



# Deep Learning

## Variational Autoencoder (VAE)

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir



<https://www.aparat.com/mehran.safayani>



[https://github.com/safayani/deep\\_learning\\_course](https://github.com/safayani/deep_learning_course)

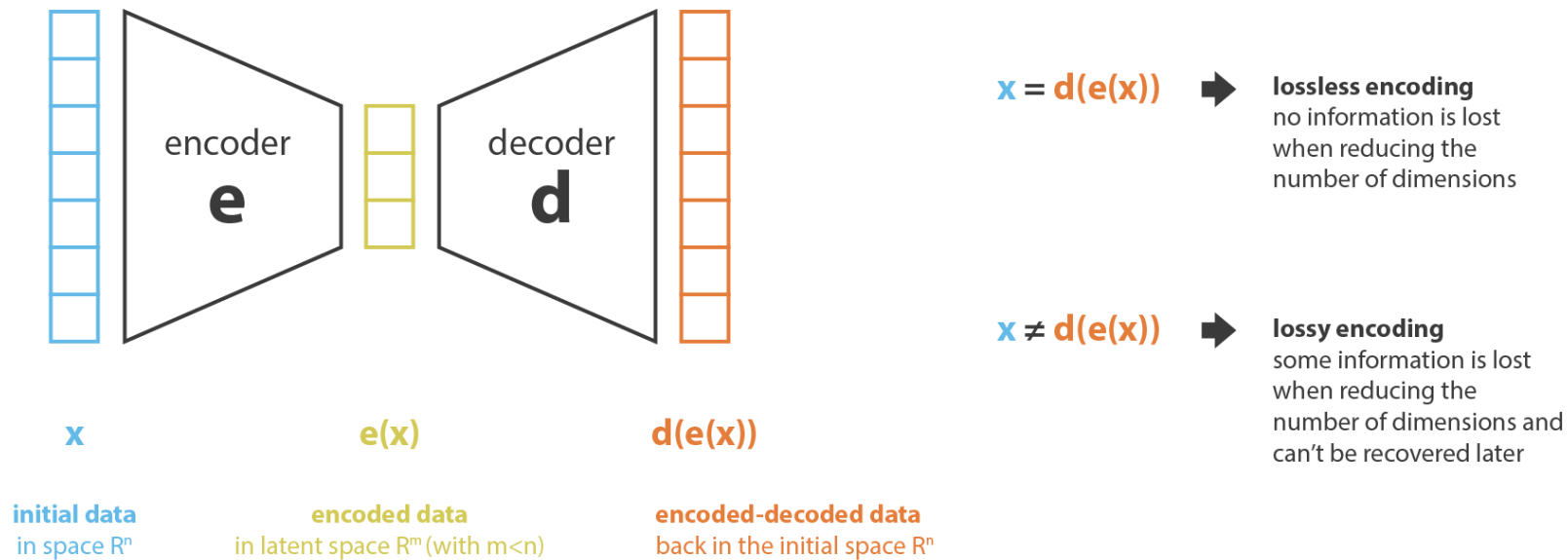


# Face images generated with a Variational Autoencoder



# Dimensionality reduction, PCA and autoencoders

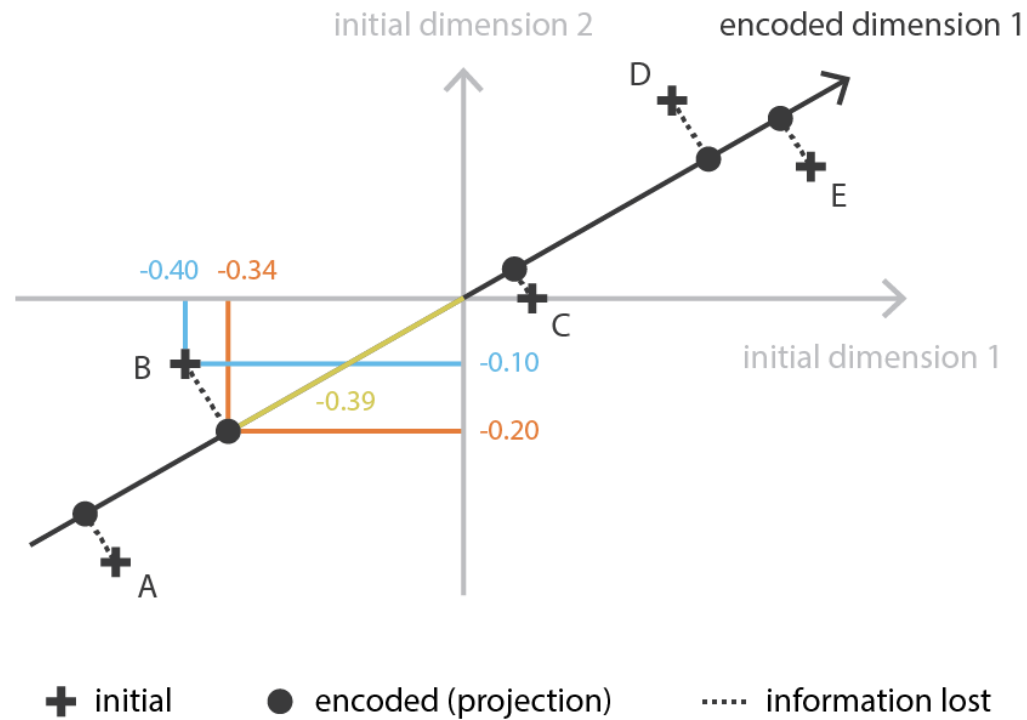
Illustration of the dimensionality reduction principle with encoder and decoder.



$$(e^*, d^*) = \arg \min_{(e, d) \in E \times D} \epsilon(x, d(e(x)))$$

$$\epsilon(x, d(e(x)))$$

# Principal components analysis (PCA)

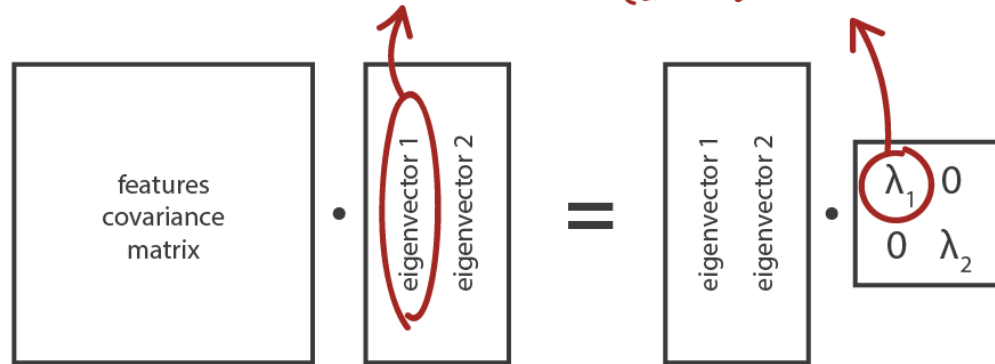


Point	Initial	Encoded	Decoded
A	$(-0.50, -0.40)$	-0.63	$(-0.54, -0.33)$
B	$(-0.40, -0.10)$	-0.39	$(-0.34, -0.20)$
C	$(0.10, 0.00)$	0.09	$(0.07, 0.04)$
D	$(0.30, 0.30)$	0.41	$(0.35, 0.21)$
E	$(0.50, 0.20)$	0.53	$(0.46, 0.27)$

# Principal components analysis (PCA)

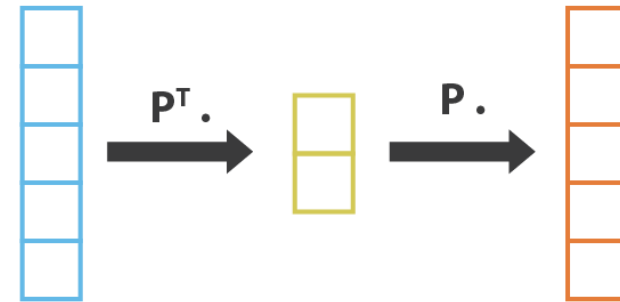
eigenvector associated to the greatest eigenvalue  $\lambda_1$  and orthogonal to other columns

greatest eigenvalue of the covariance matrix  $C$  (in absolute value)



$$C \cdot P = P \cdot \lambda$$

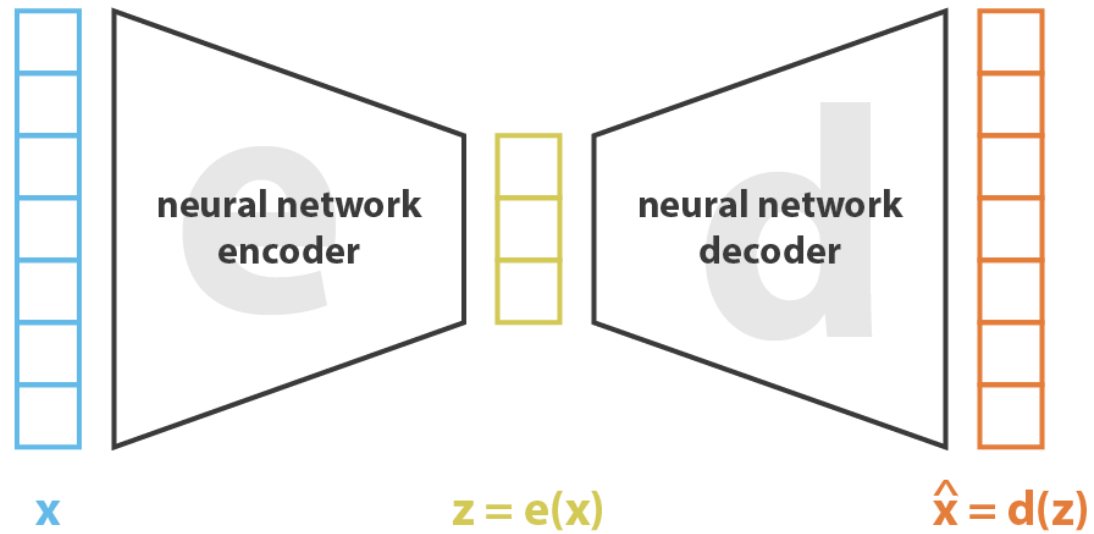
notice that  $d(e(x)) \neq x$  as soon as  $C \neq P \lambda P^T$



$x$

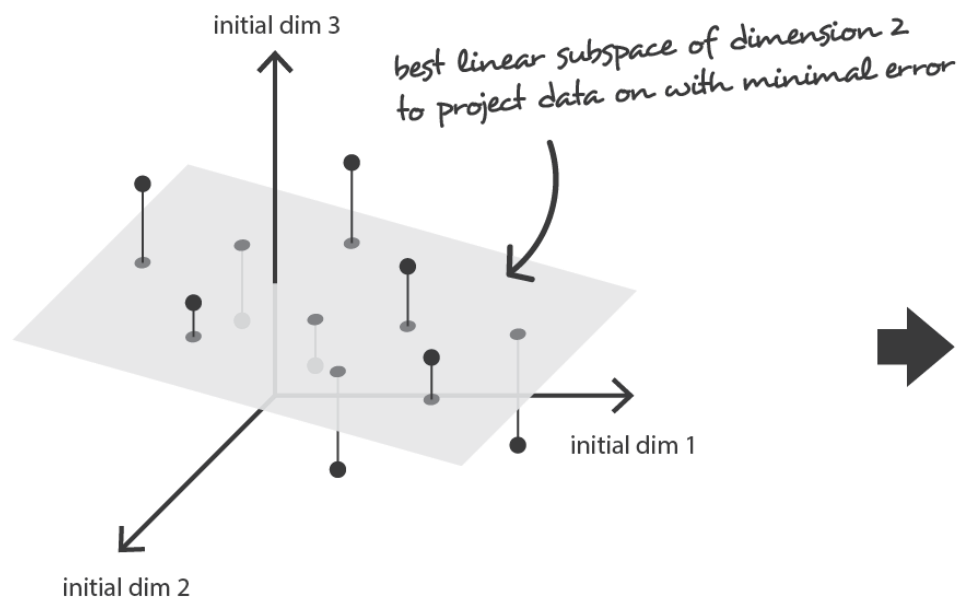
$$e(x) = P^T x \quad d(e(x)) = P P^T x$$

# Autoencoders



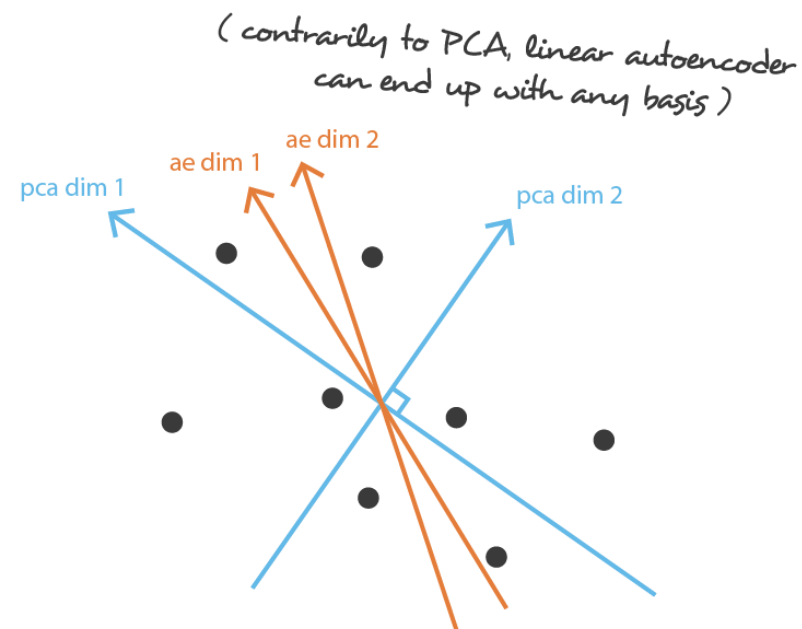
---

$$\text{loss} = ||x - \hat{x}||^2 = ||x - d(z)||^2 = ||x - d(e(x))||^2$$



**Data in the full initial space**

In order to reduce dimensionality, PCA and linear autoencoder target, in theory, the same optimal subspace to project data on...



**Data projected on the best linear subspace**

... but not necessarily with the same basis due to different constraints (in PCA the first component is the one that explains the maximum of variance and components are orthogonal)







# Autoencoder Limitations

- The lack of interpretable and exploitable structures in the latent space (**lack of regularity**)
- most of the time the final purpose of dimensionality reduction is not to only reduce the number of dimensions of the data but to reduce this number of dimensions **while keeping the major part of the data structure information in the reduced representations.**

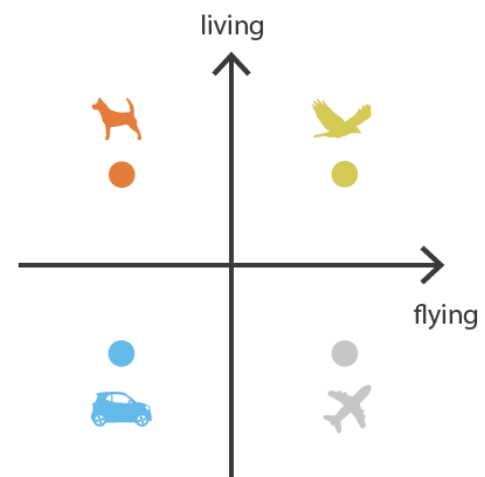


near optimal encoding  
in one dimension  
(too much information lost)



			
dog	bird	car	plane
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>
<div></div>	<div></div>	<div></div>	<div></div>

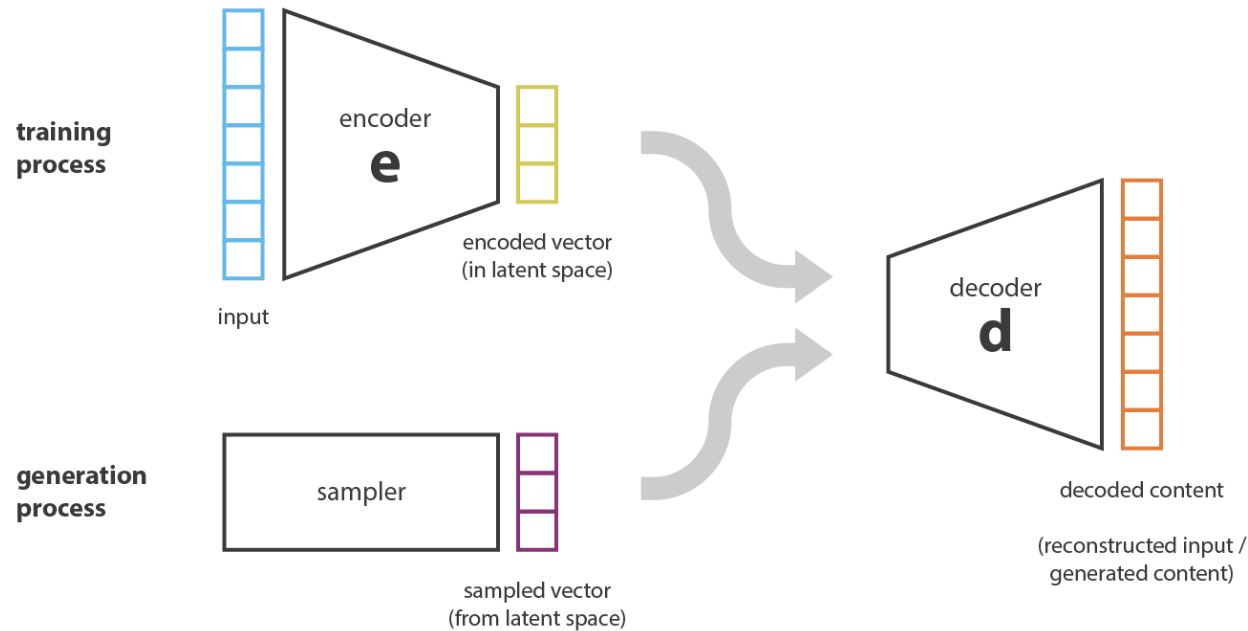
initial data with many features



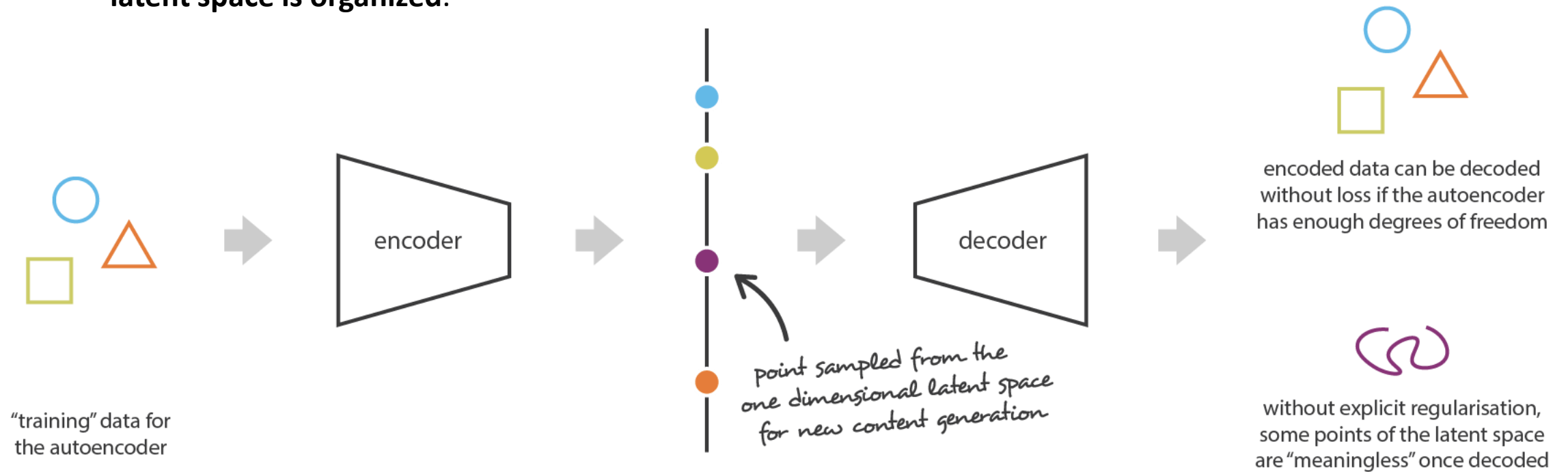
near optimal encoding  
in two dimensions  
(less information lost)

# Variational Autoencoders

- Limitations of autoencoders for content generation



**the autoencoder is solely trained to encode and decode with as few loss as possible, no matter how the latent space is organized.**



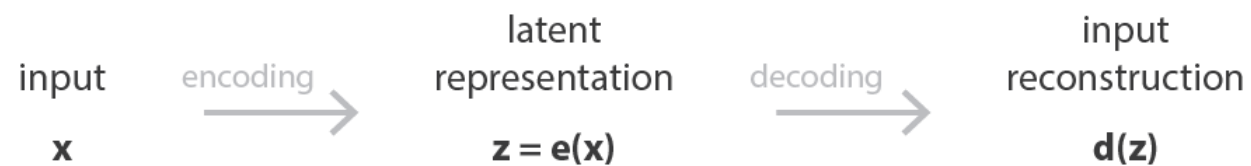
# Definition of variational autoencoders

- a variational autoencoder can be defined as being an autoencoder whose training is regularized to avoid overfitting and ensure that the latent space has good properties that enable generative process
- instead of encoding an input as a single point, we encode it as a distribution over the latent space.

The model is then trained as follows:

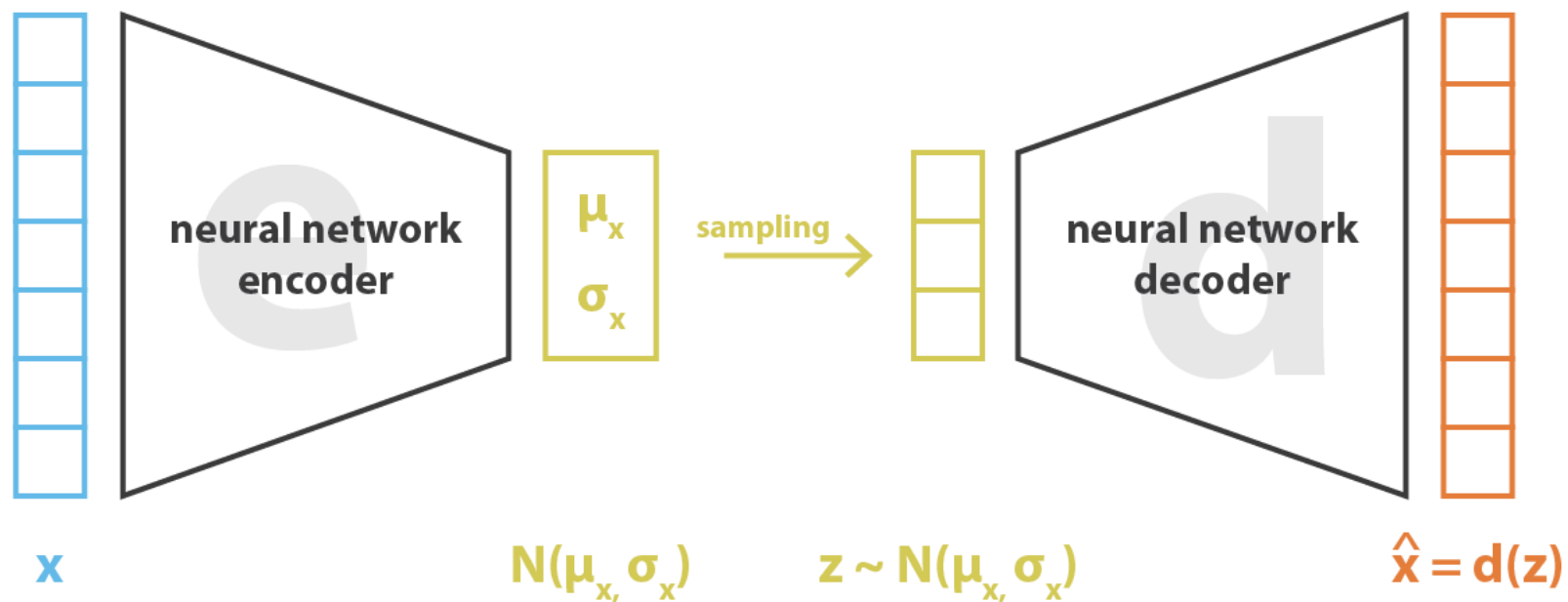
- first, the input is encoded as distribution over the latent space
- second, a point from the latent space is sampled from that distribution
- third, the sampled point is decoded and the reconstruction error can be computed
- finally, the reconstruction error is back propagated through the network

**simple  
autoencoders**



**variational  
autoencoders**





---


$$\text{loss} = ||x - \hat{x}||^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)] = ||x - d(z)||^2 + \text{KL}[N(\mu_x, \sigma_x), N(0, I)]$$

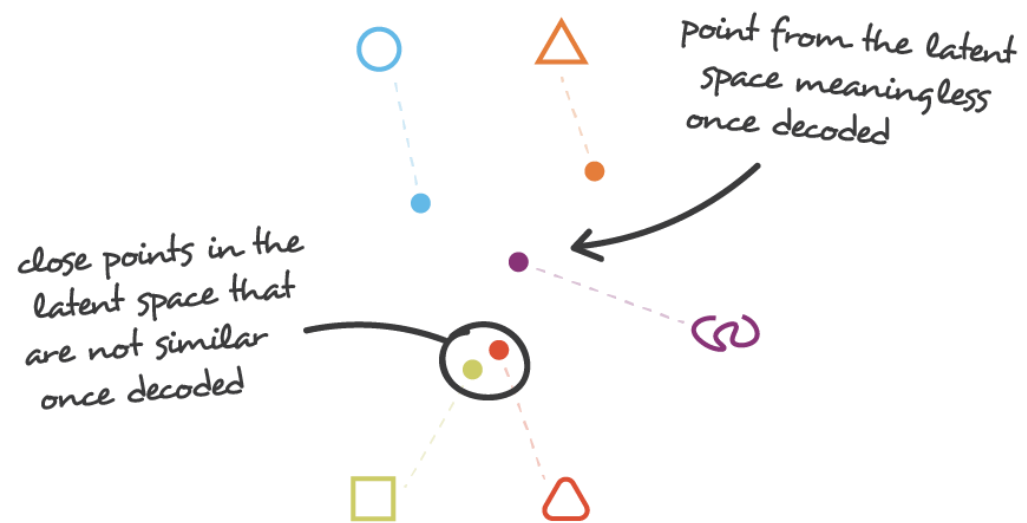
Kullback-Leibler divergence

$$D_{\text{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \left( \frac{p(x)}{q(x)} \right) dx$$

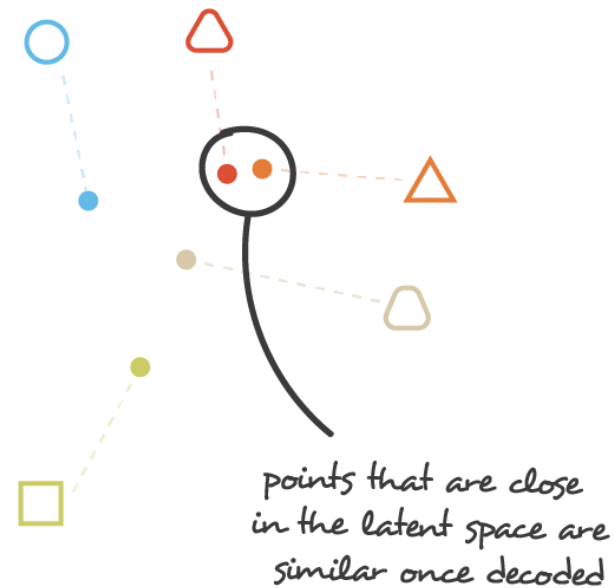


# Intuitions about the regularization

- The regularity that is expected from the latent space in order to make generative process possible can be expressed through two main properties: **continuity** (two close points in the latent space should not give two completely different contents once decoded) and **completeness** (for a chosen distribution, a point sampled from the latent space should give “meaningful” content once decoded).

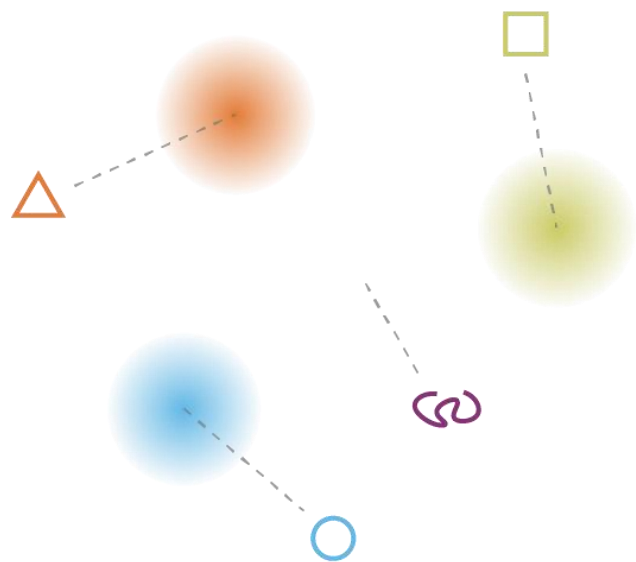


irregular latent space

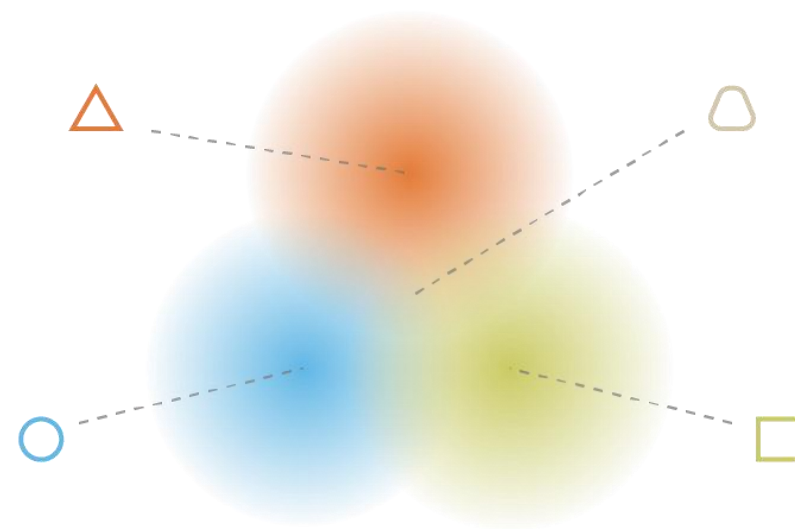


regular latent space

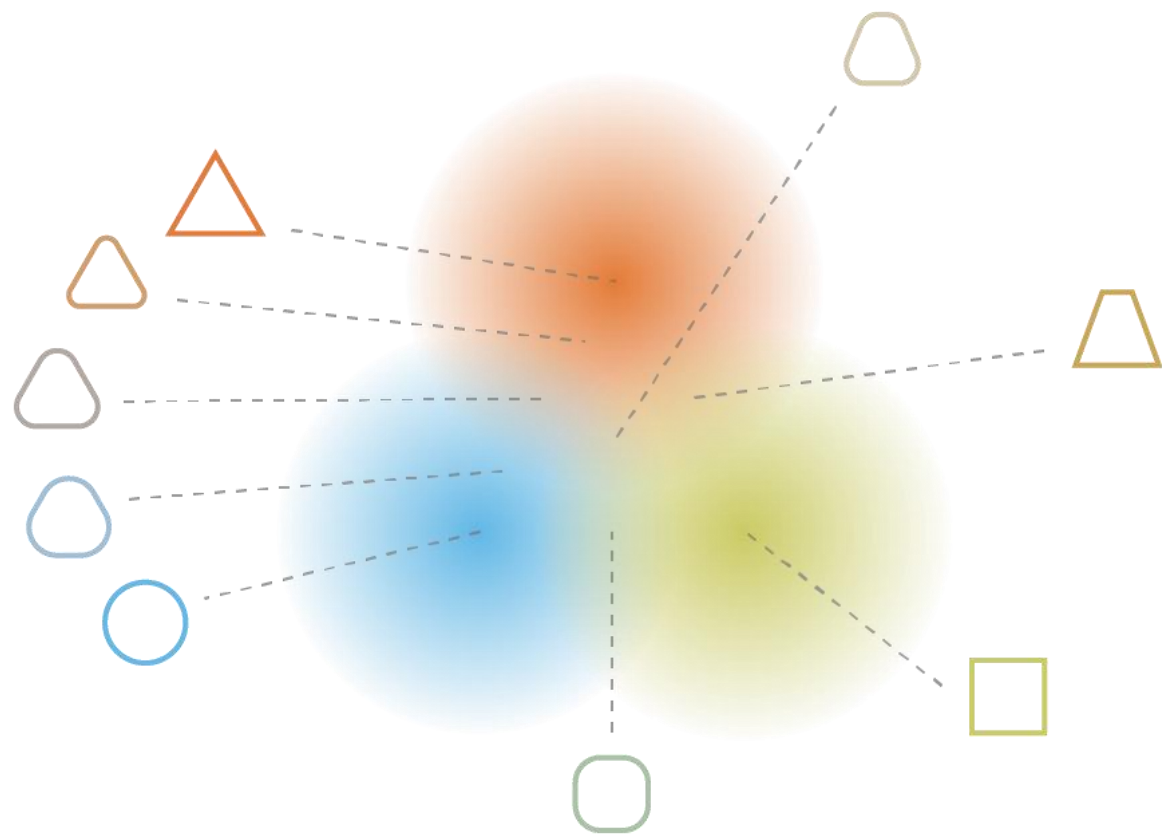




what can happen without regularisation

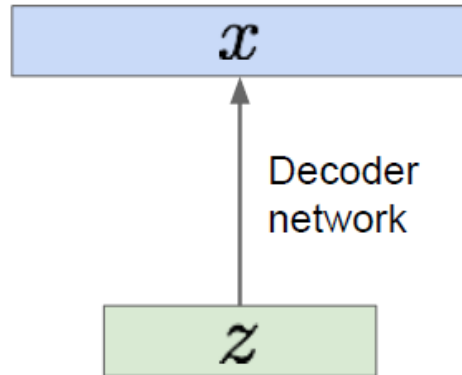


what we want to obtain with regularisation



# How to train the model?

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



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

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

Q: What is the problem with this?

Intractable!

# Intractability

Data likelihood:  $p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz$

Data likelihood:  $p_{\theta}(x) = \int p_{\theta}(z)p_{\theta}(x|z)dz$

↑  
Intractable to compute  
 $p(x|z)$  for every  $z$ !

# Intractability

Posterior density also intractable:  $p_{\theta}(z|x) = p_{\theta}(x|z)p_{\theta}(z)/p_{\theta}(x)$

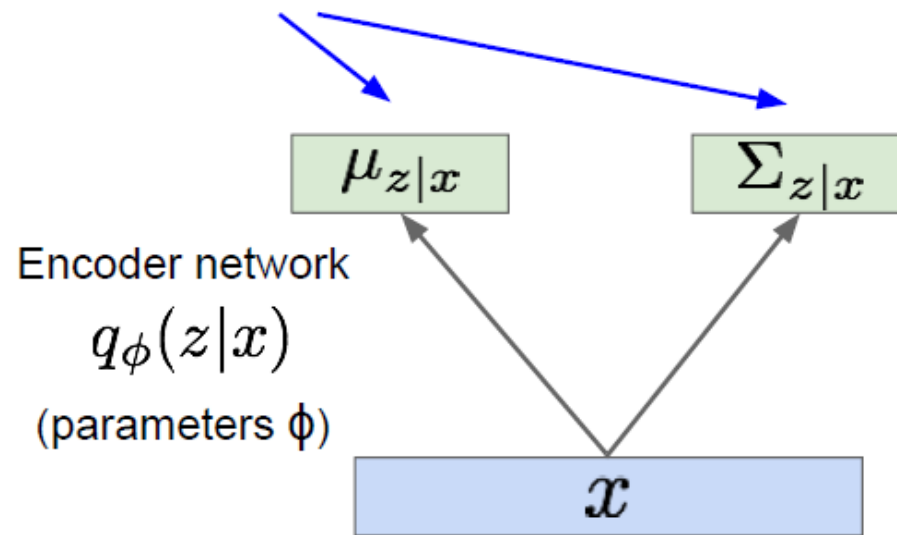
↑  
Intractable data likelihood

Solution: In addition to decoder network modeling  $p_{\theta}(x|z)$ , define additional encoder network  $q_{\phi}(z|x)$  that approximates  $p_{\theta}(z|x)$

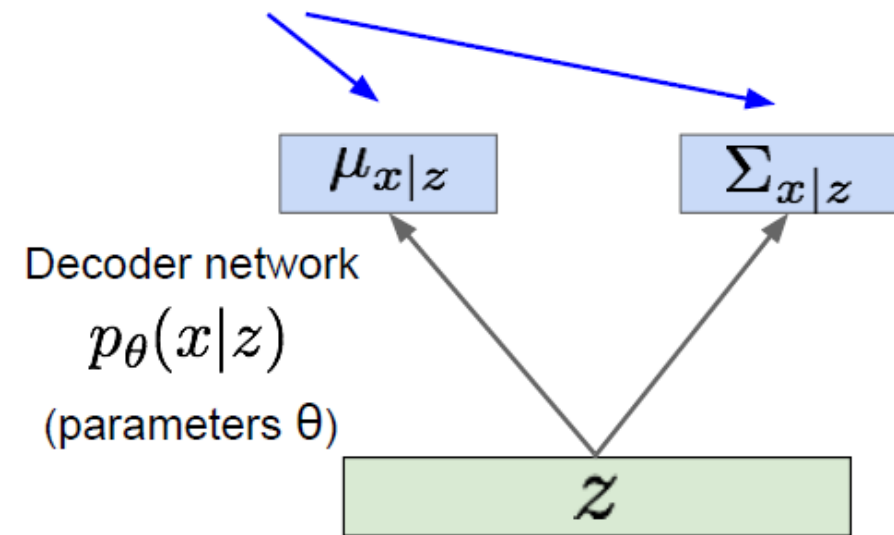
Will see that this allows us to derive a lower bound on the data likelihood that is tractable, which we can optimize

# Variational Autoencoder

Mean and (diagonal) covariance of  $\mathbf{z} | \mathbf{x}$



Mean and (diagonal) covariance of  $\mathbf{x} | \mathbf{z}$





$$\begin{aligned}
\log p_{\theta}(x^{(i)}) &= \mathbf{E}_{z \sim q_{\phi}(z|x^{(i)})} \left[ \log p_{\theta}(x^{(i)}) \right] \\
&= \mathbf{E}_z \left[ \log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z \left[ \log \frac{p_{\theta}(x^{(i)} | z) p_{\theta}(z)}{p_{\theta}(z | x^{(i)})} \frac{q_{\phi}(z | x^{(i)})}{q_{\phi}(z | x^{(i)})} \right] \quad (\text{Multiply by constant}) \\
&= \mathbf{E}_z \left[ \log p_{\theta}(x^{(i)} | z) \right] - \mathbf{E}_z \left[ \log \frac{q_{\phi}(z | x^{(i)})}{p_{\theta}(z)} \right] + \mathbf{E}_z \left[ \log \frac{q_{\phi}(z | x^{(i)})}{p_{\theta}(z | x^{(i)})} \right] \quad (\text{Logarithms}) \\
&= \mathbf{E}_z \left[ \log p_{\theta}(x^{(i)} | z) \right] - D_{KL}(q_{\phi}(z | x^{(i)}) || p_{\theta}(z)) + D_{KL}(q_{\phi}(z | x^{(i)}) || p_{\theta}(z | x^{(i)}))
\end{aligned}$$

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

↑  
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  $\geq 0$ .

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

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

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

Variational lower bound ("ELBO")

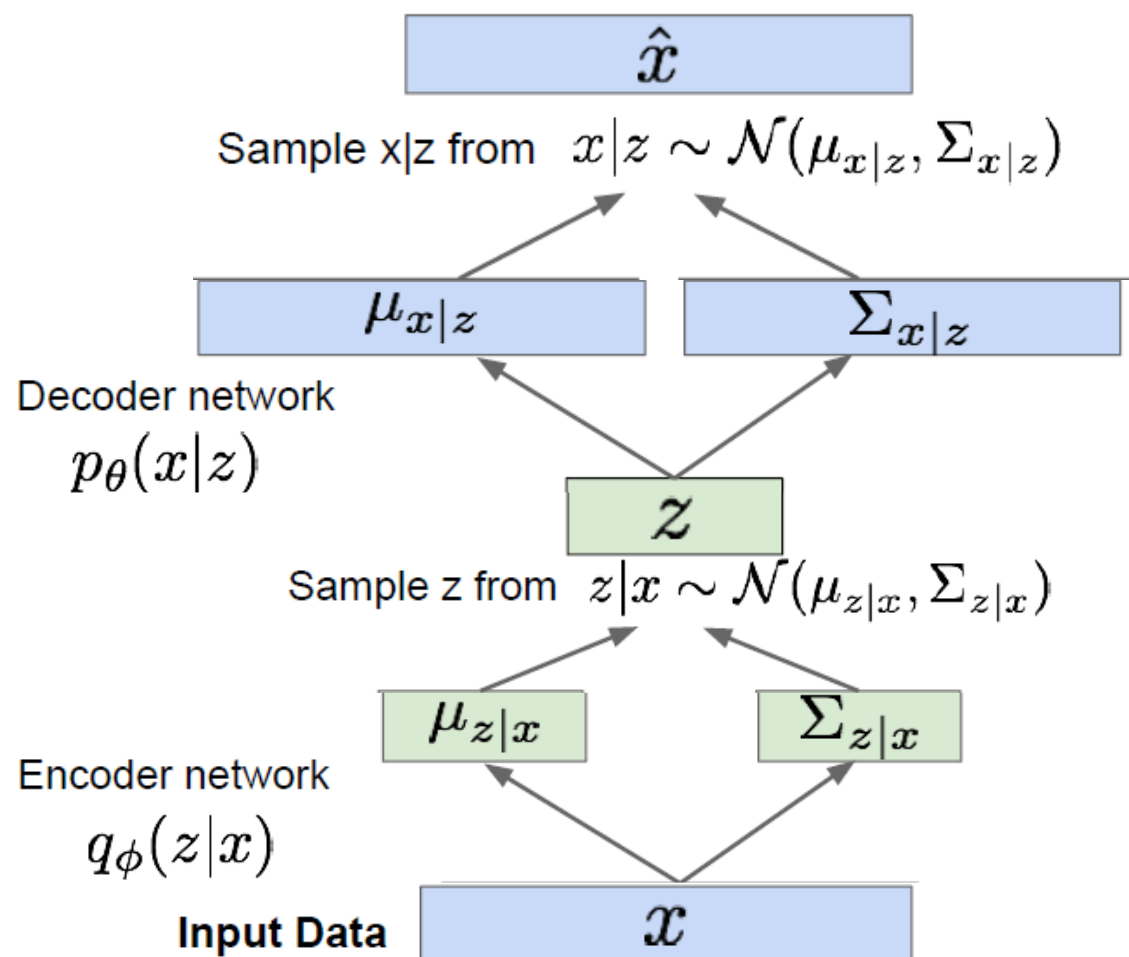
$$\theta^*, \phi^* = \arg \max_{\theta, \phi} \sum_{i=1}^N \mathcal{L}(x^{(i)}, \theta, \phi)$$

Training: Maximize lower bound

Putting it all together: maximizing the likelihood lower bound

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

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



# Objective Function

$$\tilde{L}(\theta, \phi, \mathbf{x}^{(i)}) = -D_{KL}[q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\theta}(\mathbf{z})] + \frac{1}{L} \sum_{l=1}^L (\log p_{\theta}(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)}))$$

$$q_{\phi}(\mathbf{z}|\mathbf{x}^{(i)}) = \mathcal{N}(\mathbf{z}, \mu^{(i)}, \sigma^{2(i)} \mathbf{I})$$

$$p_{\theta}(\mathbf{z}) = \mathcal{N}(\mathbf{z}, 0, \mathbf{I})$$



$$\tilde{L}(\theta, \phi, \mathbf{x}^{(i)}) = \frac{1}{2} \sum_{j=1}^J (1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2) + \frac{1}{L} \sum_{l=1}^L (\log p_{\theta}(\mathbf{x}^{(i)} | \mathbf{z}^{(i,l)}))$$

Regularization

Reconstruction  
Error

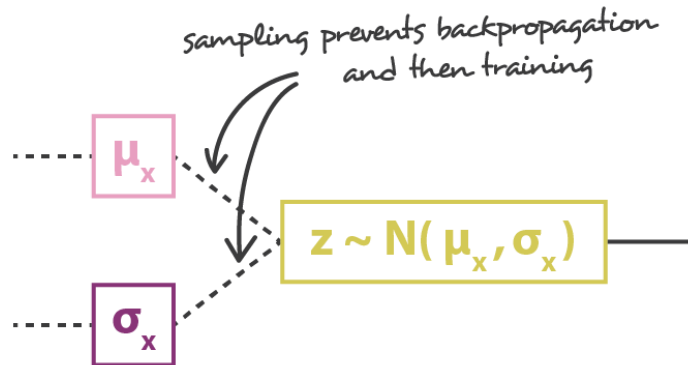
# reparametrization trick

$$z = h(x)\zeta + g(x)$$

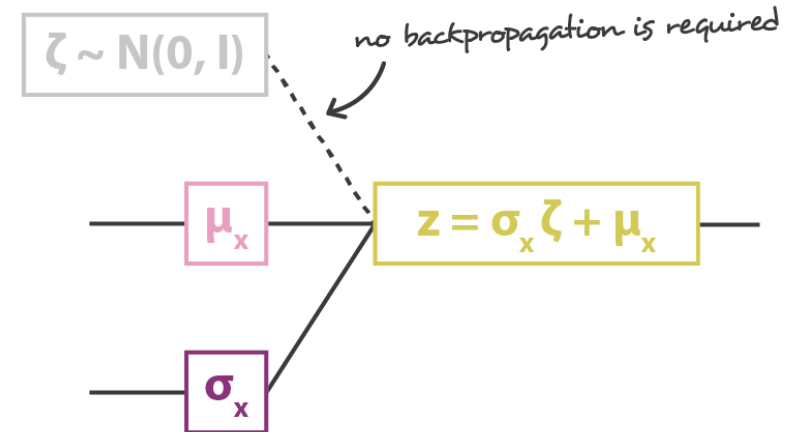
$$\zeta \sim \mathcal{N}(0, I)$$

—— no problem for backpropagation

..... backpropagation is not possible due to sampling



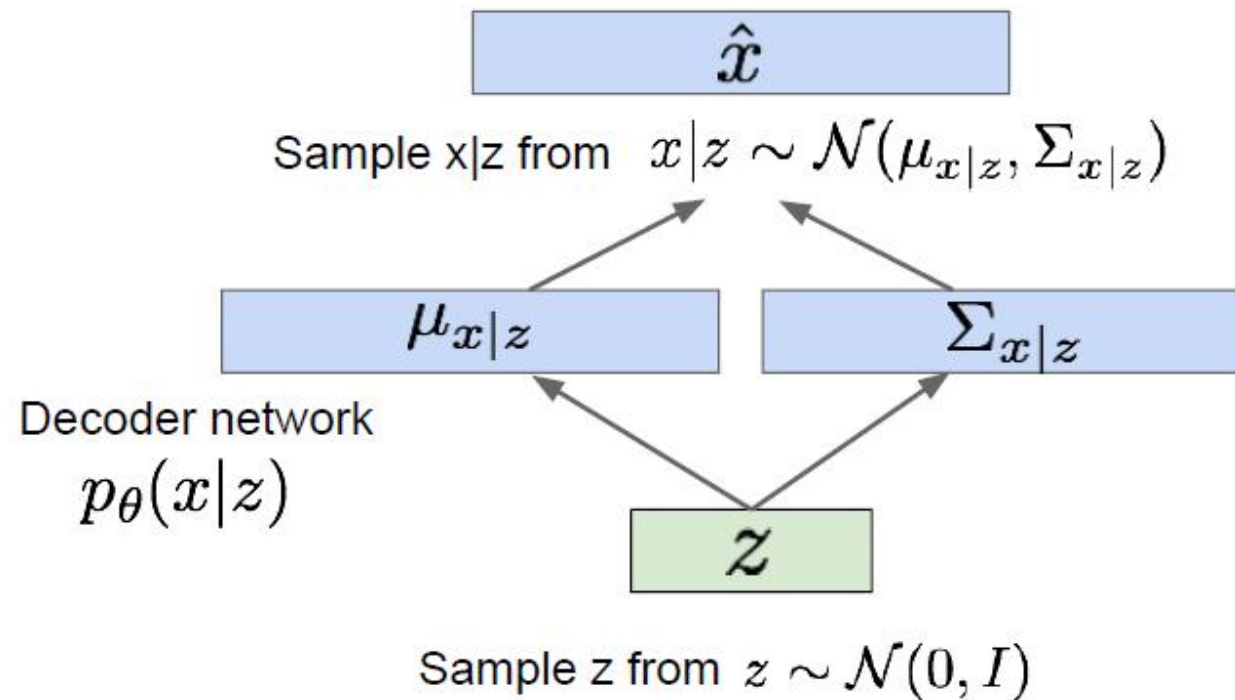
sampling without reparametrisation trick



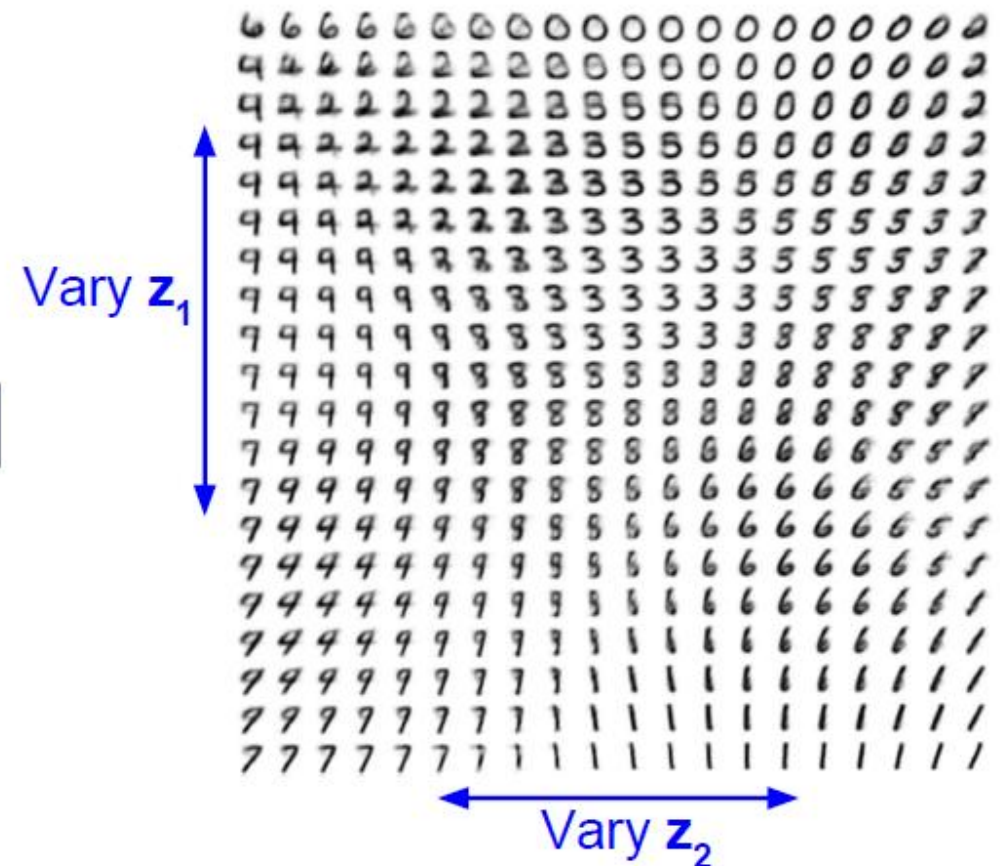
sampling with reparametrisation trick

# Generating data

Use decoder network. Now sample  $z$  from prior!



Data manifold for 2-d  $z$



# Generating data

Diagonal prior on  $\mathbf{z}$   
=> independent  
latent variables

Different  
dimensions of  $\mathbf{z}$   
encode  
interpretable factors  
of variation

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

Degree of smile

Vary  $\mathbf{z}_1$



Vary  $\mathbf{z}_2$

Head pose



# Generating data



32x32 CIFAR-10



Labeled Faces in the Wild



# Variational Autoencoder

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:

- 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

# References

<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>

<https://www.slideshare.net/ckmarkohchang/variational-autoencoder>

Fei-Fei Li & Justin Johnson & Serena Yeung Convolutional Neural Networks for Visual Recognition Lecture 12, Generative models, May 15, 2018