Autoencoders: Multi-Modal Generative AI

Presentation · December 2024

1 author:

Praveen Kumar Chandaliya
Sardar Vallabhbhai National Institute of Technology Surat

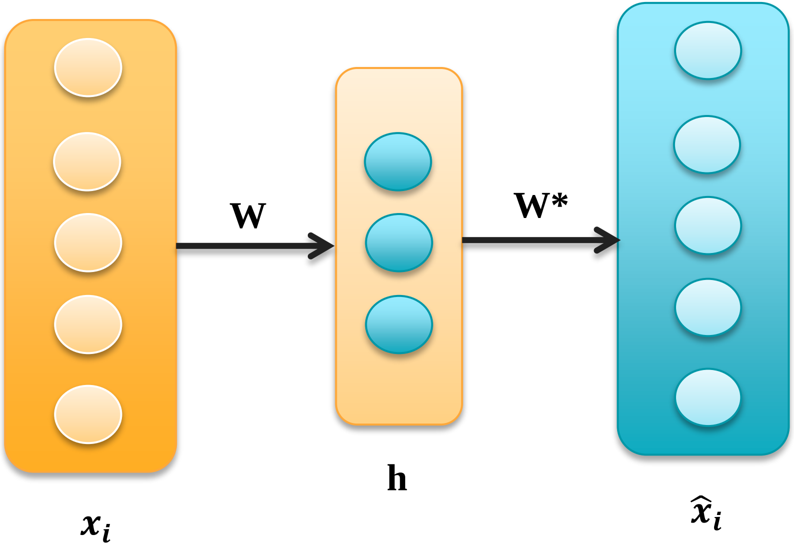
Sardar Vallabhbhai National Institute of Technology, Surat, India

Two-weeks Short-term Training Programme
On
Multi-Modal Generative AI
16-25 December, 2024

Autoencoders

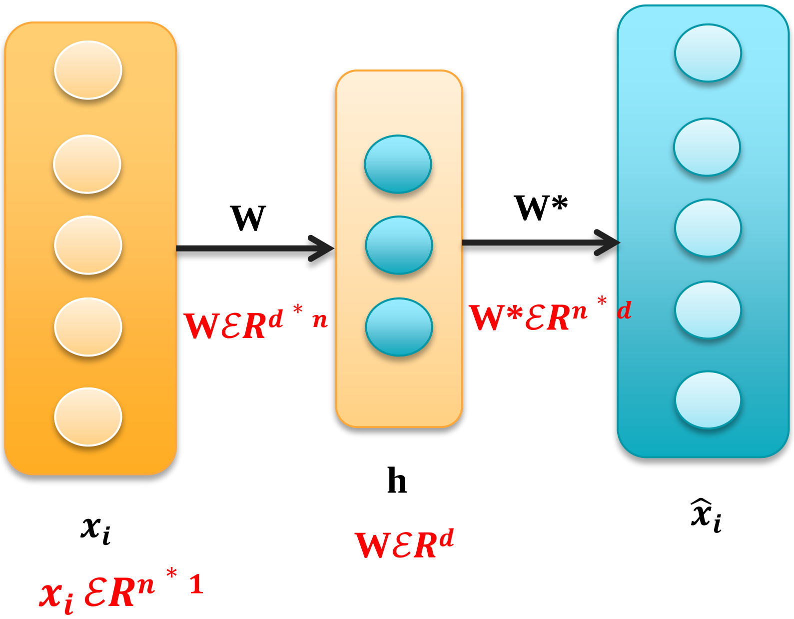
Praveen Kumar Chandaliya, PhD
https://scholar.google.com/citations?user=cx-vENIAAAAJ&hl=en
Department of Artificial Intelligence
pkc@aid.svnit.ac.in

Introduction to linear-Autoencoder



- An autoencoder is a special type of feed forward neural network which does the following.
- Encodes its input x_i into a hidden representation h

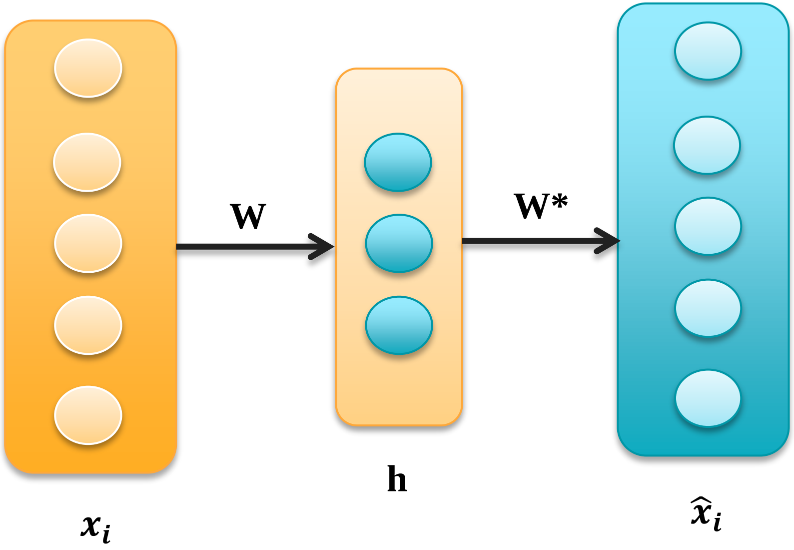
Introduction to linear-Autoencoder



$$h = g(Wx_i + b)$$

- An autoencoder is a special type of feed forward neural network which does the following.
- Encodes its input x_i into a hidden representation h

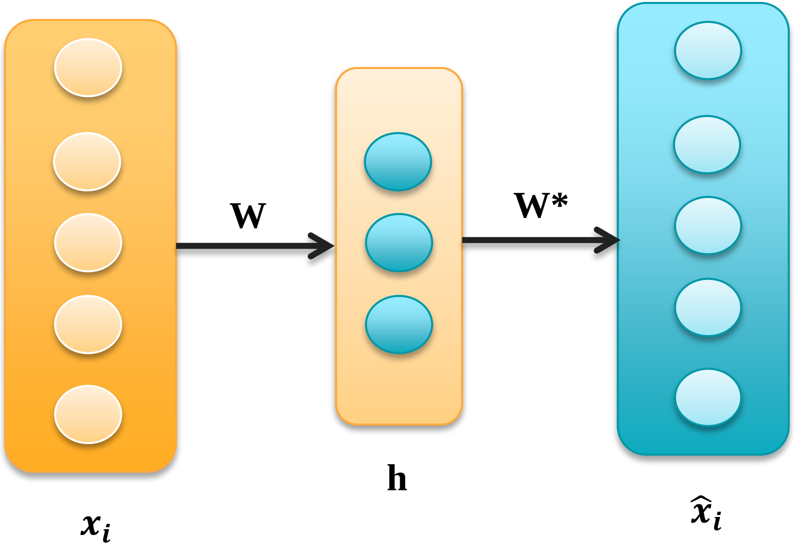
Introduction to linear-Autoencoder



h = g(Wx_i + b)

- An autoencoder is a special type of feed forward neural network which does the following.
- Encodes its input xi into a hidden representation h

Introduction to Autoencoder

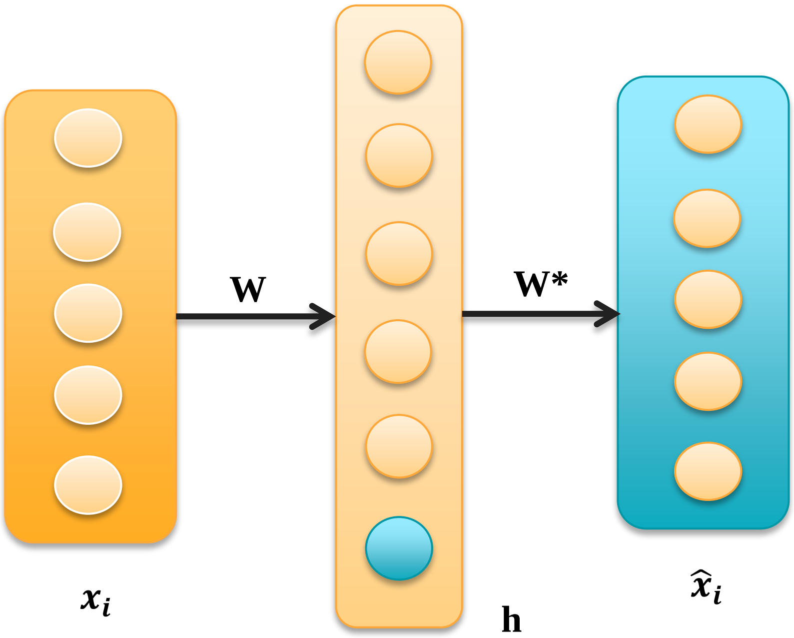


h = g(Wx_i + b)

$\hat{x}_i$ = f(W*h + c)

- An autoencoder is a special type of feed forward neural network which does the following.
- **Encodes** its input $\mathbf{x_i}$ into a hidden representation **h**
- **Decodes** the input again from this hidden representation
- The model is trained to minimize a certain loss function which will ensure $\hat{x}_i$ that is close to x_i
- An autoencoder where dim(h) < dim(x_i) is called an under complete autoencoder.
- h is a loss-free encoding of x_i. It captures all the important characteristics of x_i

Introduction to Autoencoder

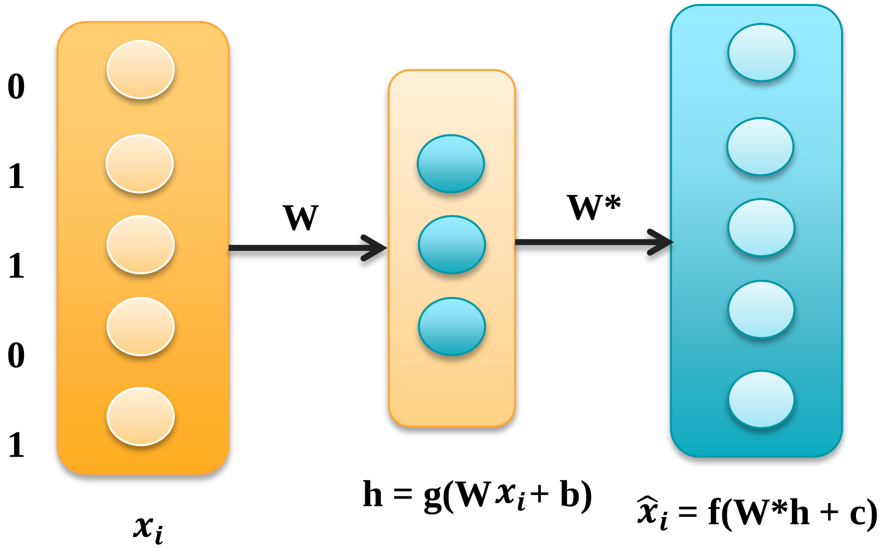


h = g(Wx_i + b)

$\hat{x}_i$ = f(W*h + c)

- case when **dim(h) >= dim(x_i)**
- In such a case the autoencoder could learn a trivial encoding by simply copying xi into h and then copying h into $\hat{x}_i$
- Such an identity encoding is useless in practice as it does not really tell us anything about the important characteristics of the data.
- An autoencoder where dim(h) dim(x_i) is called an **over complete autoencoder**

Choice of $f(x_i)$ and $g(x_i)$



- Suppose all our inputs are binary (0,1).
- Which of the following functions would be most apt for the decoder?

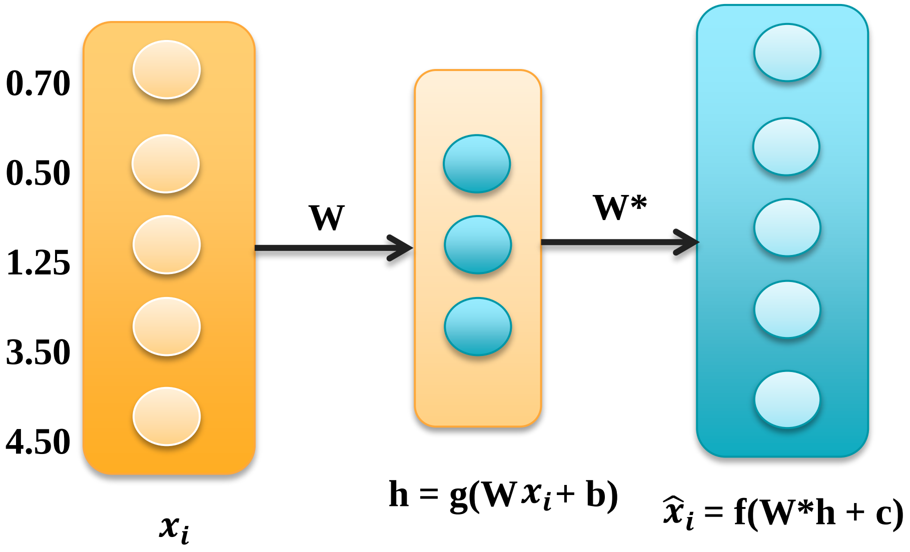
  $\hat{x}_i = \tanh(W^*h + c)$

  $\hat{x}_i = W^*h + c$

  $\hat{x}_i = \text{sigmoid}(W^*h + c)$

- **sigmoid as it naturally restricts all outputs to be between 0 and 1**

Choice of $f(x_i)$ and $g(x_i)$



- Suppose all our inputs are real number.
- Which of the following functions would be most apt for the decoder?

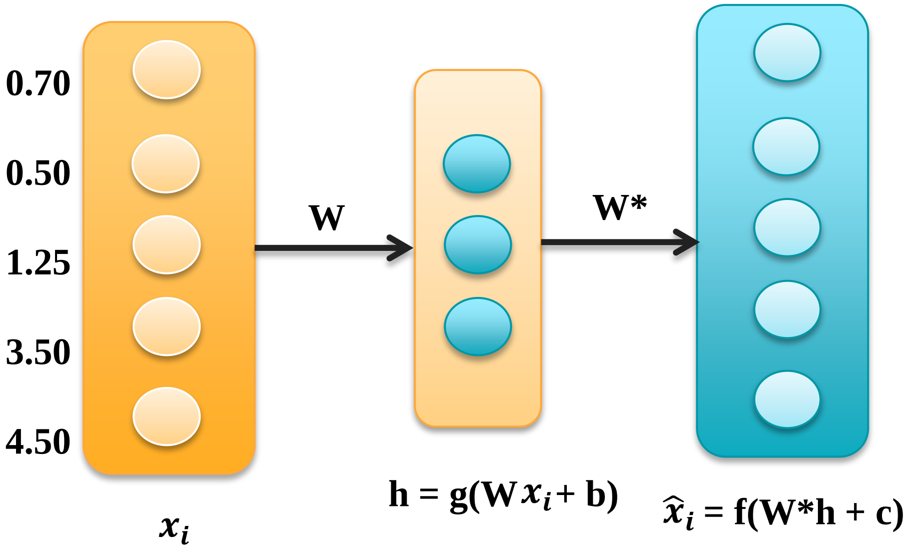
$$\hat{x}_i = \tanh(W^*h + c)$$

$$\hat{x}_i = W^*h + c$$

$$\hat{x}_i = \text{sigmoid}(W^*h + c)$$

- **sigmoid and tanh will restrict the reconstructed output to lie between [0,1] or [-1,1].**
- **Again, g is typically chose as the sigmoid function.**

Choice of loss function



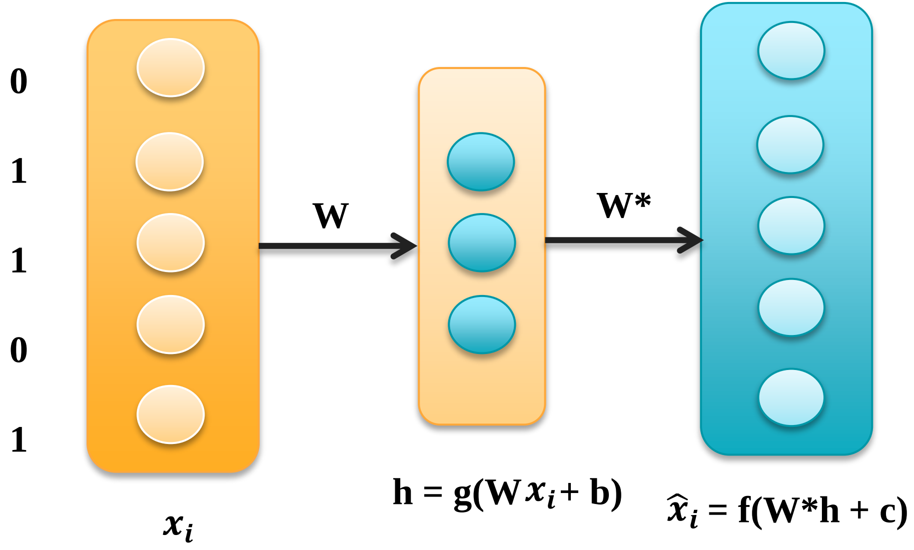
- Suppose all our inputs are real number.
- The objective of the autoencoder is to reconstruct $\hat{\boldsymbol{x}}_i$ to be as close to $\boldsymbol{x}_i$ as possible

$$L = \min_{w, w*, c, b} \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} (\widehat{x_{ij}} - xi_j)^2$$

$$L = \min_{w, w*, c, b} \frac{1}{m} \sum_{i=1}^{m} (\hat{x}_i - xi)^T (\hat{x}_i - xi)$$

- We can then train the autoencoder just like a regular feedforward network using back propagation

Choice of loss function



- Suppose all our inputs are binary.
- We use a sigmoid decoder which will produce outputs between 0 and 1, and can be interpreted as probabilities.

$$L = \min\{-\sum_{j=1}^{n}(x_{ij} log\widehat{x_{ij}} + (1 - x_{ij}) log(1\text{-}\widehat{x_{ij}}\ )\}$$

PCA vs AE

1. Methodology

PCA: PCA is a linear transformation technique.

It identifies the directions (principal components) in which the data varies the most.

It projects the data onto a lower-dimensional linear subspace that retains the most variance.

Autoencoder:

Autoencoders are neural network-based models.

They can capture both linear and non-linear relationships in the data.

The model consists of two parts: an encoder (compresses data) and a decoder (reconstructs the data).

The goal is to minimize the reconstruction loss between the input and output.

2. Linearity

PCA: Limited to linear transformations.

Autoencoder: Can model non-linear relationships due to the use of activation functions and non-linear architectures.

3. Learning Approach

PCA: Analytical method, with eigenvalue decomposition or singular value decomposition (SVD).

Autoencoder: Uses optimization techniques (like gradient descent) to train the neural network.

4. Dimensionality Reduction

PCA: Provides an explicit transformation matrix (principal components) for dimensionality reduction.

Autoencoder: Embedding is learned implicitly in the hidden layer(s) of the neural network.

6. Interpretability

PCA: Principal components are linear combinations of original features, making them relatively interpretable.

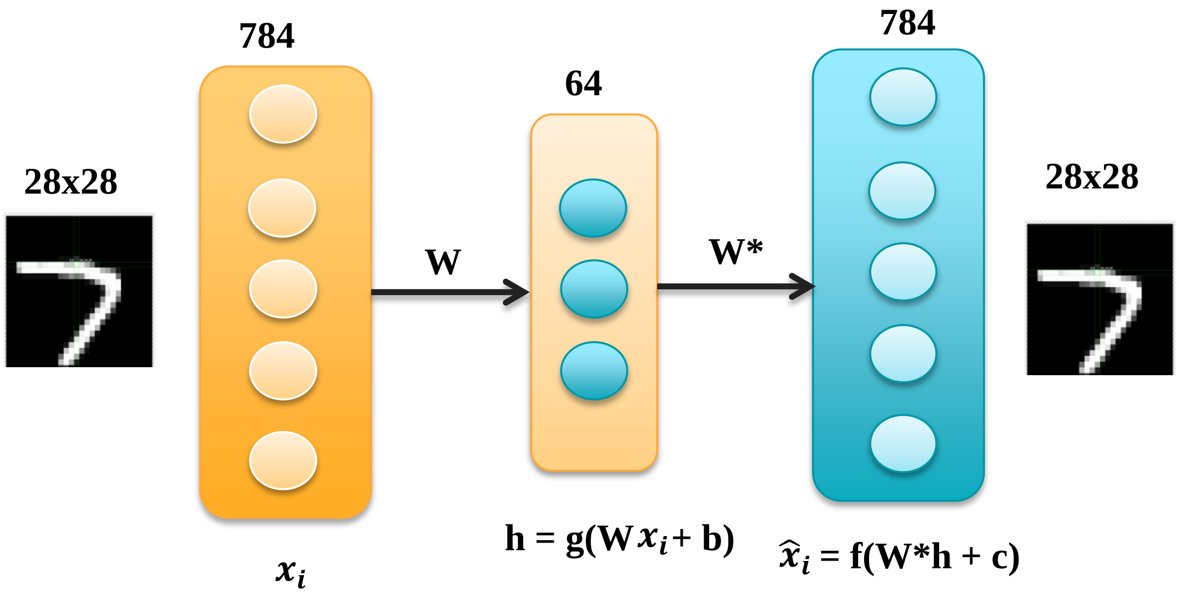
Autoencoder: The features in the latent space are less interpretable because they are learned features without explicit mathematical representation.

7. Scalability

PCA: Computationally efficient for small to moderate-sized datasets.

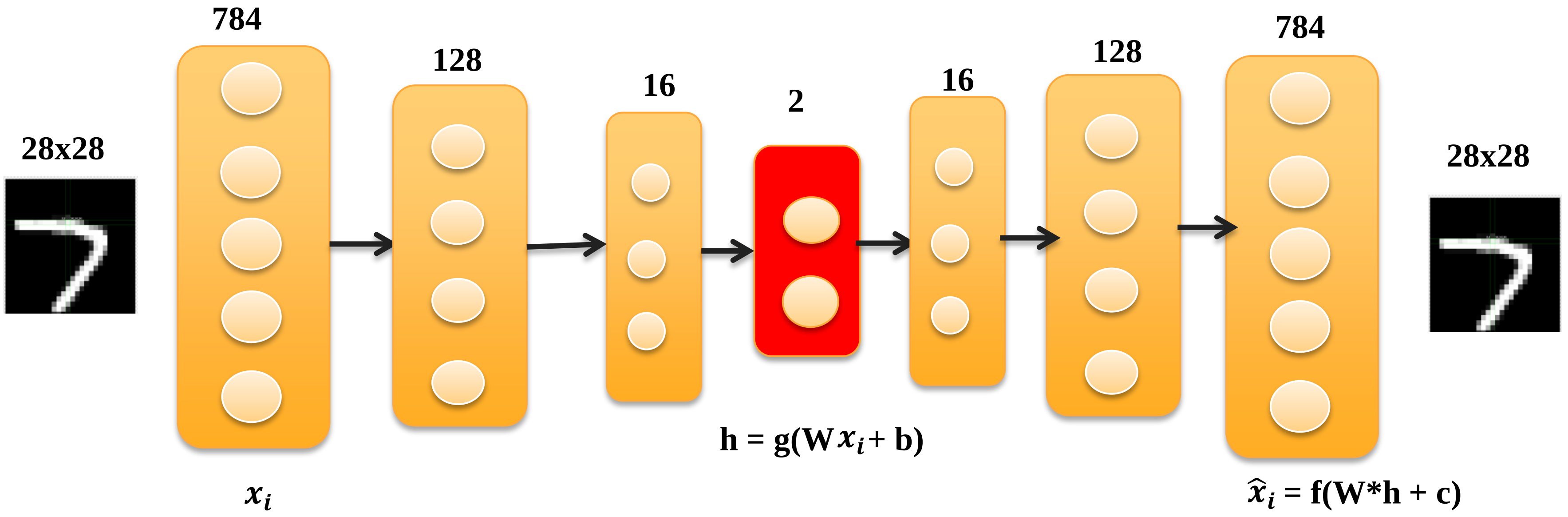
Autoencoder: Scales better to large datasets and can leverage GPUs for training.

Implementation-1



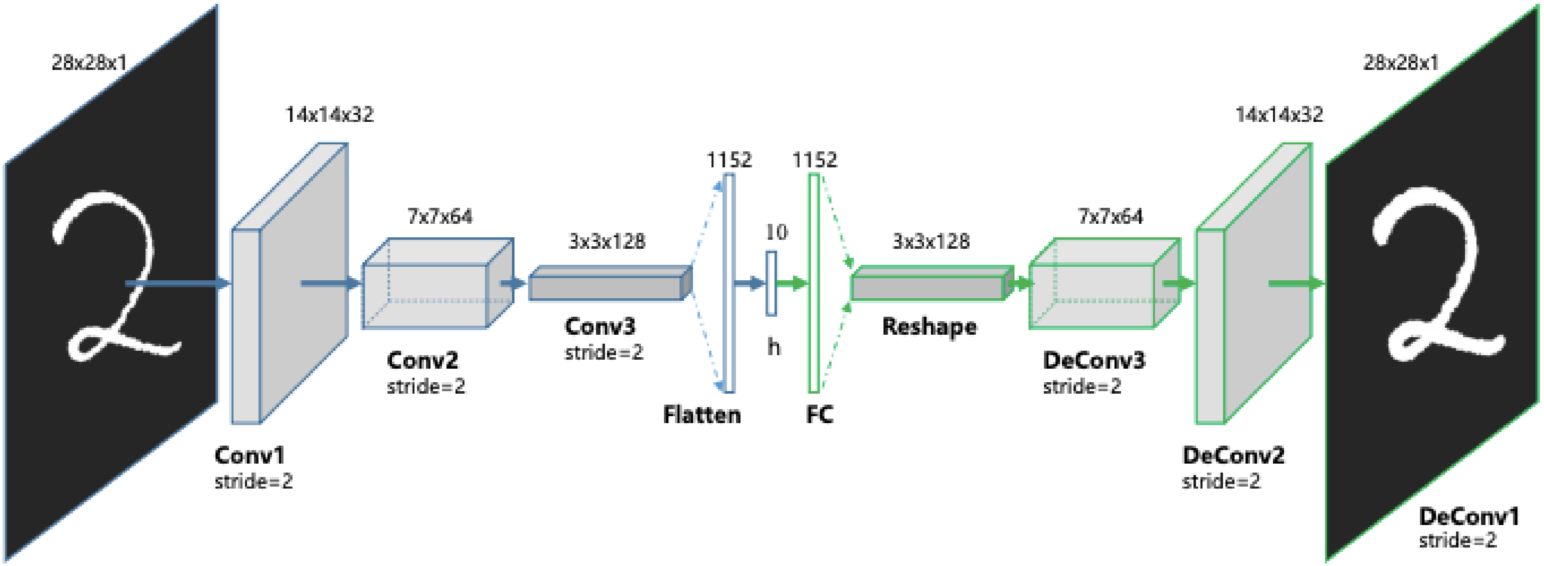
Colab: STTP_Linear_Autoencoder.ipynb

Implementation-2



Colab: STTP_Linear_Autoencoder_2.ipynb

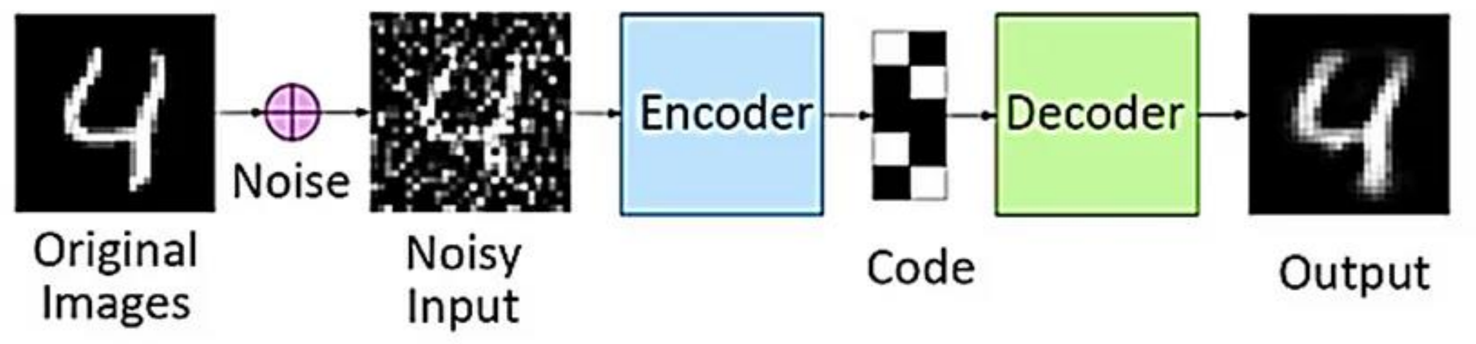
Convolution AutoEncoder



A convolutional autoencoder (CAE) is a deep learning model that uses convolutional layers to extract features and generate a compact representation of data. CAEs are a type of autoencoder, which are neural networks that learn to encode input data into a lower-dimensional representation and then decode it back to its original form.

STTP_CNN_AE.ipynb

Denoising Autoencoder (DAE)



STTP_Denoising_AE.ipynb

Autoencoder Application

1. Image Denoising

Autoencoders can remove noise from images by learning to reconstruct clean images from noisy inputs. Paired with Generative AI, this technique can:

- Enhance low-quality or noisy datasets.
- Improve the quality of generated synthetic images.

2. Image Compression

- Autoencoders can learn efficient low-dimensional representations of images, reducing storage requirements.
- Generative AI can reconstruct high-quality images from these compressed representations, making it ideal for real-time image transmission and storage systems.

3. Image Inpainting

- Autoencoders can fill in missing parts of images, making them useful for restoration tasks.
- Generative AI enhances this by ensuring the reconstructed sections blend seamlessly with the original image.

4. Style Transfer

- Autoencoders combined with generative adversarial techniques can modify the style of images while preserving content (e.g., converting photos to paintings or changing image textures).

Autoencoder Application

5. Face Aging and De-Aging

- Autoencoders can learn the latent representations of facial features to simulate aging or rejuvenation.
- Generative AI models such as GANs can further refine these representations to produce realistic results.

6. Localized Image Manipulation

- **Autoencoders** can embed local facial or image modifications in latent spaces, allowing for fine-grained adjustments (e.g., changing the shape of eyes or mouth).
- Integration with GANs ensures realistic alterations, leveraging latent spaces to manipulate specific regions.

7. Anomaly Detection

Autoencoders can detect anomalies by measuring reconstruction errors. This is useful in:

- Identifying fake or tampered images in forensic applications.
- Detecting subtle flaws in generated synthetic images.

8. Image-to-Image Translation

Paired with Generative AI, autoencoders can perform complex translations, such as:

- Transforming day-to-night scenes.
- Converting sketches to photorealistic images.

Autoencoder Application

9. Synthetic Dataset Generation

- Autoencoders can create diverse variations of images within a dataset by reconstructing and modifying images in the latent space.
- This is especially valuable in domains requiring large labeled datasets, like medical imaging.

10. Deepfake Detection

- Autoencoders can analyze the latent space of images and videos to detect artifacts introduced by deepfake generation processes.

11. Domain-Specific Image Augmentation

For medical imaging, autoencoders can:

- Denoise and enhance low-quality scans.
- Generate realistic augmented samples to improve diagnosis accuracy.

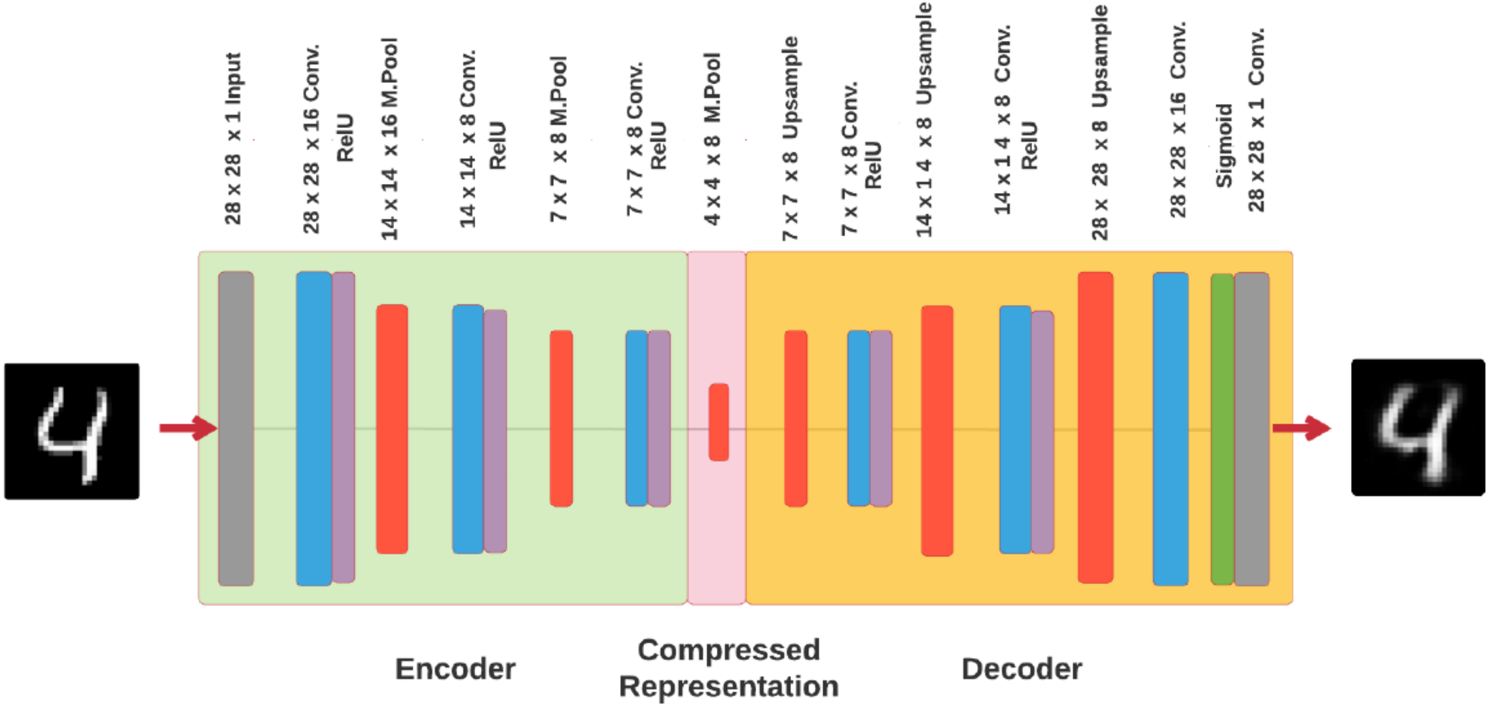
In biometrics, they can:

- Enhance face image quality for recognition.
- Generate synthetic identities for privacy-preserving applications.

12. Explainable AI

- Autoencoders provide insight into how an image is reconstructed, highlighting the features that are most critical for generation or recognition. This can assist in creating interpretable Generative AI models.

Lab Assignment: CAE



Dr. Praveen Kumar Chandaliya
Department of Artificial Intelligence
pkc@aid.svnit.ac.in
SVNIT, Surat, India
https://scholar.google.com/citations?user=cx-vENIAAAAJ&hl=en