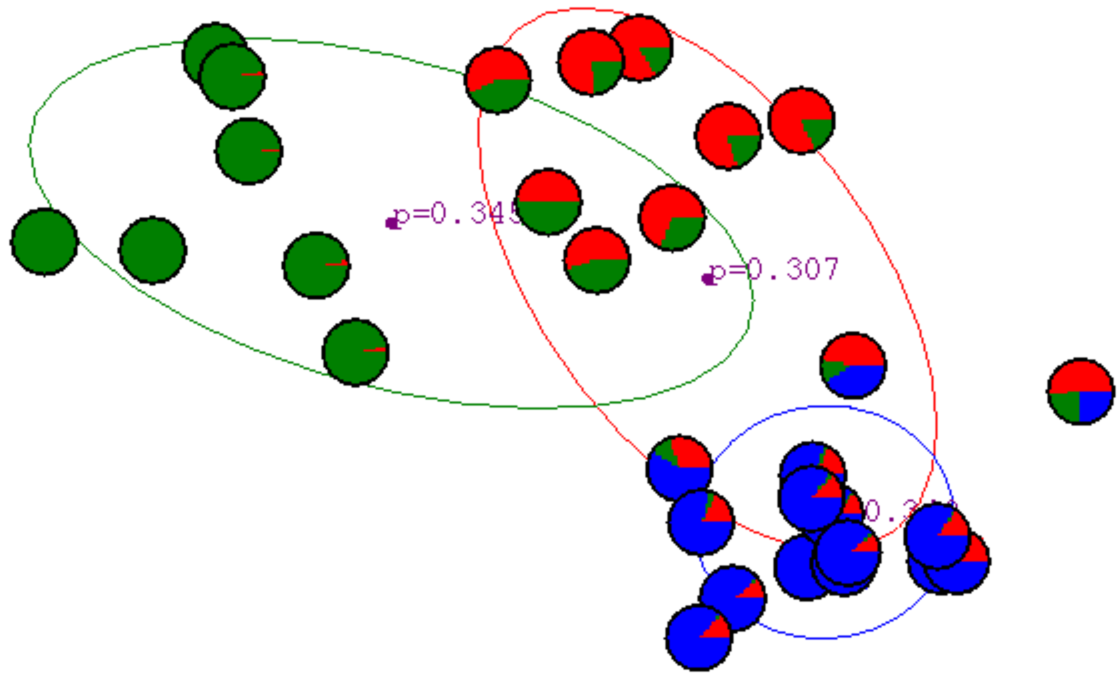
After 1st iteration



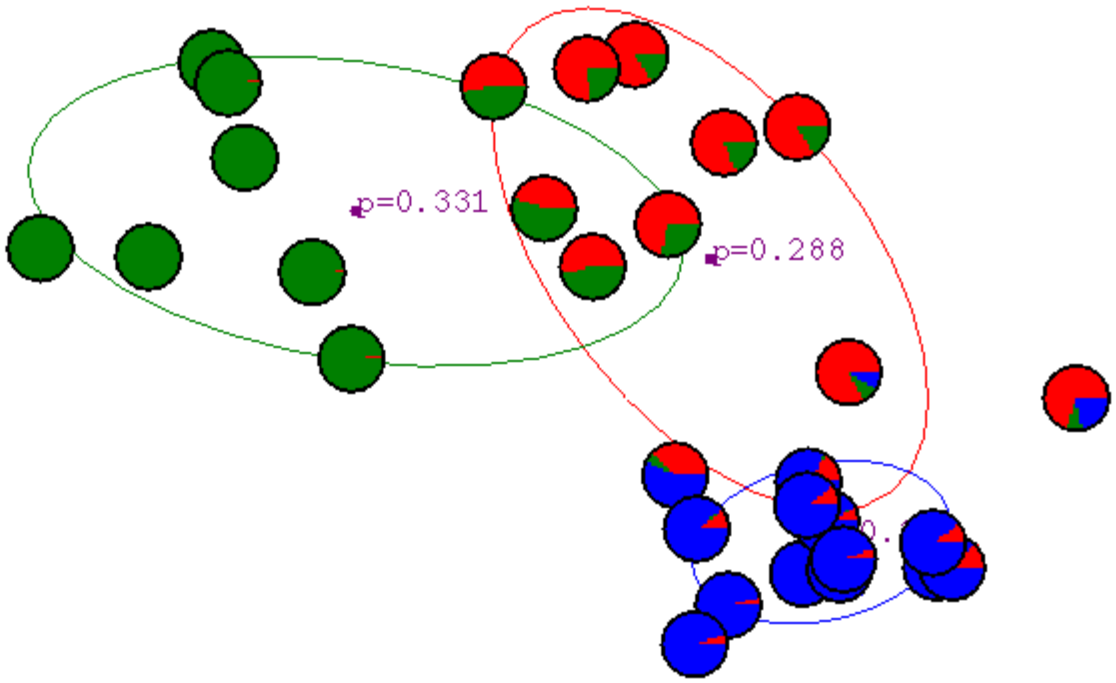
After 2nd iteration



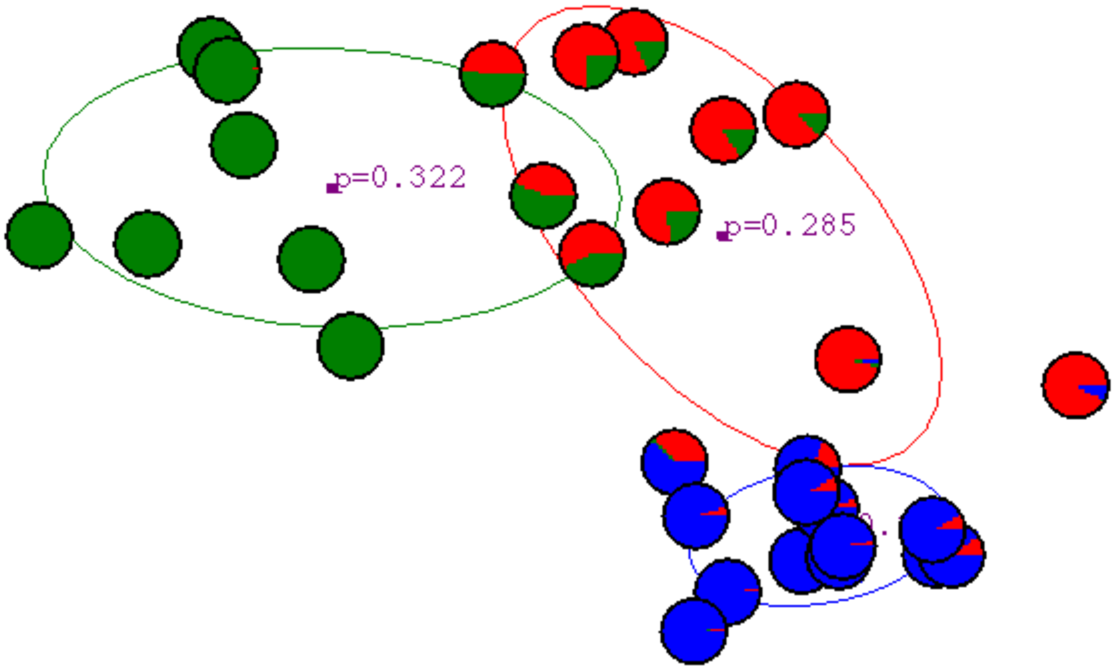
After 3rd iteration



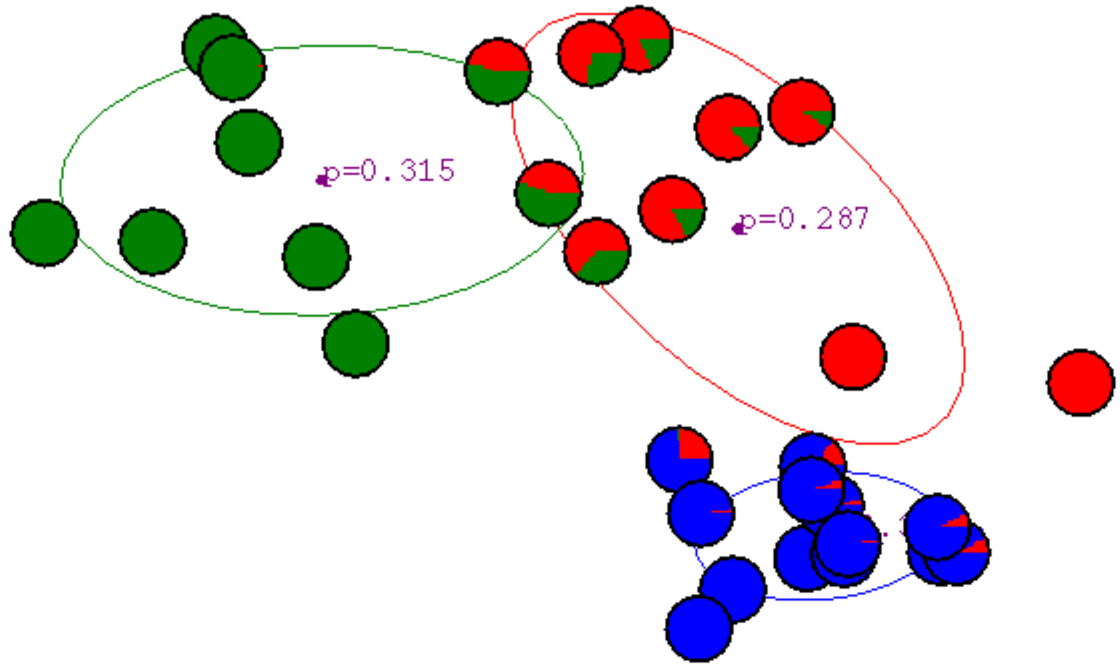
After 4th iteration



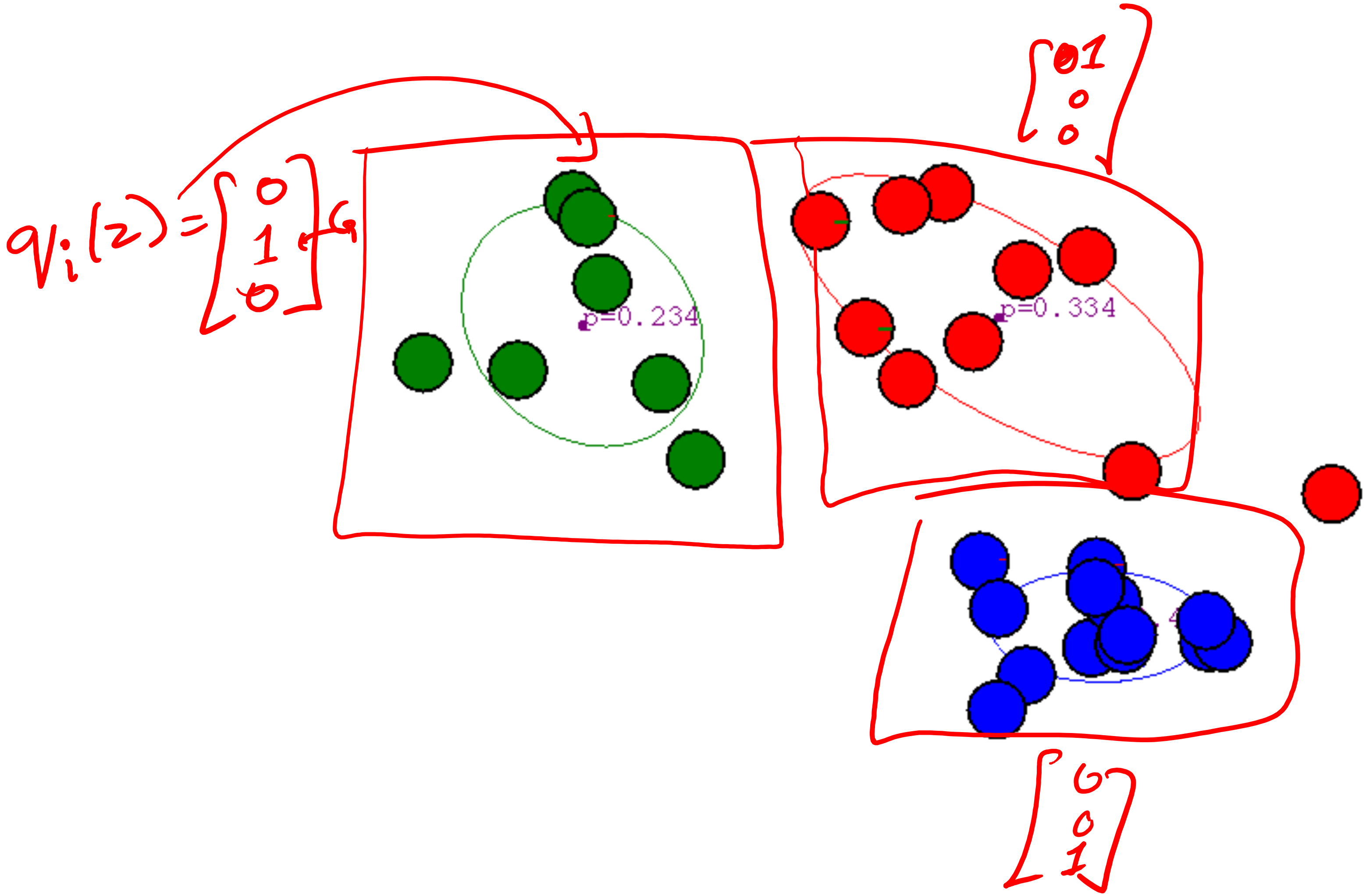
After 5th iteration



After 6th iteration



After 20th iteration



ELBO: Factorization #2 (VAEs)

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

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

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

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

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

"Explain the data"

"Regularizer"

"Be simple"

(VAE objective)

maximize

Variational Auto Encoders

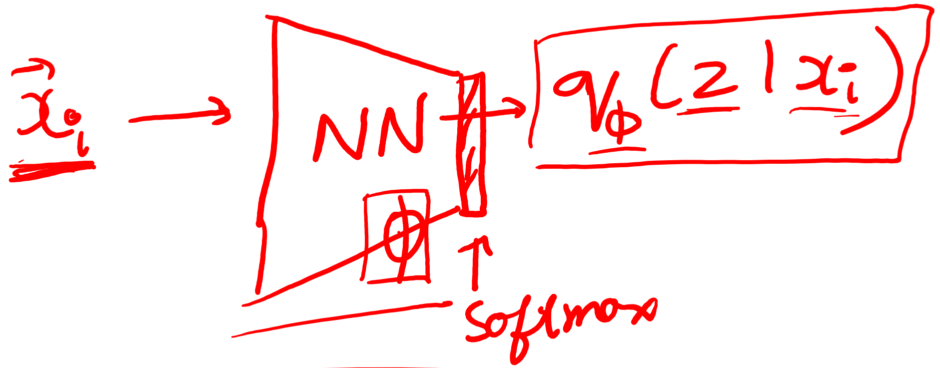
VAEs are a combination of the following ideas:

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

Amortized Inference Neural Networks

"Let's shove a NN into a Bayesian inf prob"

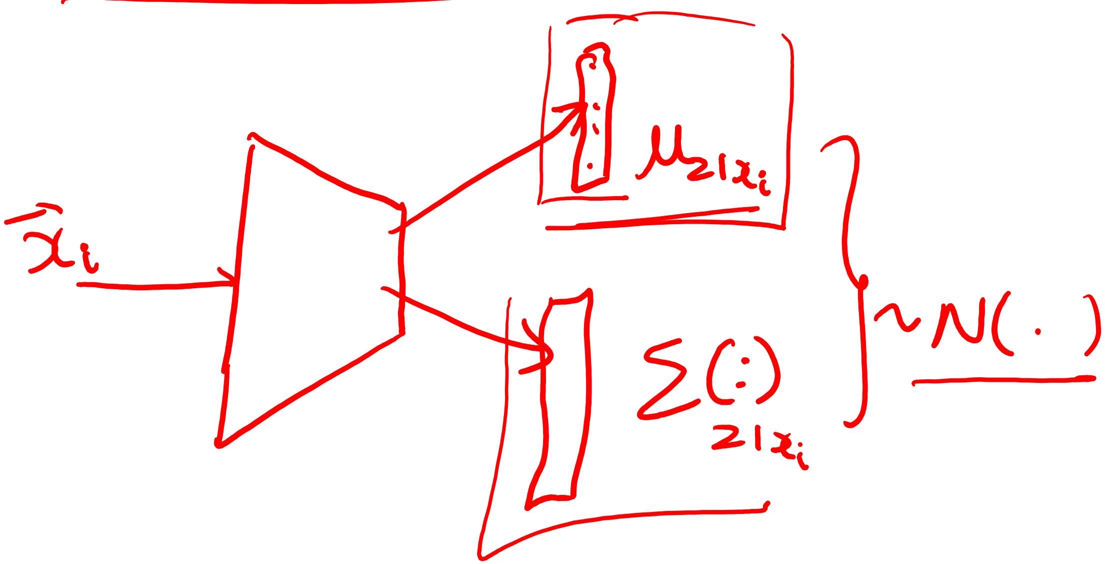
$$\boxed{Q_{\boxed{i}}(z)} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.7 \end{bmatrix}$$



if z discrete

---

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



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

Variational Autoencoders

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

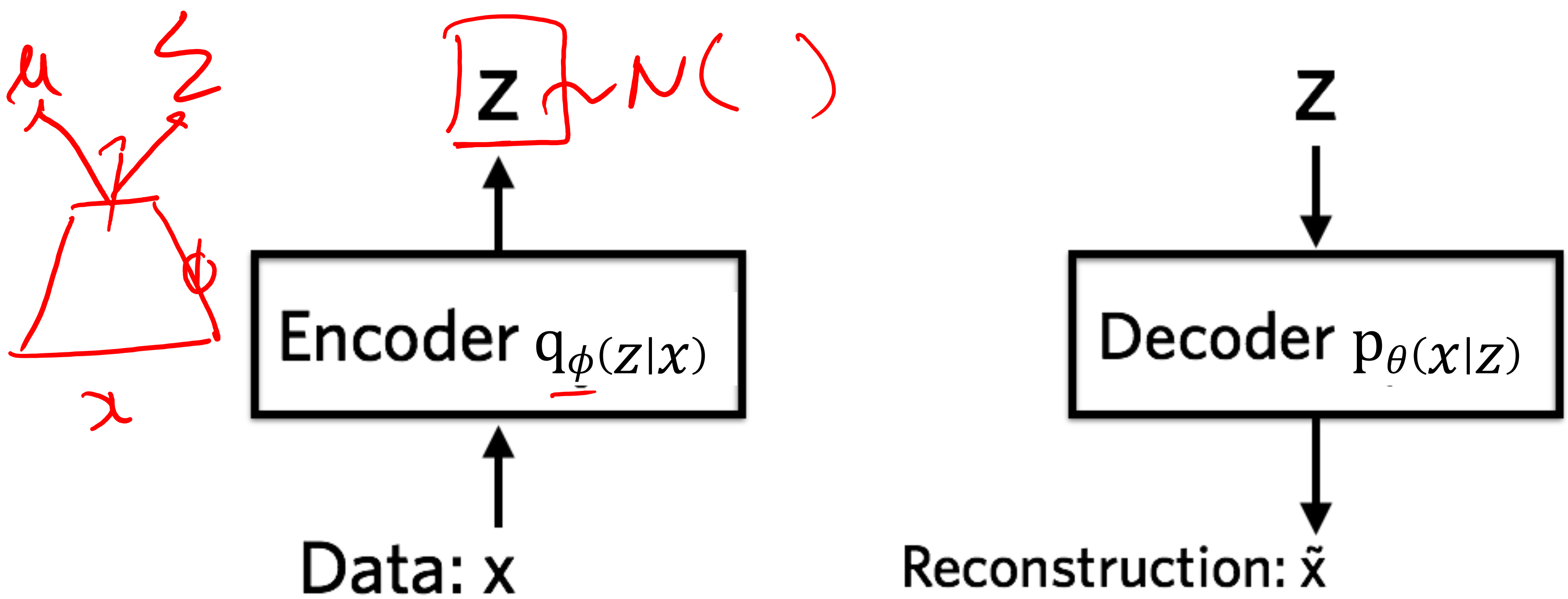
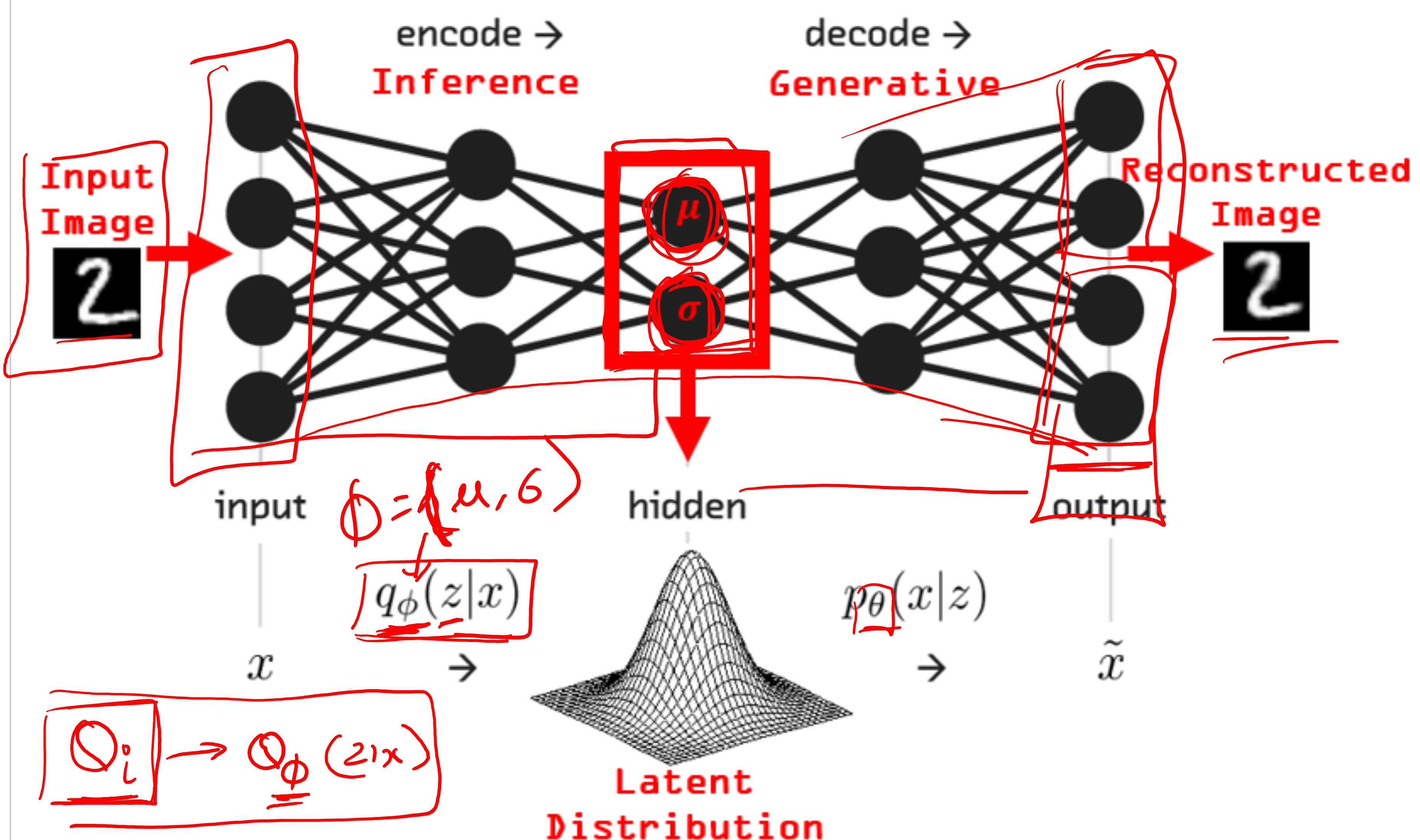


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

VAEs



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

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

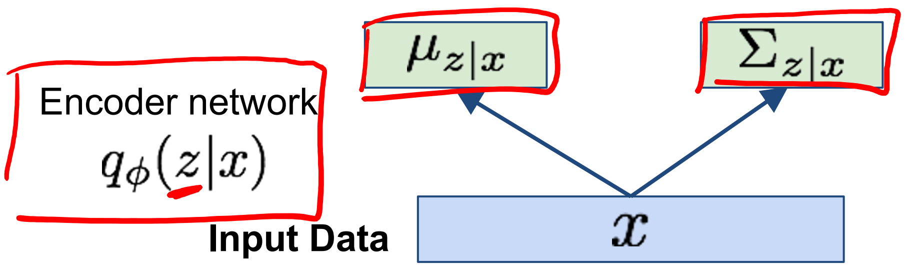
Let's look at computing the bound (forward pass) for a given minibatch of input data

Input Data x

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

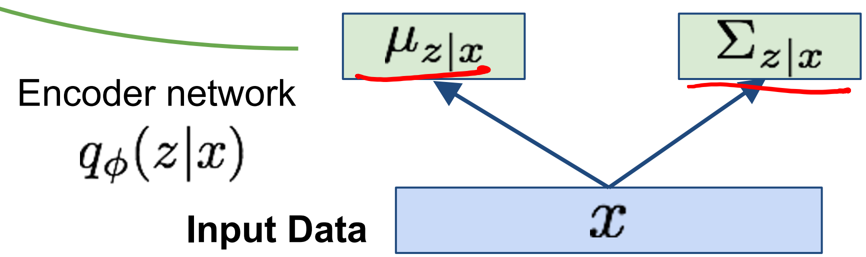


Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

Make approximate posterior distribution close to prior

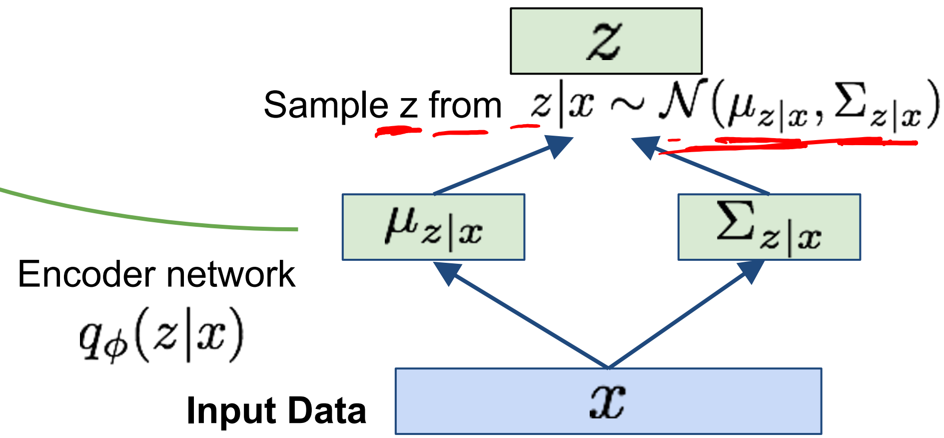


Variational Auto Encoders

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

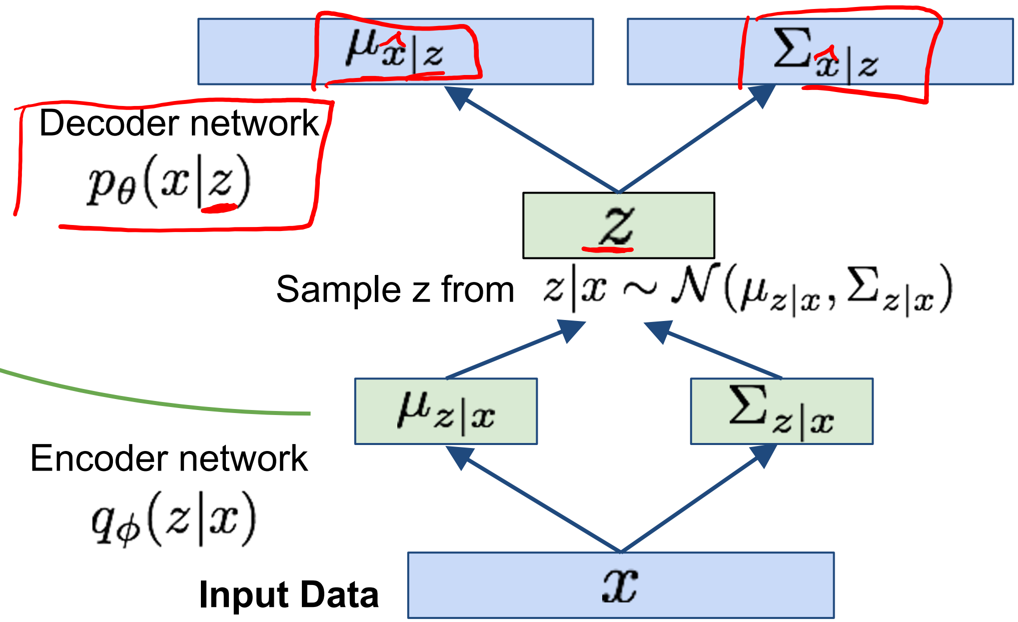


Variational Auto Encoders

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



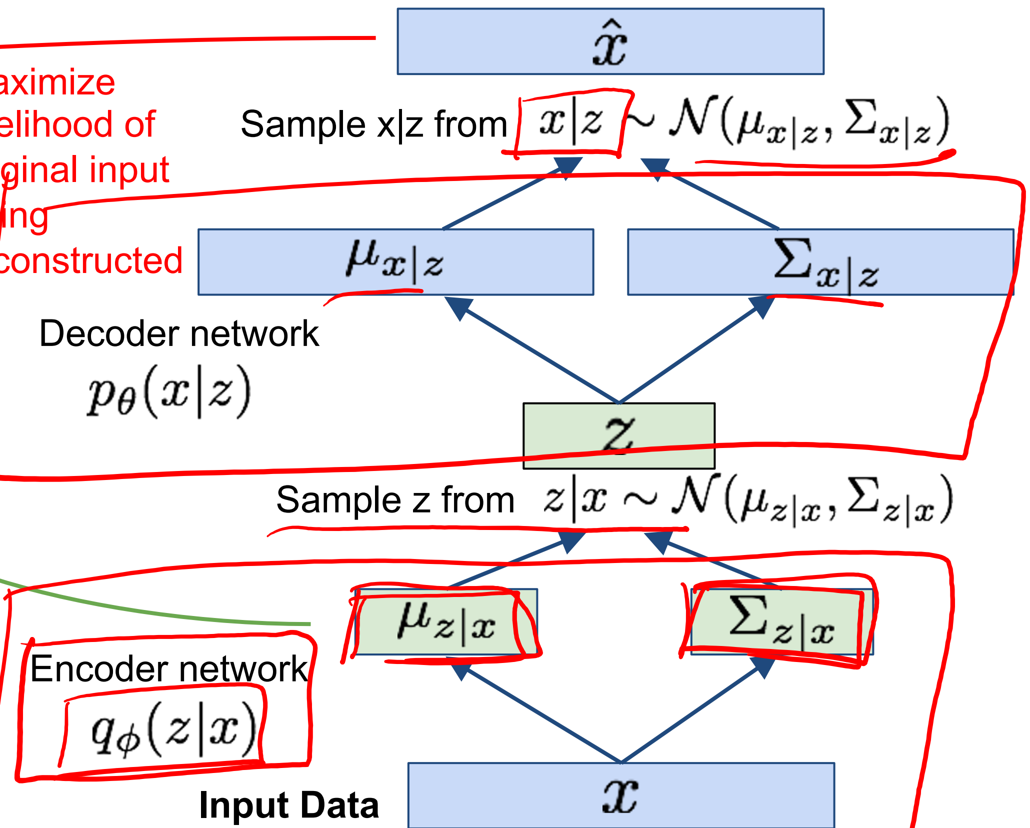
Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

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

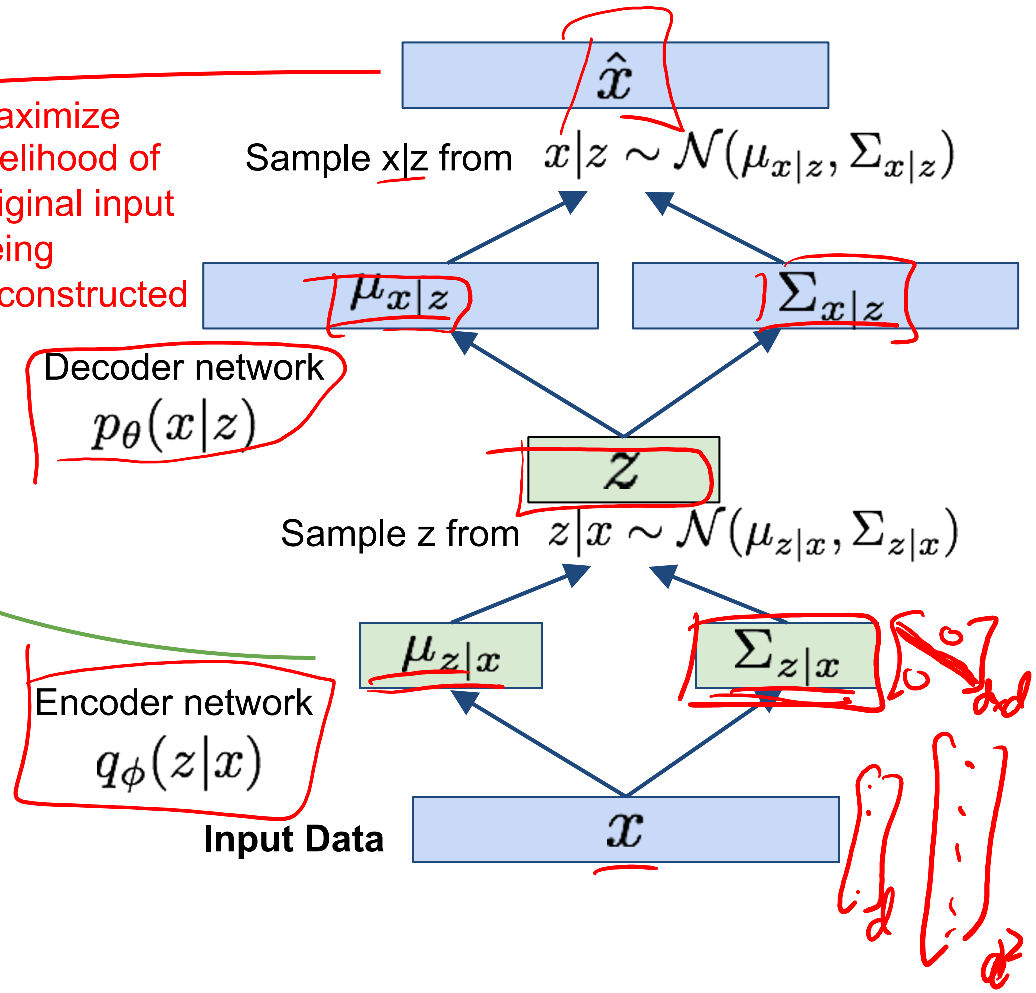
$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \boxed{\theta, \phi})}$$

Maximize likelihood of original input being reconstructed

$$\frac{\partial(\quad)}{\partial \phi} \qquad \frac{\partial(\quad)}{\partial \theta}$$

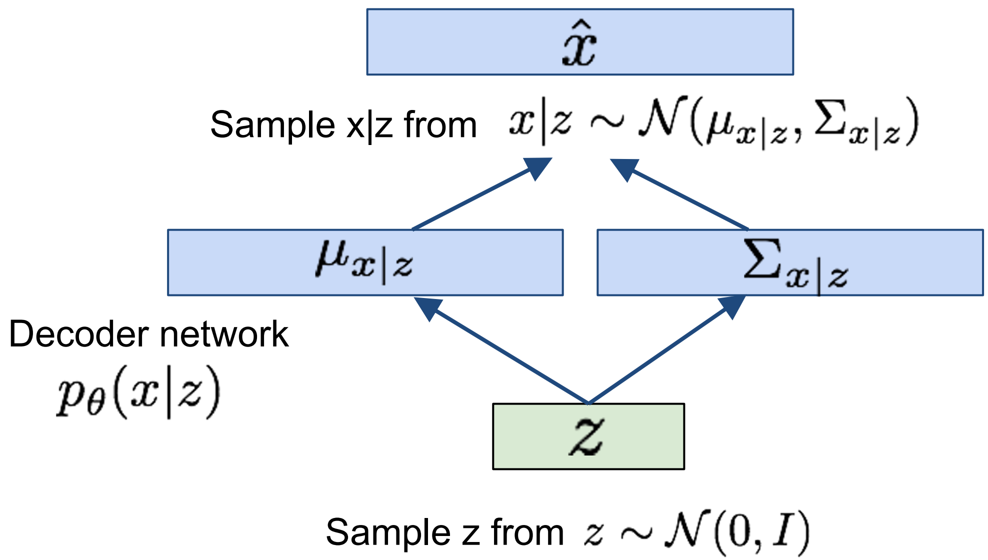
Make approximate posterior distribution close to prior

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



Variational Auto Encoders: Generating Data

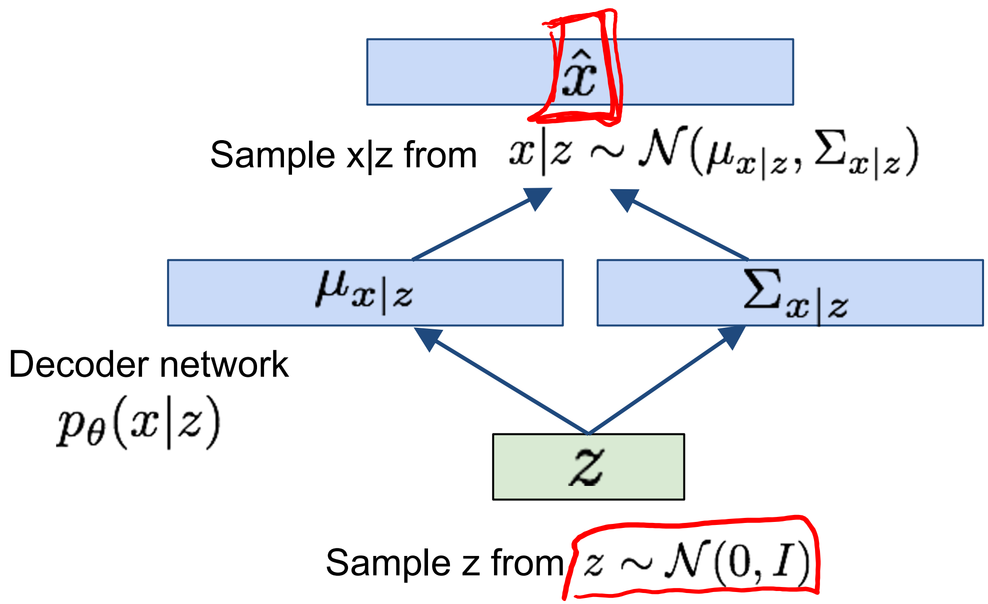
Use decoder network. Now sample z from prior!

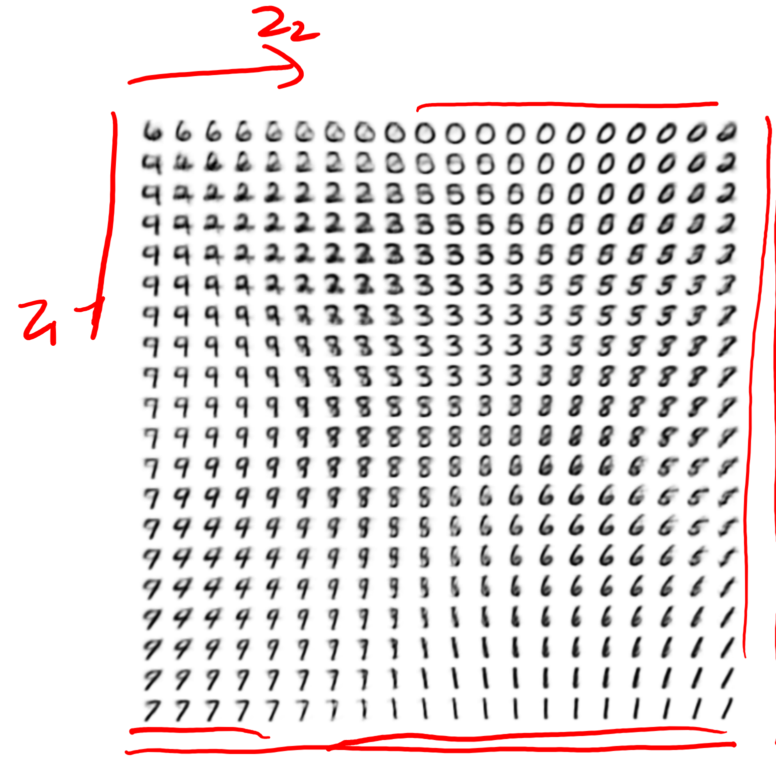


Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Variational Auto Encoders: Generating Data

Use decoder network. Now sample z from prior!

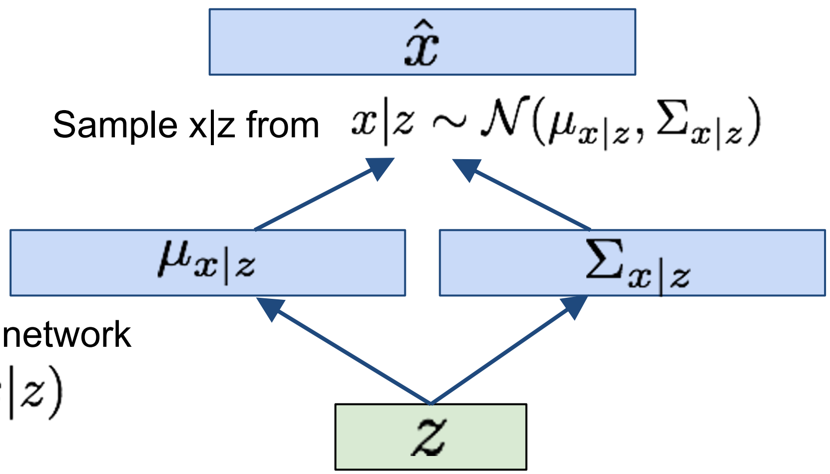




Kingma and Welling, “Auto-Encoding Variational Bayes”, ICLR 2014

Variational Auto Encoders: Generating Data

Use decoder network. Now sample z from prior!



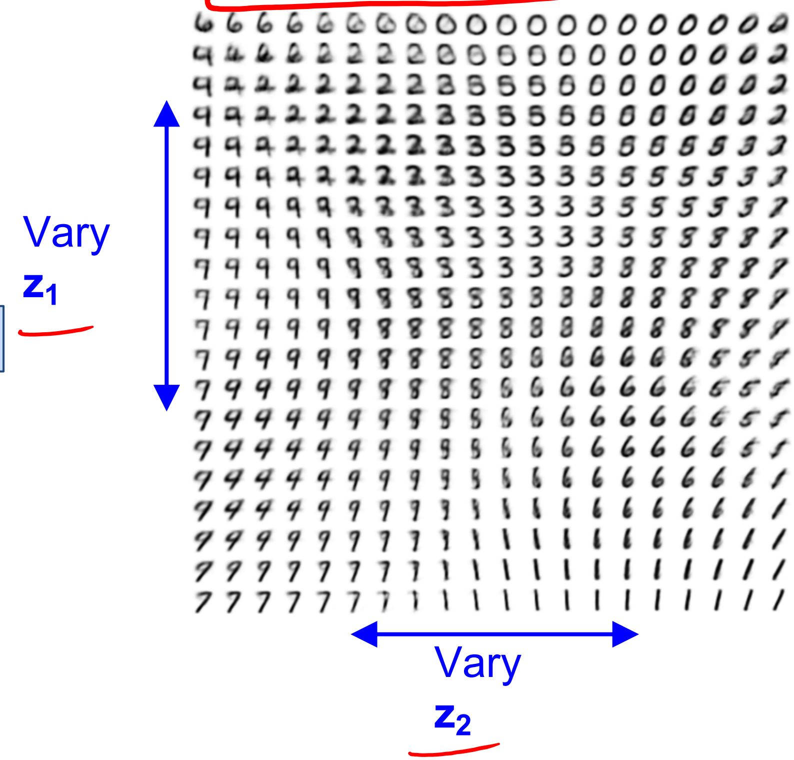
Decoder network

$p_\theta(x|z)$

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

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

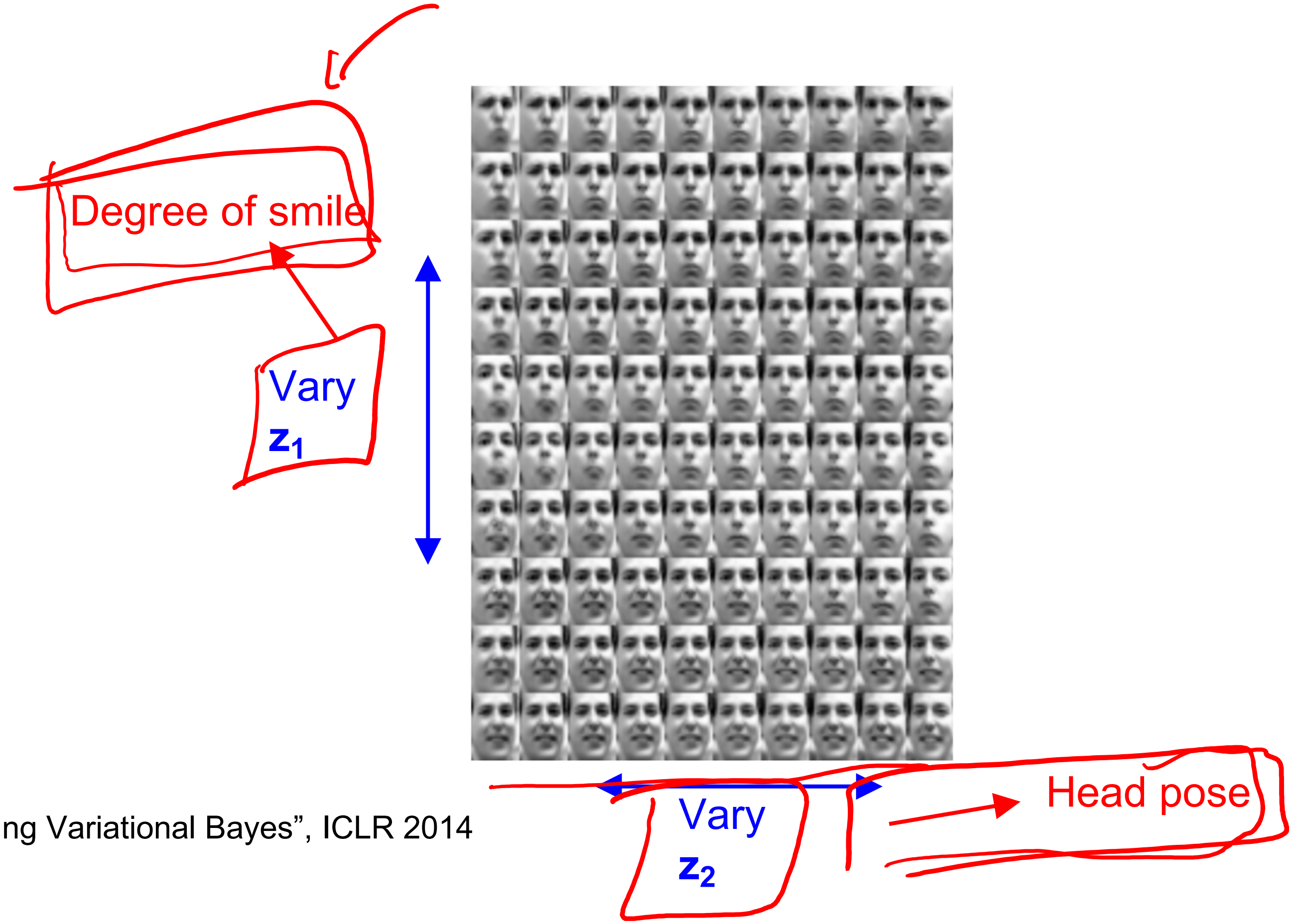
Data manifold for 2-d **z**



Variational Auto Encoders: Generating Data

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

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



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 $\mathbf{z_1}$

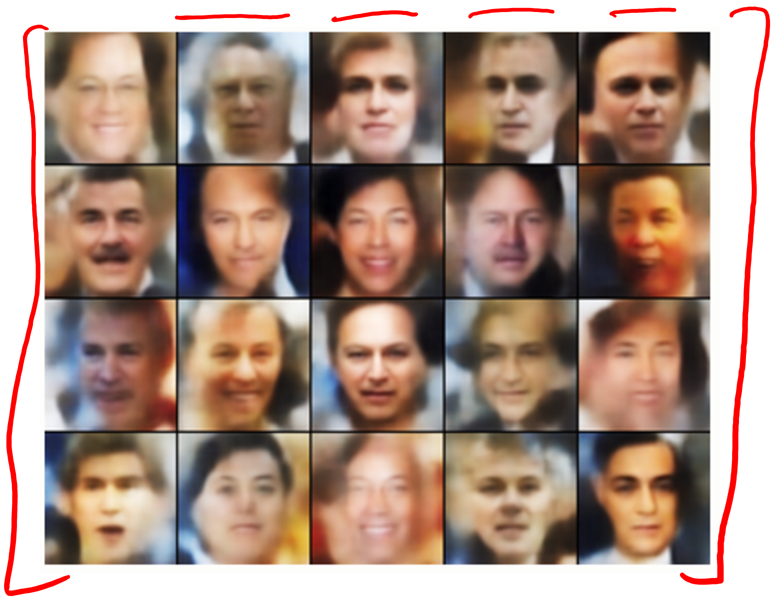
Vary $\mathbf{z_2}$

Head pose

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

Variational Auto Encoders: Generating Data

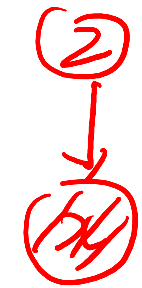
32x32 CIFAR-10



Labeled Faces in the Wild

Variational Autoencoders

$P(x, z)$



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

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