Neural Network and Deep Learning

Autoencoders & Generative Models

YANIKA KONGSOROT

Outline

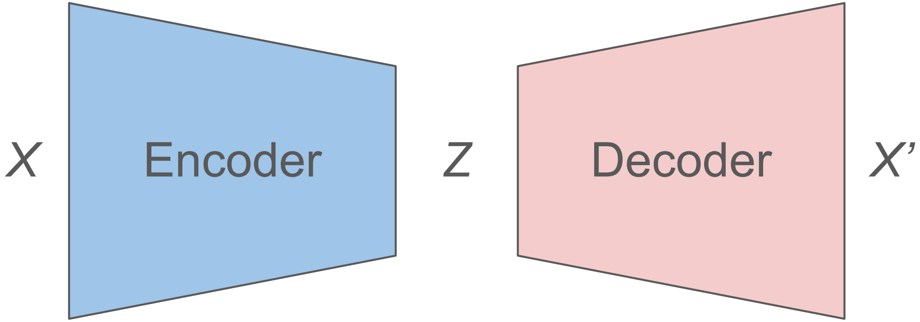
- Autoencoder
  - Vanilla Autoencoder
  - Autoencoder variants
- Generative models
  - Variational Autoencoder
  - Generative Adversarial Network

Autoencoder

Autoencoder

Autoencoders consist of two components : an **encoder** and a **decoder**

- **Encoder**: Compresses the input data X into a lower-dimensional representation, known as the **latent space** Z.
    - The *latent space* is a compact representation of the input data.
- **Decoder**: Reconstructs the original input X from the latent representation Z, aiming to make the output X' as close to X as possible.



Autoencoder

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^n$ be data

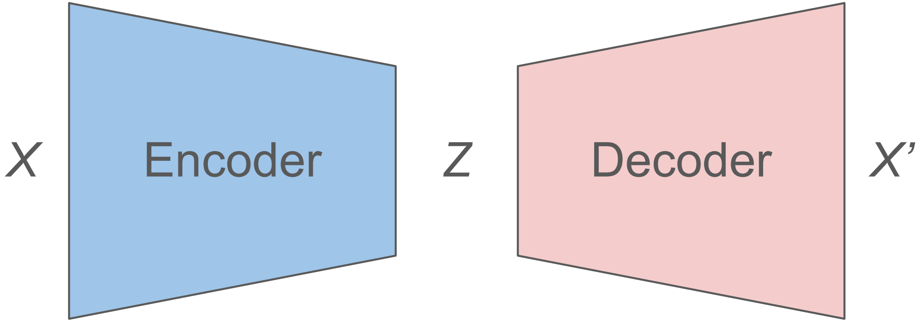
- Encoder: $f_E : \mathbb{R}^n \mapsto \mathbb{R}^d$
- Decoder: $f_D : \mathbb{R}^d \mapsto \mathbb{R}^n$

$$\boldsymbol{x}_i \approx f_D(f_E(\boldsymbol{x}_i))$$
$$\boldsymbol{x}_i \approx \boldsymbol{x}_i'$$

Truth

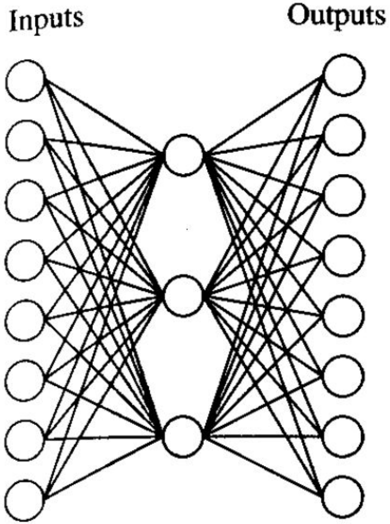
$$\boldsymbol{x}_i = \boldsymbol{x}_i'$$

Ideal



Mitchell, T. M., & Mitchell, T. M. (1997). *Machine learning* (Vol. 1, No. 9). New York: McGraw-hill.

Autoencoder



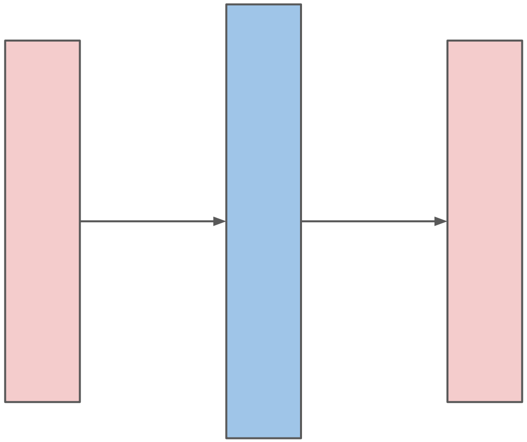
Input		Hidden Values				Output
10000000	→	.89	.04	.08	→	10000000
01000000	→	.15	.99	.99	→	01000000
00100000	→	.01	.97	.27	→	00100000
00010000	→	.99	.97	.71	→	00010000
00001000	→	.03	.05	.02	→	00001000
00000100	→	.01	.11	.88	→	00000100
00000010	→	.80	.01	.98	→	00000010
00000001	→	.60	.94	.01	→	00000001

FIGURE 4.7
Learned Hidden Layer Representation. This $8 \times 3 \times 8$ network was trained to learn the identity function, using the eight training examples shown. After 5000 training epochs, the three hidden unit values encode the eight distinct inputs using the encoding shown on the right. Notice if the encoded values are rounded to zero or one, the result is the standard binary encoding for eight distinct values.

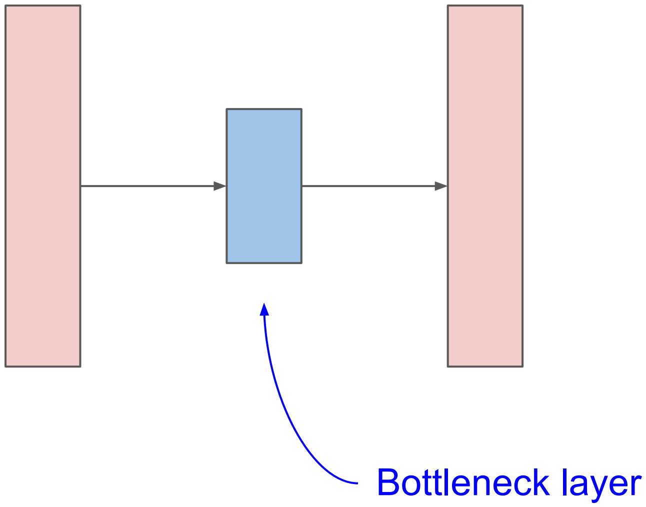
Autoencoder

General types of autoencoders based on size of hidden layer

Overcomplete



Undercomplete



Autoencoder

The goal of the autoencoder was initially to **minimize** the **reconstruction error** using *Mean Squared Error (MSE)*

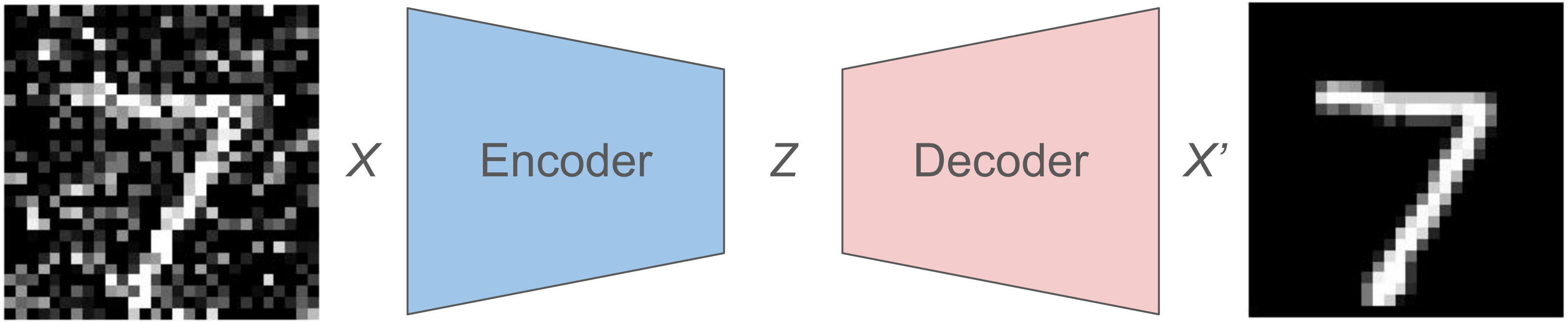
$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} (x_{ij} - x'_{ij})^2$$

But now *Binary Cross-Entropy (BCE)* is used instead (with sigmoid output layer).

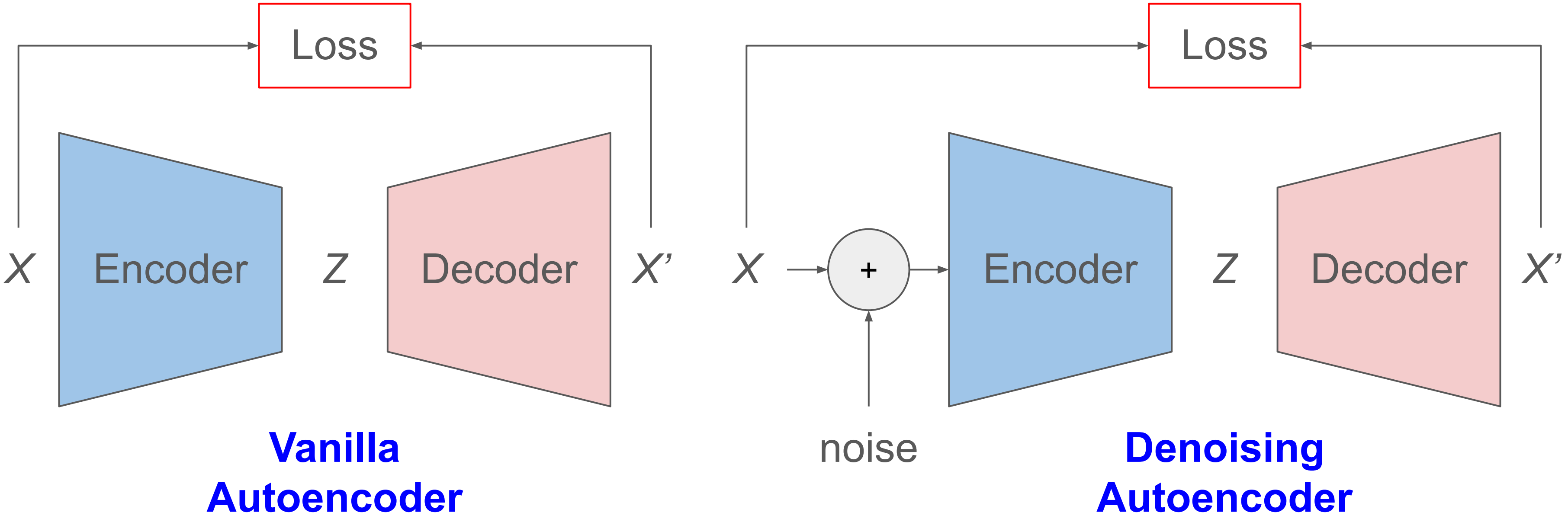
$$\mathrm{BCE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} \left[x_{ij} \cdot \log(x'_{ij}) + (1 - x_{ij}) \cdot \log(1 - x'_{ij}) \right]$$

Denoising Autoencoder

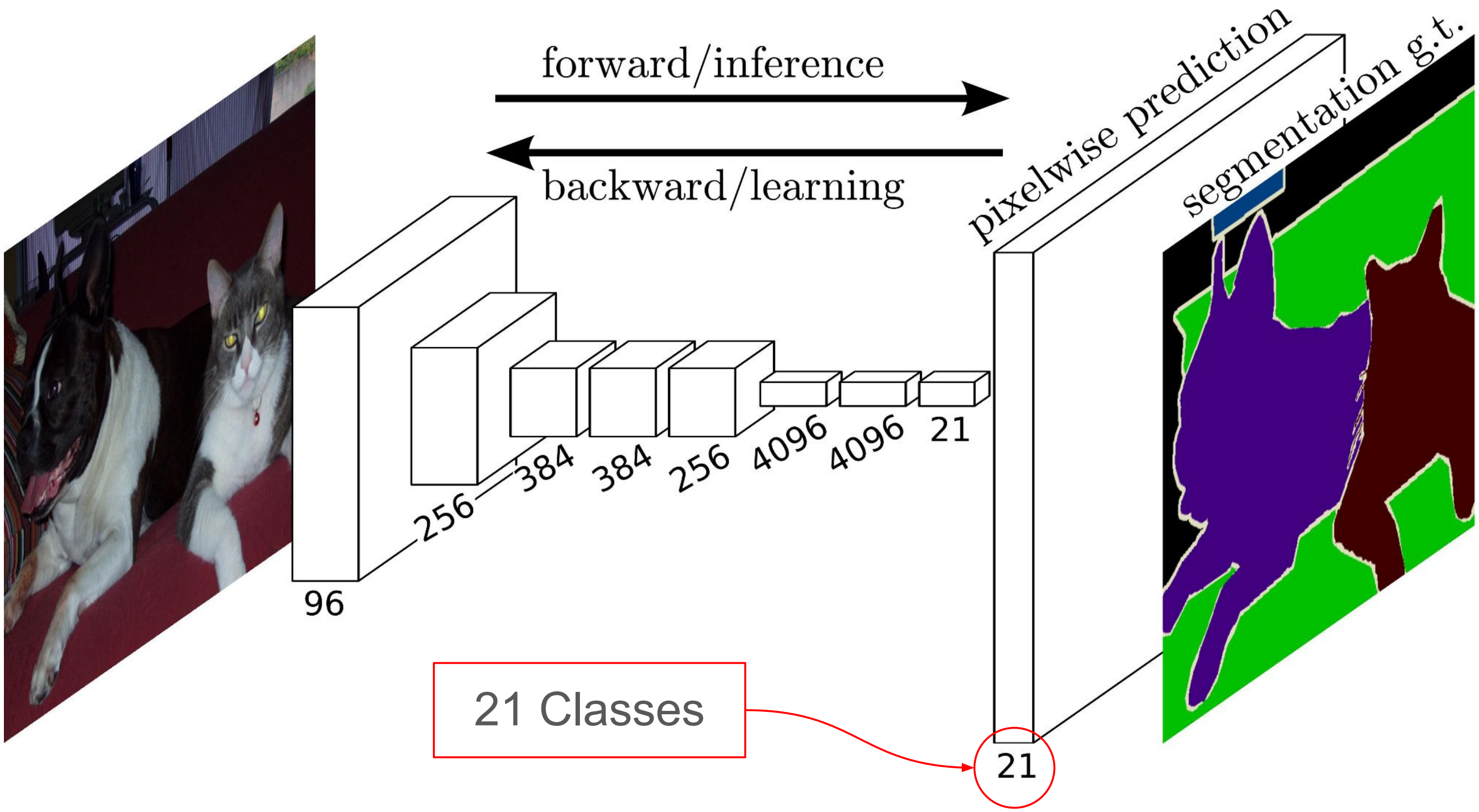
Denoising Autoencoder



Denoising Autoencoder

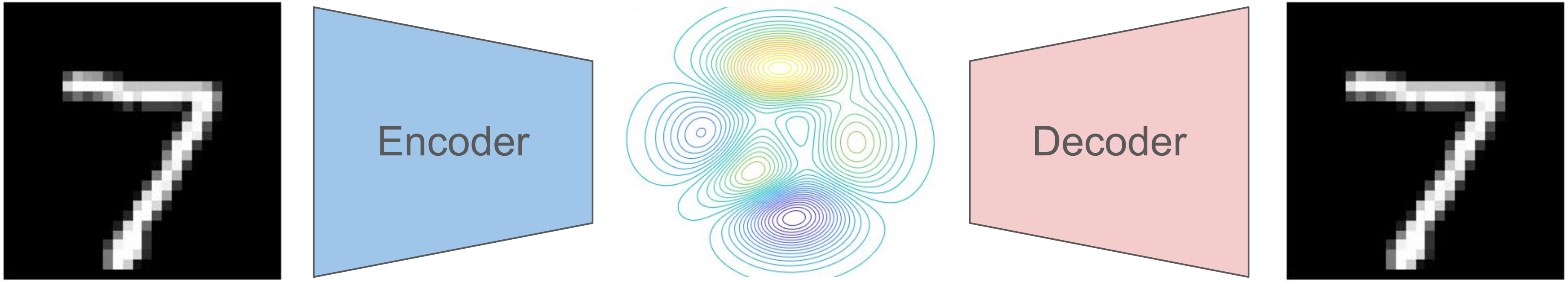


Applications of Autoencoder



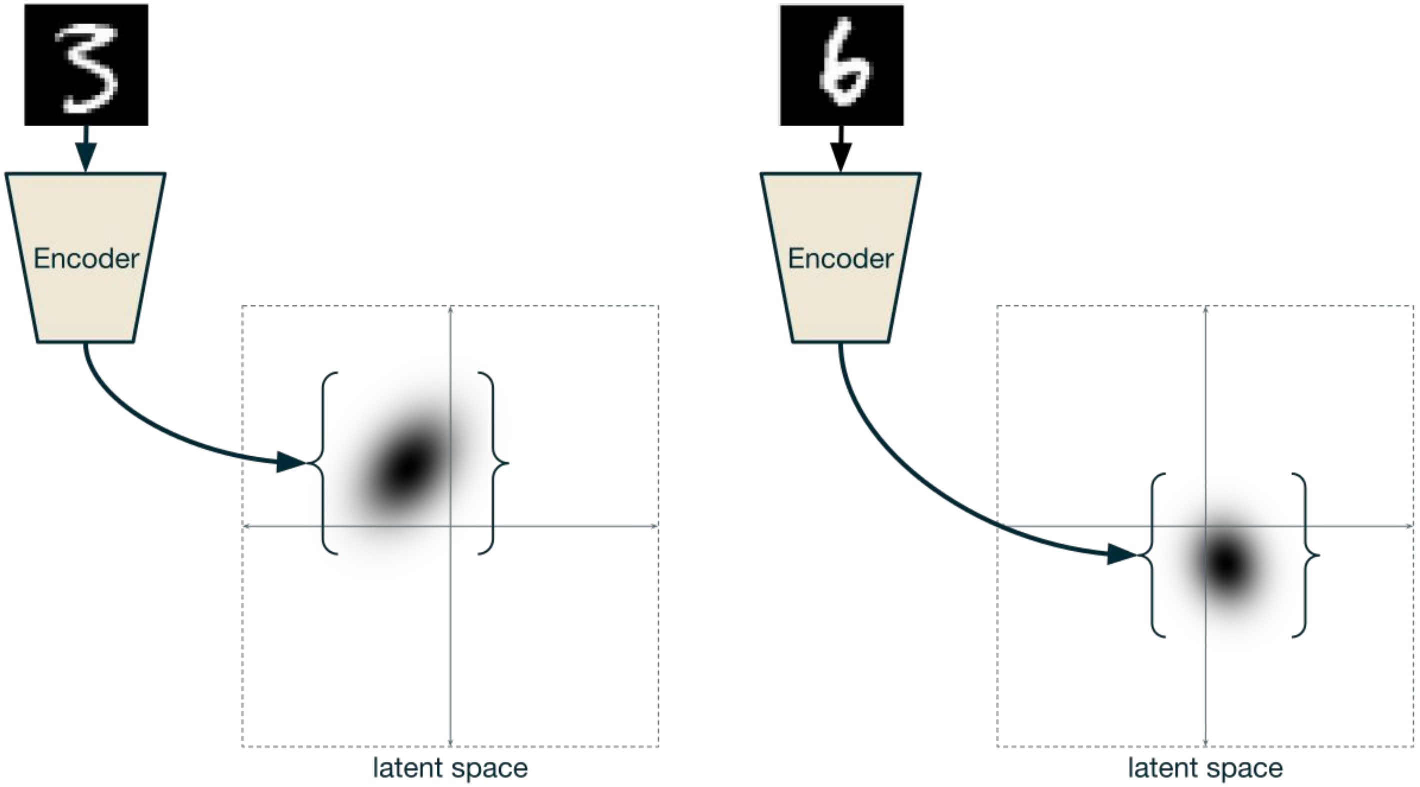
Variational Autoencoder

Variational Autoencoder

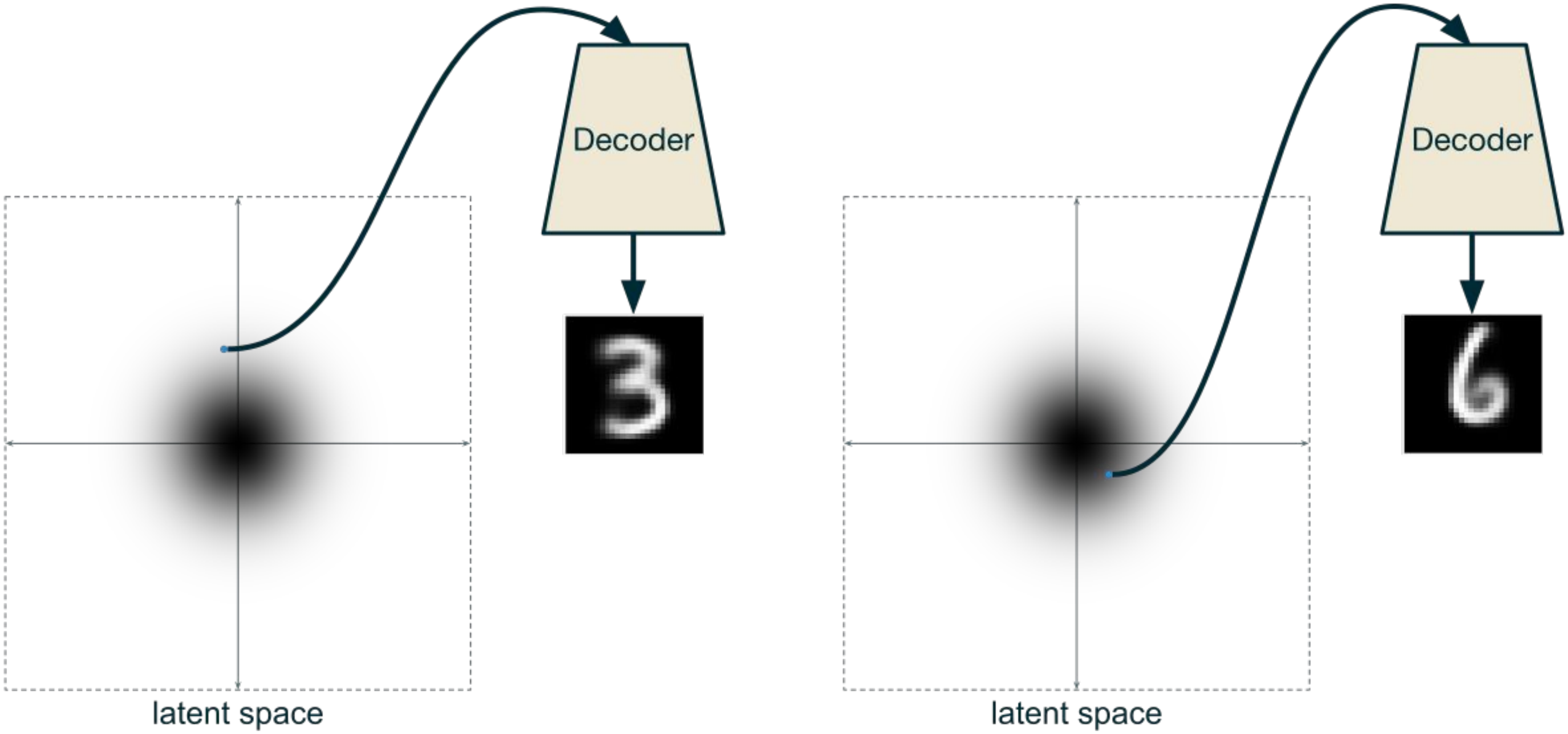


Latent space

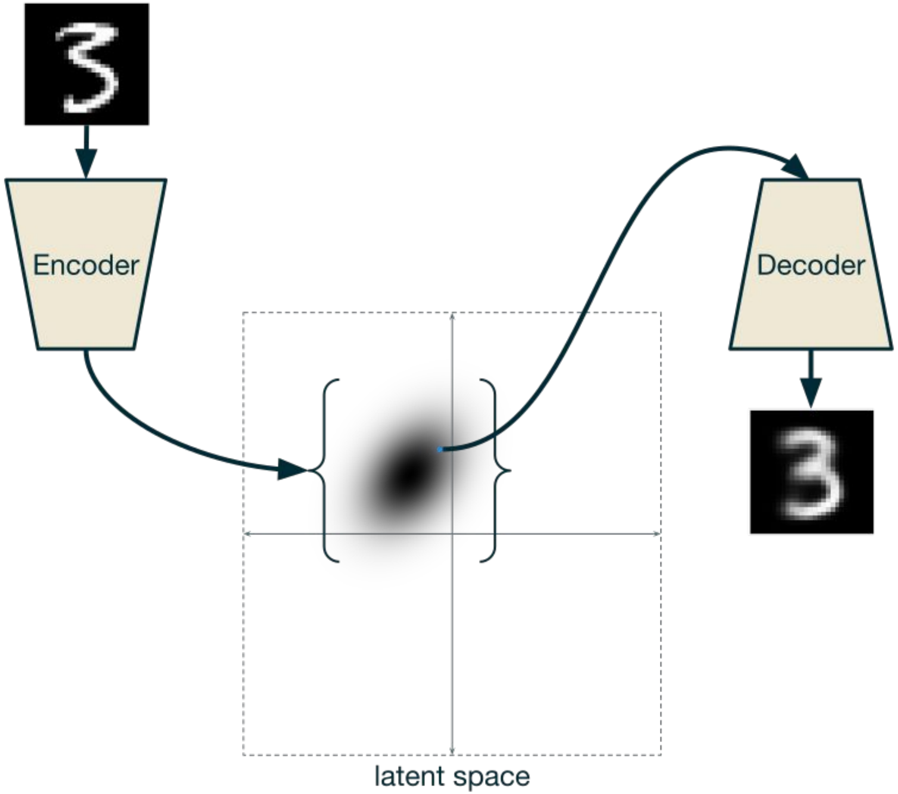
Variational Autoencoder



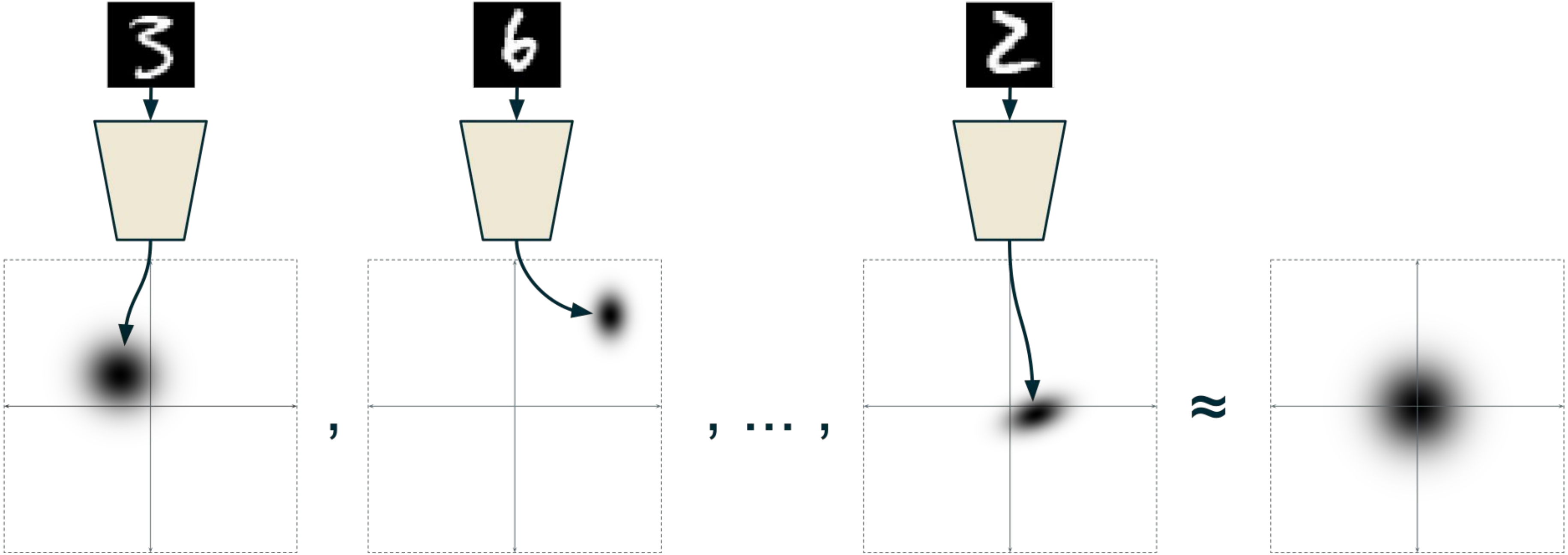
Variational Autoencoder



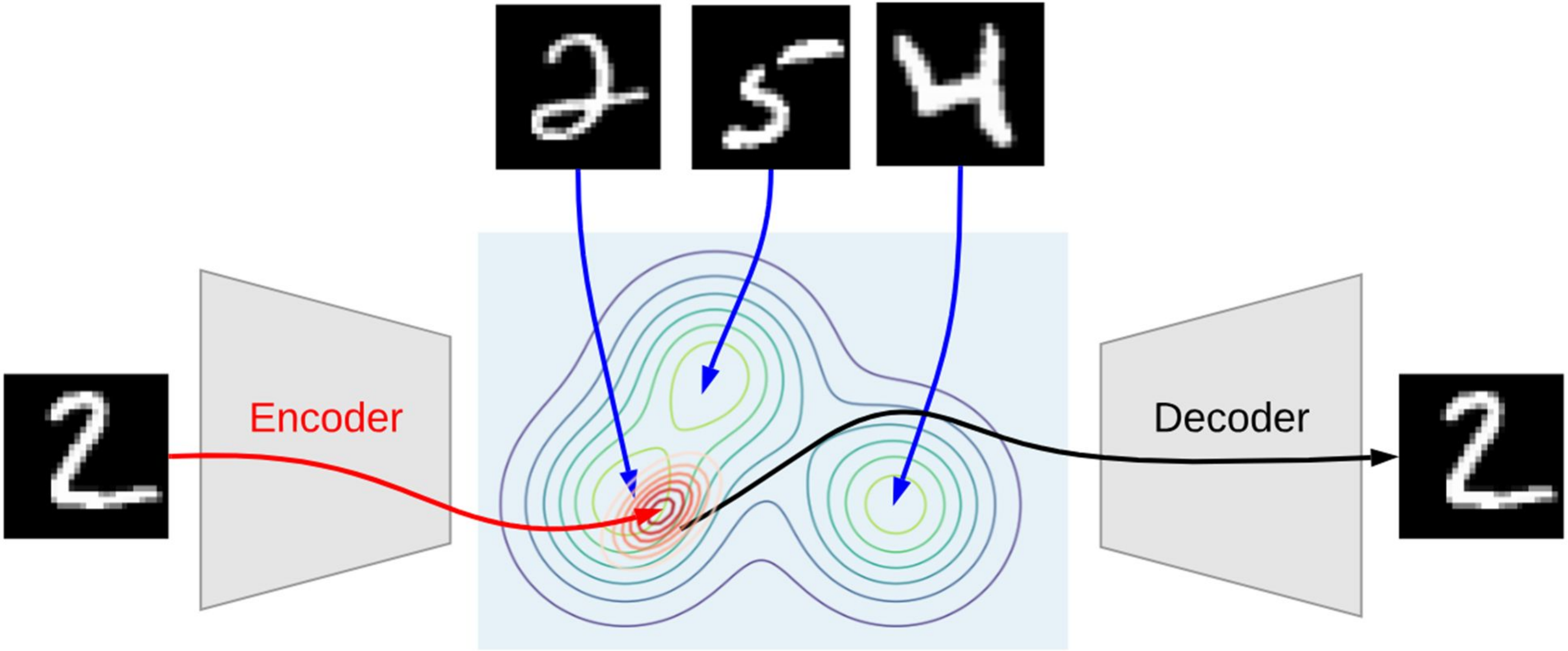
Variational Autoencoder



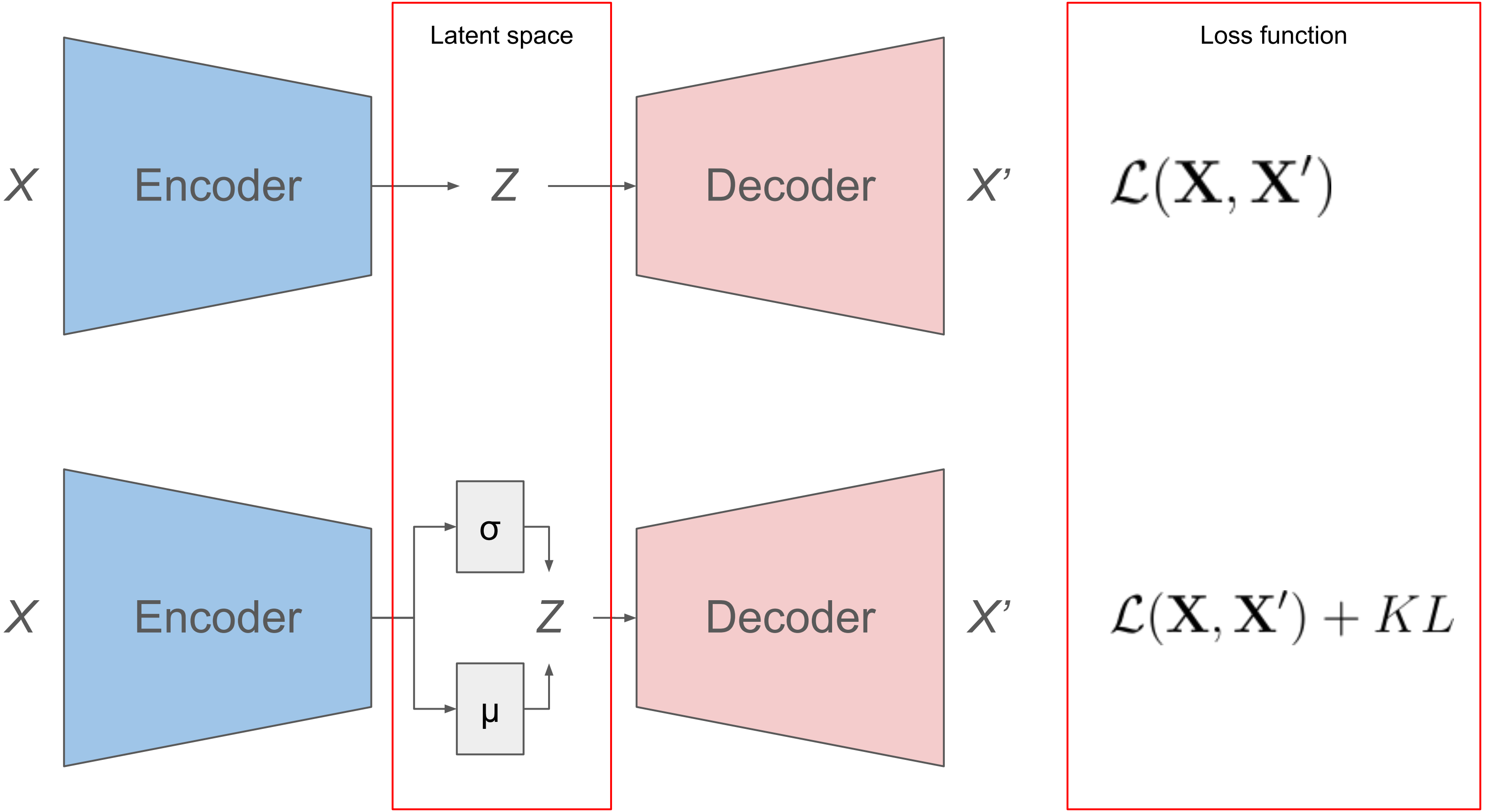
Variational Autoencoder



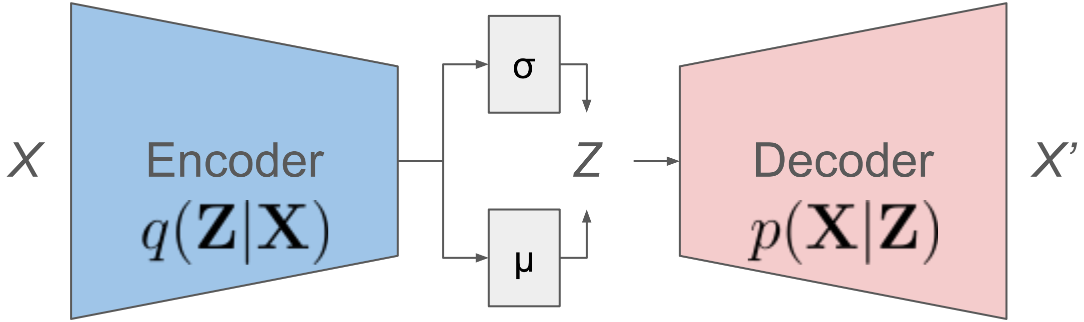
Variational Autoencoder



Variational Autoencoder



Variational Autoencoder



$$Z = \mu + \sigma \odot \varepsilon$$

$q(\mathbf{Z}|\mathbf{X})$ a probabilistic encoder

$p(\mathbf{X}|\mathbf{Z})$ a probabilistic decoder

$$\mathcal{L}(\mathbf{X}, \mathbf{X}') + KL$$

$$KL = D_{KL}(q(\mathbf{Z}|\mathbf{X}) \| \mathcal{N}(0,1))$$

$$= -0.5 \sum_{j=1}^{d} 1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2$$

KL the Kulback-Leibler (KL) divergence

Generative Adversarial Network

Generative Adversarial Network