Variational Autoencoders

32x32 CIFAR-10

Labeled Faces in the Wild

Some slides from Fei-Fei Li, Ranjay Krishna, Danfei Xu

ECE324, Winter 2023

Michael Guerzhoy

Key ideas

- The Variational Autoencoder is trained by constraining the encoder so that training becomes tractable
- The encoder tries to approximate $p(z|x)$, the true conditional distribution of the code given an input

Pre-requisites

- Understand why “vanilla” autoencoders are difficult to train
- Understand training a model with maximum likelihood
- Understand law of total expectation

A bit of information theory

- Suppose V is a random variable with the probability distribution

$P(v = 0)$	$P(v = 1)$	$P(v = 2)$	$P(v = 3)$	$P(v = 4)$	$P(v = 5)$	$P(v = 6)$
0.1	0.002	0.52	…	…	…	…

- The *surprise* $S(V = v)$ for each value of v is defined as

$$S(V = v) = -\log_2 P(V = v)$$

  - The smaller the probability of the event, the larger the surprise if we observe the event
  - 0 surprise for events with probability 1
  - Infinite surprise for events with probability 0

Surprise and Message Length

- Suppose we want to communicate the value of v to a receiver. It makes sense to use longer binary codes for rarer values of V
  - Can use $-\log_2 P(V = v)$ bits to communicate v
    - Check that this makes sense if $P(V = 0) = 1$ (no need to transmit any information) and $P(V = 0) = P(V = 1) = \frac{1}{2}$ (need one bit to transmit v)
    - Fractional bits only make sense for longer messages
    - Example: UTF-8 uses more bytes for rare symbols
    - "Amount of information"

Message length

$P(a)=P(b)=0.25$, $P(c)=0.5$

- Use 00 for a, 01 for b, and 1 for c
- Can decode any sequence

Entropy: Average/Expected Surprise

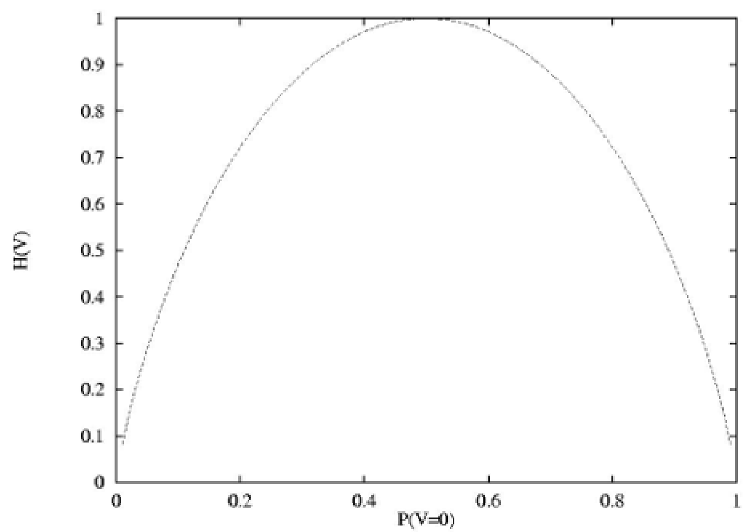
- The entropy of V, $H(V)$ is defined as

$$H(V) = \sum_{v} -P(V = v) \log_2 P(V = v)$$

- The average surprise for one "trial" of V
  - The average message length when communicating the outcome v
- The average amount of information we get by seeing one value of V (in bits)

Entropy: How "Spread Out" the distribution is

- High entropy of V means we cannot predict what the value of V might be
- Low entropy means we are pretty sure we know what the value of V is every time

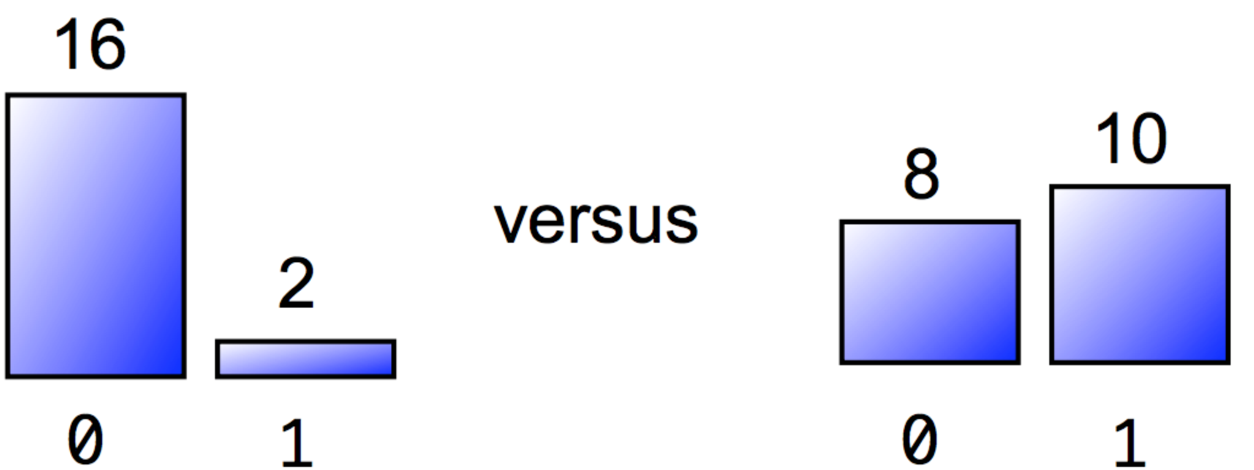


The entropy of a Bernoulli variable is maximized when $p = 0.5$

Entropy of Coin Flips

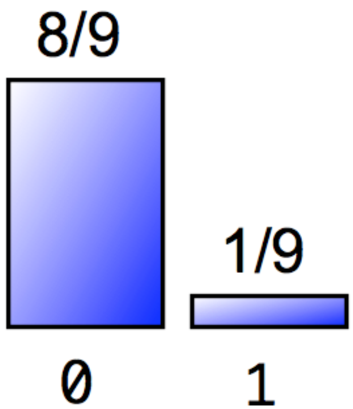
Sequence 1:
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 ... ?

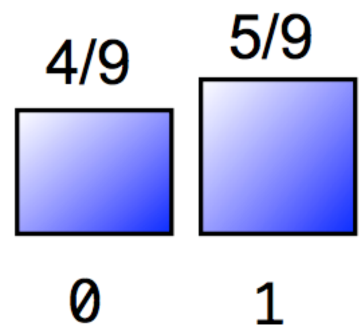
Sequence 2:
0 1 0 1 0 1 1 1 0 1 0 0 1 1 0 1 0 1 ... ?



Entropy of Coin Flips

$$H(V) = \sum_{v} -P(V = v) \log_2 P(V = v)$$





$$-\frac{8}{9}\log_2\frac{8}{9} - \frac{1}{9}\log_2\frac{1}{9} \approx \frac{1}{2}$$

$$-\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} \approx 0.99$$

Higher Entropy; more uncertainty about the outcome

Three views of Entropy

We are considering a random variable V, and a sample v from it

The Entropy is

1. Average Surprise at v
2. Average message length when transmitting v in an efficient way
3. Measure of the ”spread-out”-ness of the distribution V

Kullback-Leibler Divergence

- Expected excess surprise when sampling from P with an expected distribution Q
- Extra bits needed to communicate samples from P when using a code for Q

$$D_{KL}(P||Q) = -\sum_v (P(v) \log Q(v) - P(v) \log P(v))$$

$$D_{KL}(P||Q) = -\sum_v P(v) \log \left(\frac{Q(v)}{P(v)} \right)$$

- Continuous version: $D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(v) \log \frac{p(v)}{q(v)} dv$
- Can view as a measure of the difference between the distributions P and Q

KL divergence is non-negative

- $\log a \leq a - 1$ for $a > 0$
  - $\log a$ is concave, and only equals $a - 1$ at $a = 1$

$$
\begin{aligned}
-D(p||q) &= -\sum_x p(x) \ln \frac{p(x)}{q(x)} \\
&= \sum_x p(x) \ln \frac{q(x)}{p(x)} \\
&\overset{\text{(a)}}{\leq} \sum_x p(x) \left(\frac{q(x)}{p(x)} - 1 \right) \\
&= \sum_x q(x) - \sum_x p(x) \\
&= 1 - 1 \\
&= 0
\end{aligned}
$$

https://stats.stackexchange.com/questions/335197/why-kl-divergence-is-non-negative

KL divergence as an expectation

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(v) \log \frac{p(v)}{q(v)} dv$$

$$= E_{v \sim p(v)} \log \left(\frac{p(v)}{q(v)} \right)$$

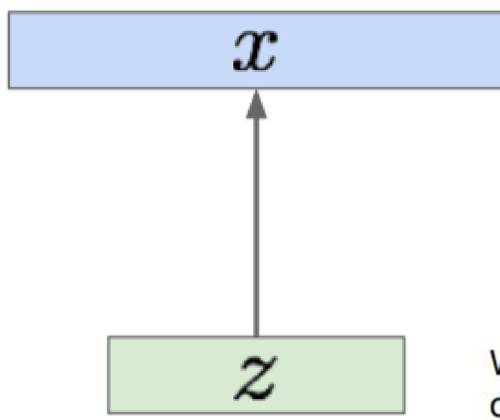
- Intuition: $D_{KL}(P||Q)$ is non-negative because $\frac{p(v)}{q(v)}$ is given more weigh for larger $p(v)$

Variational Autoeoncoders

- Assume the training data $\{x^{(i)}\}_{i=1\dots N}$ is generated from the distribution of unobserved code z

Sample from true conditional

$p_{\theta^*}(x \mid z^{(i)})$



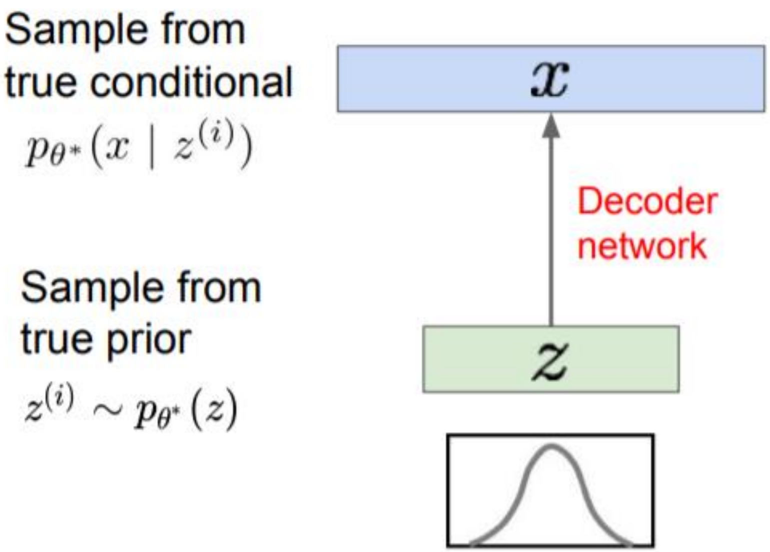
Intuition (remember from autoencoders!): **x** is an image, **z** is latent factors used to generate **x:** attributes, orientation, etc.

Sample from true prior

$z^{(i)} \sim p_{\theta^*}(z)$

We want to estimate the true parameters θ^* of this generative model given training data x.

Variational Autoeoncoders



We want to estimate the true parameters θ^* of this generative model given training data x.

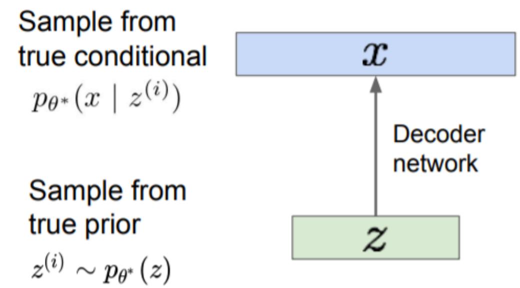
How should we represent this model?

Choose prior p(z) to be simple, e.g. Gaussian. Reasonable for latent attributes, e.g. pose, how much smile.

Conditional p(x|z) is complex (generates image) => represent with neural network

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

Training the model



- The maximum likelihood approach is to maximize the likelihood of the training data
- We observe x, but don't observe z
- Can still in principle compute the likelihood of the observed data by using the law of total probability

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

- Intuition: compute the weighted sum of the probabilities of seeing the x we see, with the weights being the probabilities of the possible code z

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

- Problem: the likelihood is intractable
  - The decoder is a complicated function so can't compute a closed-form integral of a function involving $p_\theta(x|z)$
  - Cannot compute numerically since $p_\theta(x|z)$ is mostly 0
    - For most z's, it's unlikely is the reconstruction x that we actually see
    - If $p_\theta(x|z)$ is mostly 0, need *a lot of samples* to not miss the few local maxima

Variational approach

- Learn $q_\phi(z|x)$, an approximation of
$$p_\theta(z|x) = p_\theta(x|z)p_\theta(z)/p_\theta(x)$$
- Will allow us to compute a tractable lower bound on the data likelihood

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

Decoder network gives $p_\theta(x|z)$, can compute estimate of this term through sampling (need some trick to differentiate through sampling).

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.

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

Decoder:
reconstruct
the input data

Encoder:
make approximate
posterior distribution
close to prior

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

Encoder

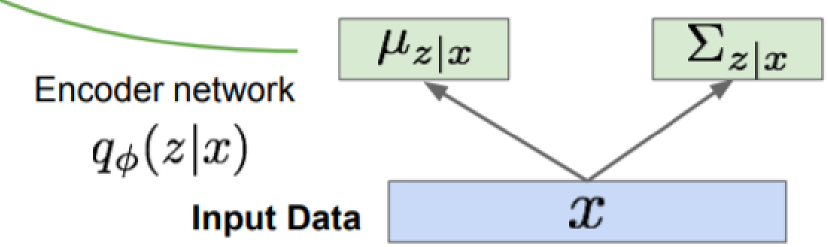
Putting it all together: maximizing the likelihood lower bound

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

$$D_{KL}(\mathcal{N}(\mu_{z|x}, \Sigma_{z|x}) || \mathcal{N}(0, I))$$

Have analytical solution

Make approximate posterior distribution close to prior



KL divergence between two Gaussians

$$D_{KL}(p||q) = \frac{1}{2}\left[\log\frac{|\Sigma_q|}{|\Sigma_p|} - k + (\boldsymbol{\mu_p} - \boldsymbol{\mu_q})^T\Sigma_q^{-1}(\boldsymbol{\mu_p} - \boldsymbol{\mu_q}) + tr\left\{\Sigma_q^{-1}\Sigma_p\right\}\right]$$

https://mr-easy.github.io/2020-04-16-kl-divergence-between-2-gaussian-distributions/

Variational Autoencoders

Putting it all together: maximizing the likelihood lower bound

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

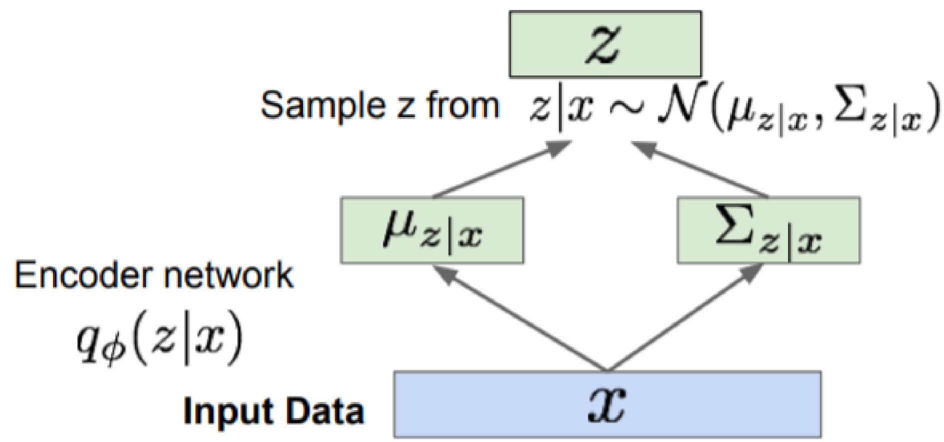
Reparameterization trick to make sampling differentiable:

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

Input to the graph

$$z = \boxed{\mu_{z|x}} + \boxed{\epsilon}\boxed{\sigma_{z|x}}$$

Part of computation graph



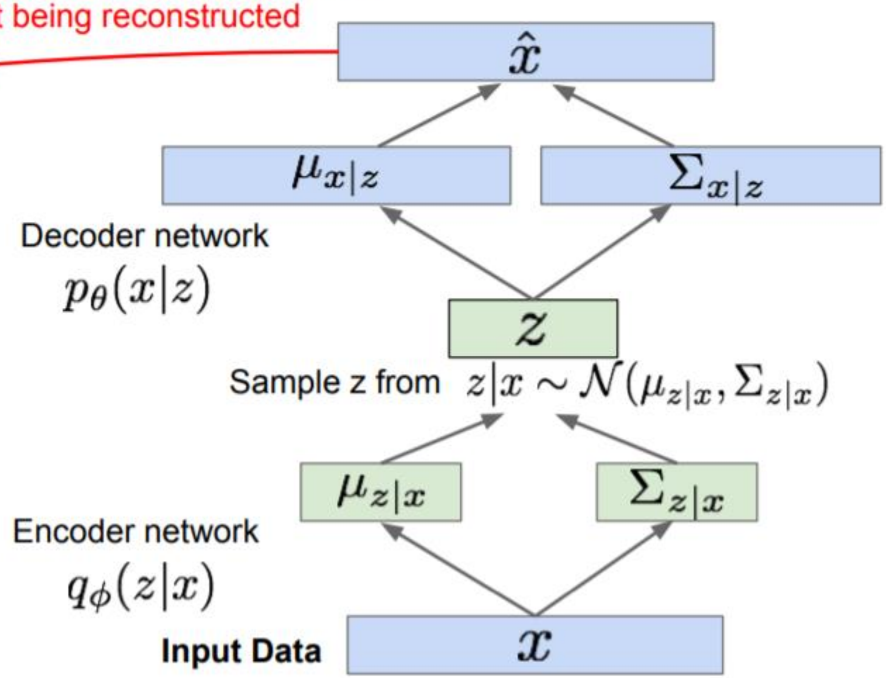
- Sample z's using the encoder
- Estimate $E_z[\log p_\theta(x^{(i)}|z)] \approx 1/N \sum_{j=1\ldots N} \log p_\theta(x^{(i)}|z_j)$
- Can differentiate $1/N \sum_{j=1\ldots N} \log p_\theta(x^{(i)}|z_j)$ w.r.t $\mu_{z|x}$ and $\sigma_{z|x}$, and then backprop to ϕ

Variational Autoencoders

Maximize likelihood of original input being reconstructed

Putting it all together: maximizing the likelihood lower bound

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



Looking back

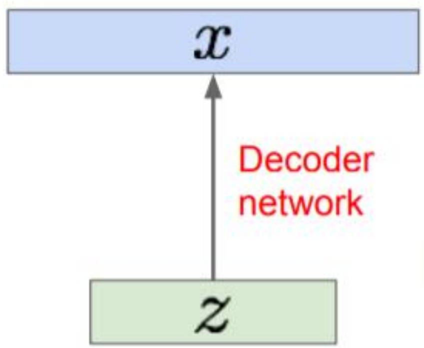
- $p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz = E_{z\sim N(0,I)} p_\theta(x|z)$
  - Not tractable: hard to find z's for which $p_\theta(x|z)$ is non-zero
- $E_{z\sim q_\phi(z|x^{(i)})} \log p_\theta(x|z)$
  - More tractable since we learned a function that gives us good z's

Generating data

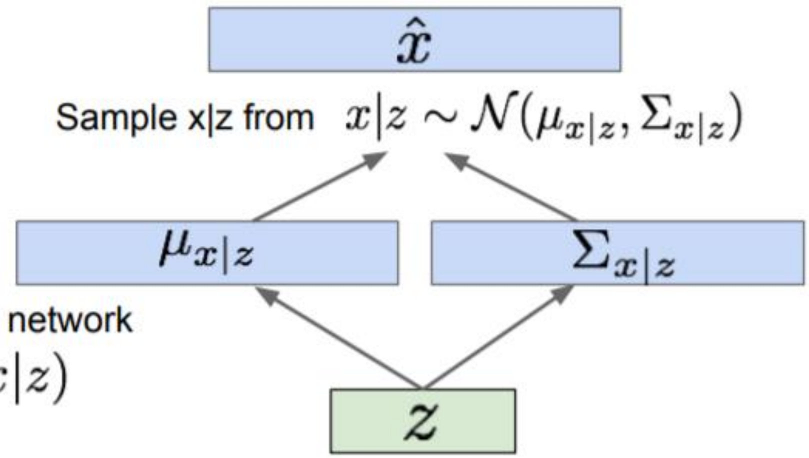
Our assumption about data generation process

Sample from true conditional $p_{\theta^*}(x \mid z^{(i)})$

Sample from true prior $z^{(i)} \sim p_{\theta^*}(z)$



Now given a trained VAE:
use decoder network & sample z from prior!

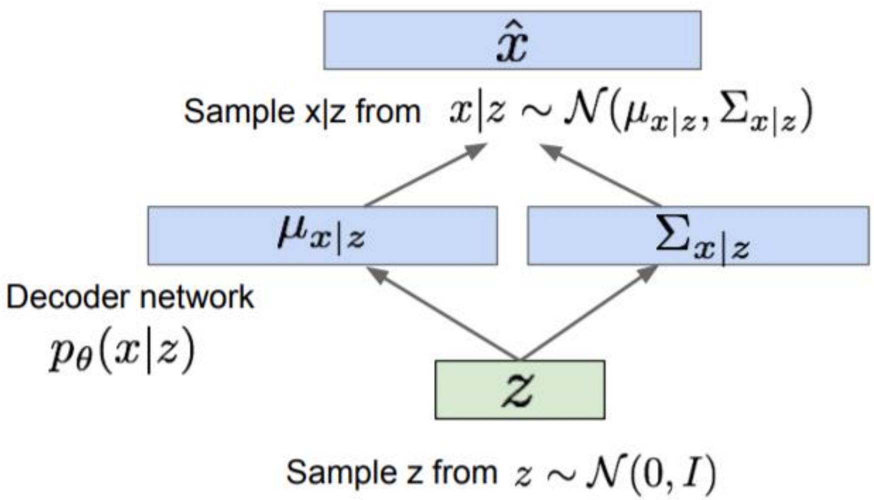


Decoder network $p_\theta(x|z)$

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

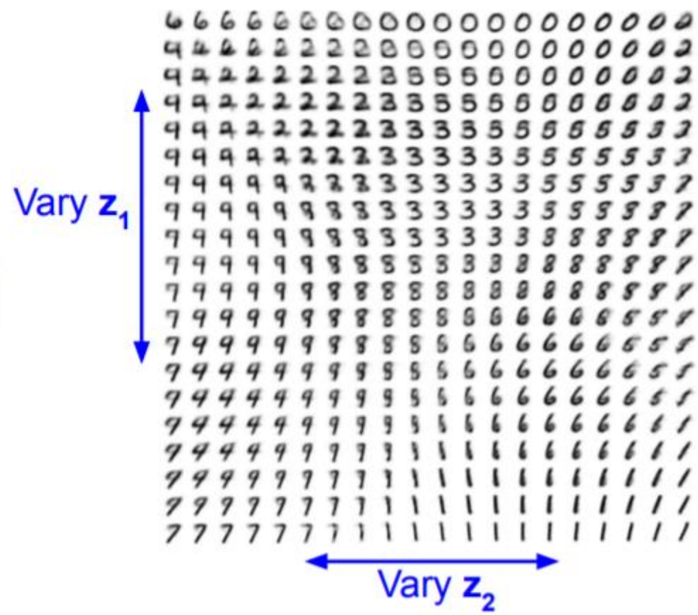
Kingma and Welling, "Auto-Encoding Variational Bayes", ICLR 2014

Use decoder network. Now sample z from prior!



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

Data manifold for 2-d **z**



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