CS 4803 / 7643: Deep Learning

Topics:

- Variational Auto-Encoders (VAEs)
  - Reparameterization trick

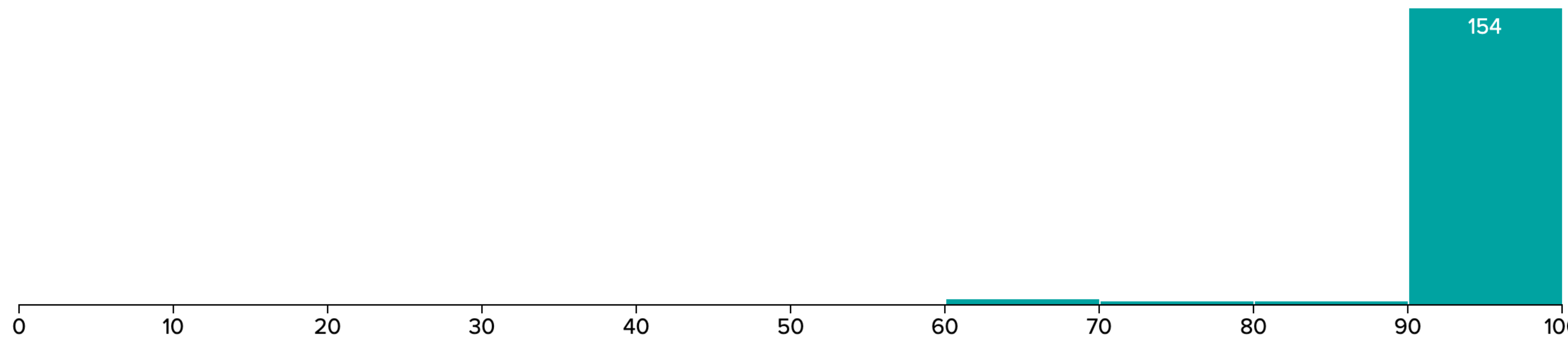
Dhruv Batra

Georgia Tech

Administrativia

- HW4 Grades Released
  - Regrade requests close: 12/03, 11:55pm
  - Please check solutions first!

- Grade histogram: 7643
  - Max possible: 100 (regular credit) + 40 (extra credit)

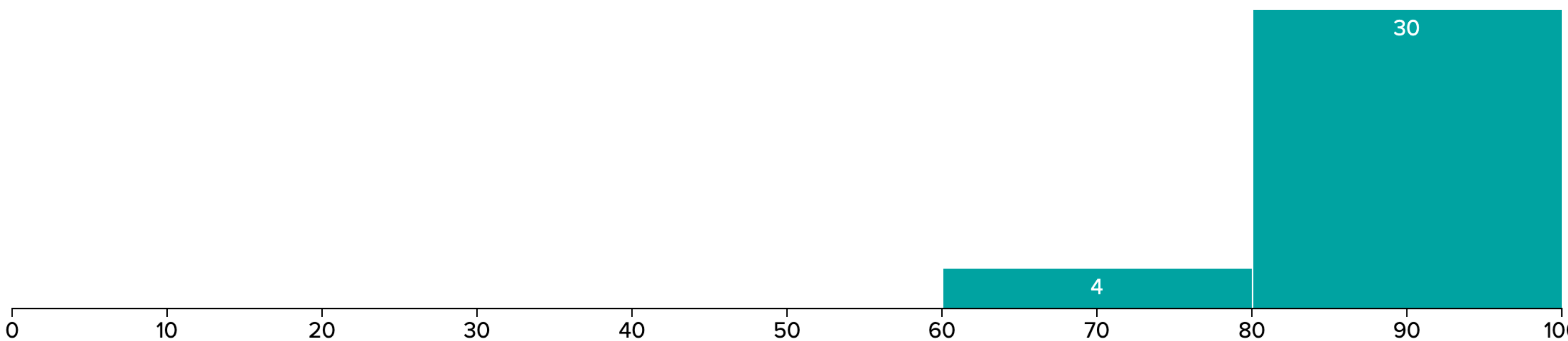


MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
63.0	111.0	138.0	110.94	12.77

Administrativia

- HW4 Grades Released
  - Regrade requests close: 12/03, 11:55pm
  - Please check solutions first!

- Grade histogram: 4803
  - Max possible: 100 (regular credit) + 40 (extra credit)



MINIMUM	MEDIAN	MAXIMUM	MEAN	STD DEV
65.5	106.75	122.0	102.32	14.32

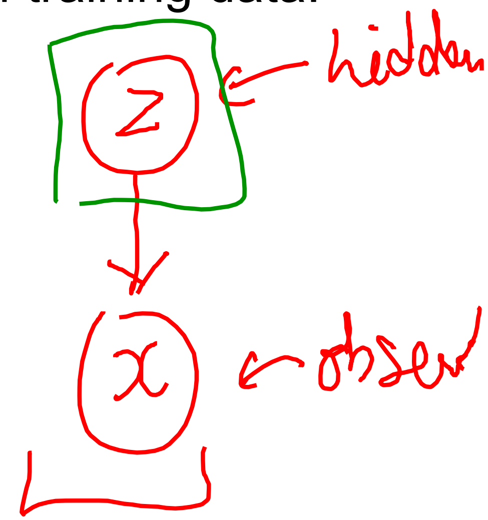
Recap from last time

Variational Autoencoders (VAE)

So far...

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

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



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

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

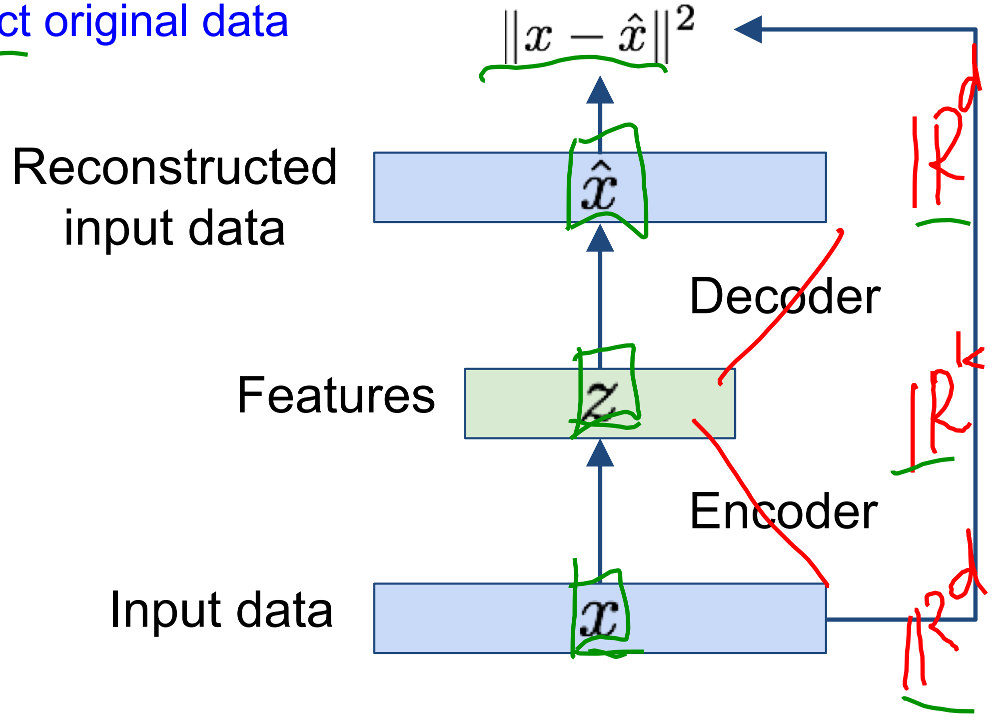
4. “Reparameterization” Trick

Autoencoders

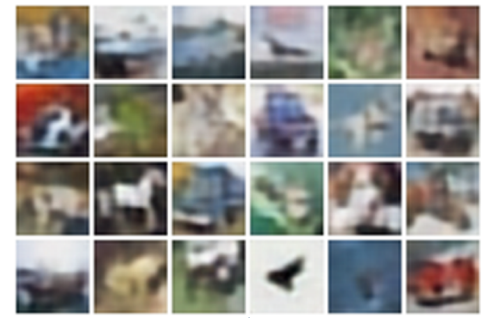
Train such that features can be used to reconstruct original data

L2 Loss function:

Doesn't use labels!



Reconstructed data



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

Input data

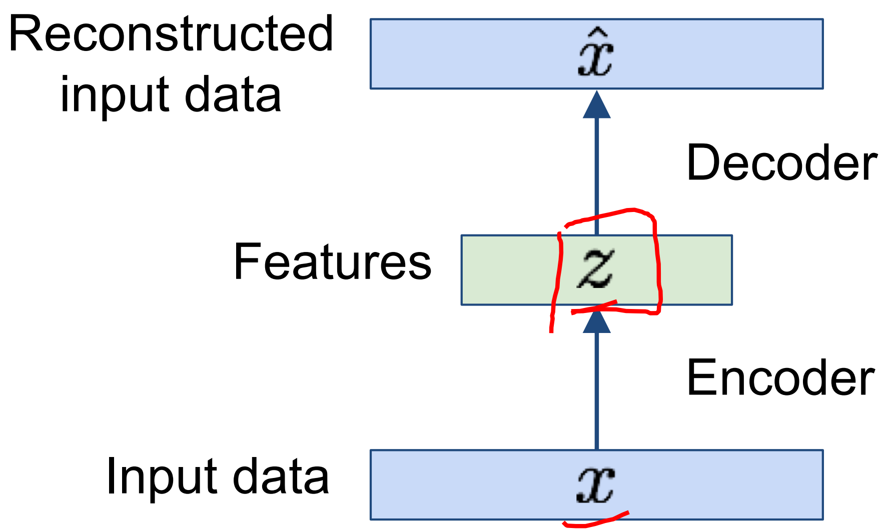
Autoencoders

$z = f_\theta(x)$

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

$p(z|x)$

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



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!

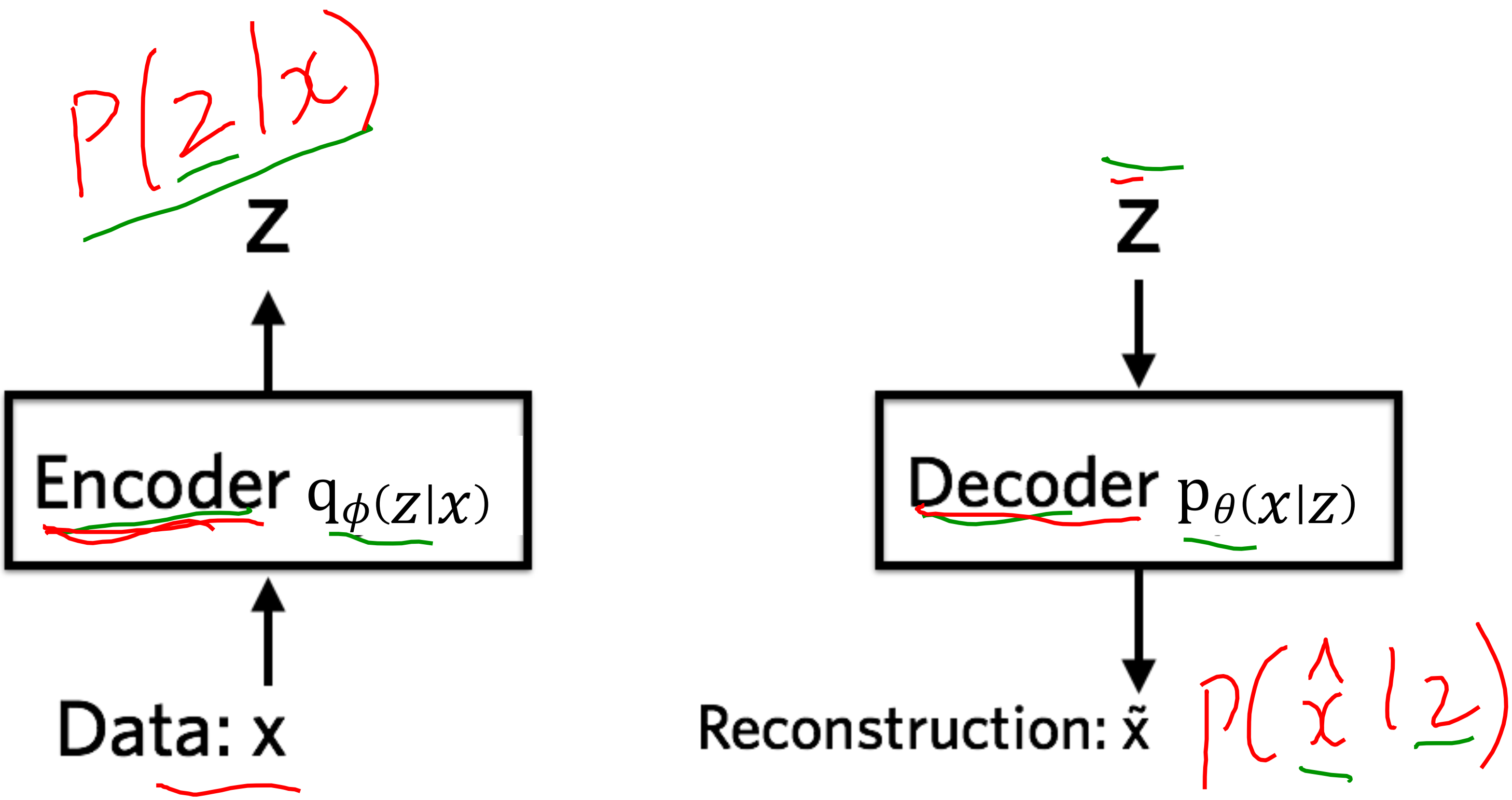


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

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

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

$q_i(z)$

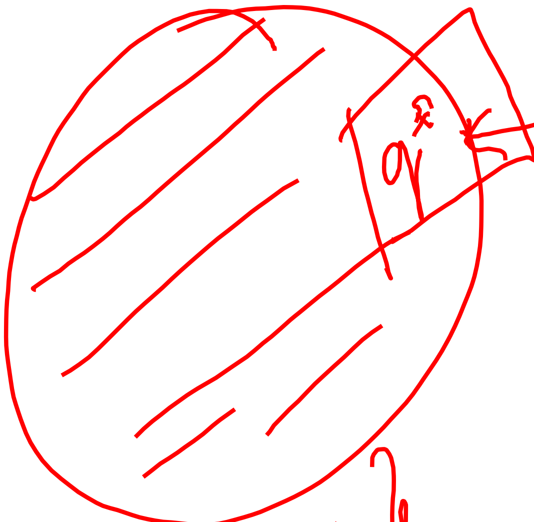
$\int$ Hard

$z \to x$

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



$\boxed{p(z)}$

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

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

$d(p, q)$

$$\boxed{\begin{array}{l} KL(p(z) \,\|\, q(z)) \\ KL(q(z) \,\|\, p(z)) \end{array}}$$

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

error

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

Support

The general learning problem with missing data

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

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

z

x

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

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

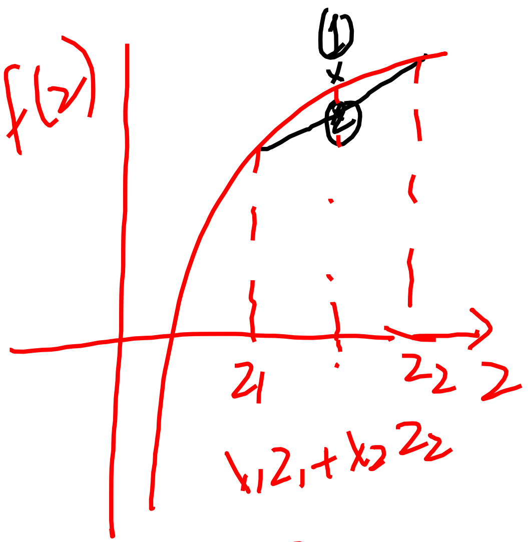
$P(x_i \mid \theta)\, P(z \mid x_i, \theta)$

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

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

Jensen's inequality

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



$\lambda_1, \lambda_2 \geq 0$

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

1 ≥ 2

$f(\lambda_1 z_1 + \lambda_2 z_2) \geq \lambda_1 f(z_1) + \lambda_2 f(z_2)$ $\quad 2 \to K$

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

$\equiv f(E[z]) \geq E_{P(z)}[f(z)]$ $\quad z \to g(z)$

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

Applying Jensen's inequality

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

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

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

$$\max_\theta l \geq \max_{\theta, Q_i} F$$

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

Variational Lower Bound

Evidence Lower Bound (ELBO)

$z \to x$

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

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

ELBO: Factorization #1 (GMMs)

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

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

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

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

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

ELBO: Factorization #2 (VAEs)

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

$\max_{\theta, Q_i}$

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

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

(VAEs)

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

"Explain the data"

"Regulariser"

Be simple

Variational Auto Encoders

VAEs are a combination of the following ideas:

1. Auto Encoders

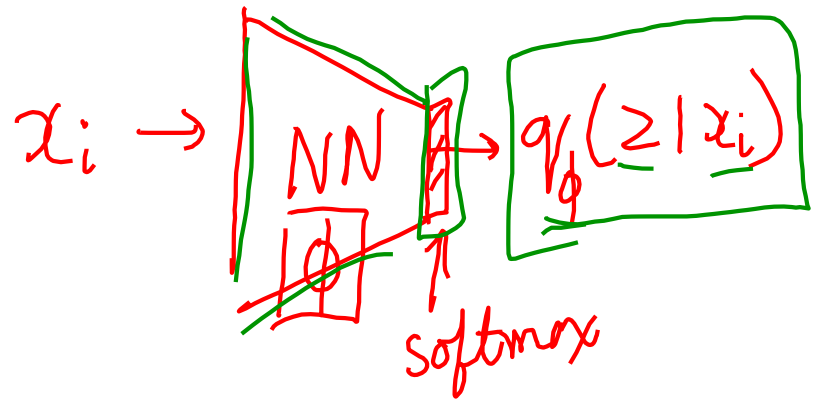
2. Variational Approximation
   - Variational Lower Bound / ELBO

3. Amortized Inference Neural Networks

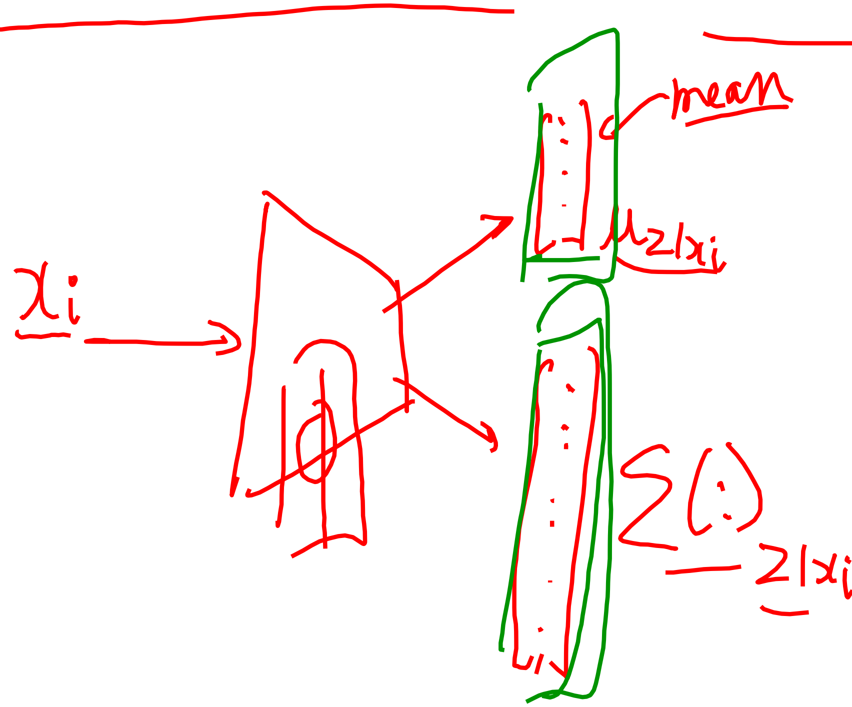
4. "Reparameterization" Trick

Amortized Inference Neural Networks

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



$z \sim Cat()$



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

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

$\sim p(z \mid x_i)$

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

VAEs

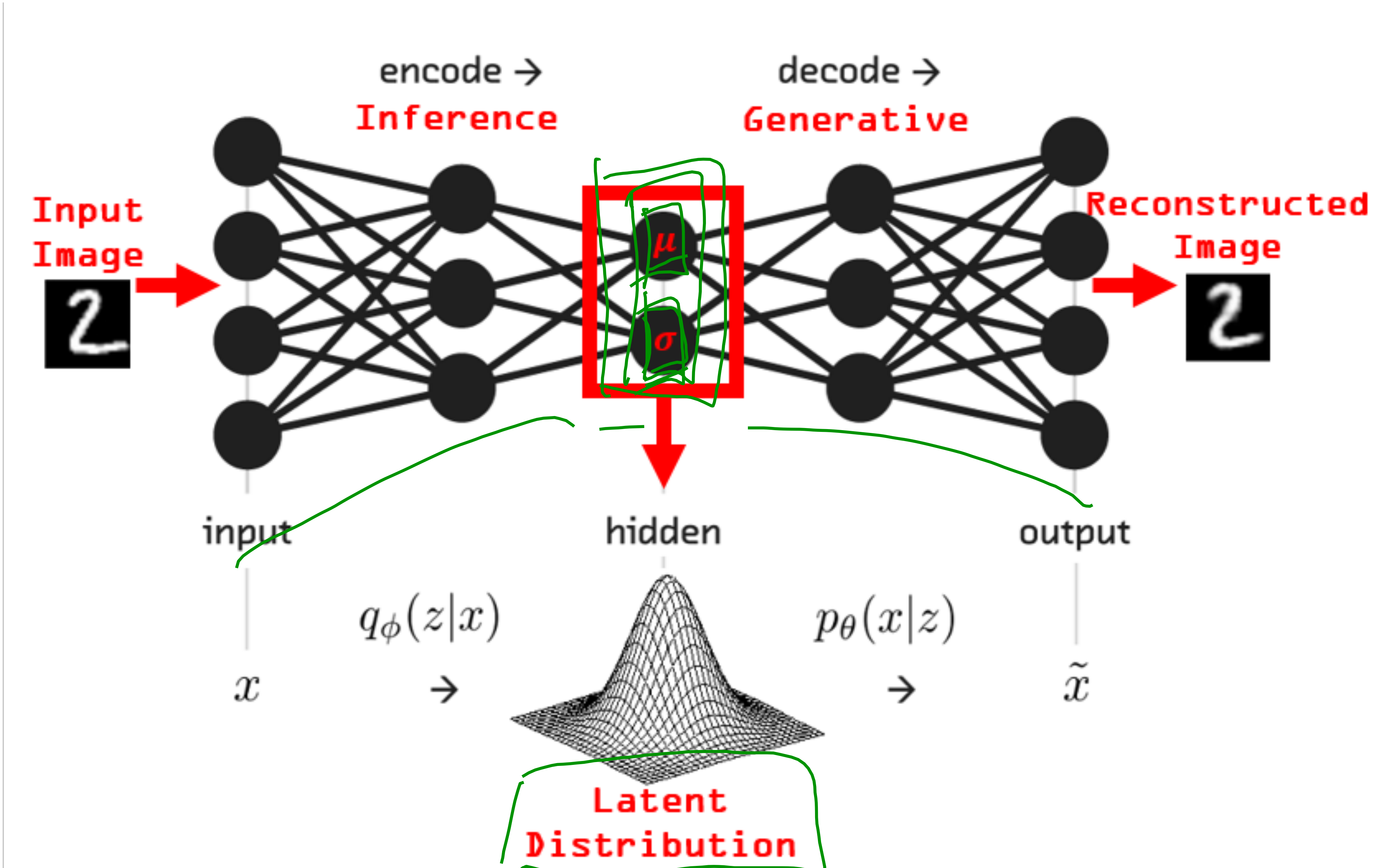


 Image Credit: https://www.kaggle.com/rvislaywade/visualizing-mnist-using-a-variational-autoencoder

Variational Autoencoders

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

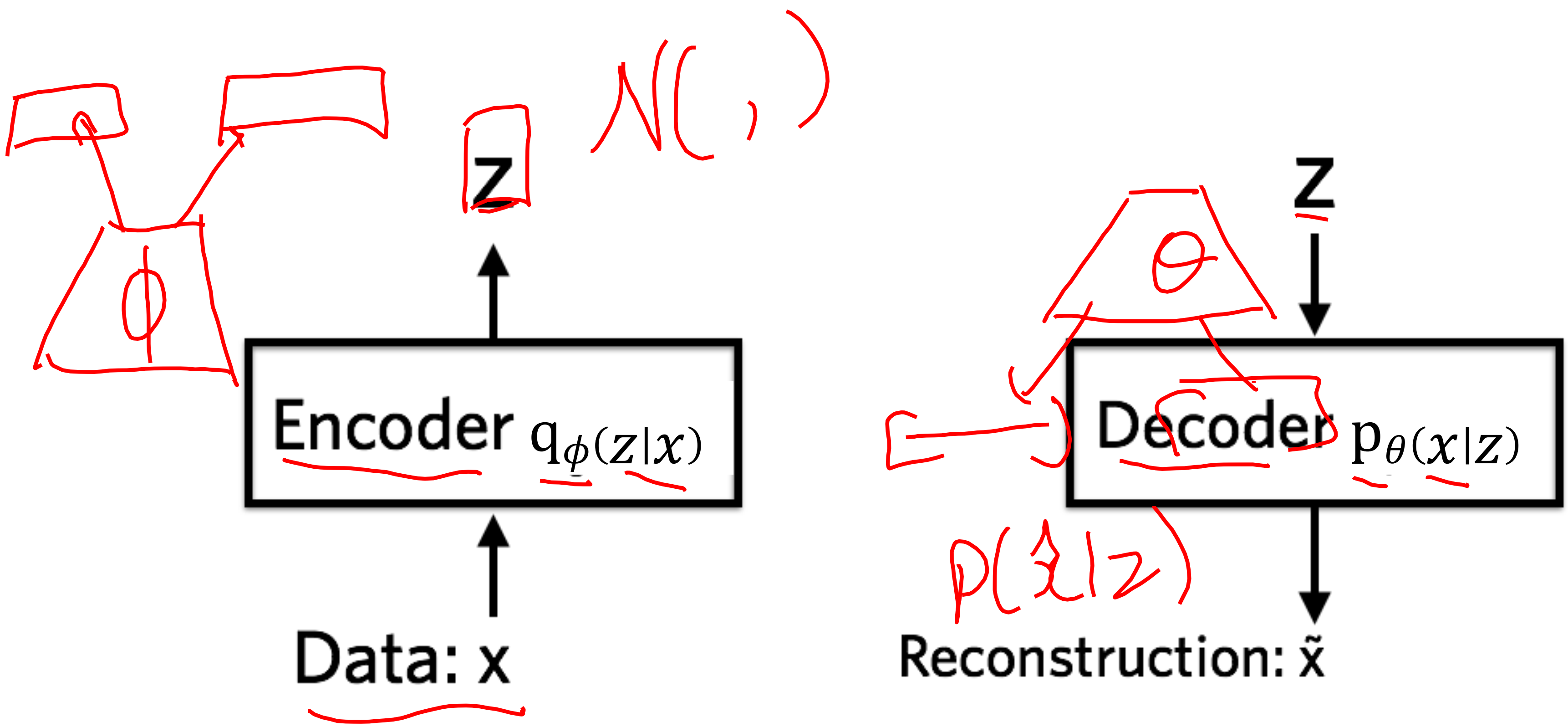


Image Credit: https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

Variational Auto Encoders

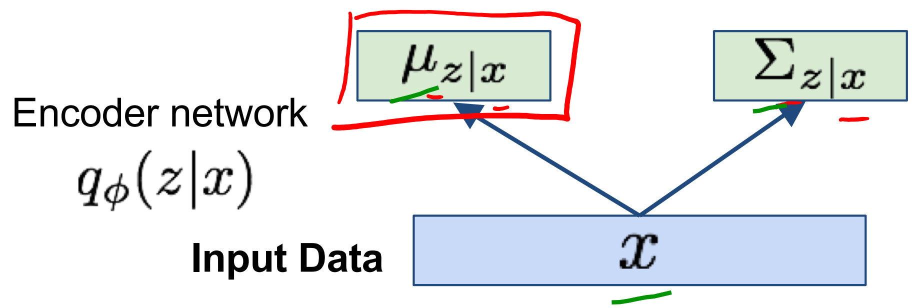
$F(\theta, \phi)$

Putting it all together: maximizing the likelihood lower bound

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

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

$\{x_i \dots\}$

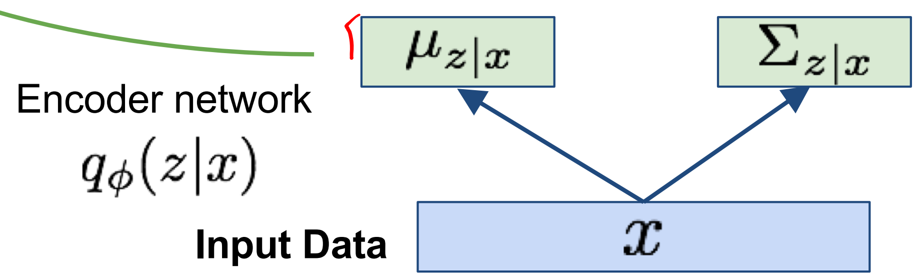


Variational Auto Encoders

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

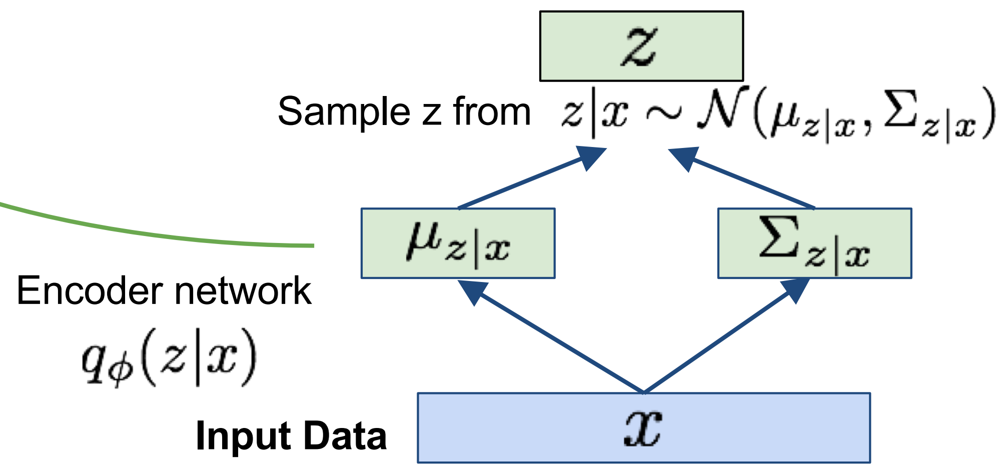


Variational Auto Encoders

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

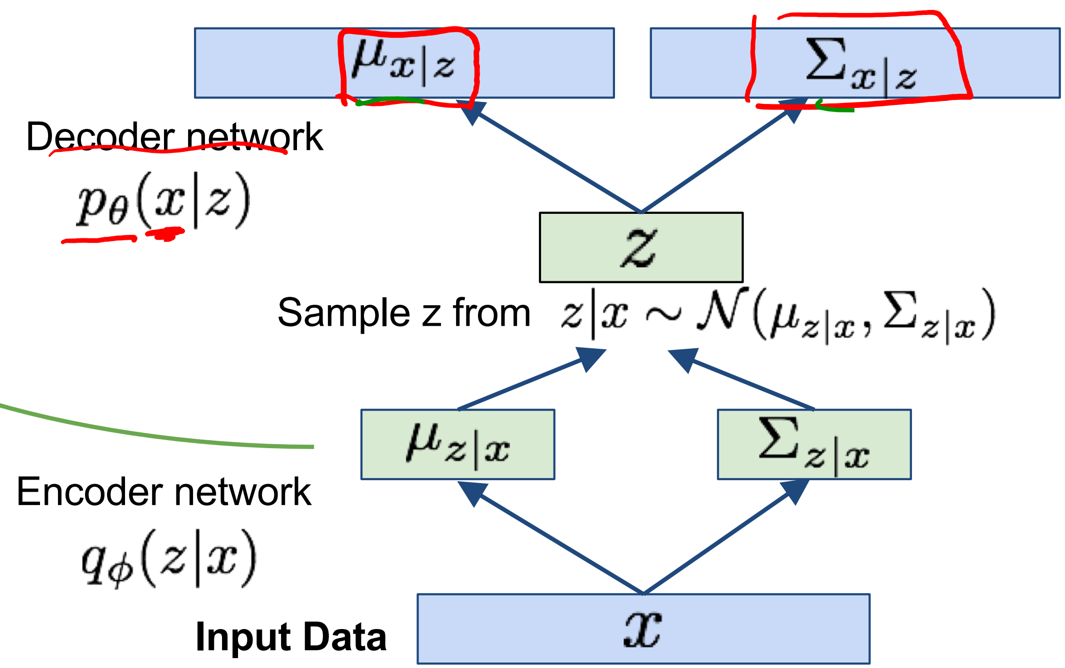


Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

Make approximate posterior distribution close to prior



Variational Auto Encoders

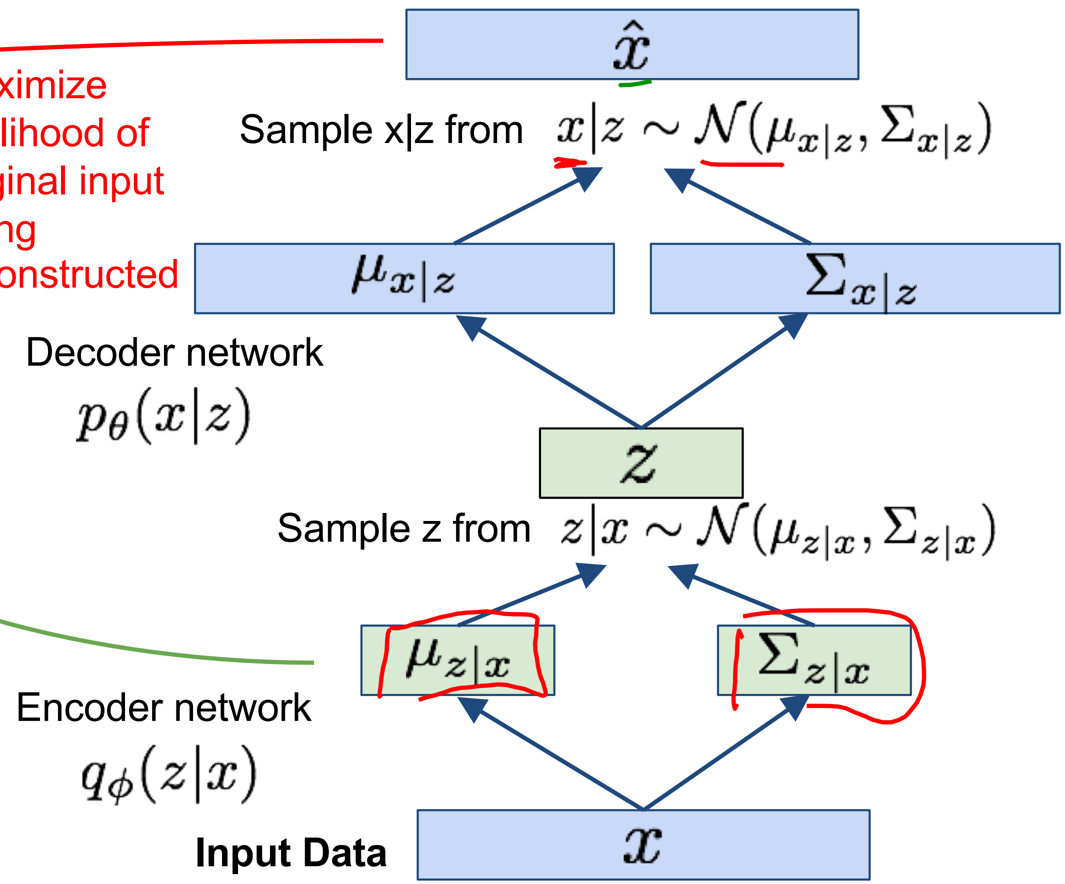
Putting it all together: maximizing the likelihood lower bound

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

Loss(θ,ϕ)

Maximize likelihood of original input being reconstructed

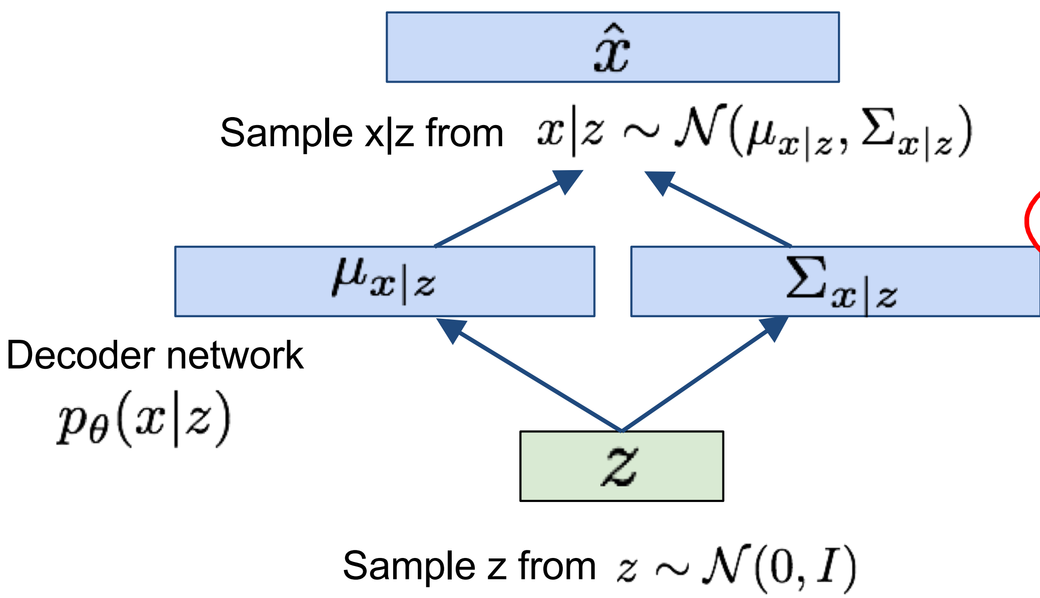
Make approximate posterior distribution close to prior

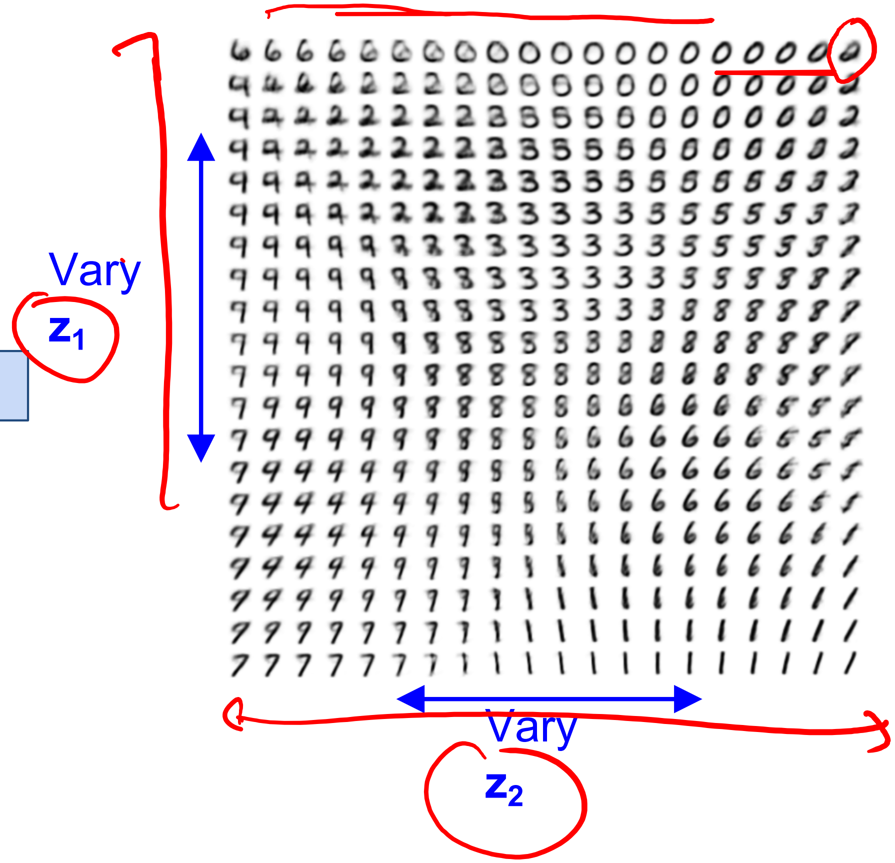


Variational Auto Encoders: Generating Data

Use decoder network. Now sample z from prior!

Data manifold for 2-d **z**





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

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

Plan for Today

- VAEs
  - Reparameterization trick

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

Variational Auto Encoders

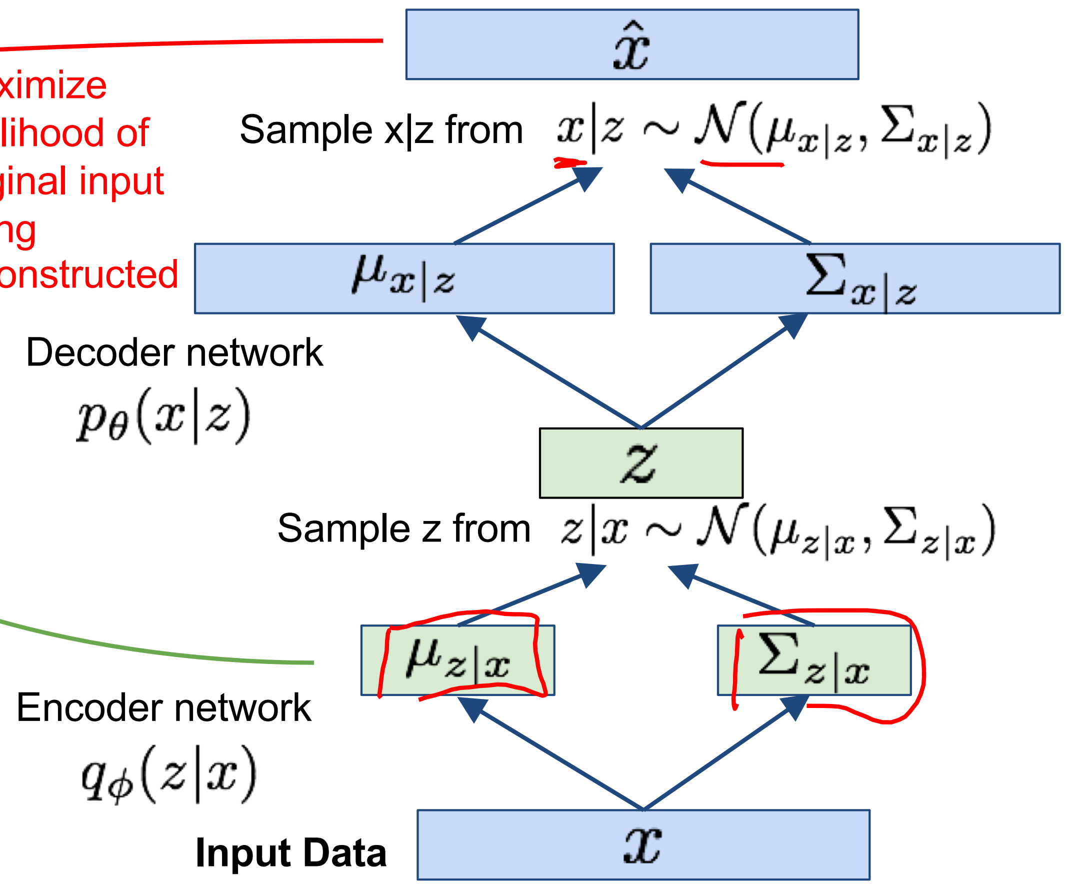
Putting it all together: maximizing the likelihood lower bound

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

Maximize likelihood of original input being reconstructed

Make approximate posterior distribution close to prior

~~log~~ loss(θ, ϕ)



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

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

q_ϕ

max

θ, ϕ

Basic Problem

$z \sim \mathrm{Cat}(\pi)$

$$\mathbb{E}_{z \sim p_{\theta}(z)}[f(z)]$$

Basic Problem

- Goal $$\boxed{\min_{\theta}} \; \mathbb{E}_{z \sim p_\theta(z)}[f(z)]$$

Basic Problem

- Goal $$\min_{\theta} \; \underline{\mathbb{E}_{z \sim p_\theta(z)}[f(z)]}$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta E_{z \sim p_\theta(z)}[f(z)]$$

- Need to compute: $$\boxed{\nabla_\theta \mathbb{E}_{z \sim p_\theta(z)}[f(z)]}$$

$f_\theta(z)$ $\quad p(z)$

$$\int \underline{\nabla_\theta f_\theta(z)} \, \underline{p(z)} \, dz$$

$$= E[\nabla_\theta f_\theta(z)]$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta f(z_i) \quad z_i \sim p(z)$$

$$\nabla_\theta \int f(z) \, p_\theta(z) \, dz$$

$$\int \nabla_\theta f(z) \, p_\theta(z) \, dz$$

$$\int \cancel{\nabla_\theta} f(z) \, \underline{\nabla_\theta p_\theta(z)} \, dz$$

Basic Problem

- Need to compute: $\nabla_\theta \, \mathbb{E}_{z \sim p_\theta(z)}[f(z)]$

Example

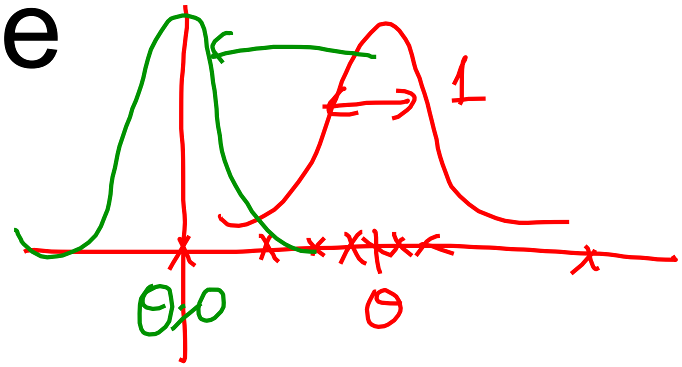
$z \sim N(\theta, 1)$

$f(z) = z^2$

$\min_{\theta} E_z[z^2]$

$\frac{\partial}{\partial \theta} \int z^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{(z-\theta)^2}{2}} dz$

"hard"



$Var(z) = E[(z-\theta)^2]$

$= E[z^2] - \theta^2$

$\min_{\theta} E[z^2] = \underbrace{\boxed{Var(z)}}_{1} + \boxed{\theta^2}$

"easy"

Does this happen in supervised learning?

- Goal $$\min_{\theta} \mathbb{E}_{z\sim p_\theta(z)}[f(z)]$$

$$\boxed{f_\theta(z)}$$

$$\min_{\theta} \nabla_\theta E_{x,y\sim \boxed{P_{data}}} \left[\ell(y^{gt}, \hat{y}(x,\theta)) \right]$$

$$\nabla_\theta E \approx \frac{1}{N} \sum \nabla_\theta \ell(y_i, \hat{y}_i(x_i, \theta))$$

$$\boxed{y_i\, x_i \sim P_{data}}$$

But what about other kinds of learning?

- Goal $$\min_{\theta} \boxed{\mathbb{E}_{\boxed{z \sim p_\theta(z)}}[f(z)]}$$

UL $\max_{\boxed{\theta, \phi}} E_{\boxed{q_\phi(z)}}\left[\underbrace{\log p(\ldots\ldots)}_{\theta,\phi}\right]$

ϕ

RL $\max_{\boxed{\theta}} E_{\boxed{a_1 \ldots a_T \sim \pi_\theta(a|s)}}\left[\sum r_t(s_t, a_t)\right]$

Two Options

$\nabla_\theta E_z[f(z)]$ $z \sim P_\theta(z)$

1. • Score Function based Gradient Estimator
aka REINFORCE (and variants)

log-ratio

$$\nabla_\theta \mathbb{E}_z [f(z)] = \mathbb{E}_z [f(z) \nabla_\theta \log p_\theta(z)]$$

2. • Path Derivative Gradient Estimator
aka “reparameterization trick”

$$\frac{\partial}{\partial \theta} \mathbb{E}_{z \sim p_\theta} [f(z))] = \frac{\partial}{\partial \theta} \mathbb{E}_\epsilon [f(g(\theta, \epsilon))] = \boxed{\mathbb{E}_{\epsilon \sim p_\epsilon} \left[\frac{\partial f}{\partial g} \frac{\partial g}{\partial \theta} \right]}$$

Option 1

- Score Function based Gradient Estimator aka REINFORCE (and variants)

$$\nabla_\theta \mathbb{E}_z\left[f(z)\right] = \mathbb{E}_z\left[f(z)\nabla_\theta \log p_\theta(z)\right]$$

$$\nabla_\theta \int f(z)\, p_\theta(z)\, dz = \int f(z)\, \nabla_\theta p_\theta(z)\, dz \cdot \frac{p_\theta(z)}{p_\theta(z)}$$

$$= \int f(z)\, \boxed{\nabla_\theta \log p_\theta(z)} \cdot p_\theta(z)\, dz$$

$$= E\left[f(z)\, \nabla_\theta \log p_\theta(z) \right]$$

$$\approx \frac{1}{N}\left(\quad \right)$$

Recall: Policy Gradients

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim p_\theta(\tau)}[\mathcal{R}(\tau)]$$

$$= \nabla_\theta \int \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Expand expectation

$$= \int \nabla_\theta \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

Exchange integration and expectation

$$= \int \nabla_\theta \pi_\theta(\tau) \cdot \frac{\pi_\theta(\tau)}{\pi_\theta(\tau)} \cdot \mathcal{R}(\tau)d\tau$$

$$= \int \pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau)\mathcal{R}(\tau)d\tau$$

$$\nabla_\theta \log \pi(\tau) = \frac{\nabla_\theta \pi(\tau)}{\pi(\tau)}$$

$$= \mathbb{E}_{\tau \sim p_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau)\mathcal{R}(\tau)]$$

Example

$$E_z\left[f(z)\,\nabla_\theta \log p_\theta(z)\right]$$

$$z \sim N(\theta, 1)$$

$$p_\theta(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(z-\theta)^2}{2}}$$

$$\frac{\partial}{\partial\theta}\left[\log p_\theta(z)\right] = \left[-\frac{(z-\theta)^2}{2} - \cancel{\frac{1}{2}\log 2\pi}\right]$$

$$\rightarrow -\frac{2(z-\theta)(-1)}{2}$$

$$= (z-\theta)$$

Gradient Estimate

$$E_z\left[z^2(z-\theta)\right]$$

$$\approx \frac{1}{N}\sum \left[z_i^2(z_i-\theta)\right]$$

$$z_i \sim N(\theta, 1)$$

Mental Break!

- VAE Demo
  - https://www.siarez.com/projects/variational-autoencoder

Two Options

$\min_\theta \mathbb{E}_{z \sim p_\theta(z)} [f(z)]$

① • Score Function based Gradient Estimator
aka REINFORCE (and variants)

$$\nabla_\theta \mathbb{E}_z \left[f(z) \right] = \mathbb{E}_z \left[f(z) \nabla_\theta \log p_\theta(z) \right]$$

② • Path Derivative Gradient Estimator
aka "reparameterization trick"

$$\frac{\partial}{\partial \theta} \mathbb{E}_{z \sim p_\theta} \left[f(z) \right] = \frac{\partial}{\partial \theta} \mathbb{E}_\epsilon \left[f(g(\theta, \epsilon)) \right] = \mathbb{E}_{\epsilon \sim p_\epsilon} \left[\frac{\partial f}{\partial g} \frac{\partial g}{\partial \theta} \right]$$

Option 2

- Path Derivative Gradient Estimator
  aka "reparameterization trick"

$$\frac{\partial}{\partial\theta}\mathbb{E}_{z\sim p_\theta}\left[f(z)\right] = \frac{\partial}{\partial\theta}\mathbb{E}_{\epsilon}\left[f(g(\theta,\epsilon))\right] = \mathbb{E}_{\epsilon\sim p_\epsilon}\left[\frac{\partial f}{\partial g}\frac{\partial g}{\partial\theta}\right]$$

$$\underline{z} \sim p_{\underline{\theta}}(z) \implies z = g(\underline{\theta}, \underline{\varepsilon}) \qquad \varepsilon \sim \text{'standard' RV}$$

$$z \sim N(\mu, \sigma^2) \qquad \varepsilon \sim N(0,1)$$

$$\varepsilon \sim N(0,1)$$
$$\sim U(0,1)$$
$$\sim \text{Bern}(0.5)$$
$$\vdots$$

$$\underline{z} = \underbrace{\mu + \sigma\underline{\varepsilon}}_{g(\underset{(\mu,\sigma)}{\theta},\ \varepsilon)}$$

Option 2

- Path Derivative Gradient Estimator
  aka "reparameterization trick"

$$\frac{\partial}{\partial \theta}\mathbb{E}_{z\sim p_\theta}\left[f(z)\right] = \frac{\partial}{\partial \theta}\mathbb{E}_{\epsilon}\left[f(g(\theta,\epsilon))\right] = \mathbb{E}_{\epsilon\sim p_\epsilon}\left[\frac{\partial f}{\partial g}\frac{\partial g}{\partial \theta}\right]$$

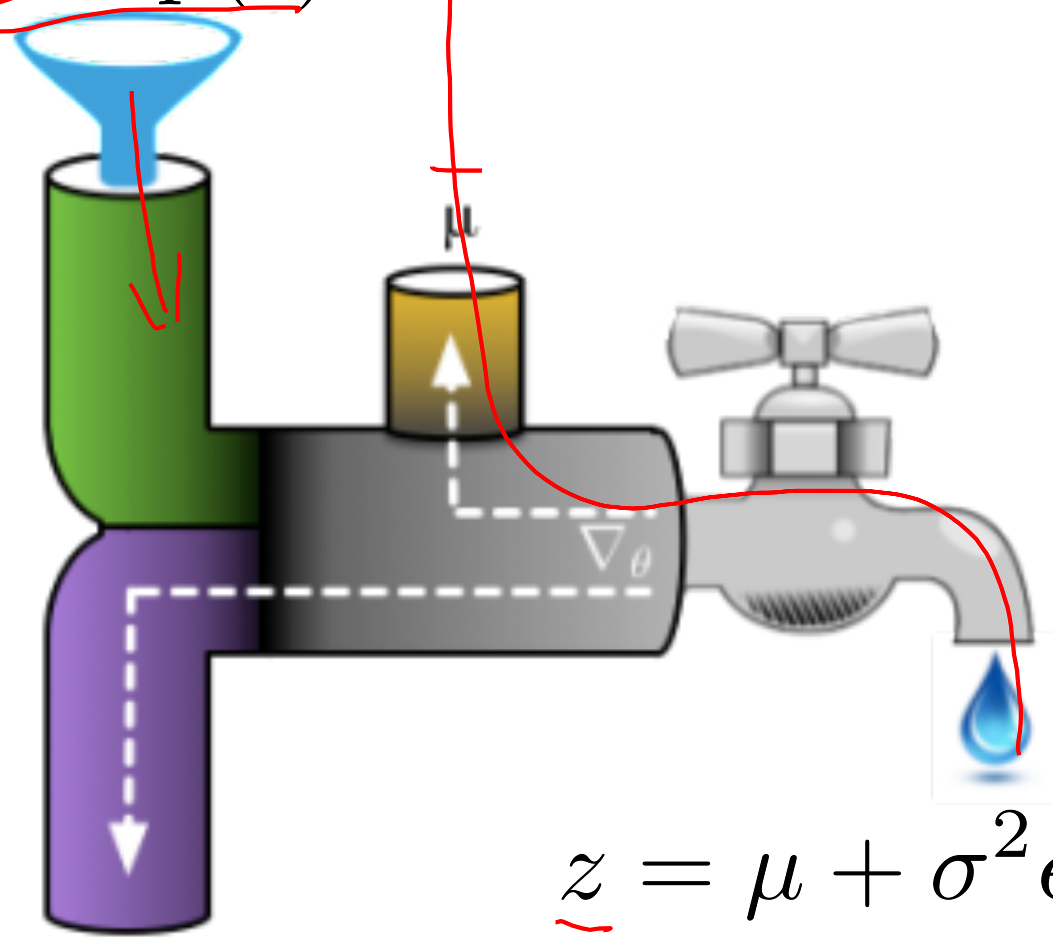
$$\frac{\partial}{\partial \theta} E_{z\sim p_\theta(z)}\left[f(z)\right] = \frac{\partial}{\partial \theta} E_{\varepsilon\sim p_\varepsilon(}\left[f(g(\theta,\varepsilon))\right]$$

$$= E_\varepsilon\left[\frac{\partial}{\partial \theta} f(g(\theta,\varepsilon))\right]$$

$$= E_\varepsilon\left[\frac{\partial f}{\partial g}\cdot\frac{\partial g}{\partial \theta}\right]$$

Reparameterization Intuition

$$\epsilon_i \sim p(\epsilon)$$

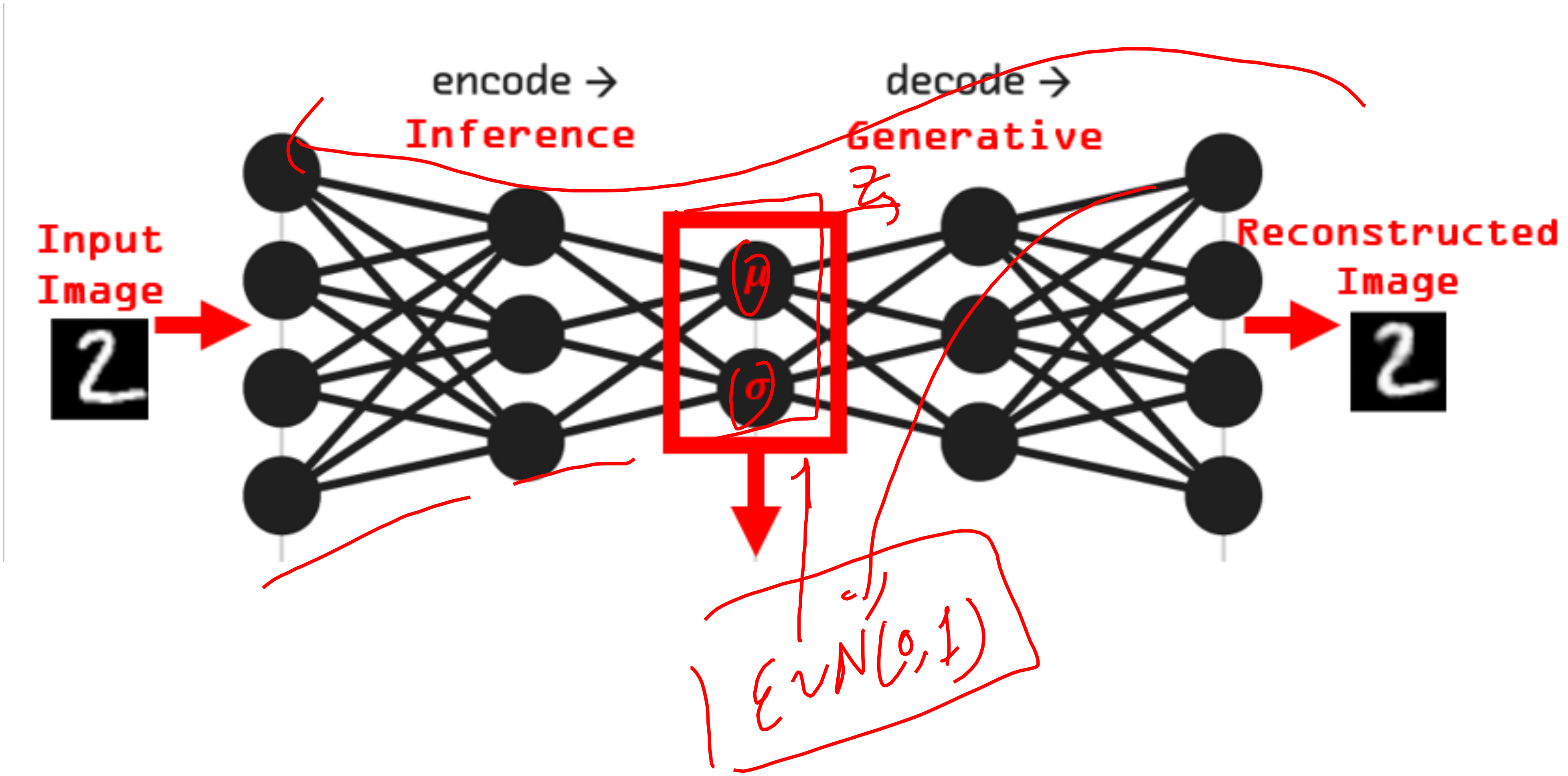


$$z = \mu + \sigma^2 \epsilon_i$$

$$\frac{\partial L}{\partial \mu}$$

$$\sigma^2$$

Reparameterization Intuition



Example

$$z \sim N(\theta, 1)$$

$$f(z) = z^2$$

$$\min_\theta E_z[z^2]$$

$$z = g(\theta, \varepsilon) \qquad \varepsilon \sim N(0,1)$$

$$z = \theta + \varepsilon$$

$$\frac{\partial}{\partial\theta} E_\varepsilon\left[(\theta+\varepsilon)^2\right]$$

$$= E_\varepsilon\left[\frac{\partial}{\partial\theta}(\theta+\varepsilon)^2\right]$$

$$= E_\varepsilon\left[2(\theta+\varepsilon)\cdot 1\right]$$

$$E\left[2\theta + 2\varepsilon\right] = \boxed{2\theta} + \boxed{\cancel{2E[\varepsilon]}}$$

$$\min_\theta \theta^2$$

$$\approx \boxed{\frac{1}{N}\sum_{i=1}^{N} 2(\theta+\varepsilon_i)} \qquad \varepsilon_i \sim N(0,1)$$

Two Options

1

- Score Function based Gradient Estimator
aka REINFORCE (and variants)

$$\nabla_\theta \mathbb{E}_z \left[f(z) \right] = \mathbb{E}_z \left[f(z) \nabla_\theta \log p_\theta(z) \right]$$

2

- Path Derivative Gradient Estimator
aka “reparameterization trick”

$$\boxed{\frac{\partial}{\partial \theta}} \mathbb{E}_{z \sim p_\theta} \left[f(z) \right] = \frac{\partial}{\partial \theta} \mathbb{E}_\epsilon \left[f(g(\theta, \epsilon)) \right] = \boxed{\mathbb{E}_{\epsilon \sim p_\epsilon} \left[\frac{\partial f}{\partial g} \frac{\partial g}{\partial \theta} \right]}$$

Example

$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta (\ \ )$

```
import numpy as np
N = 1000
theta = 2.0
x = np.random.randn(N) + theta
eps = np.random.randn(N)

grad1 = lambda x: np.sum(np.square(x)*(x-theta)) / x.size
grad2 = lambda eps: np.sum(2*(theta + eps)) / x.size

print grad1(x)
print grad2(eps)
```

$\theta^{(t)} = 2$

2θ

z

$z_i^2 (z_i - \theta)$

$2(\theta + \epsilon_i)$

```
4.46239612174
4.1840532024
```

$\approx 2\theta$

Example

```
Ns = [10, 100, 1000, 10000, 100000]
reps = 100

means1 = np.zeros(len(Ns))
vars1 = np.zeros(len(Ns))
means2 = np.zeros(len(Ns))
vars2 = np.zeros(len(Ns))

est1 = np.zeros(reps)
est2 = np.zeros(reps)
for i, N in enumerate(Ns):
    for r in range(reps):
        x = np.random.randn(N) + theta
        est1[r] = grad1(x)
        eps = np.random.randn(N)
        est2[r] = grad2(eps)
    means1[i] = np.mean(est1)
    means2[i] = np.mean(est2)
    vars1[i] = np.var(est1)
    vars2[i] = np.var(est2)

print means1
print means2
print
print vars1
print vars2
```

```
[ 3.8409546   3.97298803  4.03007634  3.98531095  3.99579423]
[ 3.97775271  4.00232825  3.99894536  4.00353734  3.99995899]

[  6.45307927e+00   6.80227241e-01   8.69226368e-02   1.00489791e-02
   8.62396526e-04]
[  4.59767676e-01   4.26567475e-02   3.33699503e-03   5.17148975e-04
   4.65338152e-05]
```

Figure Credit: http://gokererdogan.github.io/2016/07/01/reparameterization-trick/

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

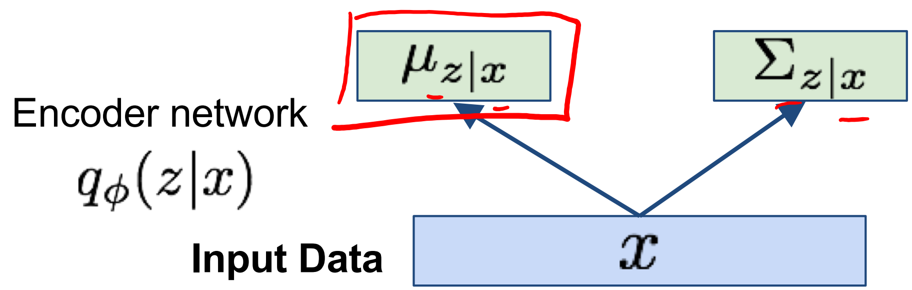
4. "Reparameterization" Trick

Variational Auto Encoders

Putting it all together: maximizing the likelihood lower bound

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

$E_{q_\phi(z|x_i)}$



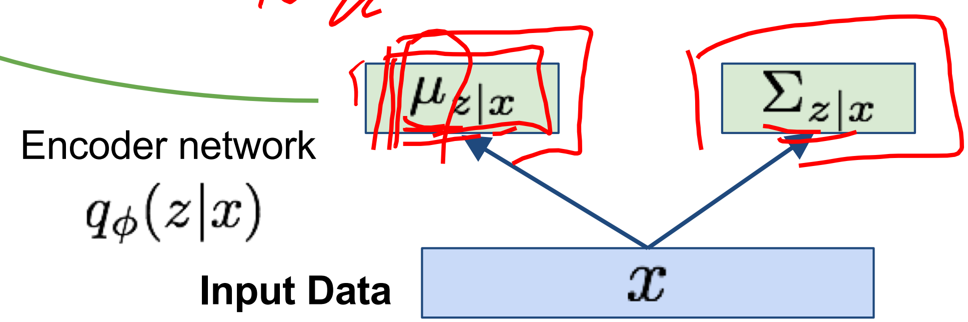
Variational Auto Encoders

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

$\sim \mathbb{E}_\epsilon[\epsilon \sim \mathcal{N}(\vec{0}, I)$

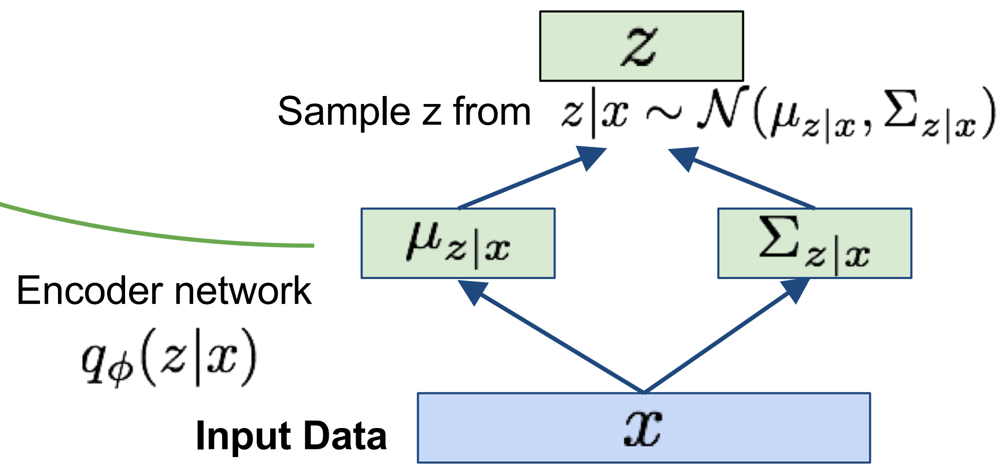


Variational Auto Encoders

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