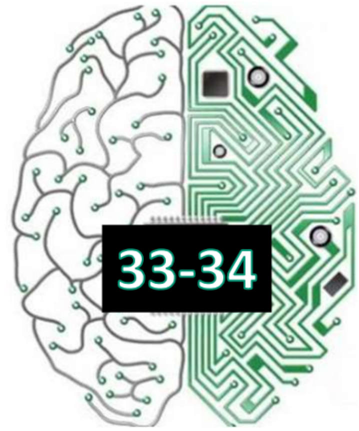
Open Elective Course [OE]
Course Code: CSO507
Winter 2023-24

Lecture#



Deep Learning

Unit-8: Generative Models (Part-II & Part-III)

Course Instructor:
Dr. Monidipa Das
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand 826004, India

Generative Models

2

- In many applications, it is desirable to **synthesise** data
  - Video post-production
  - Data augmentation
- Several types of generative models exist :
  - Restricted Bolzmann machines, Deep Belief models
  - **Variational autoencoders**
  - **Generative Adversarial Networks**;
  - Texture synthesis and style transfer models;
- The common idea in these models is the internal representation/latent space of the network

Autoencoder

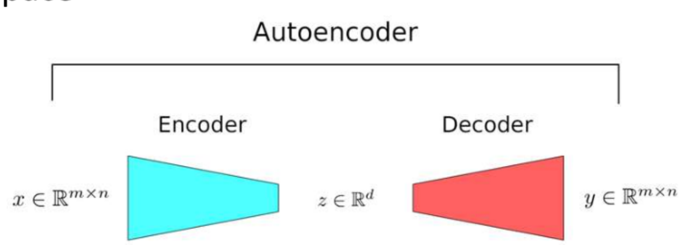
3

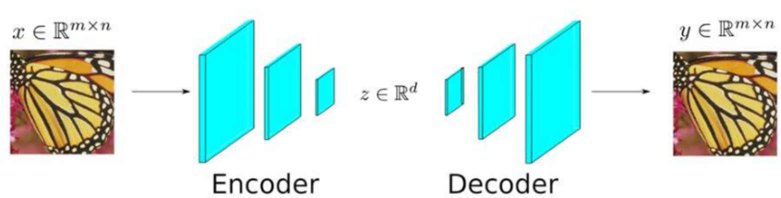
- Neural networks are often used for :
  - Classification/detection (MLPs, CNNs)
  - Modelling time-series, sequences (RNNs)
- All of these networks rely on the extraction of features to analyse data
- Idea : the network's internal representation of the data can be useful !
- Autoencoders and more generally generative models use this idea

Autoencoder

4

- Main idea : the latent space is a space where it is easier to manipulate/understand data
- More powerful and compact representation of data





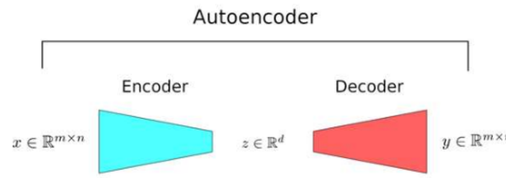
Autoencoder (AE)

- An AE is a neural network consisting of two sub-networks
  - The encoder Φ_e,

$$\Phi_e : \mathbb{R}^{mn} \to \mathbb{R}^d$$
$$x \mapsto \Phi_e(x) = z$$

  - The decoder Φ_d,

$$\Phi_d : \mathbb{R}^{d} \to \mathbb{R}^{mn}$$
$$z \mapsto \Phi_d(z) = y$$



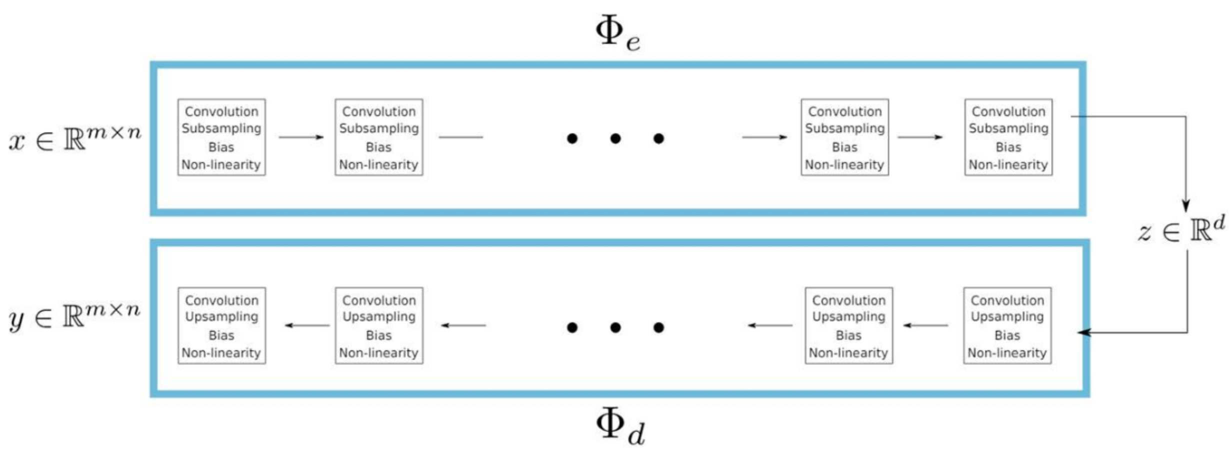
- As in other neural networks, the main components of AEs are **mlp's/convolutions**, **biases** and **non-linearities**

Autoencoders differ from General Data Compression

- Autoencoders are data-specific. This is different from, say, MP3 or JPEG compression algorithm
- Autoencoders are lossy
- Autoencoders are learnt

A generic autoencoder architecture

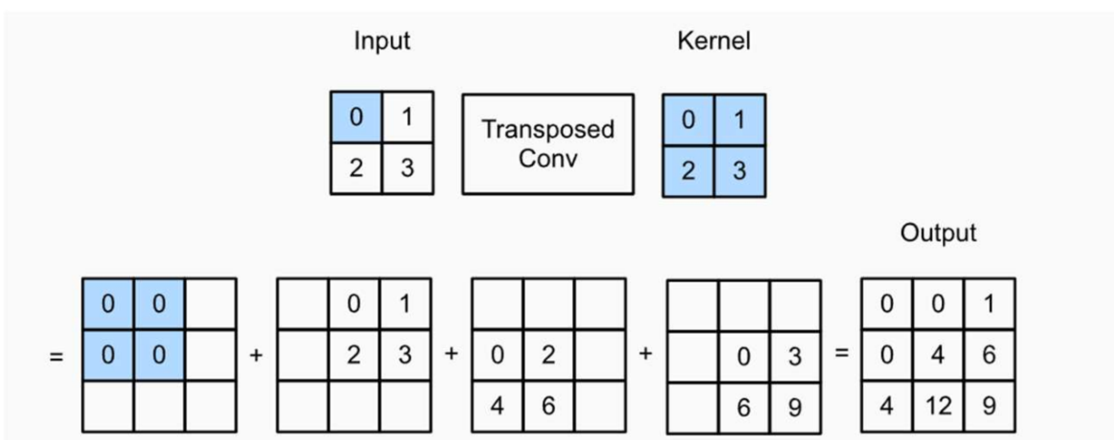
7



- $d << m * n$

Transposed Convolution

8



Autoencoder

9

The autoencoder is trained to reproduce the input x as an output y, in the sense of some norm, having gone through the bottleneck of the network

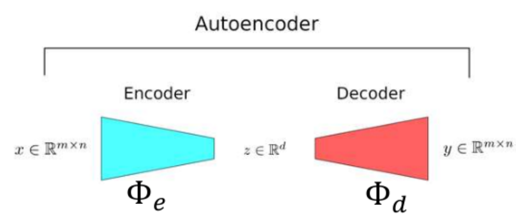
Autoencoding training minimisation problem

$$\begin{aligned}\mathcal{L}(x) &= \|y - x\|_2^2 \\ &= \sum_i^m \sum_j^n \left((\Phi_d \circ \Phi_e(x))_{i,j} - x_{i,j} \right)^2\end{aligned}$$

Autoencoder Design: Using Regularization

10

- Autoencoders come in many flavours, differ mainly by their loss functions
- Naive autoencoder loss L can lead to certain problems



- **Undercomplete:** Imagine $d = 1$ and very powerful Φ_e and Φ_d Can achieve very small reconstruction error but the learned code will not capture any interesting properties in the data
- **Overcomplete:** Imagine $d \gg mn$ and trivial (identity) functions Φ_e and Φ_d Can achieve even zero reconstruction error but the learned code will not capture any interesting properties in the data

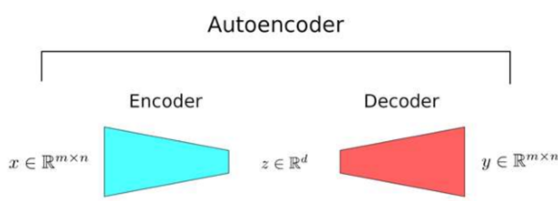
As is often the case in deep learning, this can be addressed **using regularization**

Regularized Autoencoder Properties

- Regularized AEs have properties beyond copying input to output:
  - Sparsity of representation
  - Smallness of the derivative of the representation
  - Robustness to noise
  - Robustness to missing inputs
- Regularized autoencoder can be nonlinear and overcomplete:
  - But still learn something useful about the data distribution even if model capacity is great enough to learn trivial identity function

Autoencoder variants

- **Sparse Autoencoders**



Sparse autoencoder

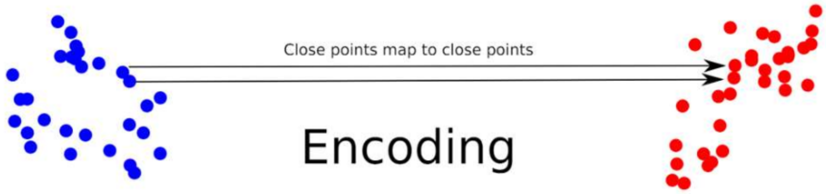
$$\mathcal{L}(x) = \|\Phi_d \circ \Phi_e(x) - x\|_2^2 + \lambda \|z\|_1$$

- The $\|z\|_1$ norm encourages sparsity in z

Autoencoder variants

13

- **Contractive Autoencoders**



$$\mathcal{L}(x) = \|\Phi_d \circ \Phi_e(x) - x\|_2^2 + \lambda \|J_x z\|_F^2$$

$$\text{Jacobian}:\ J_x(z) = \begin{pmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_{mn}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_d}{\partial x_1} & \cdots & \frac{\partial z_d}{\partial x_{mn}} \end{pmatrix}$$

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Autoencoder variants

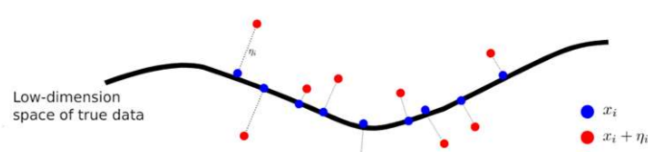
14

- **Denoising Autoencoders**

Denoising autoencoder

- Idea : add noise η to the input

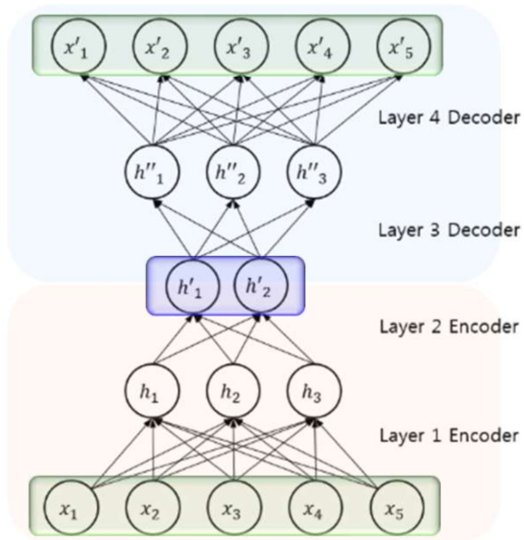
$$\mathcal{L}(x) = \|\Phi_d \circ \Phi_e(x + \eta) - x\|_2^2$$



Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Deep/Stacked Autoencoders

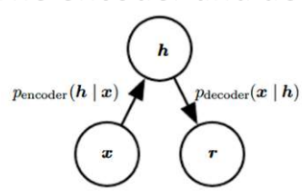
15



Stochastic Autoencoders

16

- The encoder and decoder functions defined using probability distributions



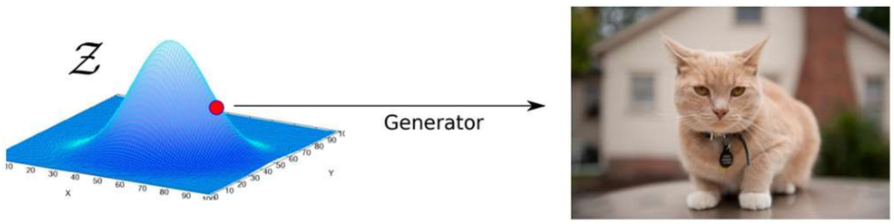
$$p_{\text{encoder}}(\boldsymbol{h}|\boldsymbol{x})$$
$$p_{\text{decoder}}(\boldsymbol{x}|\boldsymbol{h})$$

- Reconstruction error: Negative log-likelihood $-\log p_{\text{decoder}}(\boldsymbol{x}|\boldsymbol{h})$

- Can also use a prior distribution $p(\boldsymbol{h})$ on the encodings (equivalent to regularizer)
- Such ideas have been used to design generative models for autoencoders
  Variational Autoencoder (VAE)

Variational Autoencoders (VAE)

- Model the latent space in a probabilistic manner



Probabilistic model in latent space

Synthesis of random image

- the main goal here is to choose a loss function which will encourage the latent space to follow a probability distribution

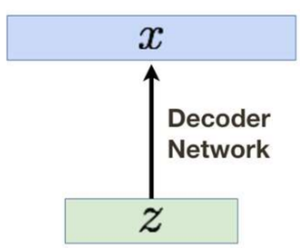
Variational Autoencoders (VAE)

Sample from true **conditional**

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

Sample from true **prior**

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



Learn model parameters to maximize likelihood of training data

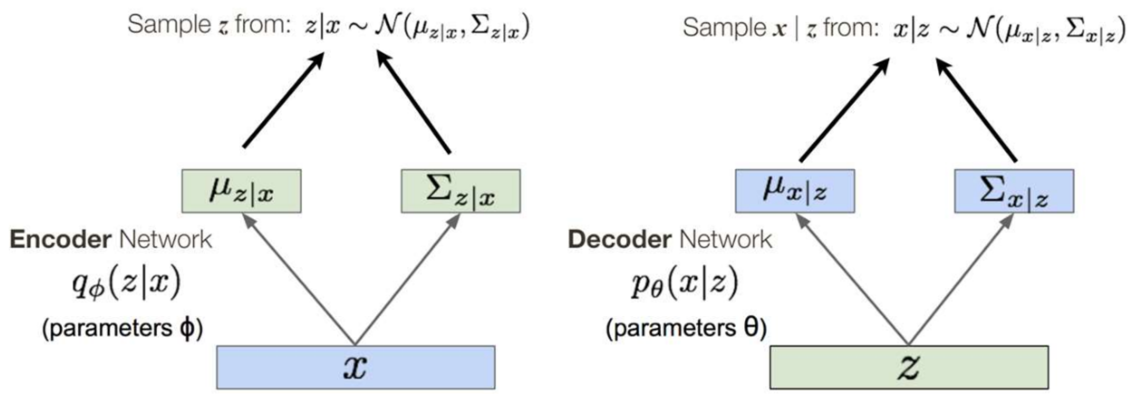
$$p_\theta(x) = \int p_\theta(z) p_\theta(x|z) dz$$

Intractable !

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

Variational Autoencoders (VAE)

19



Variational Autoencoders (VAE)

20

Now equipped with **encoder** and **decoder** networks, let's see (log) data likelihood:

$$\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \\
&= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)} + D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))
\end{aligned}$$

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

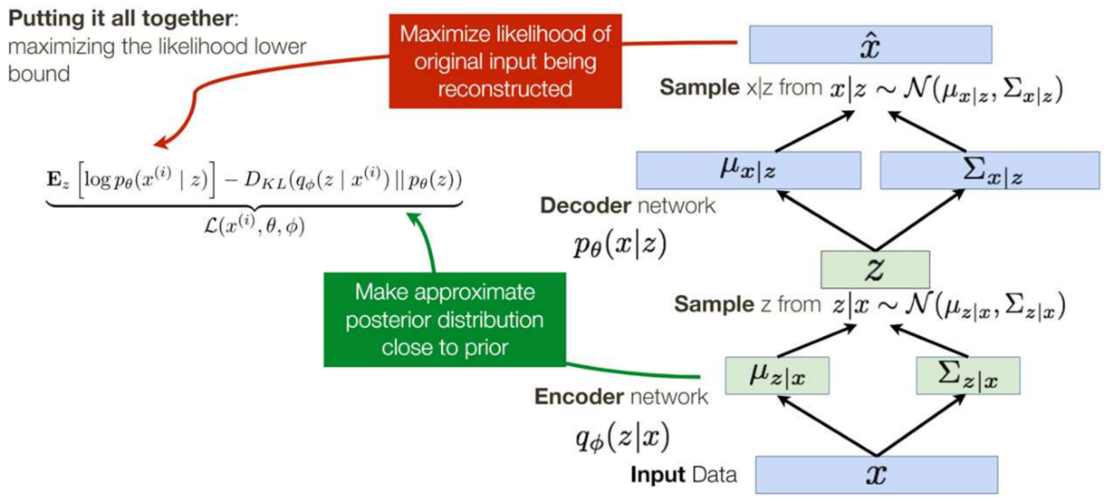
Variational lower bound ("**ELBO**")

Training: Maximize lower bound

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

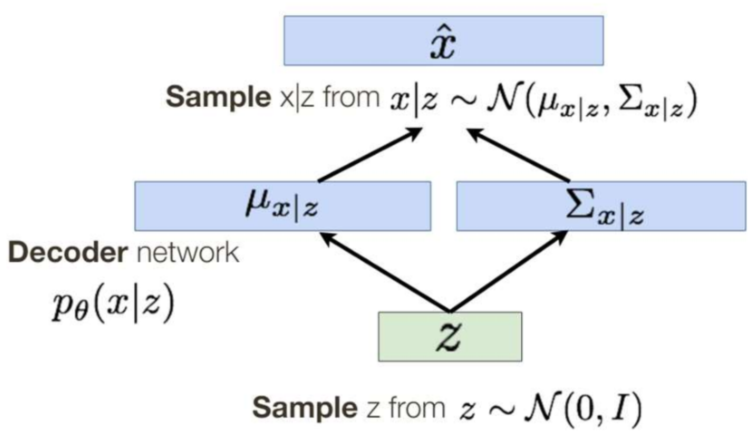
Variational Autoencoder: Learning

21



Variational Autoencoder: Generating Data

22



Questions?