Generative AI and LLM

Autoencoders, Variational autoencoders
CS5202

Course Instructor : Dr. Nidhi Goyal

Mahindra University
Global Thinkers. Engaged Leaders.

Lecture Plan

- Entropy
- Cross Entropy
- Autoencoders
- Variational Autoencoders
- Math of VAE

Entropy

- According to Shannon, **Entropy is the minimum no of useful bits required to transfer information from a sender to a receiver.**

Entropy (expressed in ‘bits’) is a measure of how unpredictable the probability distribution is. So more the individual events vary, the more is its entropy.

$$\boxed{Entropy : H(p) = -\sum_{n=1}^{n} p_i \times log(p_i)}$$

Cross Entropy

$$\boxed{\text{Cross Entropy} \;\; : H(p,q) = -\sum_{n=1}^{n} p_i \times log(q_i)}$$

- **Cross entropy is the average message length that is used to transmit the message.**

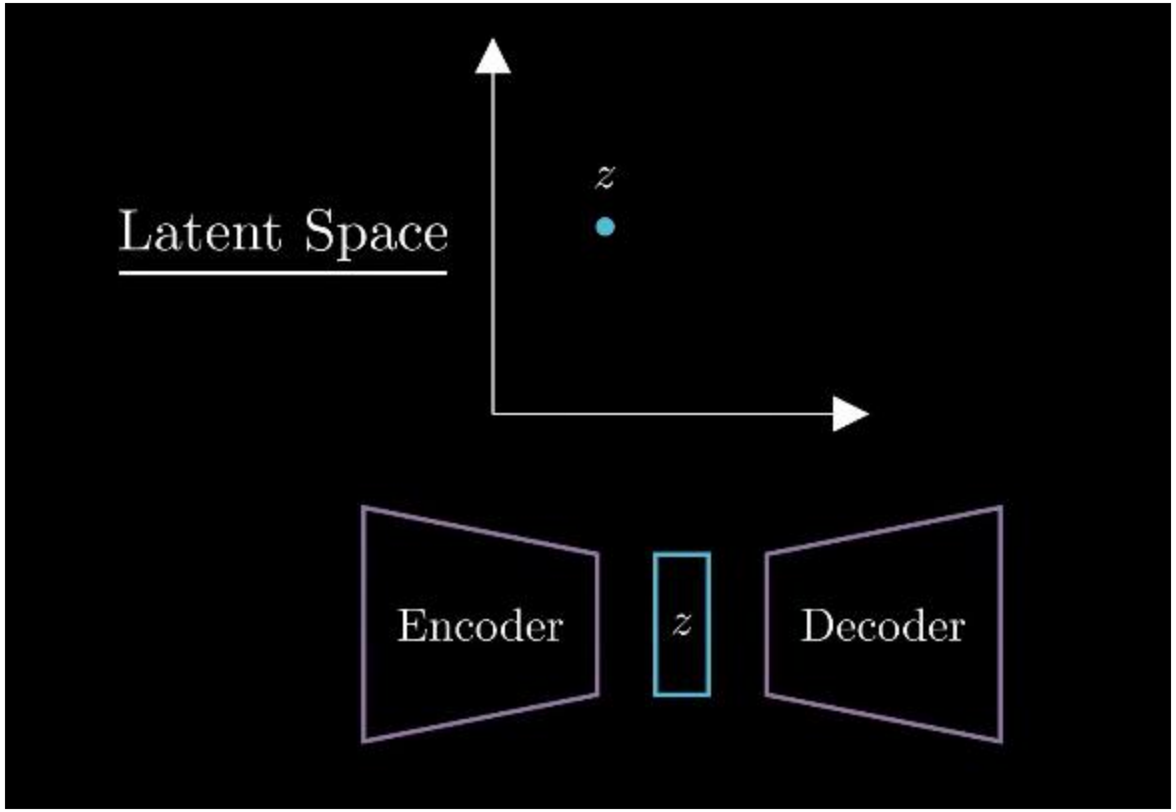
Measuring “Closeness”: KL Divergence

- **The amount by which the cross-entropy exceeds the entropy is called Relative Entropy or commonly known as Kullback-Leibler Divergence or KL Divergence.**

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{\mathcal{X}} p(x)\, \log\left(\frac{p(x)}{q(x)}\right)\, dx$$

- Used to quantify the difference between one probability distribution from a reference probability distribution

Autoencoders

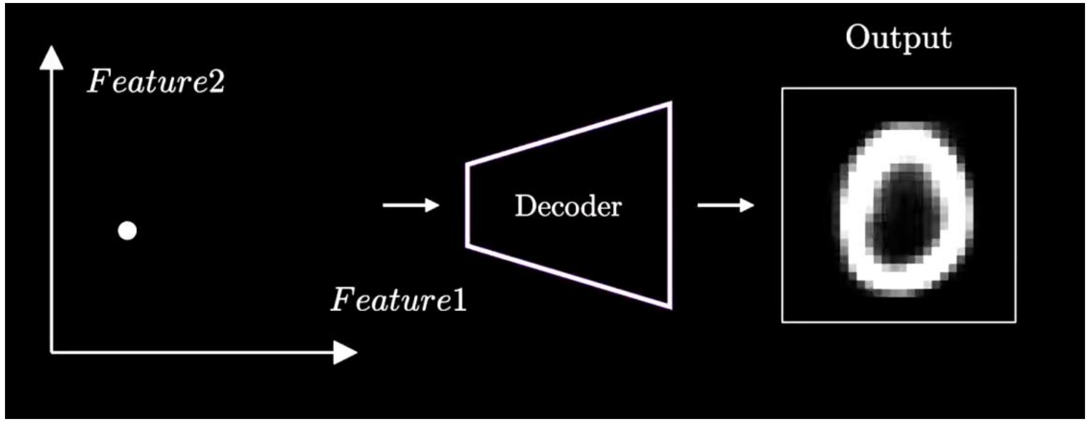


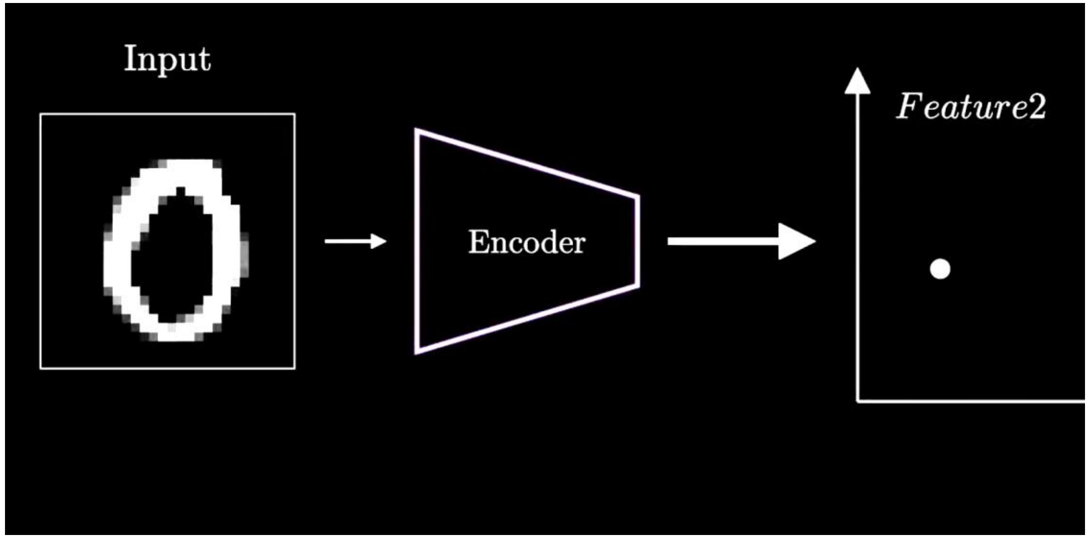
Autoencoders

- **Encoder:** Learns a compact representation of input data.

- **Decoder:** Learns to decode meaningful data from the encoded representations.

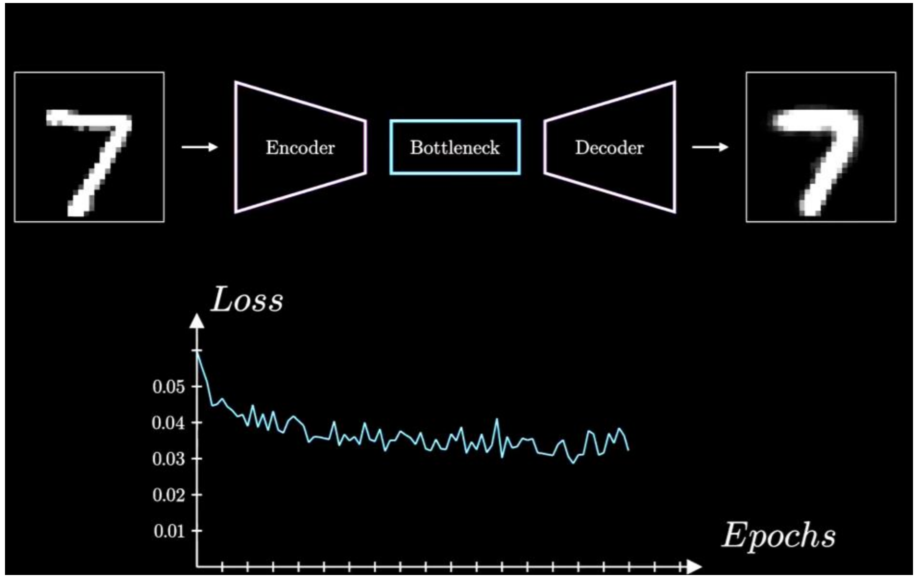
But how do we measure the quality of Latent space learnt?





Autoencoders Training

- The most common loss function used is a reconstruction loss or a **Mean Squared Error** loss.
- By learning the maximize the reconstruction loss between the input and output, the autoencoder learns a meaningful compressed Latent Space



Loss function

$$\mathcal{L}(x, y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$$

Autoencoders

$$f_{\text{enc}} : \mathbb{R}^d \to \mathbb{R}^k, \quad d \gg k$$

$$\mathbf{z} = f_{\text{enc}}(\mathbf{x}; \theta_{\text{enc}})$$

$$f_{\text{dec}} : \mathbb{R}^k \to \mathbb{R}^d \quad \hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{z}; \theta_{\text{dec}})$$

$$\min_{\theta} \ \mathcal{L}(\theta) = \sum_{i=1}^{N} \|\mathbf{x}_i - f_{\text{dec}}(f_{\text{enc}}(\mathbf{x}_i))\|_2^2$$

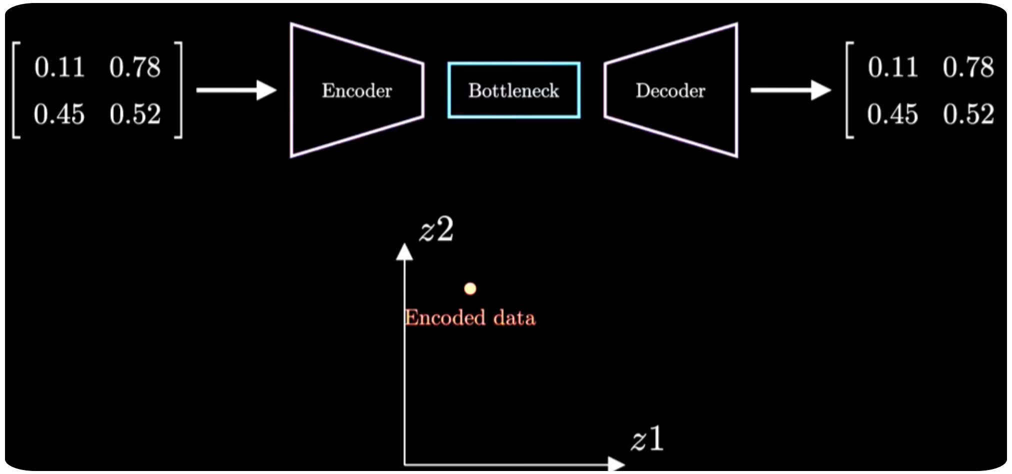
$$\theta = \{\theta_{\text{enc}}, \theta_{\text{dec}}\}$$

Limitation of Autoencoders

- Latent space has holes and discontinuities
- Autoencoders are not generative models. They do not learn p(x)
- they cannot generate new data points, as their latent representations are fixed and not probabilistic.

- Example: Try to generate a new face

Types of Autoencoders



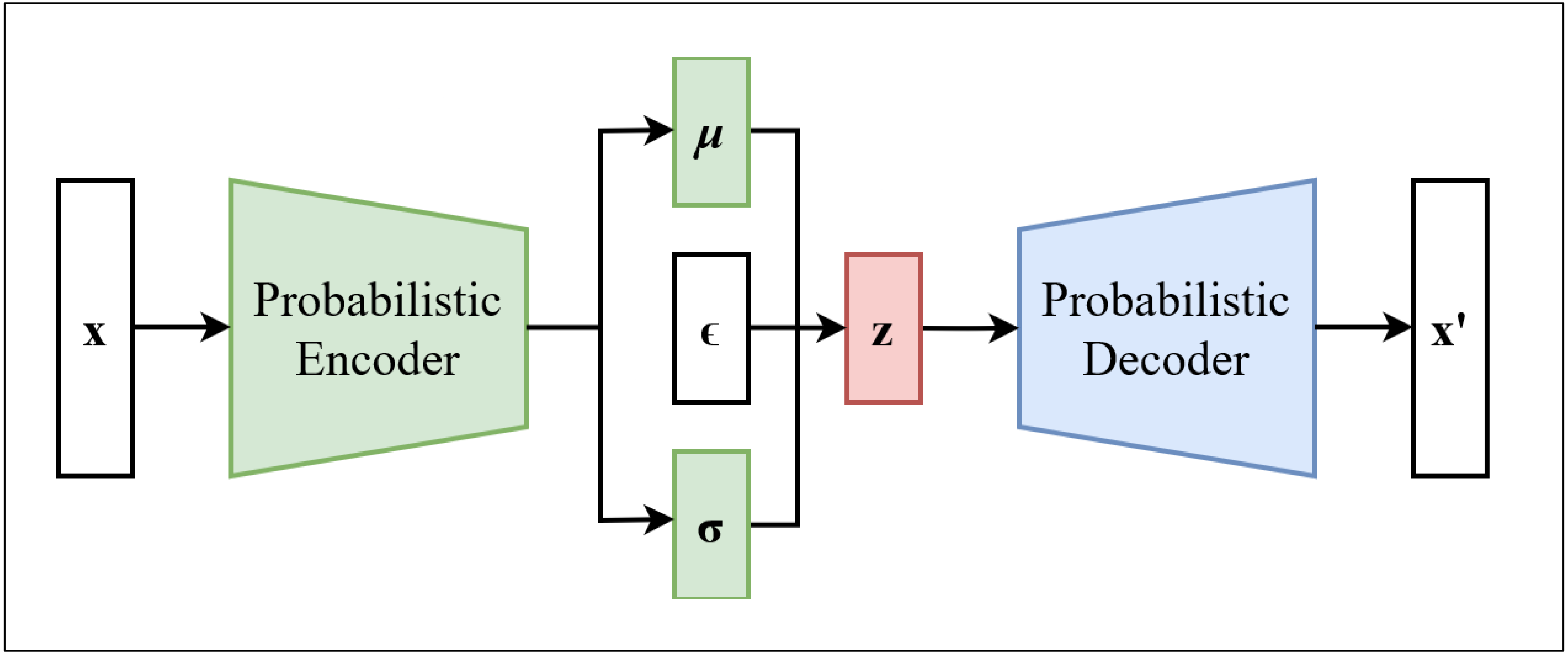
- **Autoencoder (AE):** Learns a compact representation by reconstructing the input.
- **Sparse Autoencoder (SAE):** Enforces sparse activations to learn meaningful features.
- **Variational Autoencoder (VAE):** Learns a probabilistic latent space for data generation.
- **Denoising Autoencoder (DAE):** Reconstructs clean input from noisy data for robustness.

Variational autoencoders

Introduces a **probabilistic framework**, allowing the latent space to represent distributions rather than fixed points.

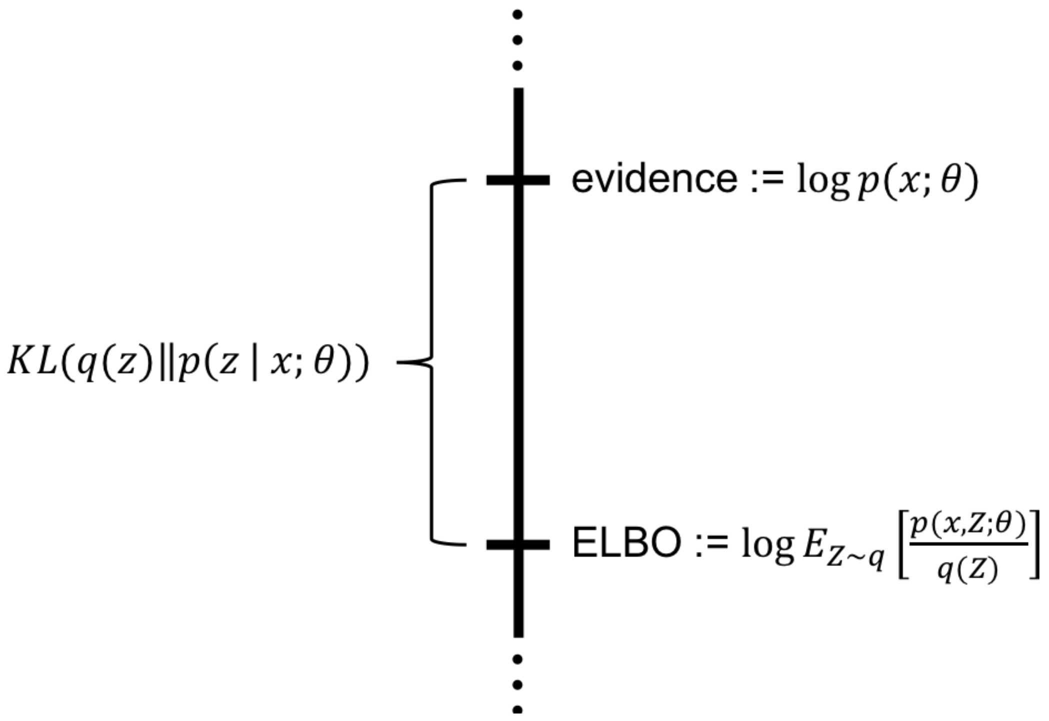
This makes it possible to **sample** new data points from the latent space, enabling applications like data generation.

Variational autoencoders



Backbone of Variational autoencoders

- **KL divergence** and **ELBO** (Evidence Lower Bound)



Expected Log likelihood

- **Expected Log Likelihood**

$$E_{q(z|x)}[\log P(x|z)]$$

$$\left[\log P\left(\frac{x}{z}\right)\right]$$

This measures how well the VAE can reconstruct the data *x* from the latent variable *z*.

A higher value means better reconstruction.

Role of KL divergence

This is the regularization term, ensuring that the approximate posterior $\boxed{q(z|x)}$ (learned by the encoder) stays close to the prior p(z) (usually a standard normal distribution).

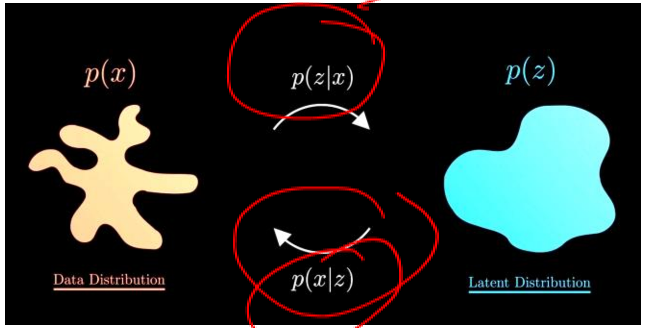
This helps prevent overfitting and ensures a structured latent space.

$q\left(\frac{z}{x}\right)$ learned by encoder

$p(z)$

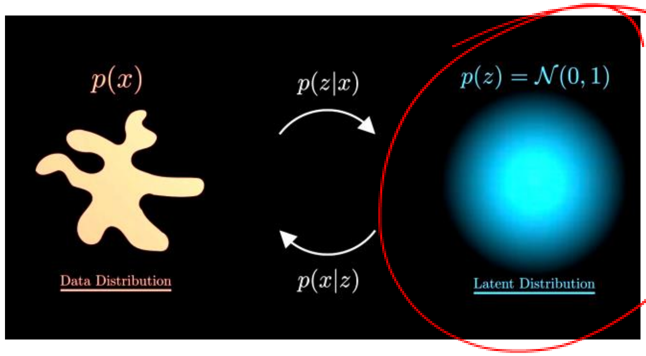
Foundation of VAE

Posterior

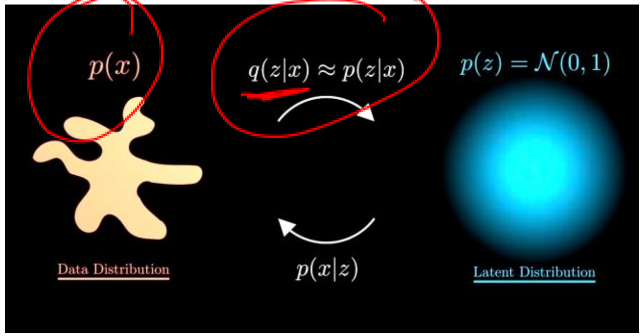


$P(\frac{x}{z})$

We define p(z) as a latent space and estimate p(x) using that

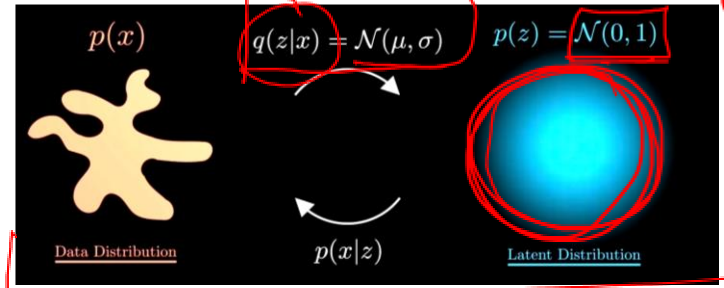


We don't know p(z) so we assume it as a normal distribution to estimate p(x) using that



P(x) is approximated using assumed q(z|x)

(μ, σ)



Parameters of the normal distribution q(x|z) are estimated using variational Inference

Loss Function in VAE

Very very high

Data consistency

Smooth

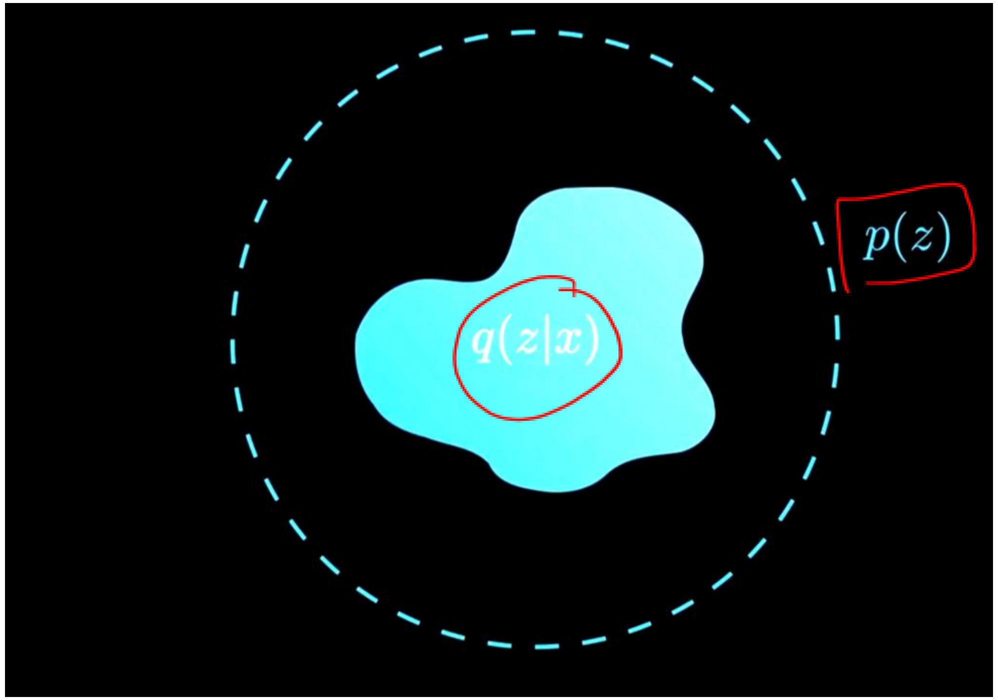
$$\mathcal{L}(x) = \underbrace{\mathbb{E}_{q(z|x)}\left[\log p(x|z)\right]}_{\text{L2}} - \overbrace{\mathrm{KL}(q(z|x) \mid p(z))}^{\text{Latent space regularization}}$$

VAE Loss function without reparametrizatrization

L2 Reconstruction loss

Regularization

$q(z/x)$



Estimating the q(z|x) space from the assumed p(z) gaussian space

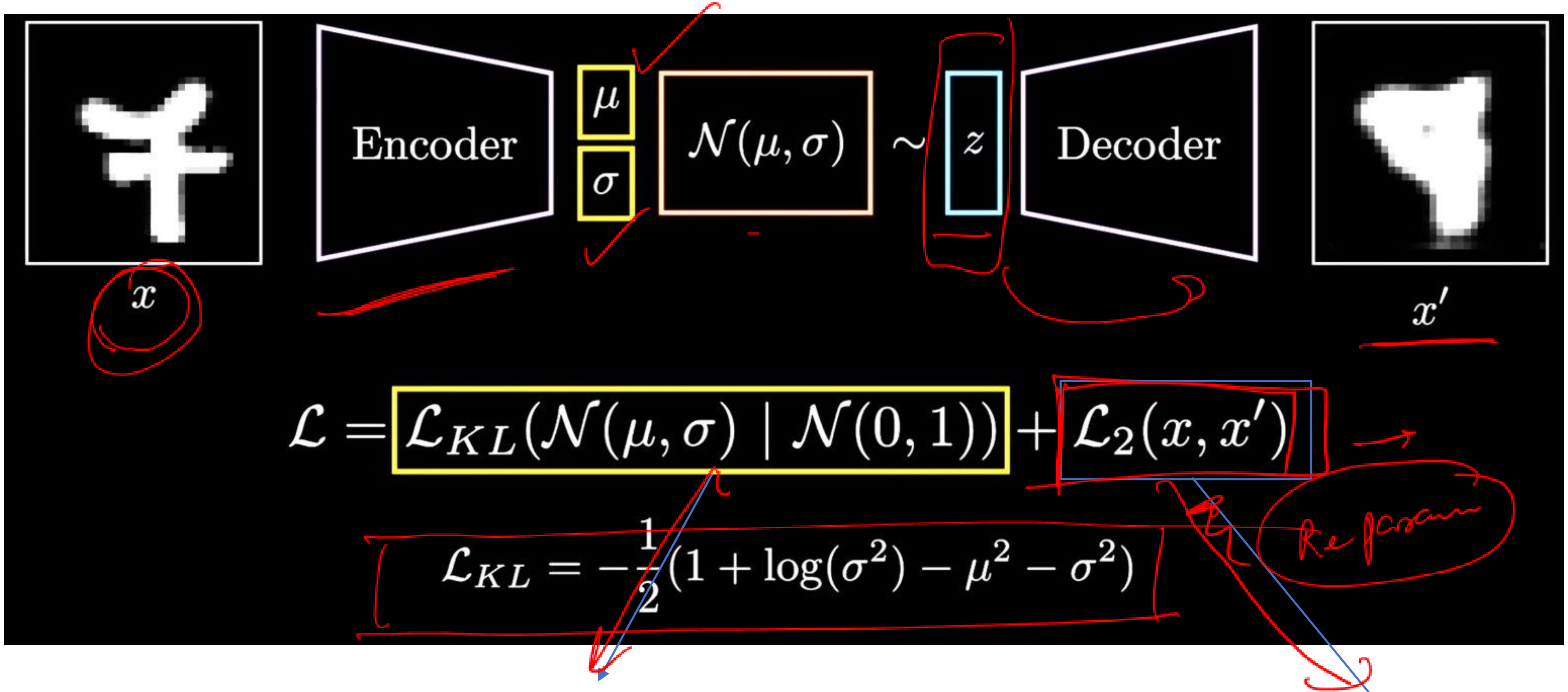
Intractable problem

A problem that can be solved in theory (given finite no. of resources) but in practice any solution that takes too many resources to be useful, is called intractable problem

How to derive the loss function in VAE?

- Refer to class notes

VAE Complete Loss Function (after reparameterization)



KL Divergence

Latent Space regularization after Reparameterization

Reparameterization Trick

z = μ + σ * ε, where ε ~ N(0, 1)

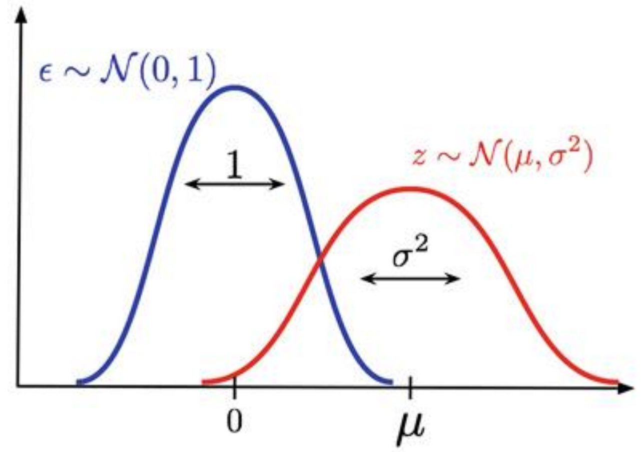


Fig. 5.3 An example of reparameterizing a Gaussian distribution: We scale ϵ distributed according to the standard Gaussian by σ, and shift it by μ.

References

- **Vaswani et al., 2017 — *Attention Is All You Need***
  https://arxiv.org/abs/1706.03762
- **Goodfellow et al., 2014 — *Generative Adversarial Networks***
  https://arxiv.org/abs/1406.2661
- **Kingma & Welling, 2013 — *Auto-Encoding Variational Bayes***
  https://arxiv.org/abs/1312.6114
- **Jumper et al., 2021 — *Highly Accurate Protein Structure Prediction with AlphaFold***
  https://www.nature.com/articles/s41586-021-03819-2
- **Shen et al., 2019 — *Deep Image Reconstruction from Human Brain Activity***
  https://www.nature.com/articles/s41593-019-0389-0