#DemocratizeAI

fuse | machines

Representation Learning

AI Fellowship

Chapters:

1. Generative Models -Roshil
2. Contrastive Learning Models -Roshil
3. Self-Supervised Learning Models -Muskan
4. Other Models for Representation Learning -Muskan

Overview

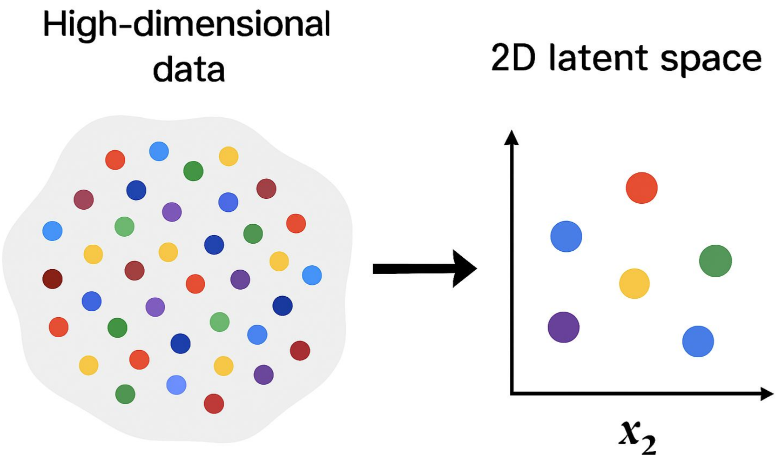
Overview

What is Latent Space?

A vector space where high dimensional input data is compressed into lower dimensional representations

Key Points:

- Captures essential features of the input data
- Used in models like autoencoders, VAEs, and GANs
- Enables efficient data processing and generation



Why & Where Is Latent Space Used?

Main Idea: Compress + Understand + Create

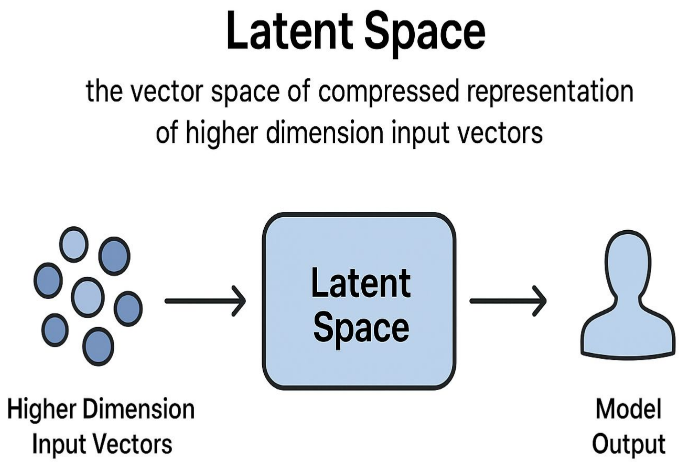
Purposes:

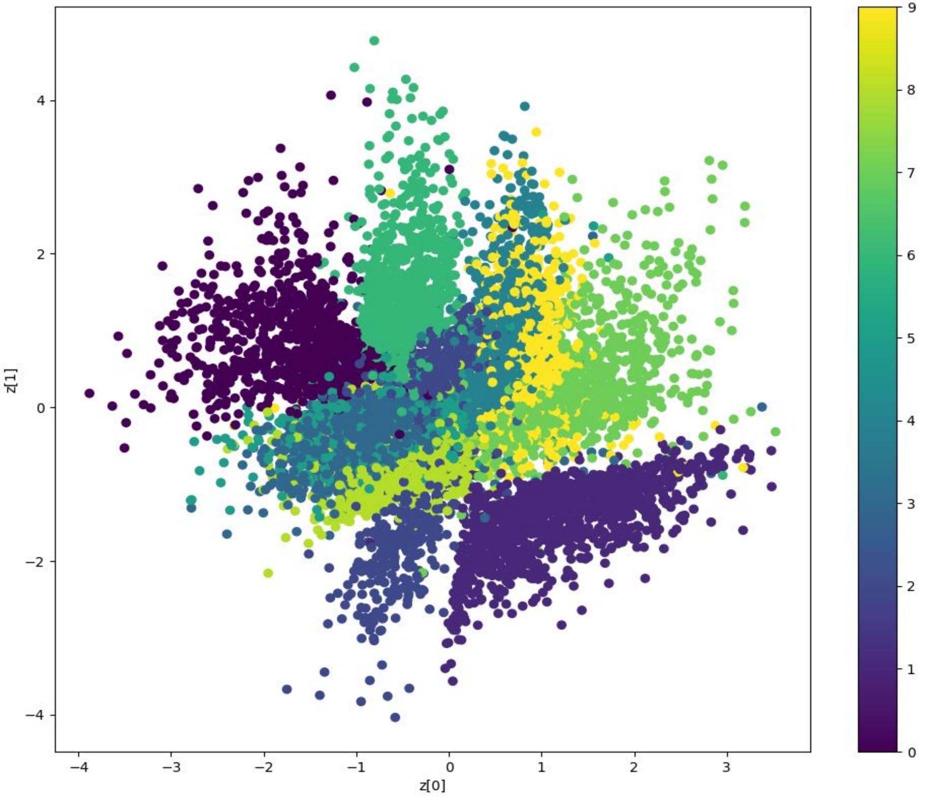
- **Data Compression –** Store big data in compact form.
- **Feature Extraction –** Find important patterns automatically.
- **Data Generation –** Create new, realistic data.
- **Pattern Recognition –** Reveal hidden relationships.

Common Uses:

- Image Generation & Editing (GANs, VAEs)
- Anomaly Detection (fraud detection, unusual data)
- Language Processing (word embeddings, NLP)
- Recommendation Systems (Netflix, Amazon)
- Medical Imaging (tumor detection, MRI analysis)

Latent Space is the vector space of compressed representation of higher dimension input vectors.





Introduction to Raw Data and Vector Representation in Machine Learning

What is Data?

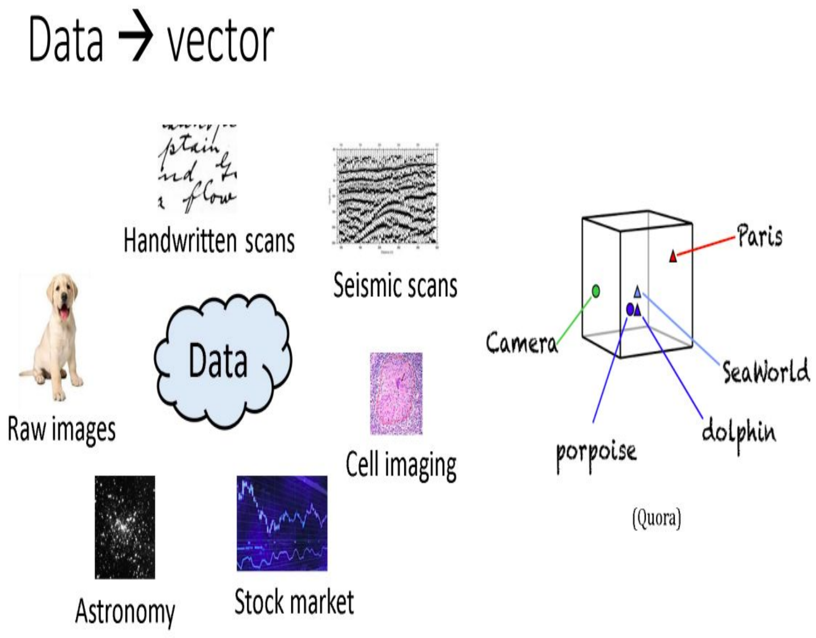
- Information collected from the real world (numbers, text, images, audio, etc.).

What is Raw Data?

- Unprocessed form of data (e.g., an image file, text document, or sensor reading).
- Needs cleaning & transformation before use in ML.

Why Represent as Vectors?

- ML models work with numbers.
- Vectors = structured numeric form of raw data.
- Enables computation, pattern detection, and learning.



Overview

1 Latent Space

2 Representation Learning

Representation Learning

Definition:

- Core ML concept: models automatically learn how to represent raw data (image, text, audio).
- Makes tasks like classification, clustering, search, prediction easier.

Traditional Approach:

- Manual feature engineering by humans.
  - Image → edges, corners, color patterns.
  - Text → keywords, n-grams.

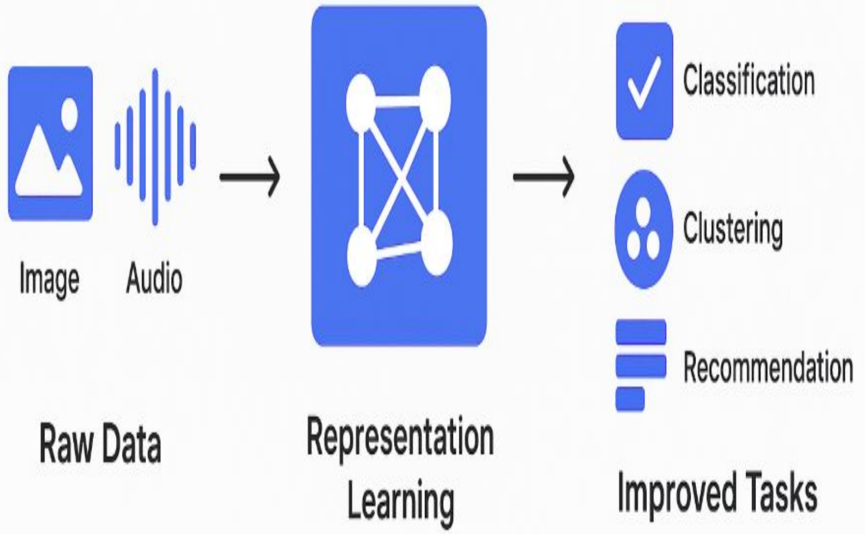
Representation Learning Approach:

- Model automatically discovers important patterns, structures & relationships.
- Leads to better performance & less manual effort.

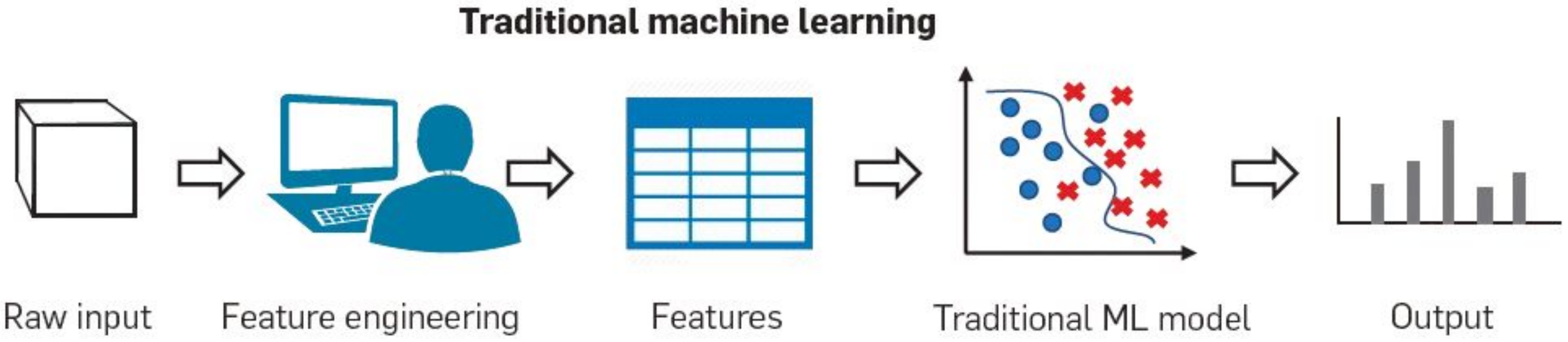
Key Idea: Models learn useful features directly from data.

Why is Representation Learning Important?

- **Handles Complex Data :**
  - Works with images, audio, and text where manual feature design is difficult.
  - Learns latent features automatically for efficient representation.
- **Better Task Performance :**
  - Improves accuracy in classification, clustering, recommendation, translation, and image recognition.
- **Reduces Human Effort**
  - Minimizes need for manual feature engineering.
  - Saves time & effort through automatic feature extraction.



Traditional Machine Learning: Manual Feature Engineering



- In traditional machine learning, **features were manually handcrafted** using domain expertise. The success of models depended heavily on the quality of these features.
- **Traditional ML = Human handcrafted features → Model**

Traditional Machine Learning: Manual Feature Engineering

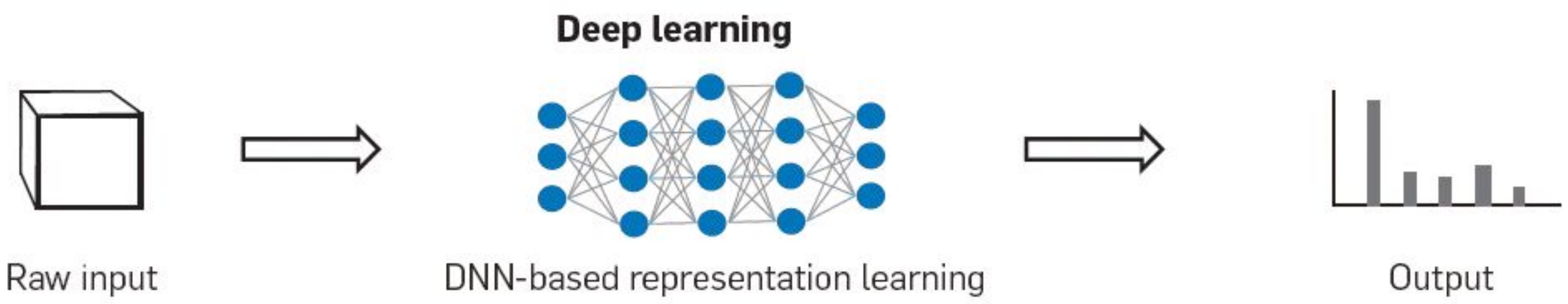
How it worked (Before Deep Learning):

- **Domain Expertise**: Experts needed deep understanding of data (finance, healthcare, images, text).
- **Manual Feature Selection**: Useful features extracted by hand.
- **Preprocessing Pipelines:** Scaling, normalization, missing value handling, one-hot encoding were done manually.
- **Model Success = Feature Quality:**
  - Good features → Model performs well.
  - Poor features → Model fails, even with advanced algorithms.

What is Feature Engineering?

- Feature engineering in machine learning is the process of **transforming raw data into meaningful features** that improve model performance.
- It involves **selecting, creating, transforming variables (features), and extracting data features** to improve model performance
- **Goal: Make data more predictive, understandable, and model friendly.**

Deep Learning for Feature Extraction



- Deep learning models **automatically learn useful features** from raw data using layers of neural networks a process called representation learning.
- No need for manual feature engineering the model finds the best representations during training.

How Deep Learning Works

Layered Structure:

- Data passes through multiple layers.
- Each layer learns different feature levels.

  Images → Edges → Shapes → Objects

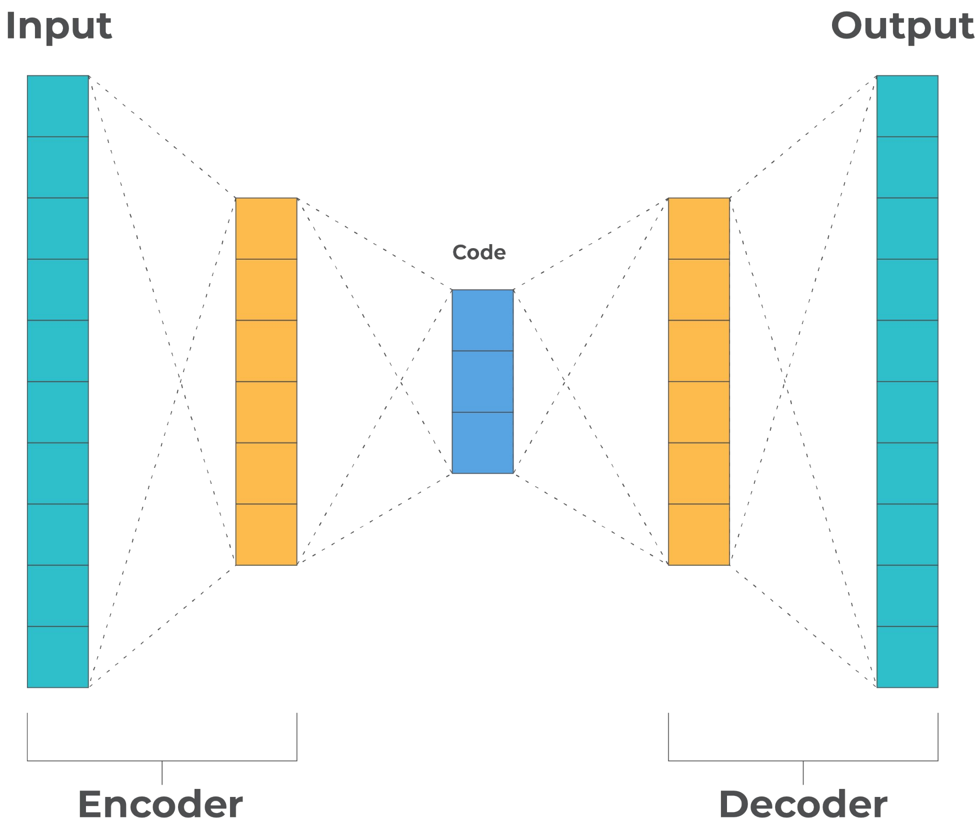
Automatic Feature Learning:

- Learns patterns & correlations automatically (no manual design).

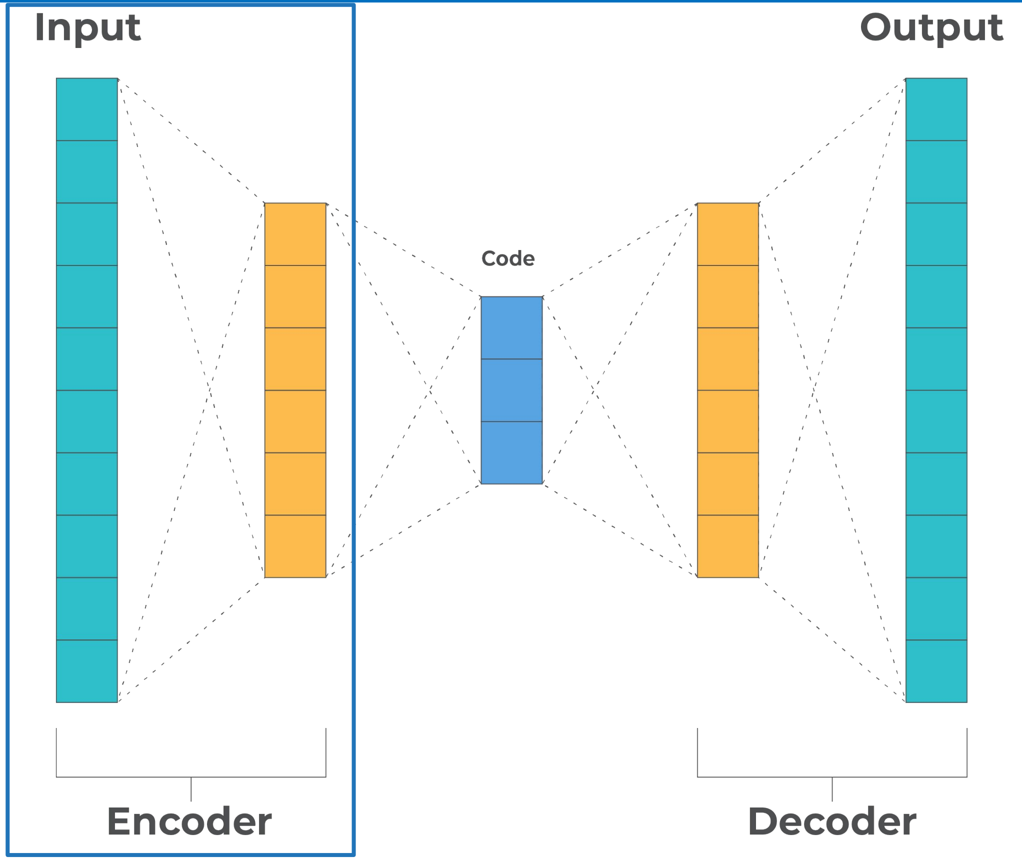
End-to-End Training:

- Raw input → Neural Network → Output
- Features + Task learned together in one process.

Autoencoders and its Components



Autoencoders and its Components



Encoder

Position in Network:

- Found on the left side of the architecture.
- First step in transforming raw input into a compact representation.

Role:

- Converts raw data → compressed, meaningful features.
- Acts as an information filter, keeping important patterns while removing noise.

Step-by-Step Compression:

- Data flows through successive hidden layers.
- Each layer extracts higher-level features while reducing dimensionality.
  - **Example**: 784 pixels (28×28 image) → 256 features → 64 features → Latent Space.

Encoder

Why Needed?

- Raw data (images, audio, text) is often high-dimensional, noisy, and redundant.
- Neural networks struggle with direct raw inputs because they contain unnecessary details.
- The encoder simplifies data into a smaller, more structured representation.
- This compressed representation makes learning faster, more efficient, and more accurate.
- Encoded data can be easily used for transfer learning (reusing knowledge in new tasks).
- Encoders also help in data visualization, making hidden structures in data visible.

Encoder

Mathematical View:

- The encoder is a mapping function:

$$z=f(x)$$

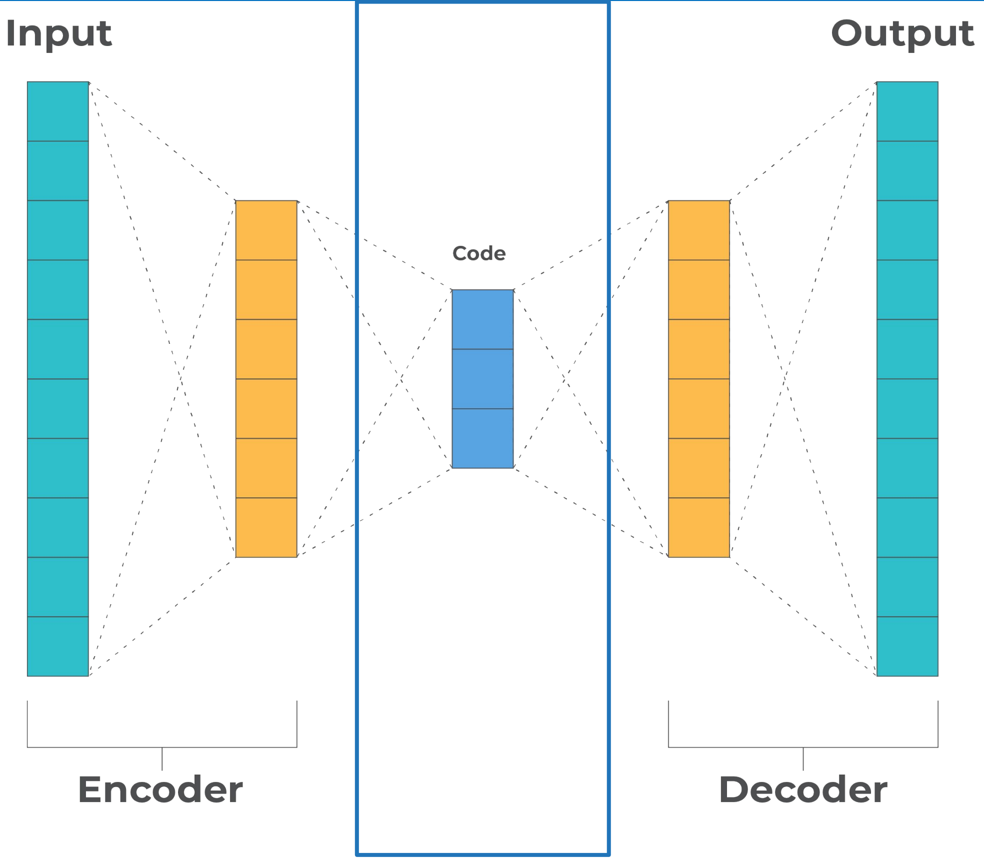
Where

x = Original input data (image, audio, text)

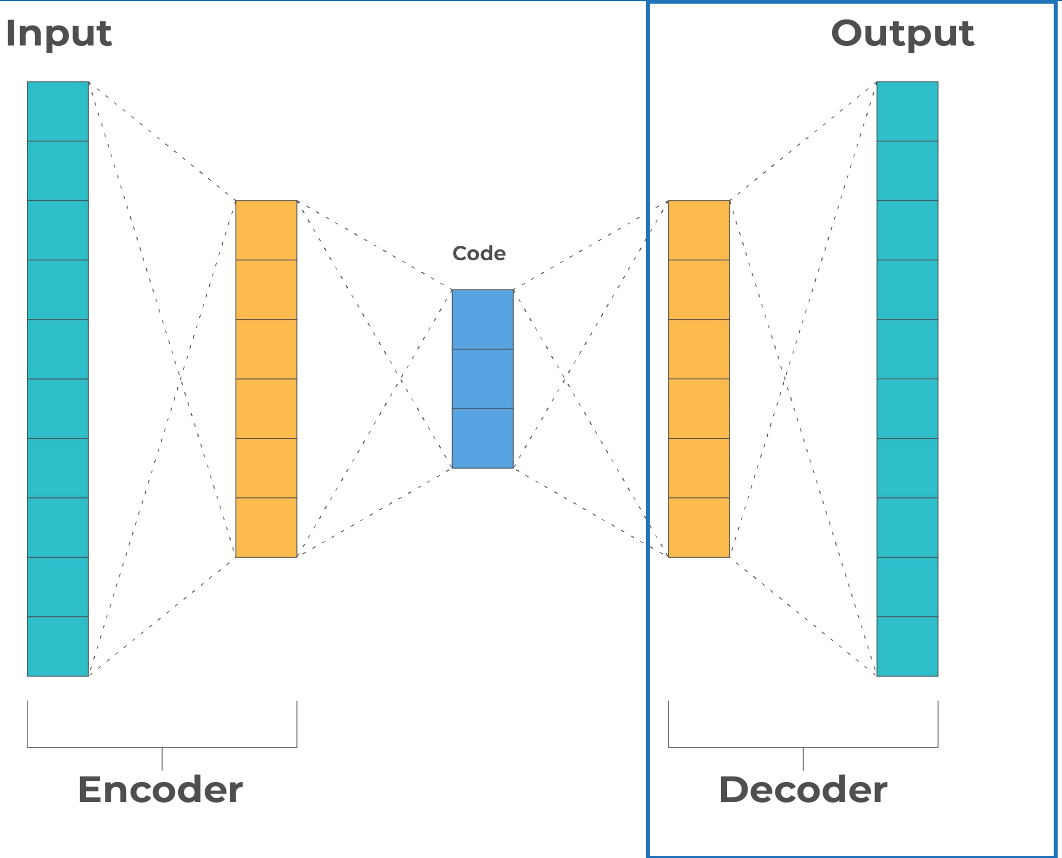
f(x) = Encoder function (neural network layers)

z = Latent vector → compressed representation, smaller dimension

Autoencoders and its Components



Autoencoders and its Components



Properties of Autoencoders

- Unsupervised Learning
- Dimensionality Reduction
- Non-linear Transformations
- Feature Learning

Types of Autoencoders

Based on Structure

- Undercomplete Autoencoder
- Overcomplete Autoencoder

Based on Functionality

- **Sparse Autoencoder**
- **Convolutional Autoencoder**
- **Variational Autoencoder**

Key Takeaways

- Latent space = a compressed hidden representation of complex input → which makes machines faster, smarter, and more creative.
- Representation Learning is learning representation of the input data by extracting features from it.
- Representation Learning has been successfully used in various domains.