Autoencoders and Deep Representation Learning

Center for Advanced Studies in Adaptive Systems

by Alireza Ghods

Outline

- Quick review
- Background
- Autoencoder
- Autoencoder + regularizer
- Variational Autoencoder
- Questions

Review: Feature Engineering vs. Representation Learning

Feature engineering

Edges?
Texture?

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ \vdots \\ X_n \end{bmatrix}$$

Machine Learning

"Dog in tree"

Review: Feature engineering vs. representation learning

Dumb feature extraction

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{pmatrix}$$

Representation Learning

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_m \end{pmatrix}$$

Machine Learning

"Dog in tree"

Review: Typical Neural Net Characteristics

So far, Deep Learning Models have things in common:

- Input Layer: (maybe vectorized), quantitative representation
- Hidden Layer(s): Apply transformation with nonlinearity
- Output Layer: Result for classification, regression, translation, segmentation, etc.
- Models used for ***supervised learning***

Changing the Objective

Today's lecture: ***unsupervised learning*** with neural network.

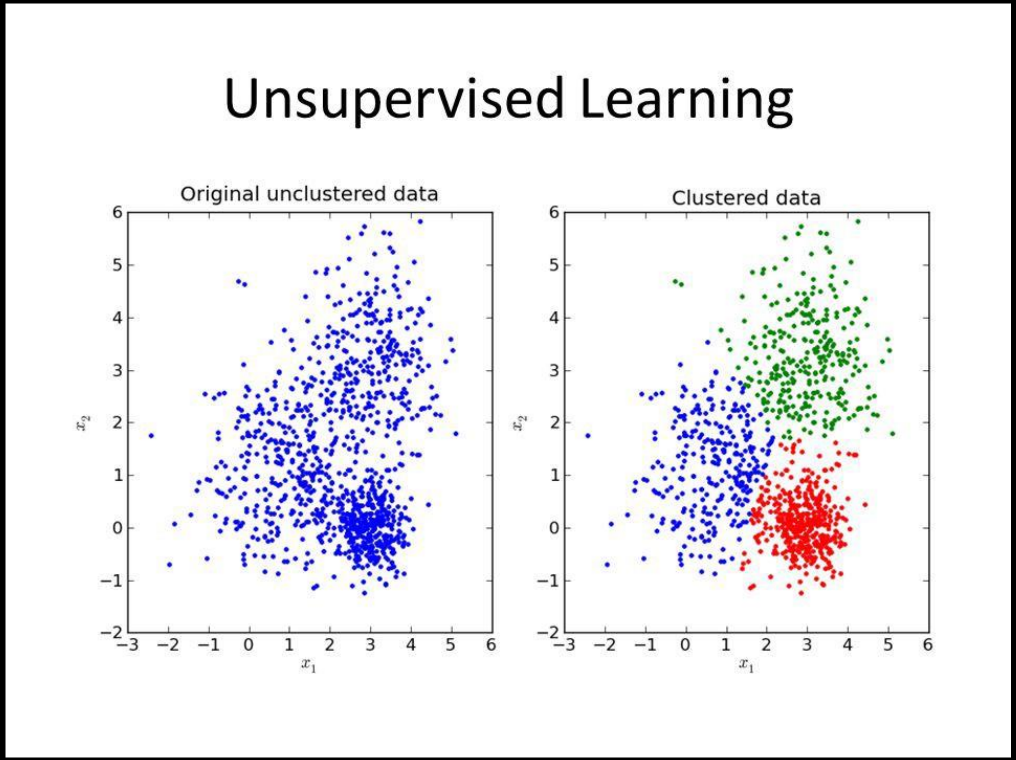


Image source: https://medium.com/the-21st-century/machine-learning-a-strategy-to-learn-and-understand-chapter-3-9daaad4afc55

Properties of Representations

- Smoothness
  - Small change to input → small change in representation
- Compactness … or overcompleteness
  - Compress input / remove redundancy
  - Multiple possible representations, chosen via other considerations (like sparsity)
- Sparsity
  - For any given input, most features are zero
- Disentangle factors of variation
  - Feature tend to vary independently of each other
- Abstraction / Invariance
  - Recognizing high-level properties shared between superficially dissimilar inputs

Credit: Galen Andrew

Relationship between Principal Component Analysis and Autoencoder

- PCA takes a N-dimensional data and represented using a much lower dimensional code.
- This happens when data lies near a linear manifold in the high dimensional space.
  - Manifold → is a topological space that all along that space there is very high probability of our data belong to that area.

Visualizing Manifolds

Extract 2D manifold of data which exist in 3D:

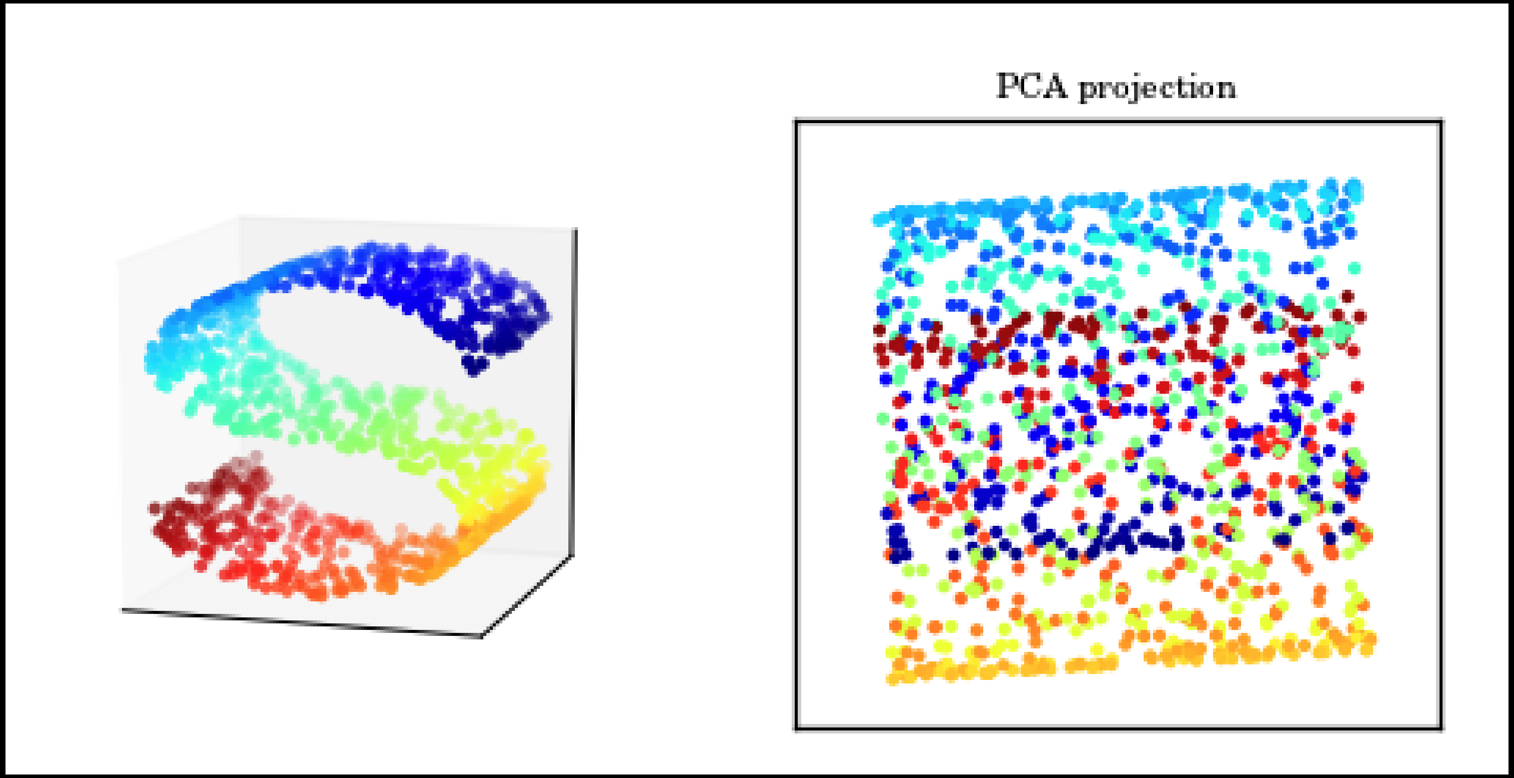


Image source: http://www.astroml.org/book_figures/chapter7/fig_S_manifold_PCA.html

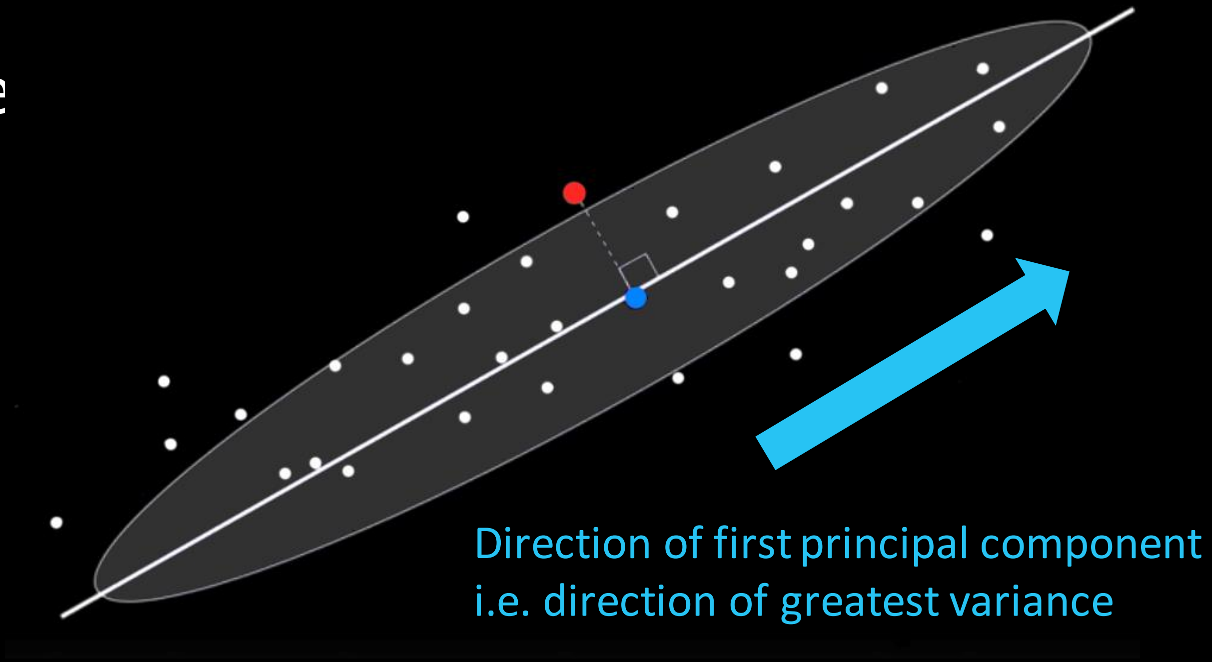
Relationship between Principal Component Analysis and Autoencoder

- PCA takes a N-dimensional data and represented using a much lower dimensional code.
- This happens when data lies near a linear manifold in the high dimensional space.
  - Manifold → is a topological space that all along that space there is very high probability of our data belong to that area.
- You can do this efficiently by using PCA or you can do it ineffecently by using linear Autoencoder.

A picture of PCA with N=2 and M=1

- This takes 2-dimensional data and finds the 1 orthogonal directions in which the data have the most variance.

- The red point is represented by the of the red point has an error equal and blue point.



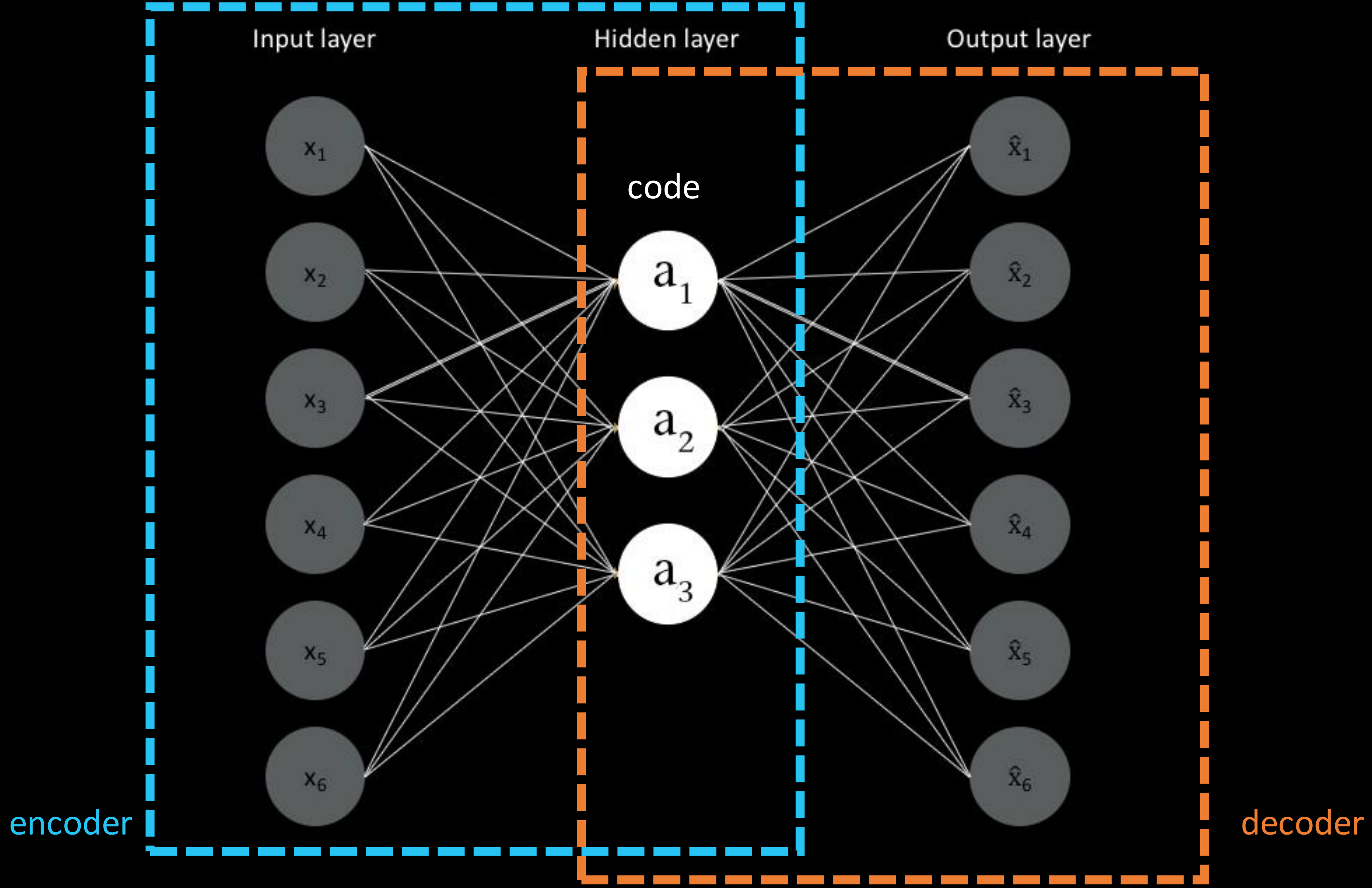
Credit: Geoffrey Hinton

An inefficient Autoencoder to implement PCA

Autoencoders are neural networks that are trained to copy their inputs to their outputs. Therefore, the objective function is $\min |x - \hat{x}|$.

- If hidden and output layers are linear it will learn hidden units that are a linear function of the data and minimize the squerd reconstruction error.
  - This is exactly what PCA does.

Visualization of a simple Autoencoder



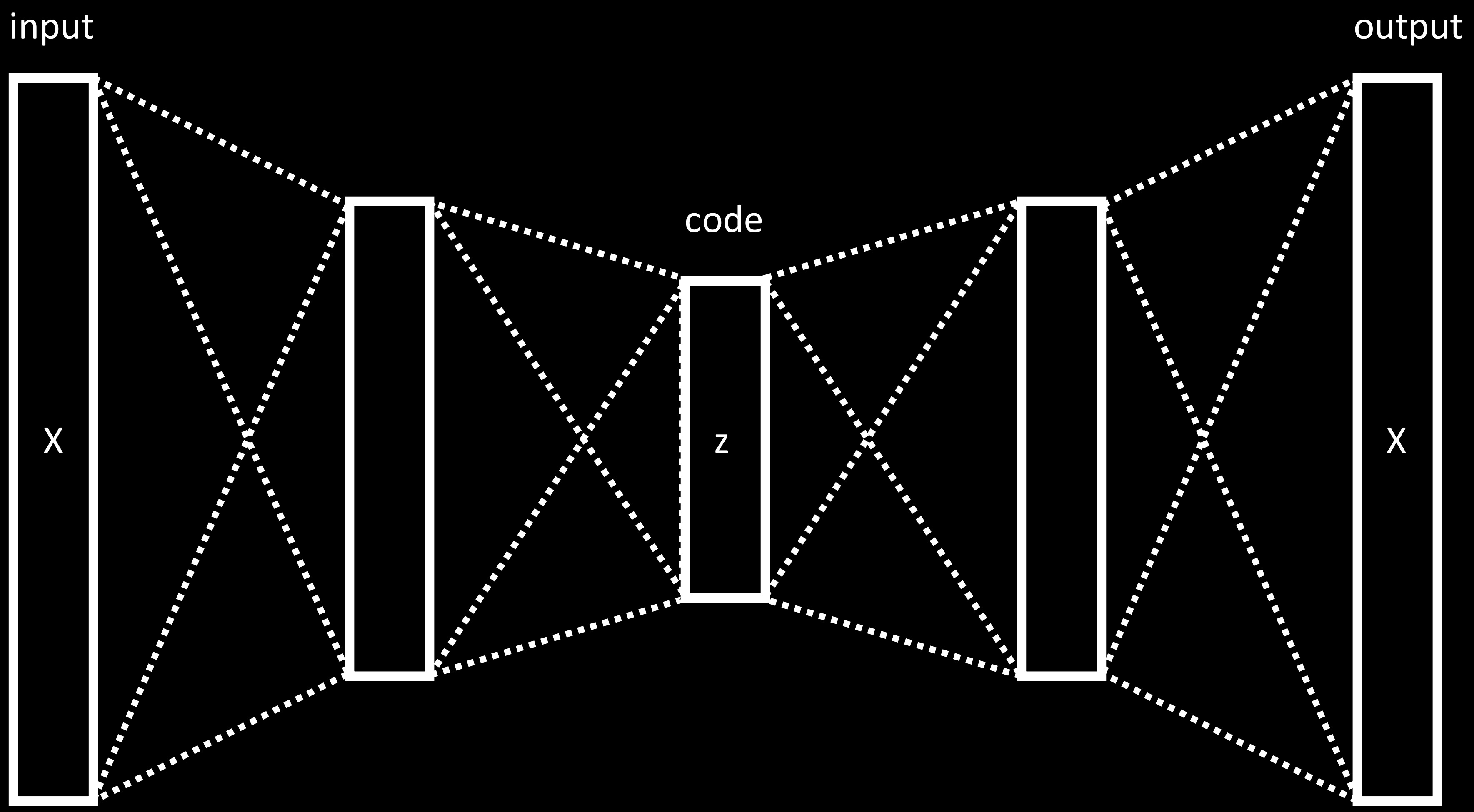
Code Demo of a simple Autoencoder

Let’s implement a simple linear autoencoder and compare it to PCA.

Deep Autoencoder

We learn how to implement a linear auto encoder, let’s try to make it nonlinear by adding activation function and more hidden layers.

Visualizing Deep Autoencoder



Demo Deep Autoencoder

Let’s look at the implimentation.

Caveats and Danger

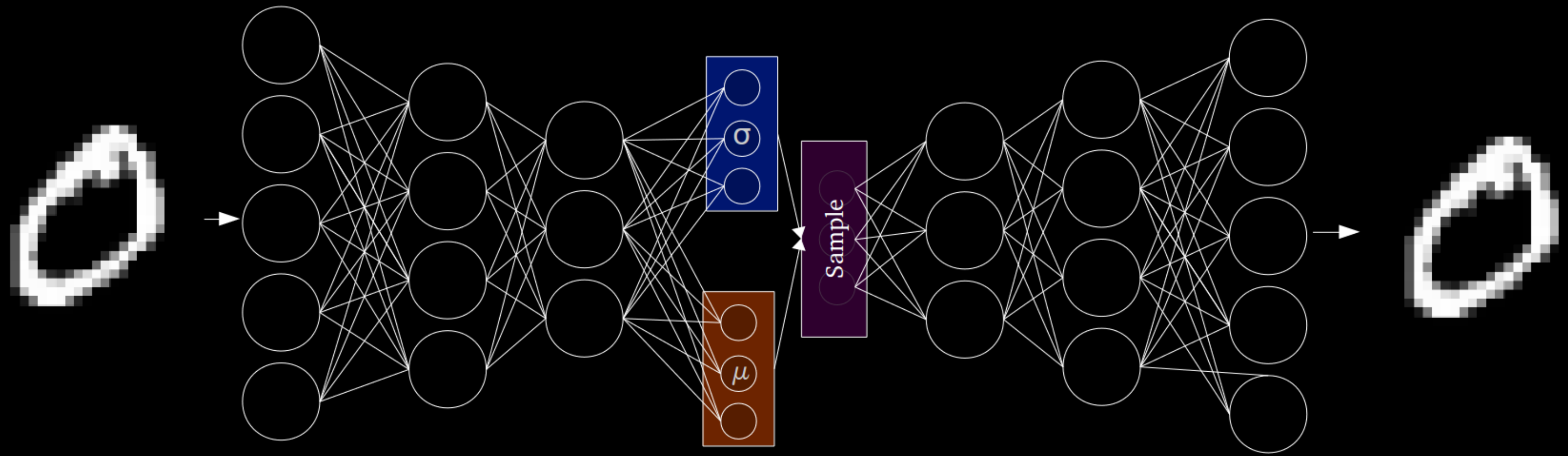
Unless careful, autoencoders will not learn meaningful representations.

- Reconstruction loss: ***indifferent to latent space (code)*** characteristic. (not true for PCA)
- Higher representational power gives flexibility for suboptimal encoding.
- Pathological case: hidden layer is only one dimension, learns index mappings: $\boldsymbol{x}^{(i)} \rightarrow i \rightarrow \boldsymbol{x}^{(i)}$
  - Not very realistic, but completely possible.

Variational Autoencoder

Idea: Latent space explicitly encodes distribution! Typically made to encode unit Gaussian.

- Flow: Input → encode to statistic vectors → sample a latent vector → decode for reconstruction
- Loss: Reconstruction + KL Divergence



Autoencoder vs. Varaiational Autoencoder

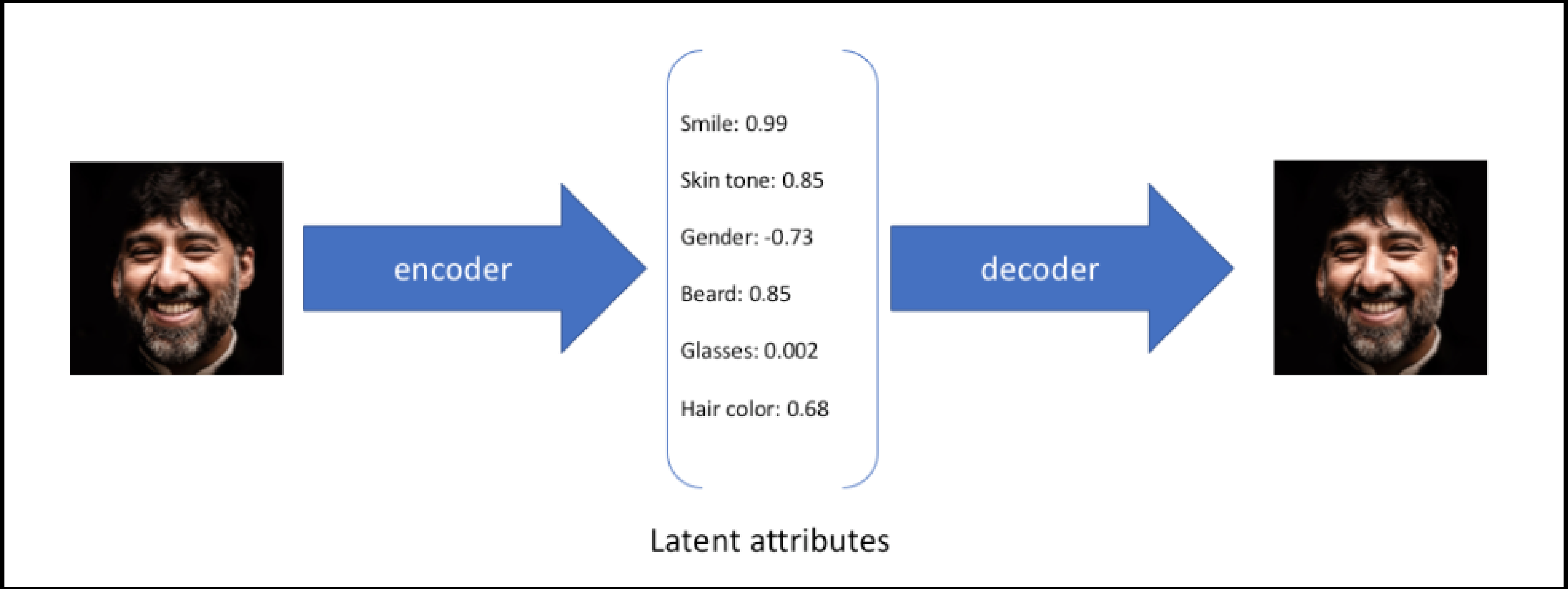


Image sources: https://www.jeremyjordan.me/variational-autoencoders/

Autoencoder vs. Varaiational Autoencoder

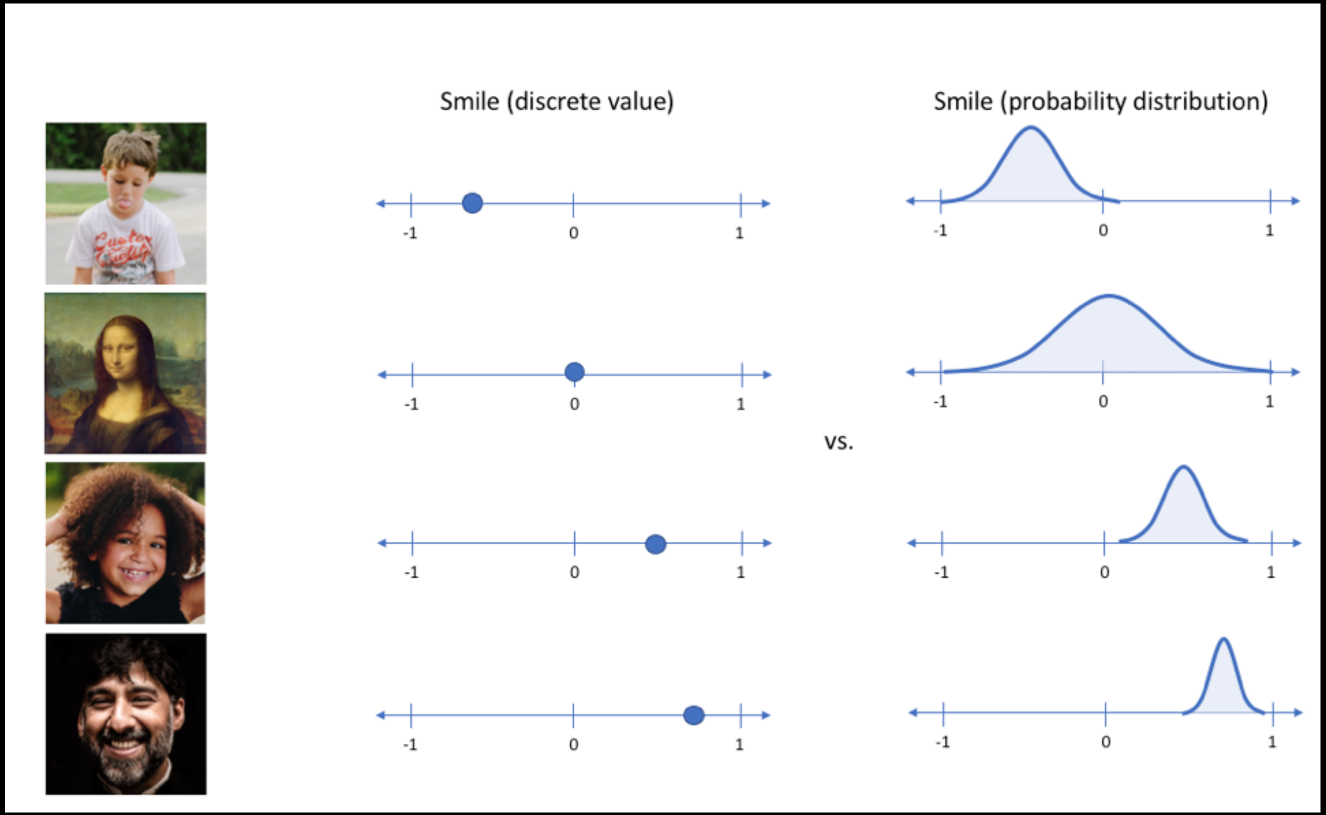


Image sources: https://www.jeremyjordan.me/variational-autoencoders/

Autoencoder vs. Varaiational Autoencoder

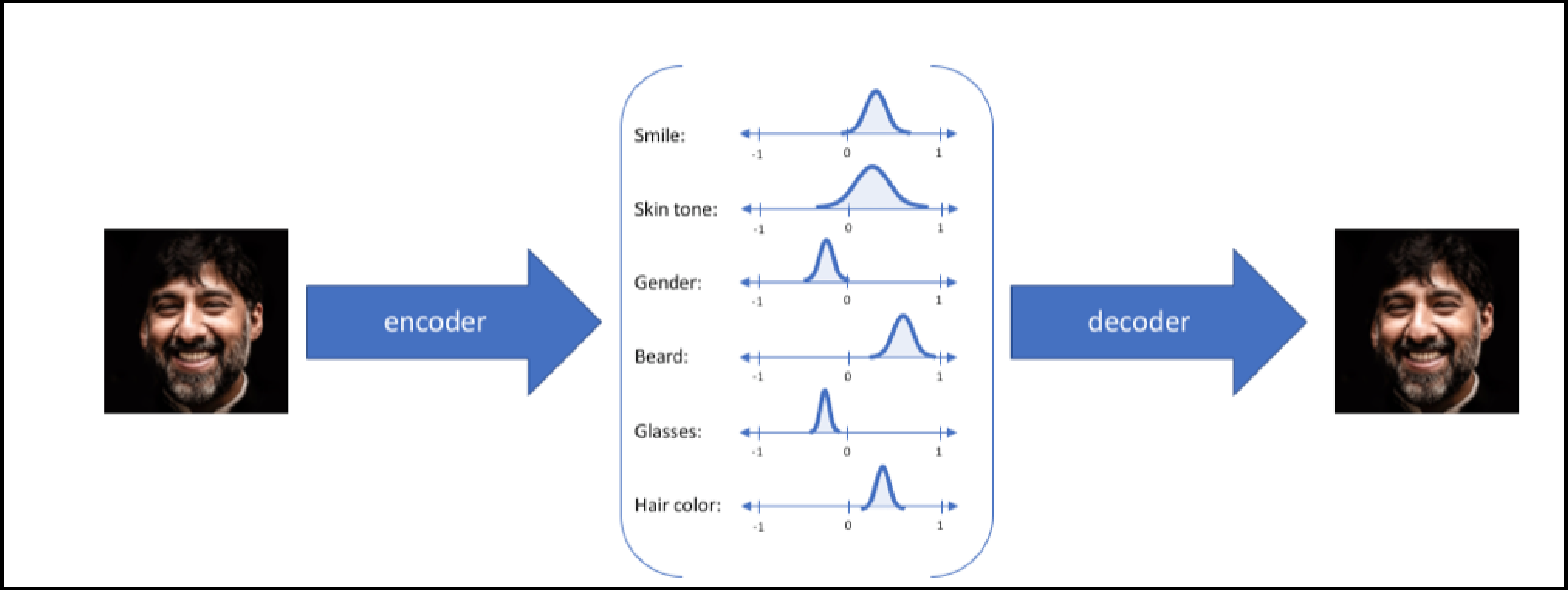


Image sources: https://www.jeremyjordan.me/variational-autoencoders/

Autoencoder vs. Varaiational Autoencoder

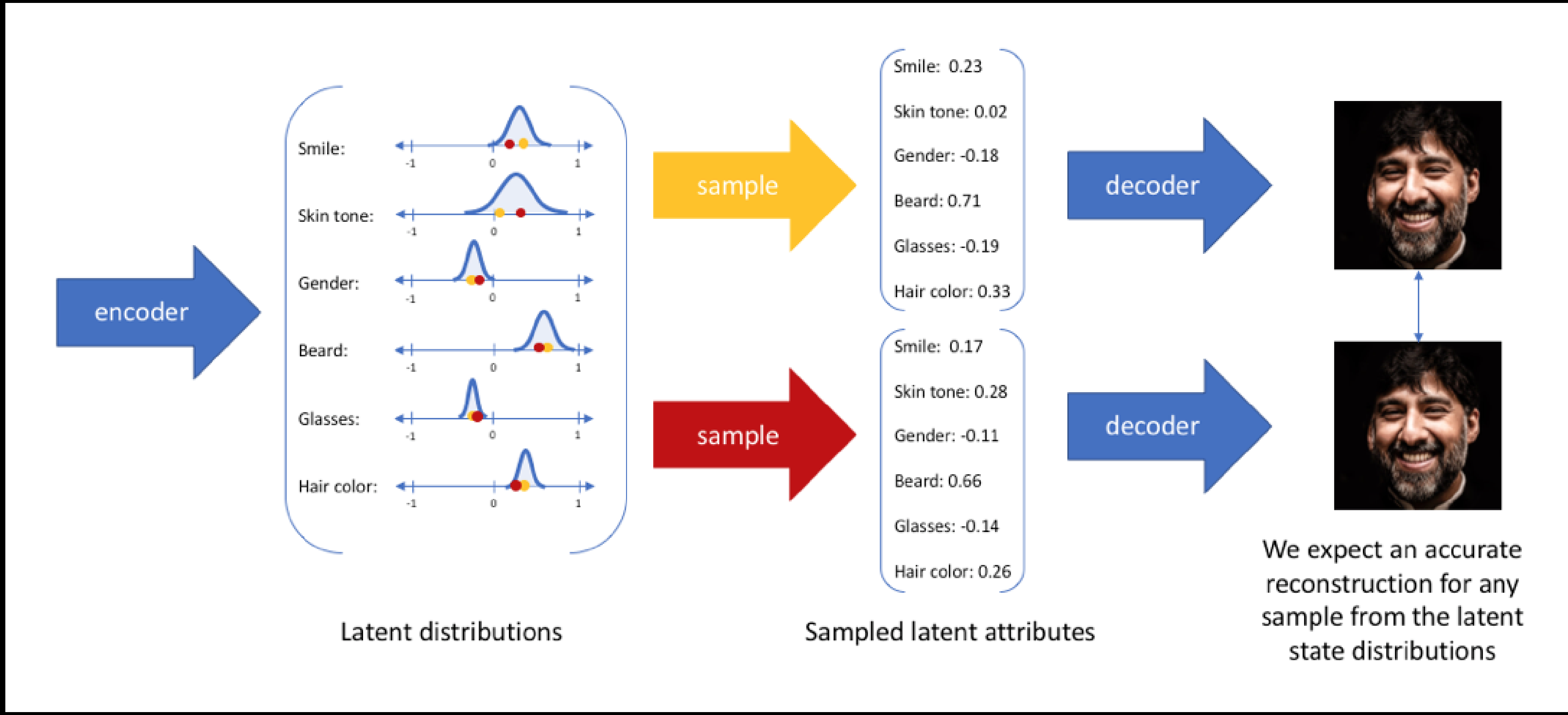


Image sources: https://www.jeremyjordan.me/variational-autoencoders/

KL Divergence

- Information Gain

$$I = -\log p(x), \qquad x \in event$$

- Entropy

$$H = -\sum P(x) \log P(x)$$

- KL Divergence
  - let’s have two distribution p and q, KL divergence calculates dissimilarity between these two distribution

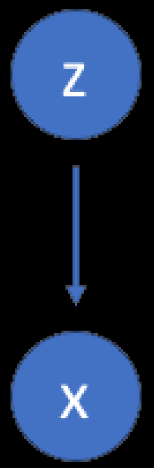
$$KL(p||q) = -\sum p(x) \log q(x) + \sum p(x) \log p(x)$$

$$KL > 0 \qquad KL(p||q) \neq KL(q||p)$$

Variational Inference

Problem definition

- Observation Data: $x = \{x_1, x_2, \ldots, x_n\}$
- Hidden Variable: $z = \{z_1, z_2, \ldots, z_n\}$



$$p(z|x) = \frac{p(z,x)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|z)p(z)dz}$$

Unfortunately, computing p(x) is quite difficult.

Variational Inference

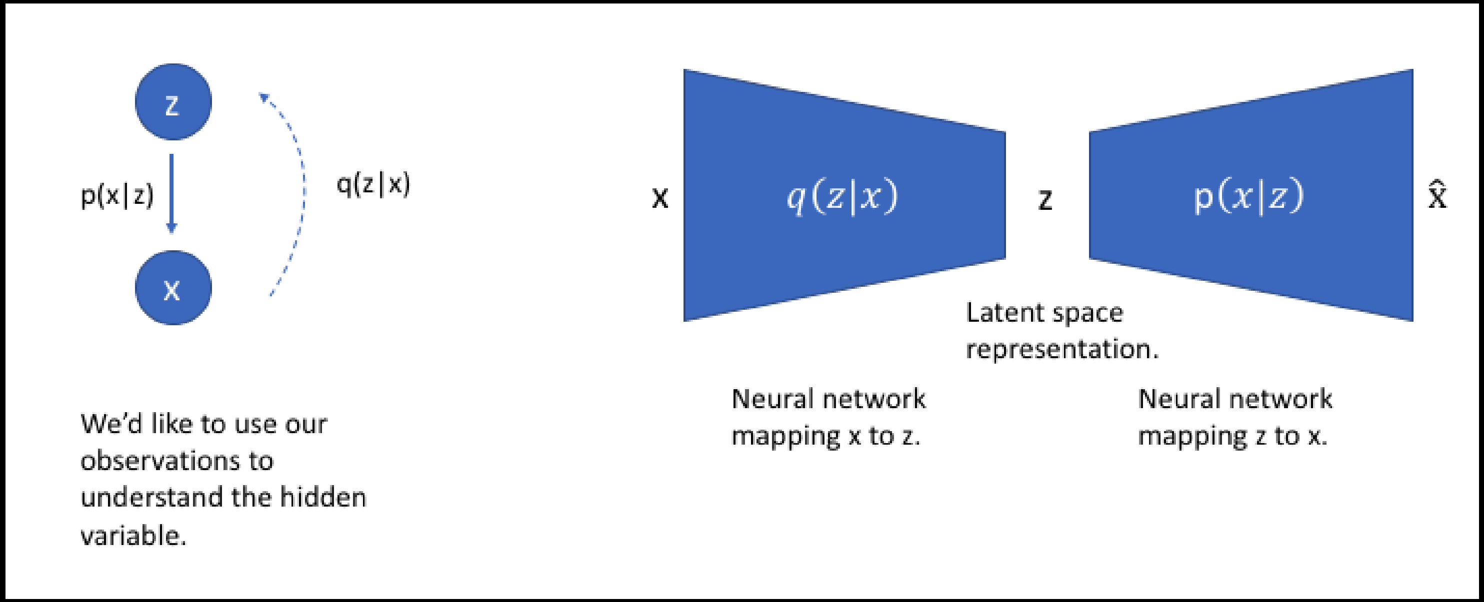
- To solve the problem, we are going to apply variational inference.
  - Let's approximate p(z|x) by another distribution q(z|x) which we'll define such that it has a tractable.
- Our goal is to make q(z|x) similar to p(z|x)
  - *min KL(q(z|x)||p(z|x))*
  - We can minimize the above expression by maximizing the following:

$$E_{q(z|x)} \log p(x|z) - KL\ (q(z|x)\ ||\ p(z))$$

    - The first term represents the reconstruction likelihood.
    - The second term ensures that our learned distribution q is similar to the true prior distribution p.

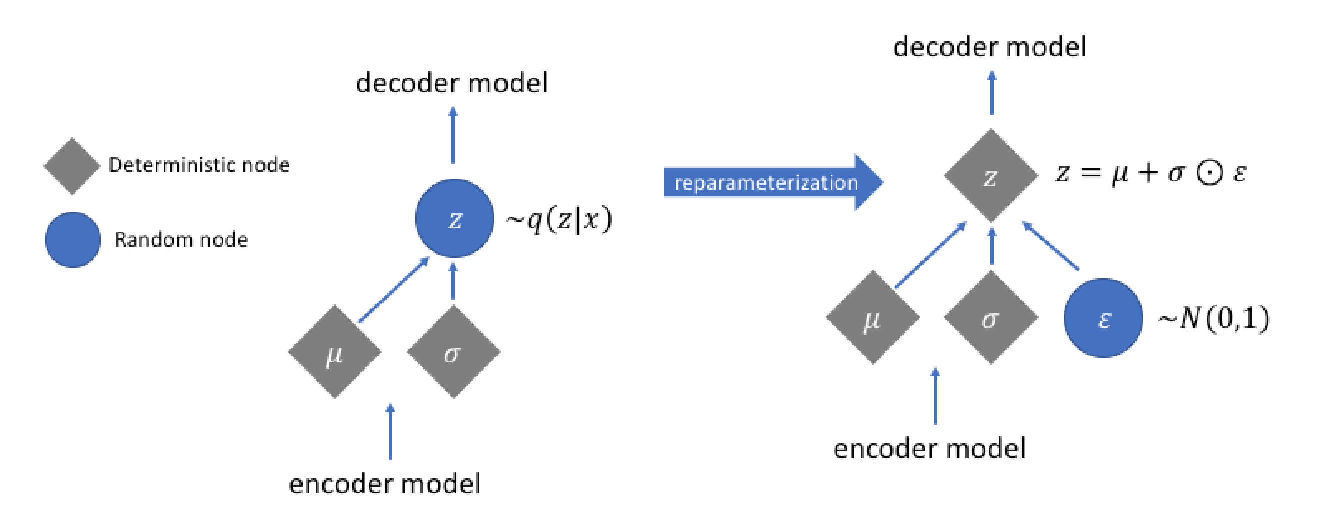
Variational Autoencoder

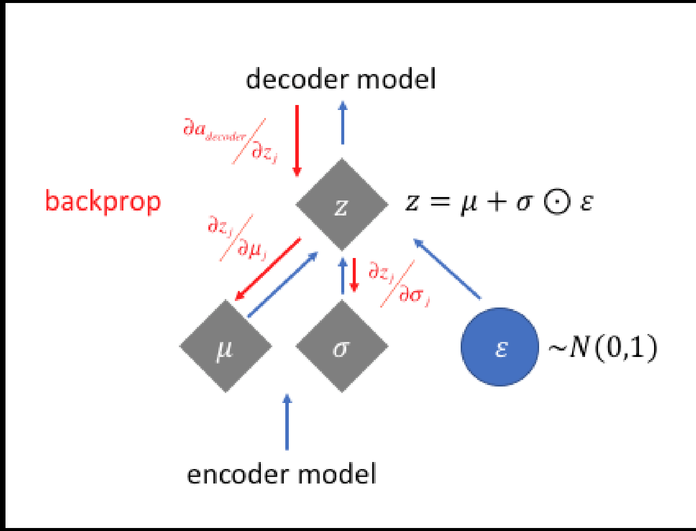
Let's revisit our network



Loss = $J(\theta) = L(x, \hat{x}) + \sum_j KL\left(q_j(z|x) \,||\, p(z)\right)$

Representation Trick





Code Demo

Let’s implement a variational autoencoder.

Questions