Deep Generative Models

Changyou Chen

Department of Computer Science and Engineering
Universitpy at Buffalo, SUNY
changyou@buffalo.edu

April 9, 2019

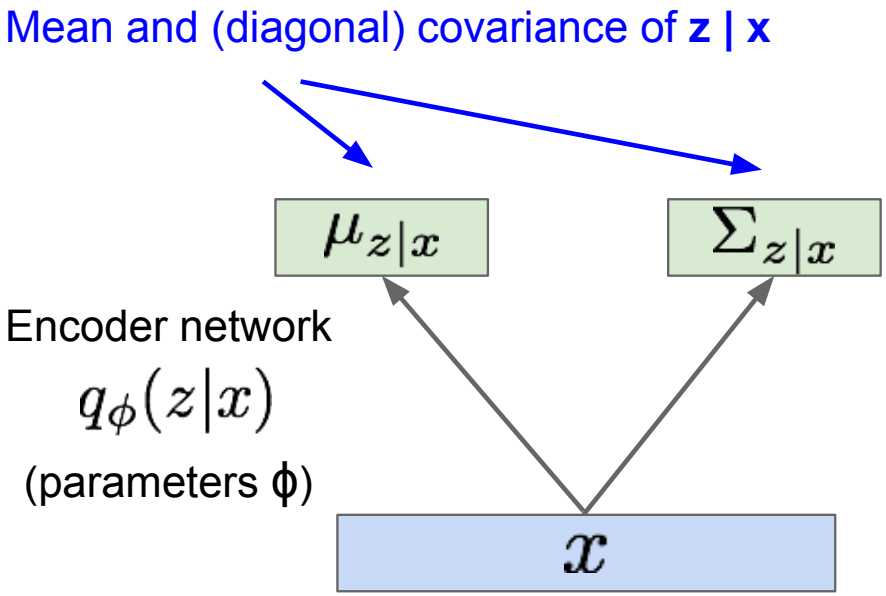
Variational Autoencoder: Intractability

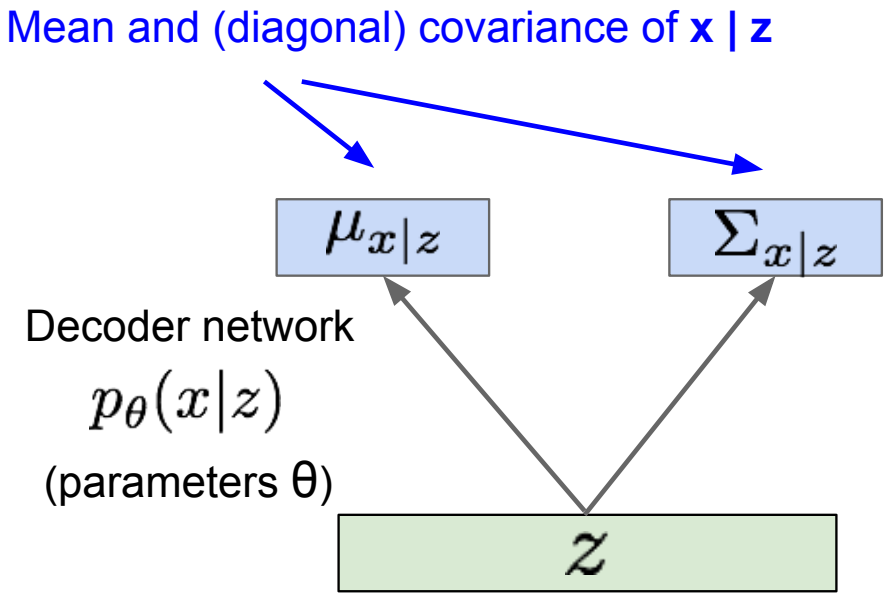
- Data likelihood: $p_\theta(\mathbf{x}) = \underbrace{\int}_{\text{intractible } \times} p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z}) \mathrm{d}\mathbf{z}$
- Posterior also intractable: $p_\theta(\mathbf{z} \mid \mathbf{x}) = \underbrace{p_\theta(\mathbf{x} \mid \mathbf{z})}_{\checkmark} \underbrace{p_\theta(\mathbf{z})}_{\checkmark} / \underbrace{p_\theta(\mathbf{x})}_{\times}$
- Solution:
  - In additional to define the decoder network $p_\theta(\mathbf{x} \mid \mathbf{z})$, define an additional *encoder network* $q_\phi(\mathbf{z} \mid \mathbf{x})$ that approximates the posterior $p_\theta(\mathbf{z} \mid \mathbf{x}) \Rightarrow$ variational inference with variational distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$!.
  - We will see this allows us to derive a lower bound on the data likelihood, which can be optimized tractably.

Variational Autoencoder

Encoder and decoder networks are probabilistic

- Encoder also called inference/recognition network.
- Decoder also called generation network.
- Amortized structure.

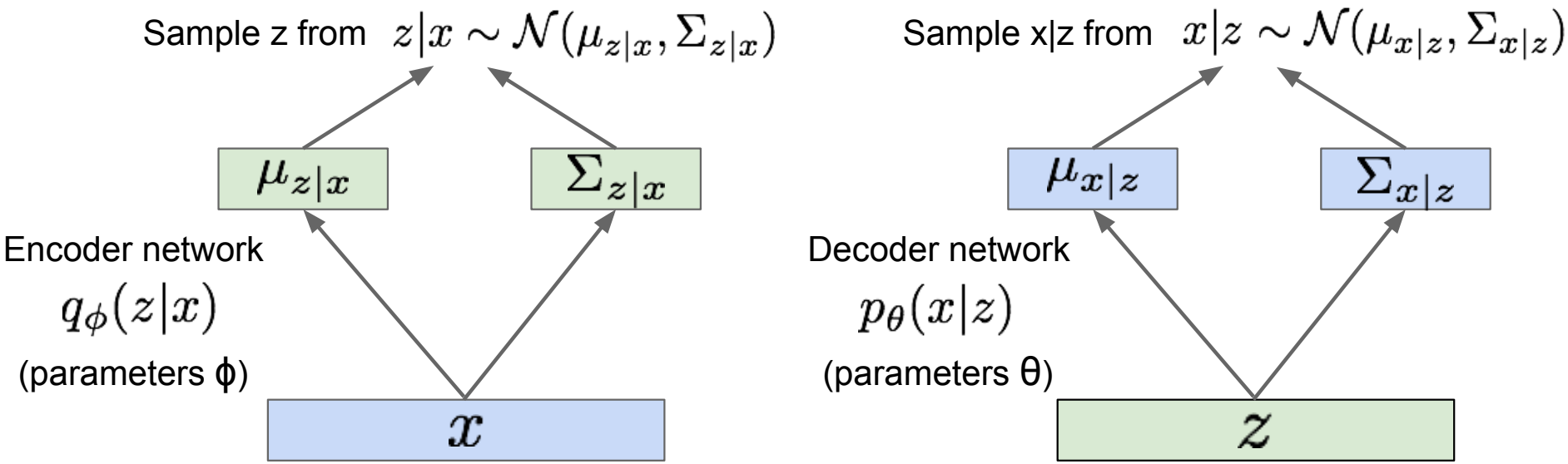




Variational Autoencoder

Encoder and decoder networks are probabilistic

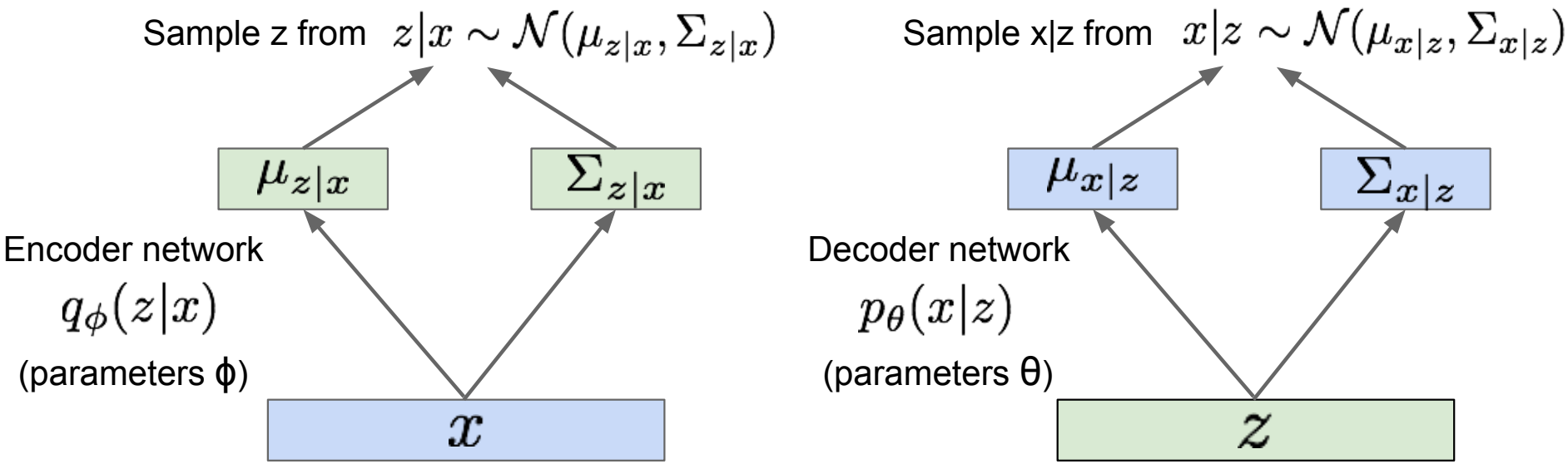
- Encoder also called inference/recognition network.
- Decoder also called generation network.
- Amortized structure.



Variational Autoencoder

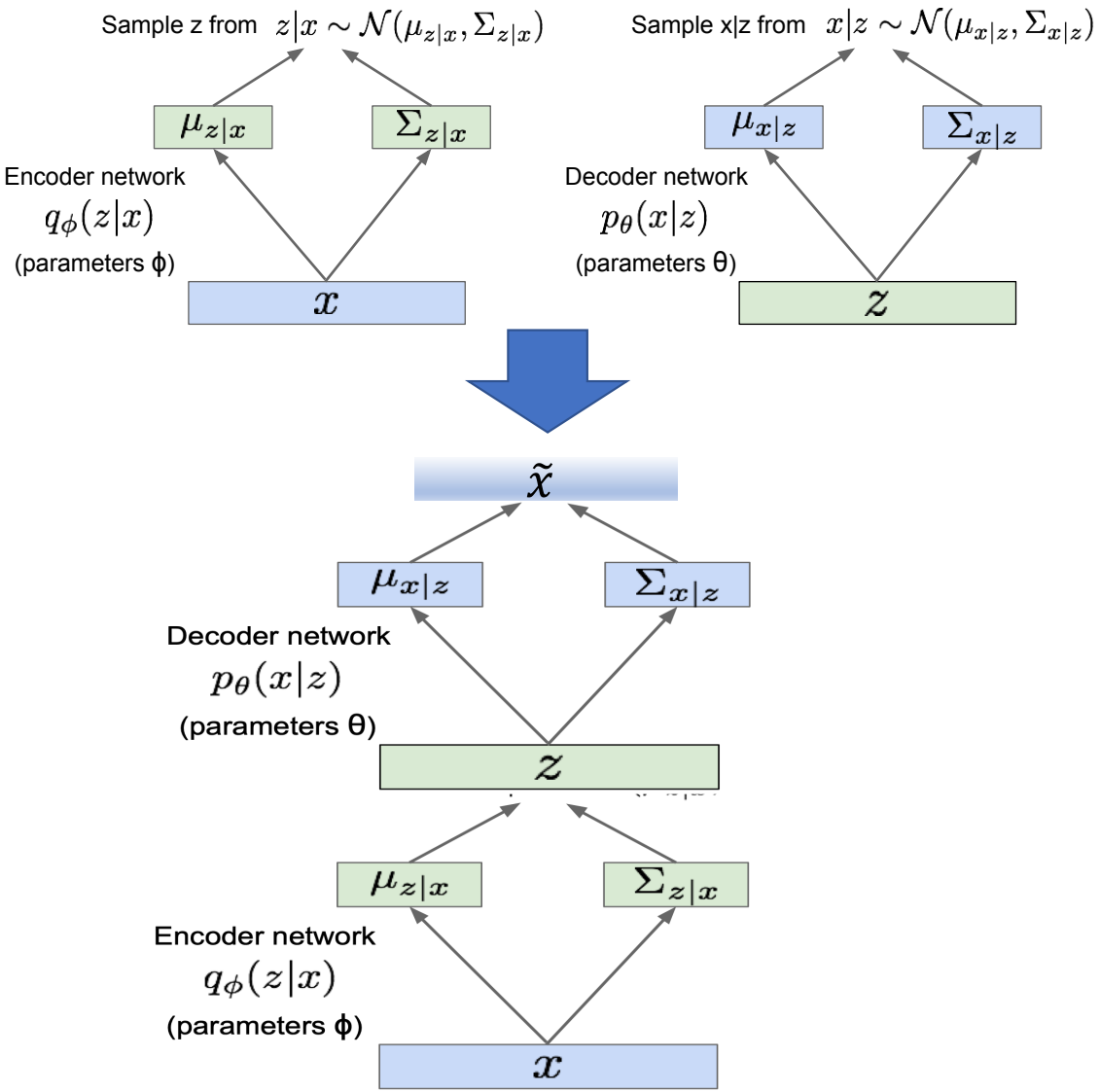
Encoder and decoder networks are probabilistic

- Encoder also called inference/recognition network.
- Decoder also called generation network.
- Amortized structure.



Want to match $q_\phi(z|x)$ and $p(z|x) \propto p(z)p(x|z)$.

Doesn't Look Like Autoencoder?

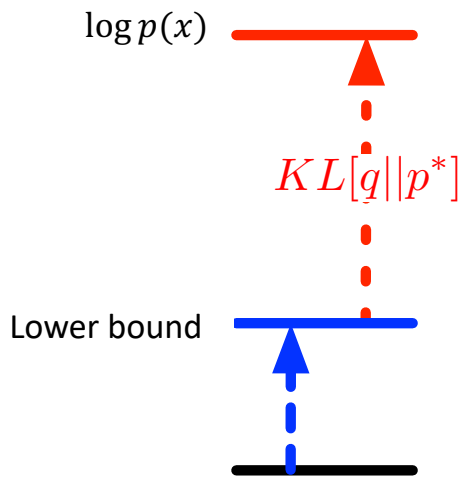


Variational Autoencoder: Log Data Likelihood

Derive from a different but equivalent way as in variational inference:

- Don't need to apply Jensen's inequality, or equivalently, prove Jensen's inequality.
- We will show: $\log p(x) = KL(q(z|x)||p(z|x)) + \text{lower bound}$

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$



Variational Autoencoder: Log Data Likelihood

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \qquad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

Taking expectation wrt. z (using encoder network) will come in handy later

Variational Autoencoder: Log Data Likelihood

$$\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] && (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] && (\text{Bayes' Rule})
\end{aligned}$$

Variational Autoencoder: Log Data Likelihood

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] && (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] && (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] && (\text{Multiply by constant})
\end{aligned}
$$

Variational Autoencoder: Log Data Likelihood

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] && (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] && \text{(Bayes' Rule)} \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] && \text{(Multiply by constant)} \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] && \text{(Logarithms)}
\end{aligned}
$$

Variational Autoencoder: Log Data Likelihood

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] && (p_\theta(x^{(i)}) \text{ Does not depend on } z)\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] && \text{(Bayes' Rule)}\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] && \text{(Multiply by constant)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] && \text{(Logarithms)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}
$$

Variational Autoencoder: Log Data Likelihood

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] && (p_\theta(x^{(i)}) \text{ Does not depend on } z)\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] && \text{(Bayes' Rule)}\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] && \text{(Multiply by constant)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] && \text{(Logarithms)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z)) + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}
$$

The expectation wrt. z (using encoder network) let us write nice KL terms

Variational Autoencoder: Log Data Likelihood

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Bayes' Rule)}$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad \text{(Multiply by constant)}$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad \text{(Logarithms)}$$

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

Decoder network gives $p_\theta(x|z)$, can compute estimate of this term through sampling. (Sampling differentiable through reparam. trick, see paper.)

This KL term (between Gaussians for encoder and z prior) has nice closed-form solution!

$p_\theta(z|x)$ intractable (saw earlier), can't compute this KL term :(But we know KL divergence always >= 0.

Variational Autoencoder: Log Data Likelihood

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] && (p_\theta(x^{(i)}) \text{ Does not depend on } z)\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] && \text{(Bayes' Rule)}\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] && \text{(Multiply by constant)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] && \text{(Logarithms)}\\
&= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}
\end{aligned}
$$

Tractable lower bound which we can take gradient of and optimize! ($p_\theta(x|z)$ differentiable, KL term differentiable)

Variational Autoencoder: Log Data Likelihood

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] && (p_\theta(x^{(i)}) \text{ Does not depend on } z)\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] && \text{(Bayes' Rule)}\\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] && \text{(Multiply by constant)}\\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] && \text{(Logarithms)}\\
&= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{>0}
\end{aligned}
$$

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)},\theta,\phi)$$

Variational lower bound ("ELBO")

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)},\theta,\phi)$$

Training: Maximize lower bound

Variational Autoencoder: Log Data Likelihood

$$\log p_\theta(x^{(i)}) = \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z)$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule})$$

$$= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant})$$

$$= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms})$$

$$= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{>0}$$

Reconstruct the input data

Make approximate posterior distribution close to prior

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)},\theta,\phi)$$

Variational lower bound ("ELBO")

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)},\theta,\phi)$$

Training: Maximize lower bound

Variational Autoencoder: Log Data Likelihood

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 Autoencoder: Log Data Likelihood

Putting it all together: maximizing the likelihood lower bound

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

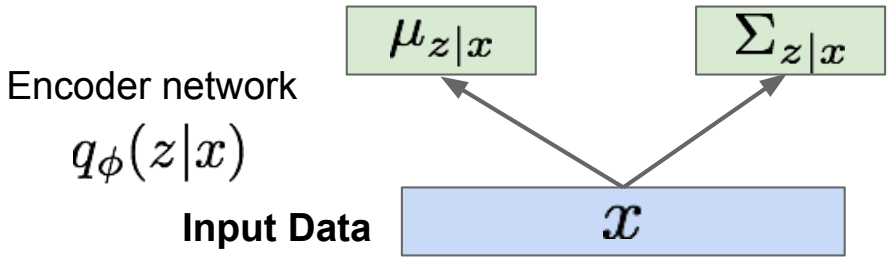
Let's look at computing the bound (forward pass) for a given minibatch of input data

Input Data x

Variational Autoencoder: Log Data Likelihood

Putting it all together: maximizing the likelihood lower bound

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

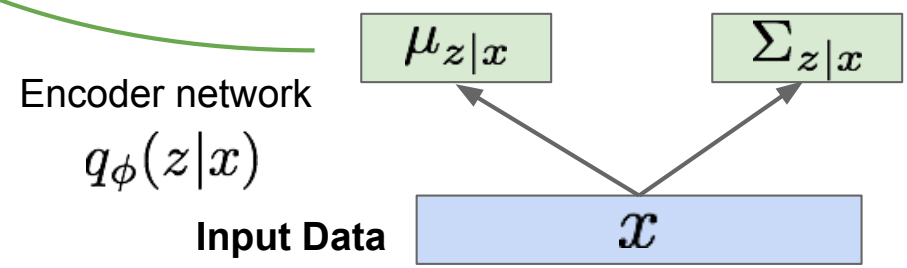


Variational Autoencoder: Log Data Likelihood

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

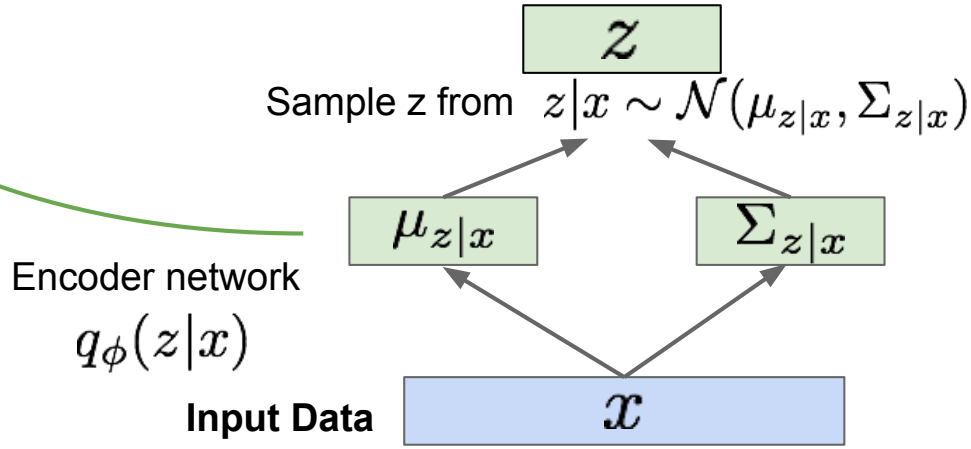


Variational Autoencoder: Log Data Likelihood

Putting it all together: maximizing the likelihood lower bound

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

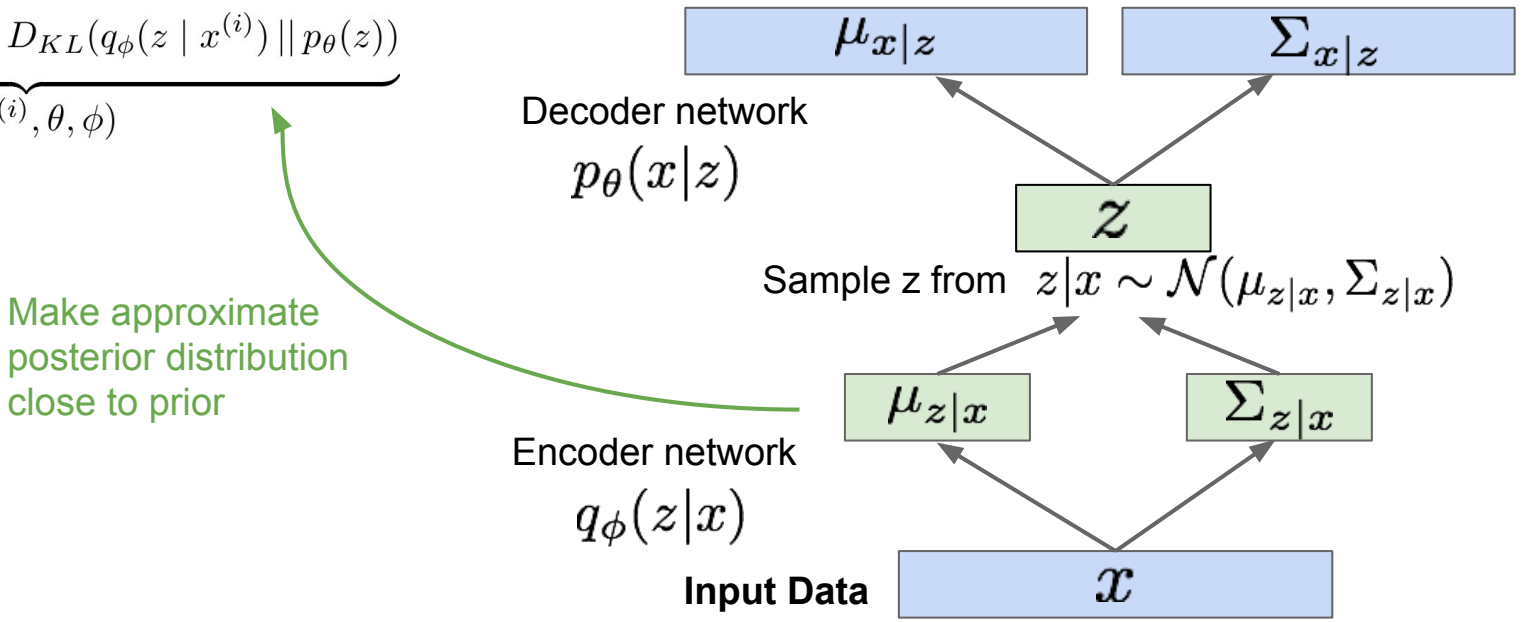
Make approximate posterior distribution close to prior



Variational Autoencoder: Log Data Likelihood

Putting it all together: maximizing the likelihood lower bound

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

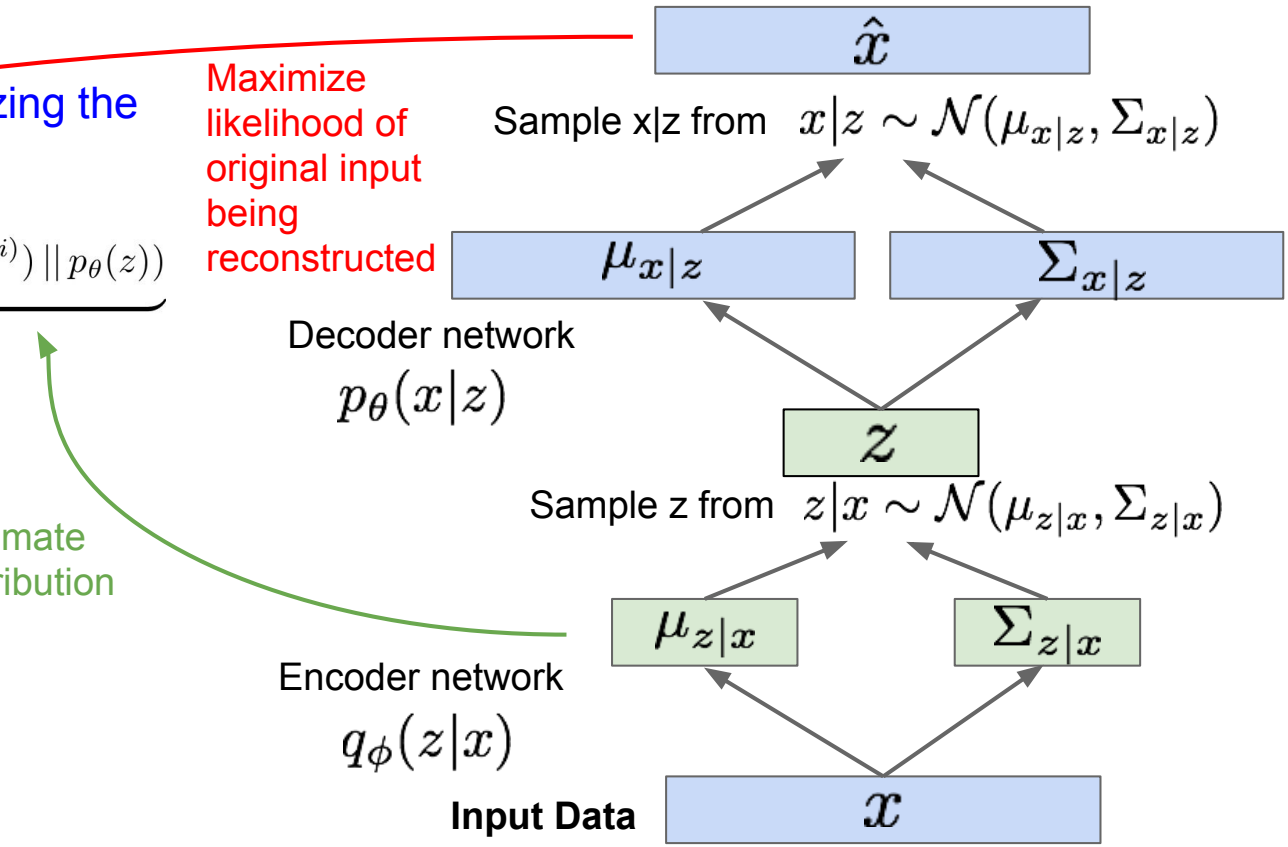


Variational Autoencoder: Log Data Likelihood

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 Autoencoder: Log Data Likelihood

Putting it all together: maximizing the likelihood lower bound

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

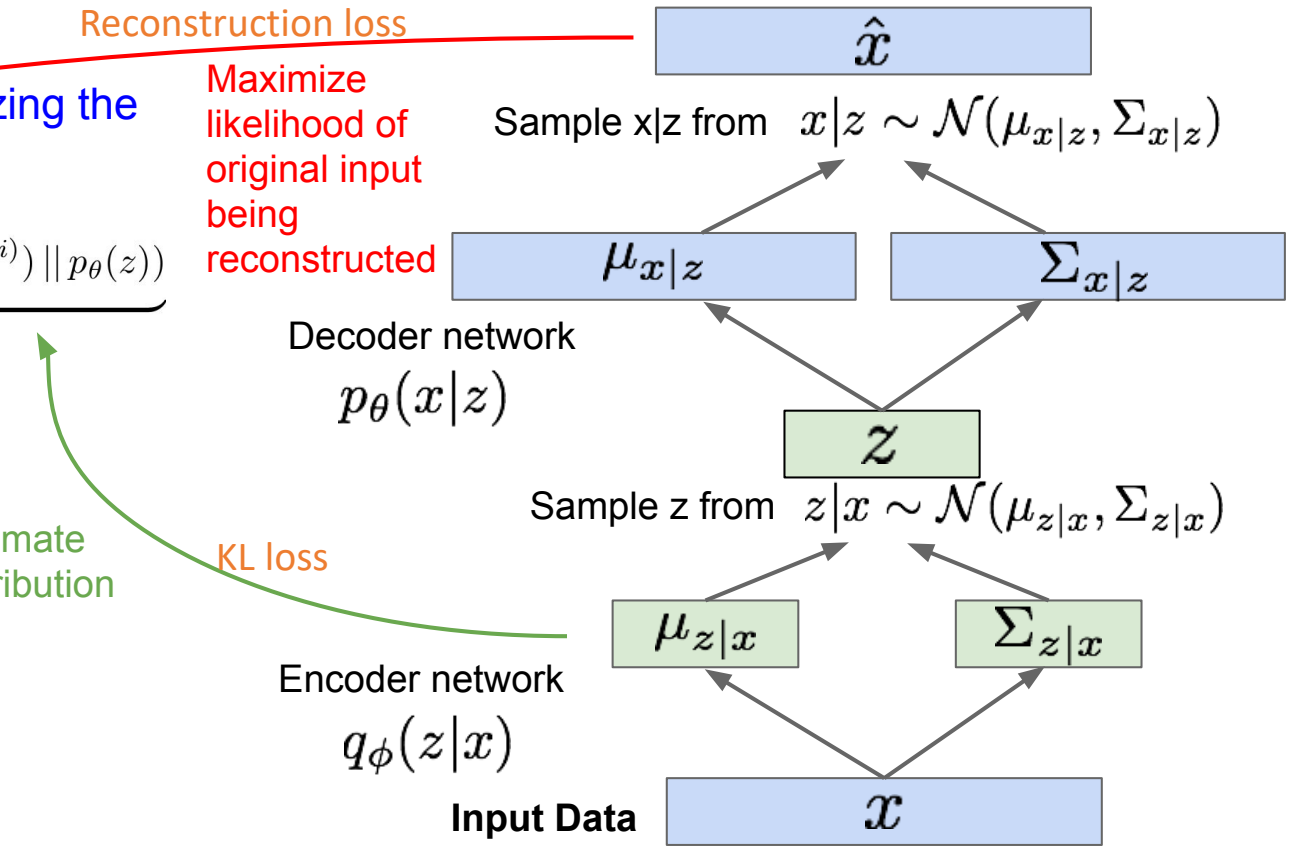
Reconstruction loss

Maximize likelihood of original input being reconstructed

Make approximate posterior distribution close to prior

KL loss

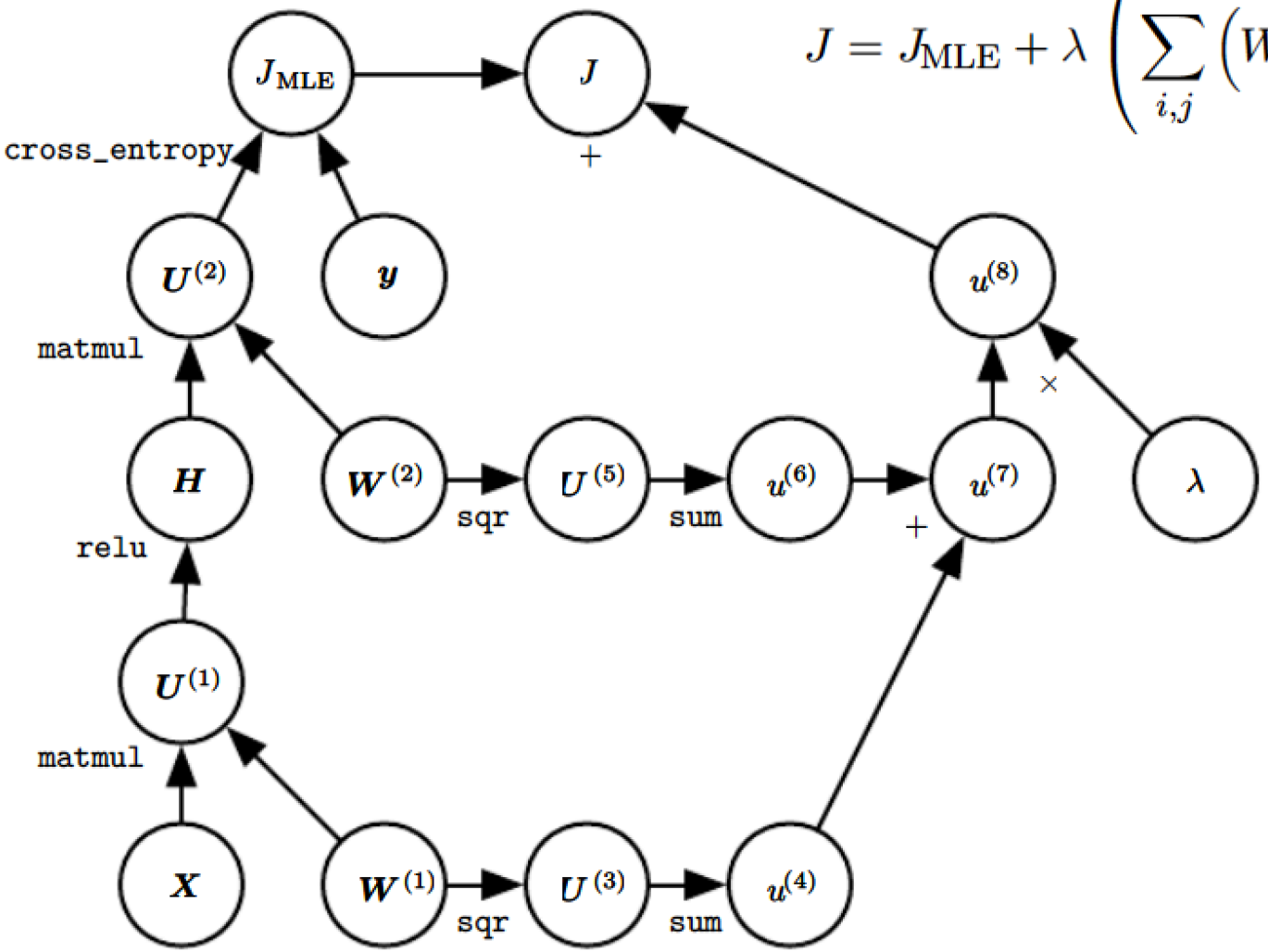
For every minibatch of input data: compute this forward pass, and then backprop!



Recap: FNN Computational Graph

Total cost:

$$J = J_{\text{MLE}} + \lambda \left(\sum_{i,j} \left(W^{(1)}_{i,j} \right)^2 + \sum_{i,j} \left(W^{(2)}_{i,j} \right)^2 \right)$$

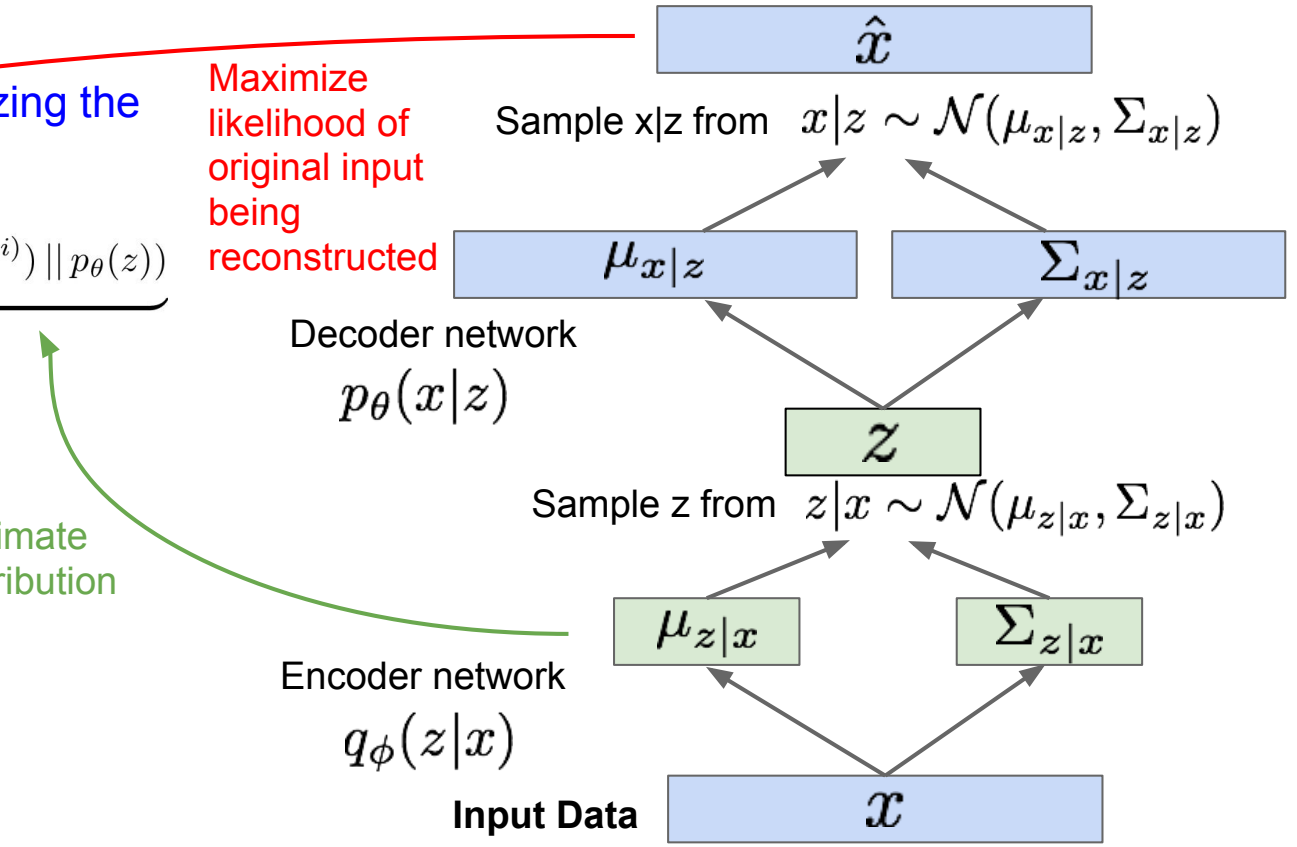


Recap: VAE Computational Graph

Putting it all together: maximizing the likelihood lower bound

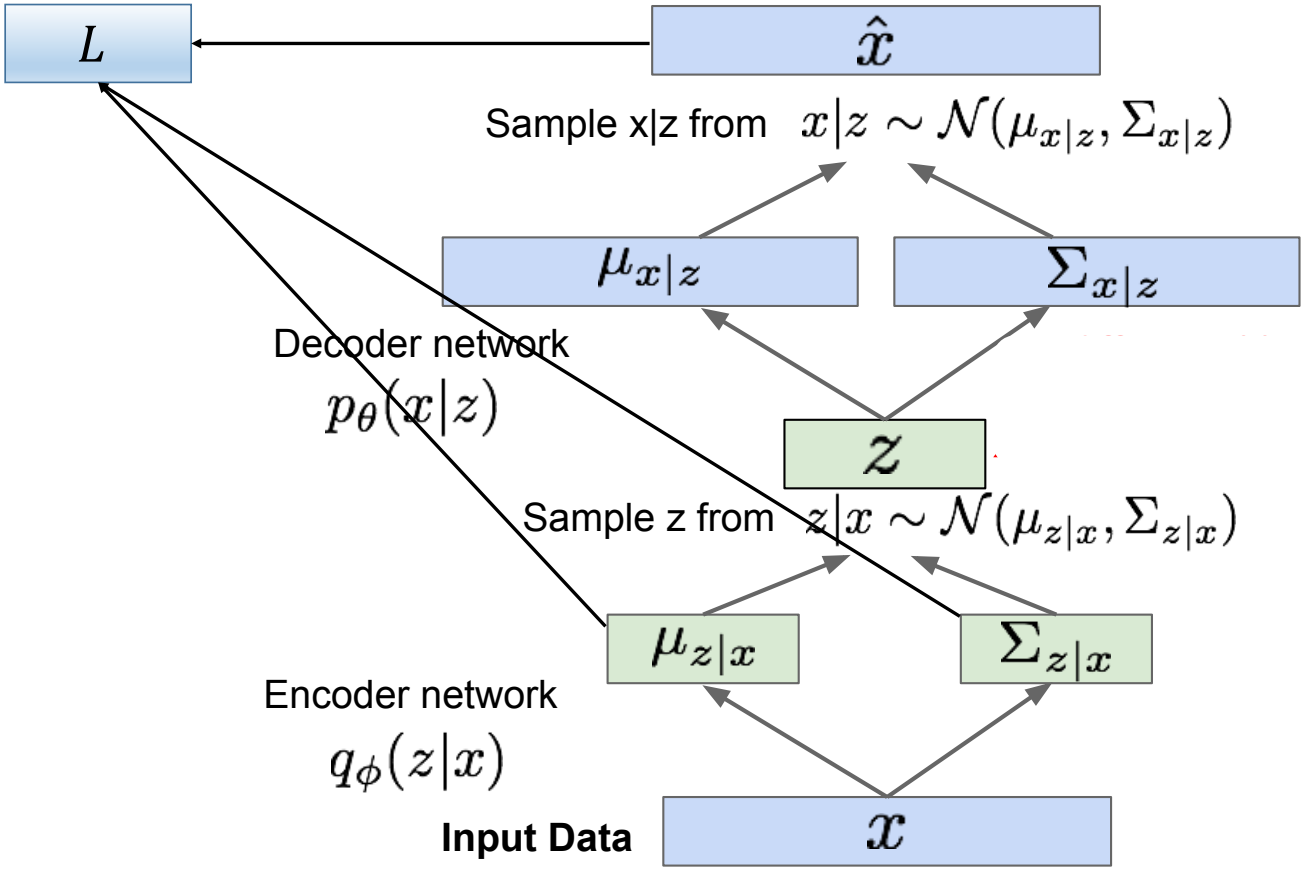
$$\underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)}, \theta, \phi)}$$

Maximize likelihood of original input being reconstructed



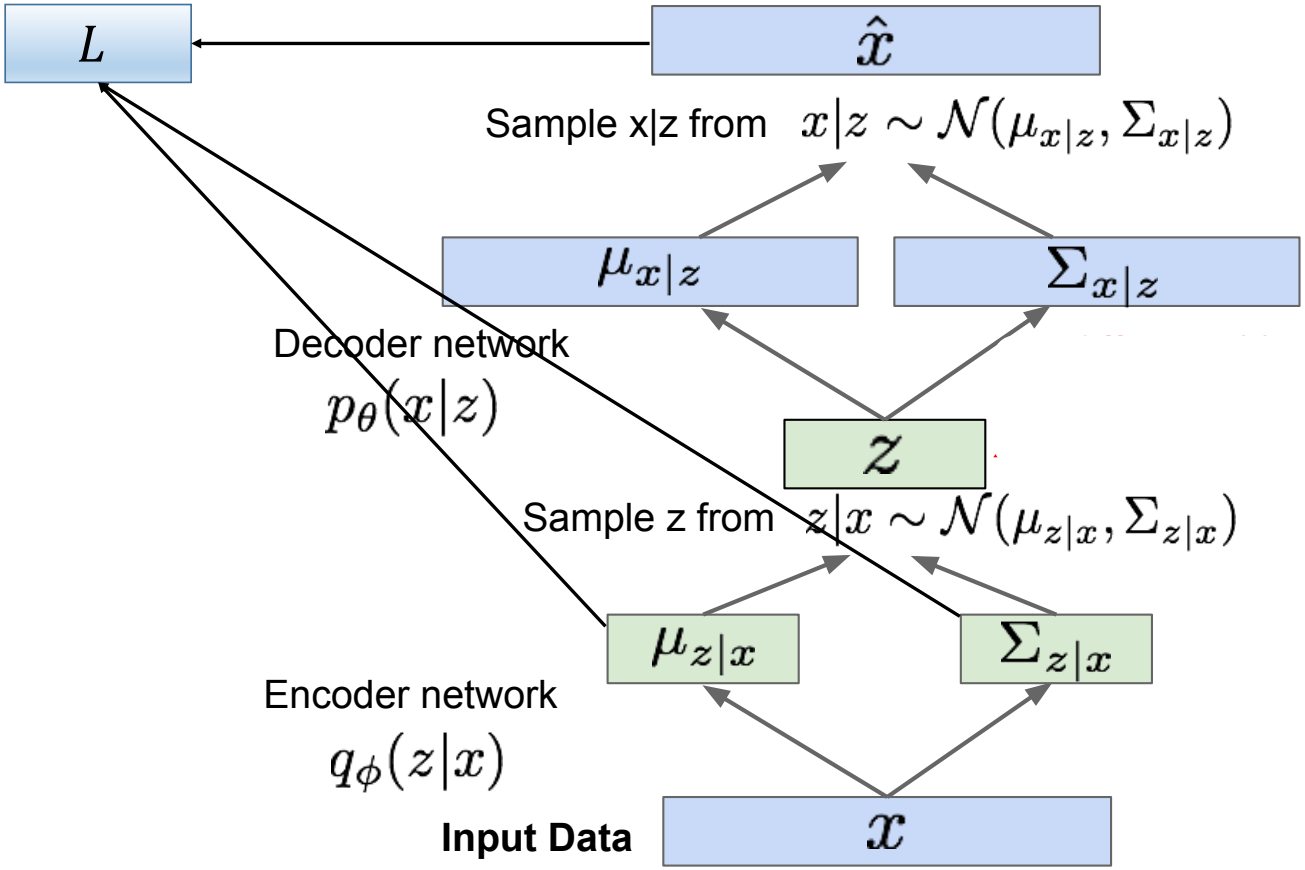
Make approximate posterior distribution close to prior

Variational Autoencoder: How to Calculate Gradients?



- Can we directly apply BP?

Variational Autoencoder: How to Calculate Gradients?



- Can we directly apply BP?

No!

Variational Autoencoder: Reparameterization Trick

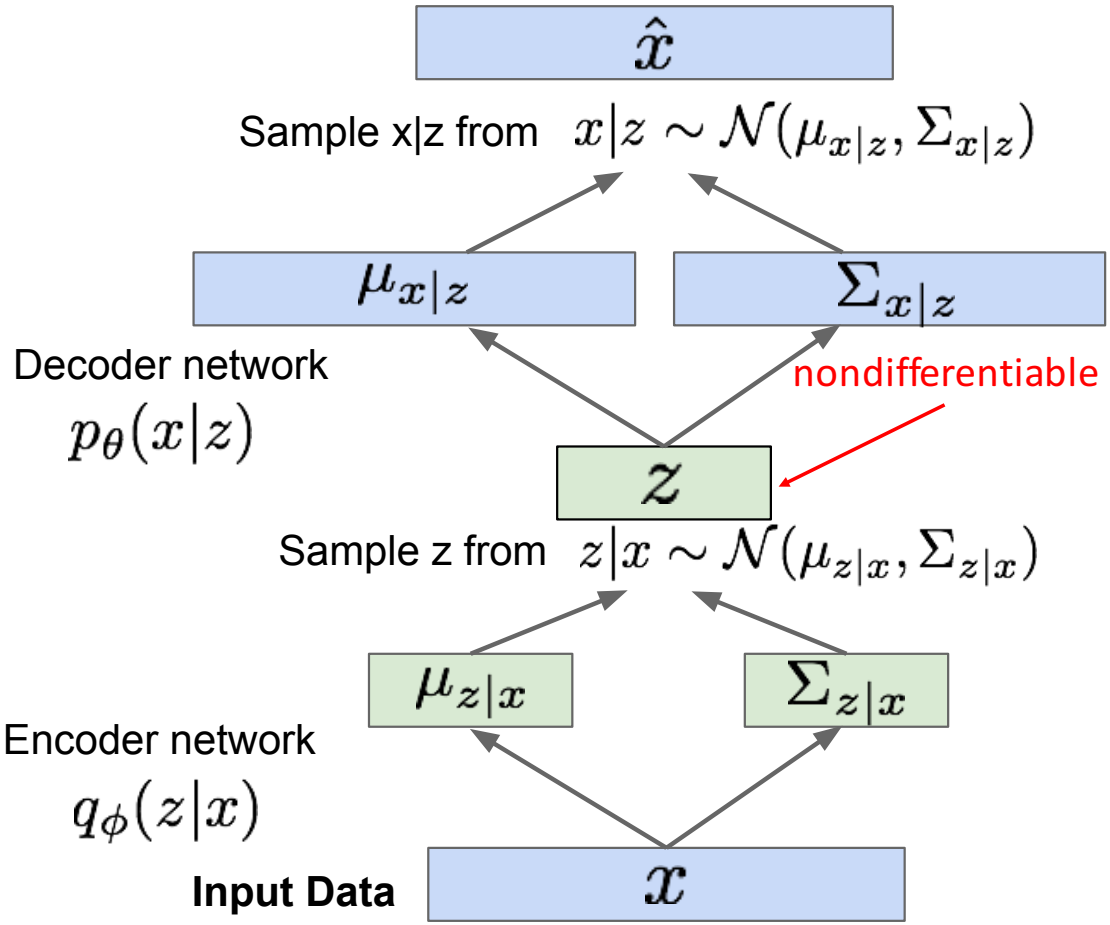
- In the computational graph, we have

$$\mathbf{z} \sim \mathcal{N}\left(\mu_{\mathbf{z}|\mathbf{x}}, \Sigma_{\mathbf{z}|\mathbf{x}}\right) .$$

- To use BP, we need to compute $\frac{\partial \mathbf{z}}{\partial \mu_{\mathbf{z}|\mathbf{x}}}$ and $\frac{\partial \mathbf{z}}{\partial \Sigma_{\mathbf{z}|\mathbf{x}}}$:
  - nondifferentiable!
- Reparameterization:

$$\mathbf{z} = \mu_{\mathbf{z}|\mathbf{x}} + \Sigma_{\mathbf{z}|\mathbf{x}} \xi$$
$$\xi \sim \mathcal{N}(0, 1)$$

  - differentiable w.r.t. both $\mu_{\mathbf{z}|\mathbf{x}}$ and $\Sigma_{\mathbf{z}|\mathbf{x}}$



Variational Autoencoder: Reparameterization Trick

- In the computational graph, we have

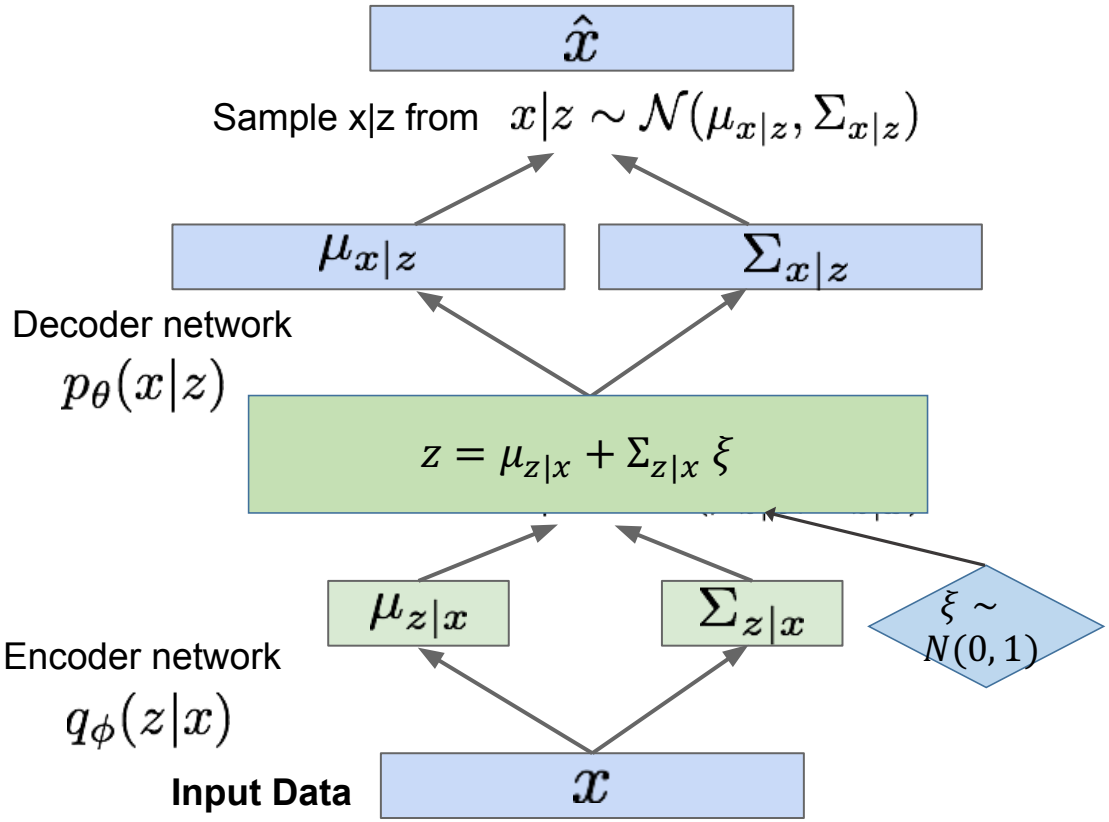
$$\mathbf{z} \sim \mathcal{N}\left(\mu_{\mathbf{z}|\mathbf{x}}, \Sigma_{\mathbf{z}|\mathbf{x}}\right) .$$

- To use BP, we need to compute $\frac{\partial \mathbf{z}}{\partial \mu_{\mathbf{z}|\mathbf{x}}}$ and $\frac{\partial \mathbf{z}}{\partial \Sigma_{\mathbf{z}|\mathbf{x}}}$:
  - nondifferentiable!
- Reparameterization:

$$\mathbf{z} = \mu_{\mathbf{z}|\mathbf{x}} + \Sigma_{\mathbf{z}|\mathbf{x}} \xi$$

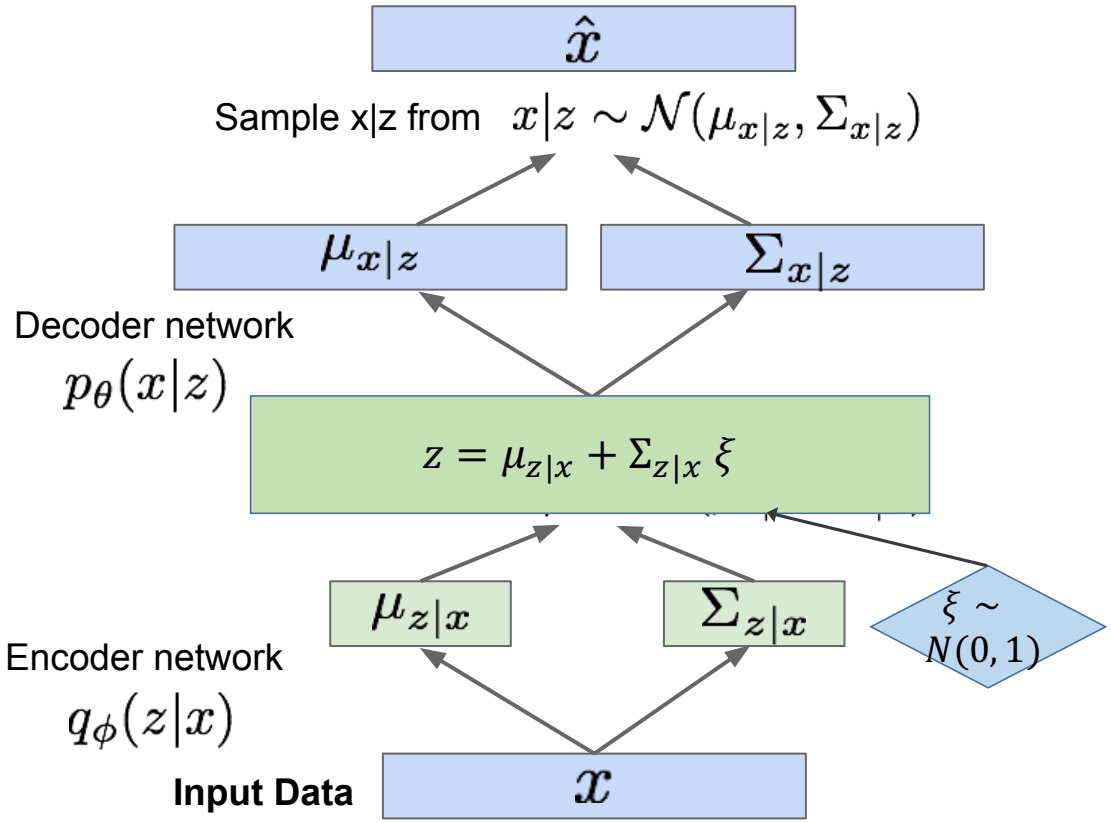
$$\xi \sim \mathcal{N}(0, 1)$$

  - differentiable w.r.t. both $\mu_{\mathbf{z}|\mathbf{x}}$ and $\Sigma_{\mathbf{z}|\mathbf{x}}$



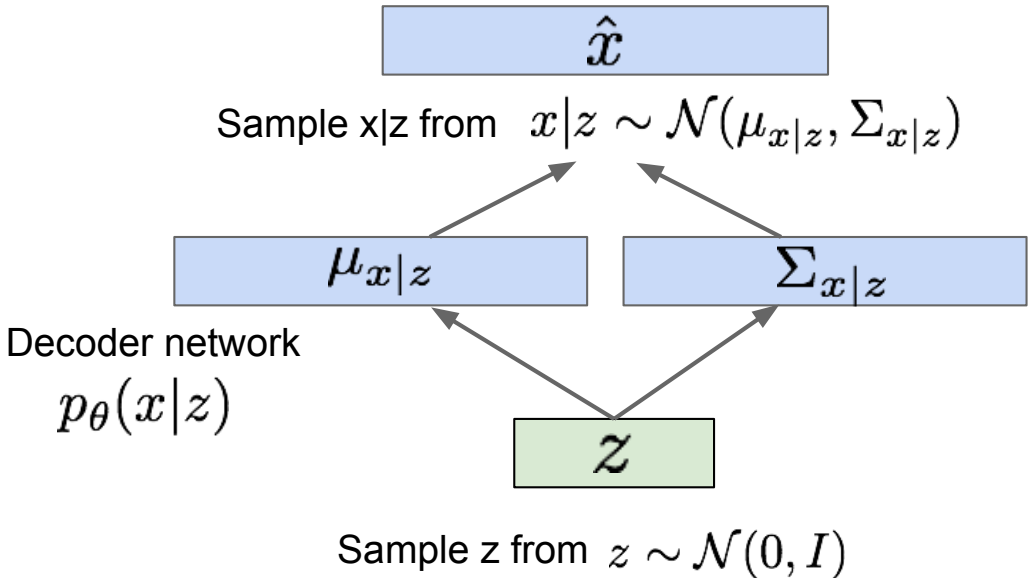
Variational Autoencoder: Implementation

- Define a forward graph as shown on the right.
- Define the loss as the ELBO.
- Associate the loss with an optimizer.



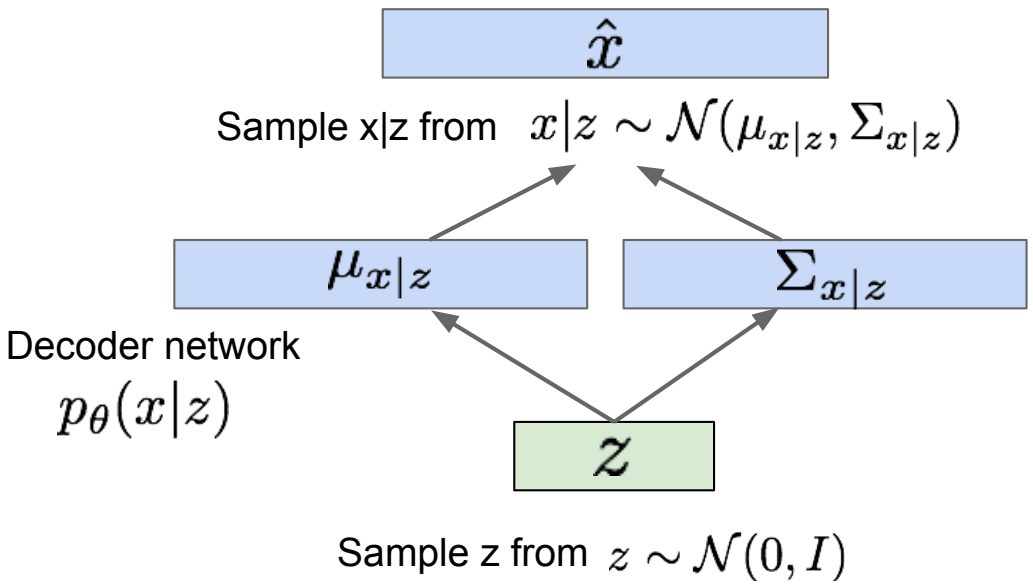
Variational Autoencoder: Generating Data

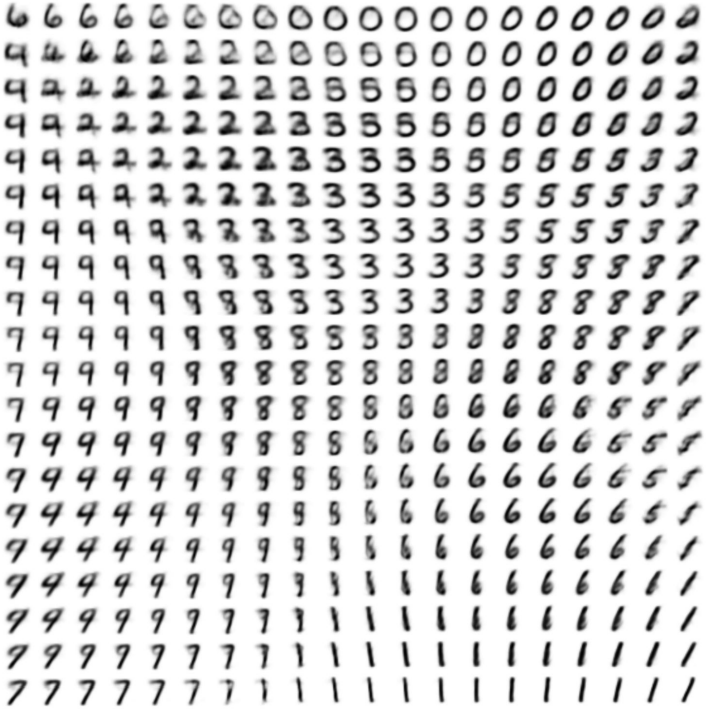
Use deconder network. Sample z from the prior.



Variational Autoencoder: Generating Data

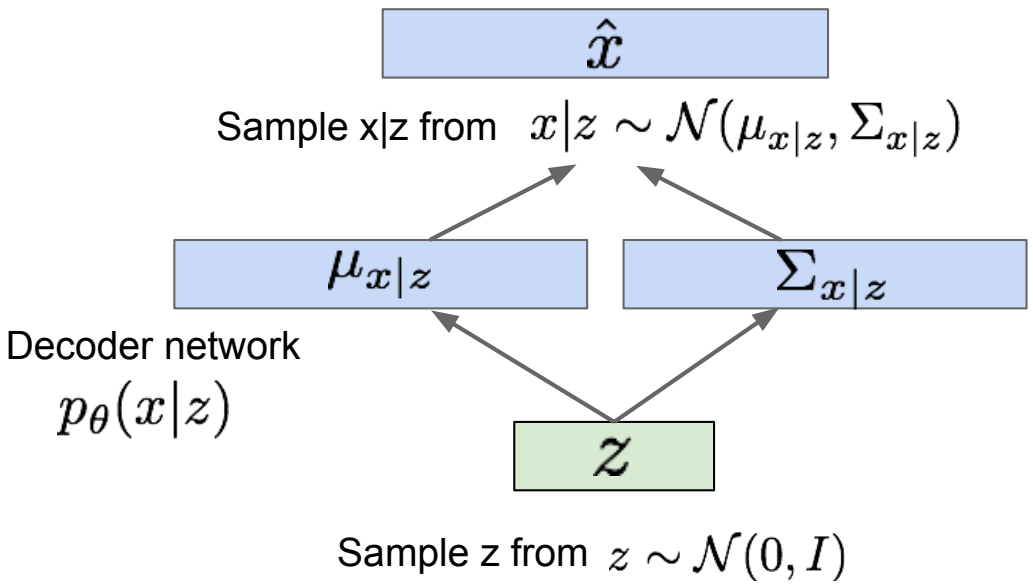
Use deconder network. Sample z from the prior.



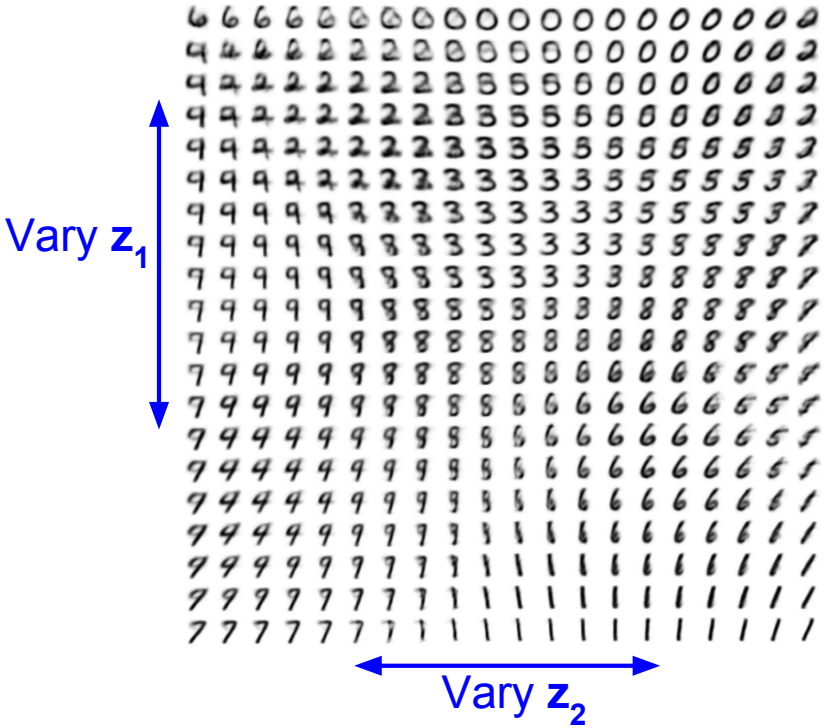


Variational Autoencoder: Generating Data

Use deconder network. Sample z from the prior.



Data manifold for 2-d **z**



Extension: Conditioned VAE

ELBO of standard VAE

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z} \sim q_\phi} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \right] - D_{KL} \left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}) \right)$$

- Can I ask VAE to generate a specific type of data, *e.g.*, generate an image of type "9"?
  - No!
- Solution: extend VAE to conditioned VAE.

Extension: Conditioned VAE

ELBO of standard VAE

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z} \sim q_\phi} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \right] - D_{KL} \left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}) \right)$$

- Can I ask VAE to generate a specific type of data, *e.g.*, generate an image of type "9"?
  - No!
- Solution: extend VAE to conditioned VAE.

Extension: Conditioned VAE

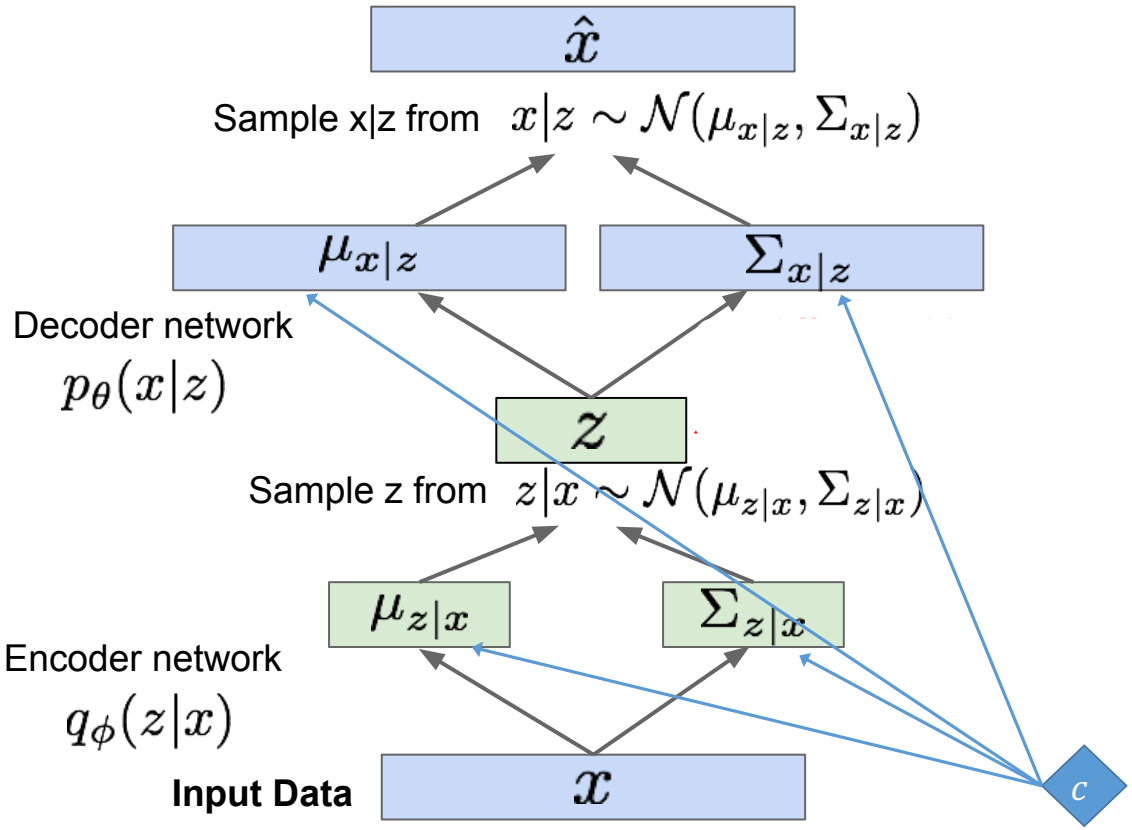
ELBO of standard VAE

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z} \sim q_\phi} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}) \right] - D_{KL} \left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}) \right)$$

- Can I ask VAE to generate a specific type of data, *e.g.*, generate an image of type "9"?
  - No!
- Solution: extend VAE to conditioned VAE.

Extension: Conditioned VAE

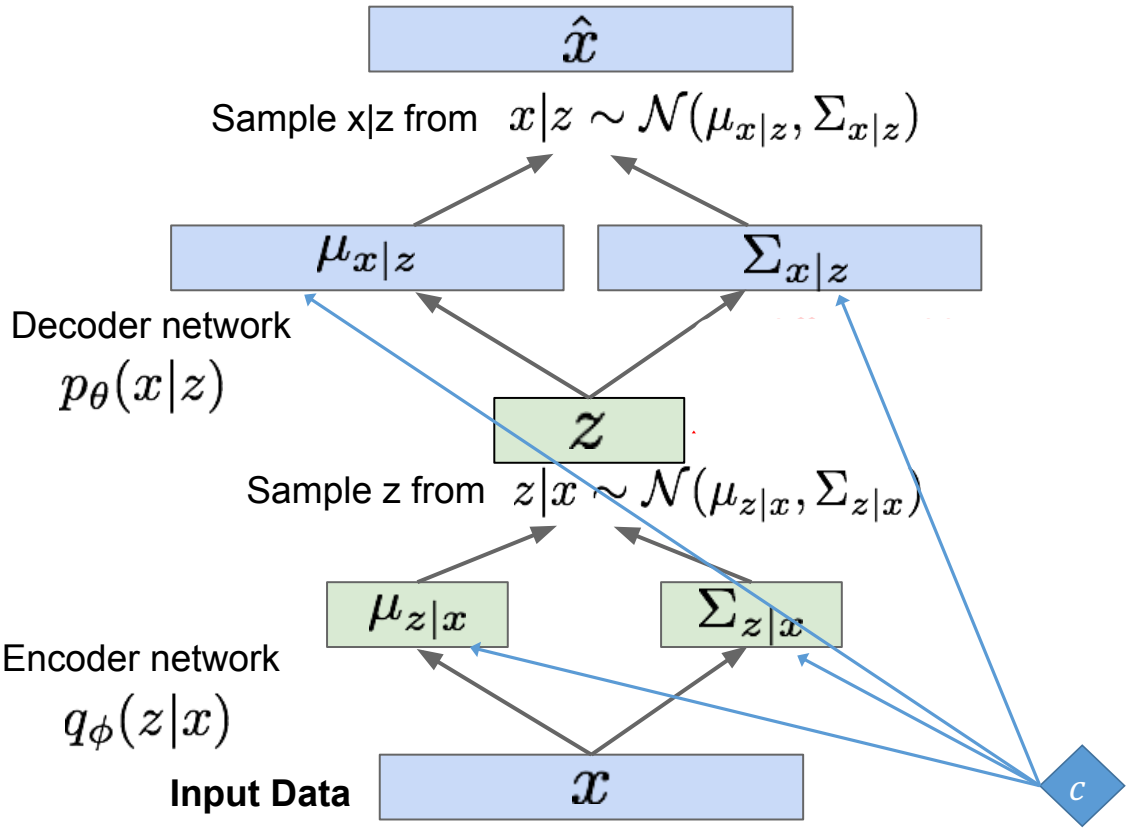
1. Implement by conditioning the encoder and decoder to something:
   - Represented by some variable **c**, *e.g.*, it could be a one-hot vector to represent a class.



Extension: Conditioned VAE

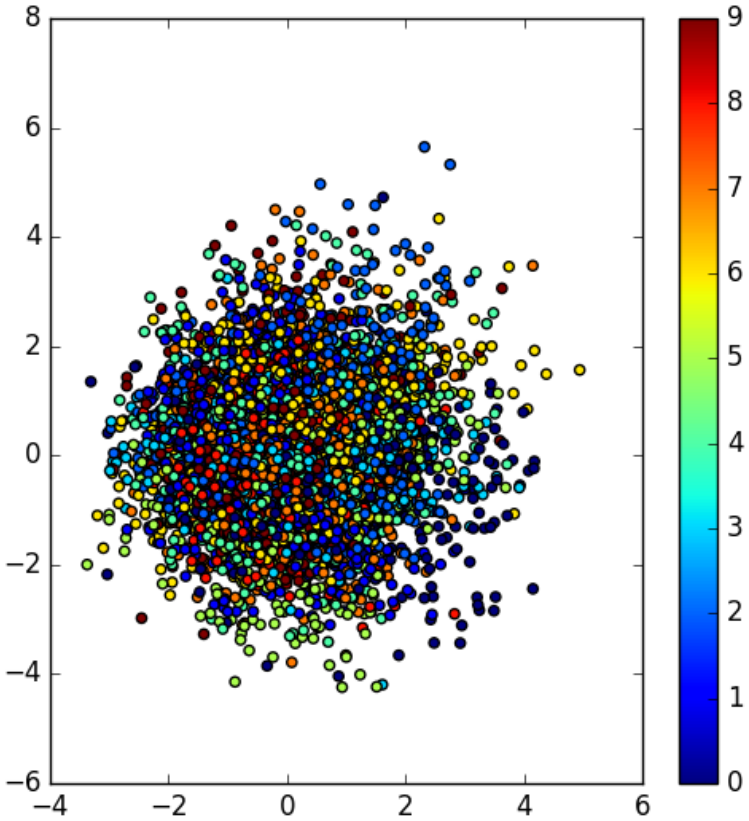
1. Implement by conditioning the encoder and decoder to something:

$$\mathcal{L}(\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z} \sim q_\phi} \left[\log p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z}, \mathbf{c}) \right] - D_{KL} \left(q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c}) \| p_{\boldsymbol{\theta}}(\mathbf{z} \mid \mathbf{c}) \right)$$



Conditional VAE on MNIST

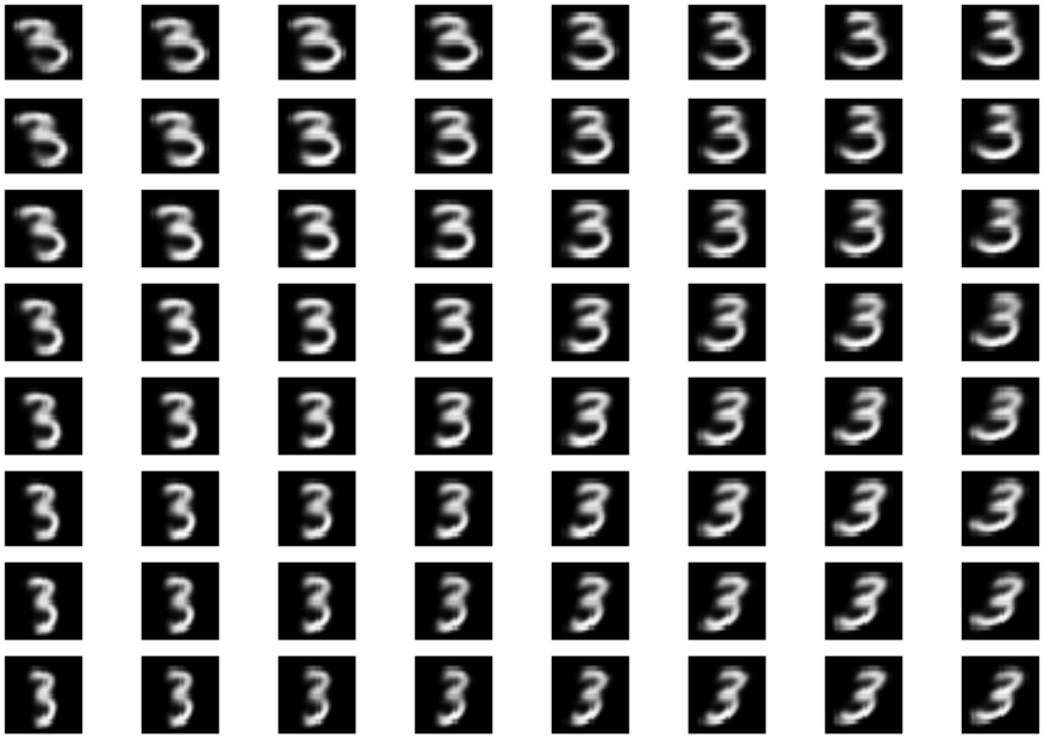
- Assume $p_\theta(\mathbf{z} \mid \mathbf{c}) = \mathcal{N}(0, 1)$ given $\mathbf{c}$.
- The figure shows the learned $q_\phi(\mathbf{z} \mid \mathbf{x}, \mathbf{c})$ for each class (different colors)[1].



[1] Image credit: https://wiseodd.github.io/techblog/2016/12/17/conditional-vae/

Conditional VAE on MNIST: Generating "3"[2]

- Set $\mathbf{c}$ to be a one-hot vector representing 3.
- Changing z_1 (on the y-axis) makes the digit style becomes narrower.
- Varying z_2 (on the x-axis) appears to rotate the digit slightly and elongate the lower portion in relation to the upper portion.



[2] Image credit: http://nnormandin.com/science/2017/07/01/cvae.html

Extension: Semi-supervised VAE

1. What if we have both labeled and unlabeled data?
2. Implement by conditioning the encoder and decoder to something:
   - Represented by some random variable $\mathbf{y}$, *e.g.*, it could a label.

Generative process

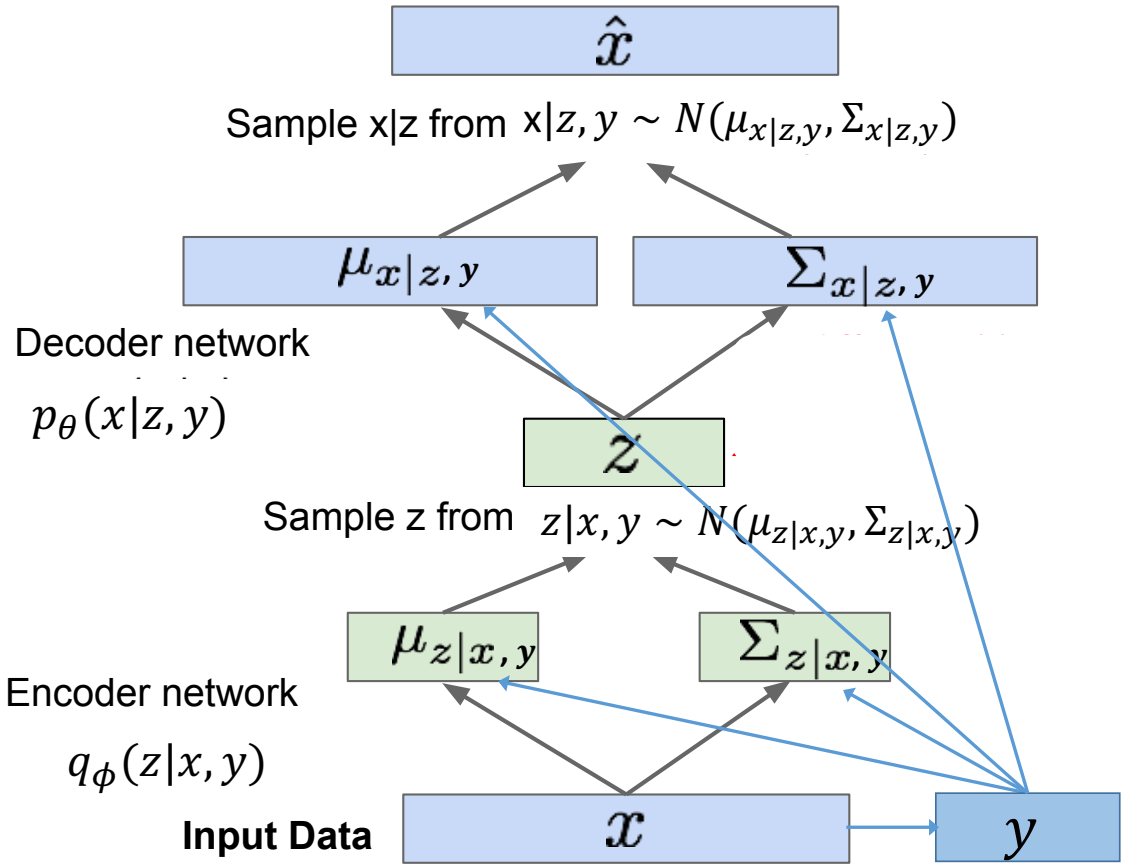
$$p(y) = \text{Cat}(y|\pi), \quad p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}),$$
$$p_\theta(\mathbf{x}|y, \mathbf{z}) = \mathcal{N}\left(\mathbf{x}; \mu_{\mathbf{x}|\mathbf{z},y}, \Sigma_{\mathbf{x}|\mathbf{z},y}\right)$$

Inference model

$$q_\phi(\mathbf{z}|y, \mathbf{x}) = \mathcal{N}\left(\mathbf{z}; \mu_{\mathbf{z}|\mathbf{x},y}, \Sigma_{\mathbf{z}|\mathbf{x},y}\right), \quad q_\phi(y|\mathbf{x}) = \text{Cat}(y|\pi_\phi(\mathbf{x}))$$

Extension: Semi-supervised VAE

1. What if we have both labeled and unlabeled data?
2. Implement by conditioning the encoder and decoder to something:



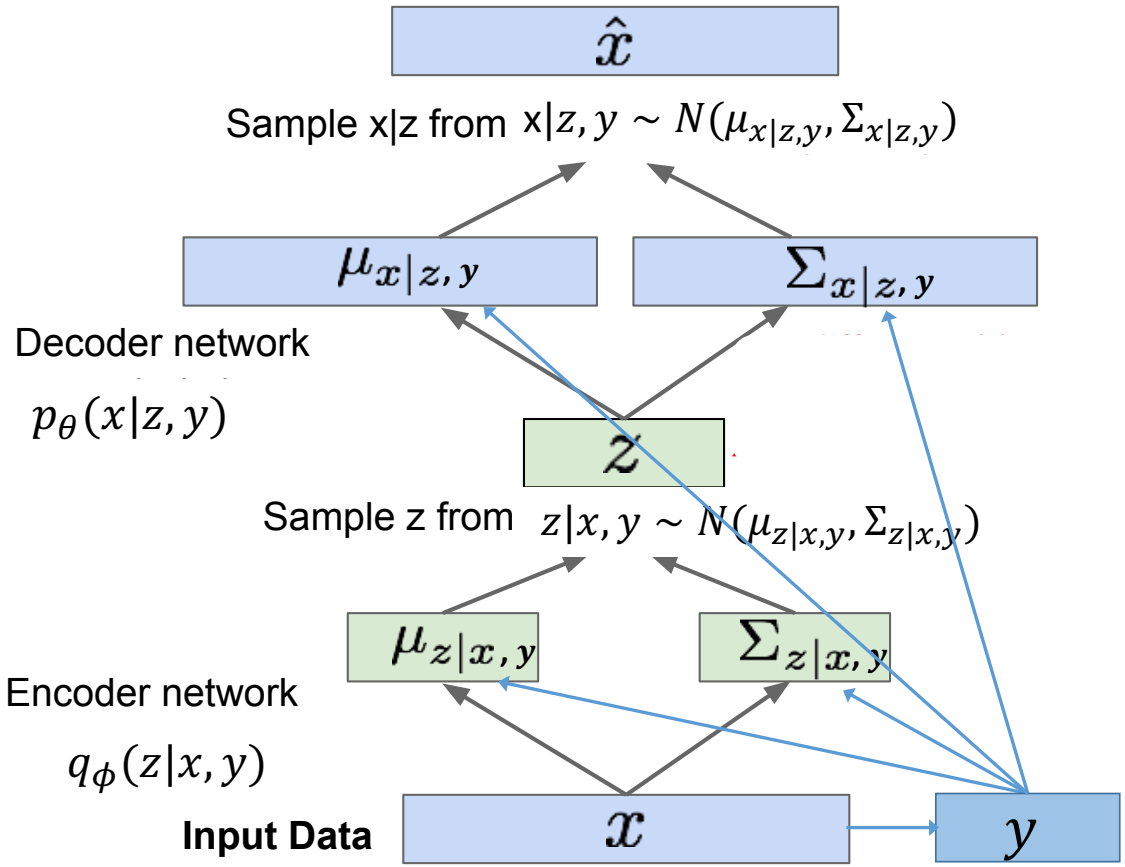
G: $p(y) = \text{Cat}(y|\pi), p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I}), p_\theta(\mathbf{x}|y,\mathbf{z}) = \mathcal{N}\left(\mathbf{x}; \mu_{\mathbf{x}|\mathbf{z},y}, \Sigma_{\mathbf{x}|\mathbf{z},y}\right)$

I: $q_\phi(\mathbf{z}|y,\mathbf{x}) = \mathcal{N}\left(\mathbf{z}; \mu_{\mathbf{z}|\mathbf{x},y}, \Sigma_{\mathbf{z}|\mathbf{x},y}\right), \quad q_\phi(y|\mathbf{x}) = \text{Cat}(y|\pi_\phi(\mathbf{x}))$

Extension: Semi-supervised VAE

1. What if we have both labeled and unlabeled data?
2. Implement by conditioning the encoder and decoder to something:

$$\mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z},y|\,\mathbf{x}\sim q_\phi}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y)\right] - D_{KL}\left(q_\phi(\mathbf{z}, y|\,\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{z}, y)\right)$$



Extension: Semi-supervised VAE

For data with labels:

$$\mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z}, y|\,\mathbf{x} \sim q_\phi}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y)\right] - D_{KL}\left(q_\phi(\mathbf{z}, y|\,\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}, y)\right)$$
$$= \mathbb{E}_{\mathbf{z}, y|\,\mathbf{x} \sim q_\phi}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y) + \log p_{\boldsymbol{\theta}}(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}, y|\,\mathbf{x})\right]$$

Extension: Semi-supervised VAE

For data with labels:

$$\mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z}, y|\,\mathbf{x} \sim q_\phi}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y)\right] - D_{KL}\left(q_\phi(\mathbf{z}, y|\,\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}, y)\right)$$
$$= \mathbb{E}_{\mathbf{z}, y|\,\mathbf{x} \sim q_\phi}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y) + \log p_{\boldsymbol{\theta}}(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}, y|\,\mathbf{x})\right]$$

For data without labels:

$$\mathcal{L}_u(\mathbf{x}, \boldsymbol{\theta}, \phi) = \sum_y \mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \phi)$$
$$= \sum_y q_\phi(y|\,\mathbf{x})\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y) + \log p_{\boldsymbol{\theta}}(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}, y|\,\mathbf{x})\right]$$

Extension: Semi-supervised VAE

For data with labels:

$$\mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \phi) = \mathbb{E}_{\mathbf{z}, y|\,\mathbf{x} \sim q_\phi}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y)\right] - D_{KL}\left(q_\phi(\mathbf{z}, y|\,\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}, y)\right)$$
$$= \mathbb{E}_{\mathbf{z}, y|\,\mathbf{x} \sim q_\phi}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y) + \log p_{\boldsymbol{\theta}}(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}, y|\,\mathbf{x})\right]$$

For data without labels:

$$\mathcal{L}_u(\mathbf{x}, \boldsymbol{\theta}, \phi) = \sum_y \mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \phi)$$
$$= \sum_y q_\phi(y|\,\mathbf{x})\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}\,|\,\mathbf{z}, y) + \log p_{\boldsymbol{\theta}}(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}, y|\,\mathbf{x})\right]$$

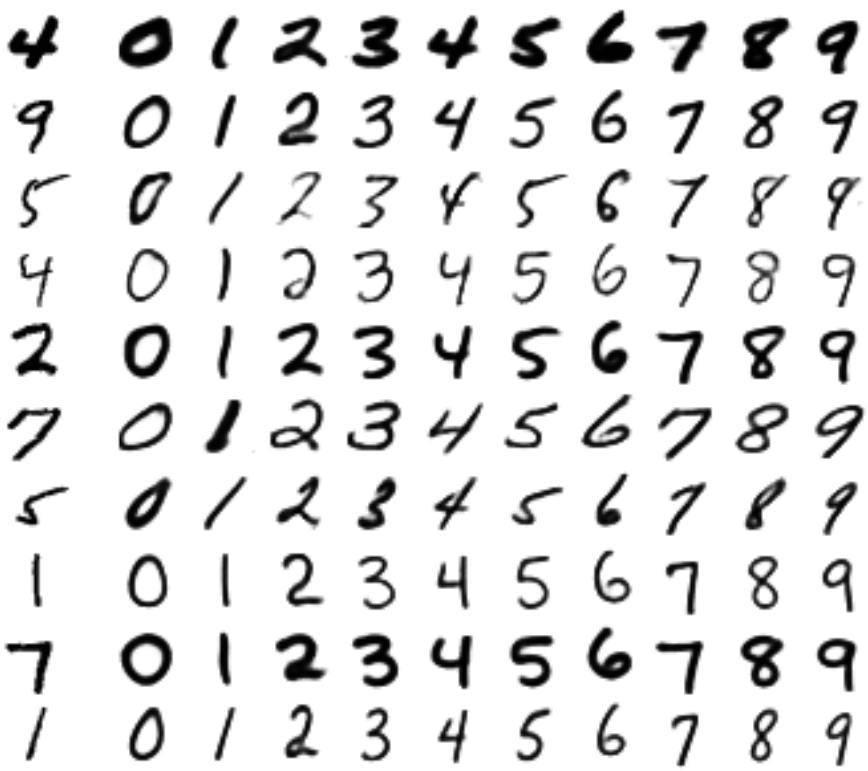
Total loss:

$$\tilde{L} = \mathcal{L}(\mathbf{x}, y, \boldsymbol{\theta}, \phi) + \mathcal{L}_u(\mathbf{x}, \boldsymbol{\theta}, \phi)$$

Extension: Semi-supervised VAE

(a) Handwriting styles for MNIST obtained by fixing the class label and varying the 2D latent variable $\mathbf{z}$

Extension: Semi-supervised VAE



(b) MNIST analogies

(c) SVHN analogies

Analogical reasoning with generative semi-supervised models using a high-dimensional $\mathbf{z}$-space. The leftmost columns show images from the test set. The other columns show analogical fantasies of $\mathbf{x}$ by the generative model, where the latent variable $\mathbf{z}$ of each row is set to the value inferred from the test-set image on the left by the inference network. Each column corresponds to a class label $\mathbf{y}$.

Questions

Can we use the VAE framework to do image segmentation? If so, how?

Summary

Pros:

- Principled approach to generative models.
- Allows inference of $q(\mathbf{z} \mid \mathbf{x})$, may 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, *e.g.*, normalizing flows.
- Incorporating structure in latent variables.