Variational Autoencoders (VAEs)

#Why VAEs?

- They're another **generative model** (like GANs), but instead of *adversarial training*, they use **probabilistic latent spaces**.
- They're actually the foundation for **diffusion models** and even some **LLM latent structures**.
- Easier to train than GANs, though images look blurrier.

1. What is a VAE?

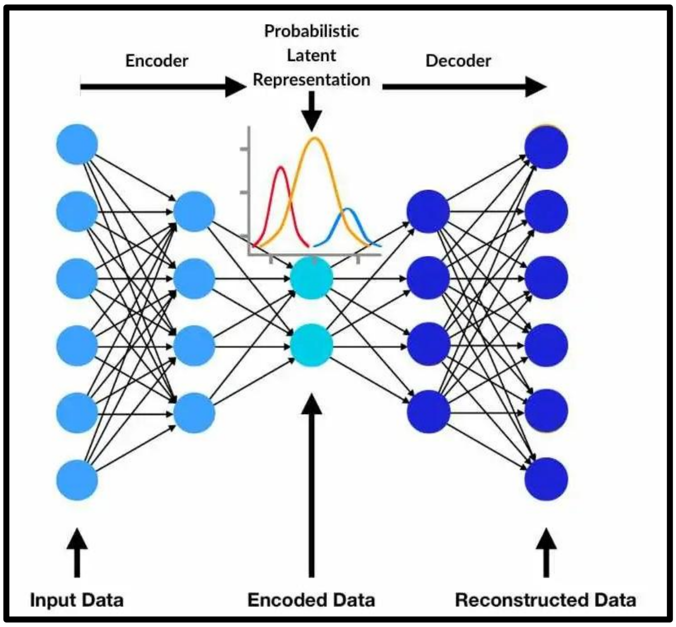
- An **Autoencoder** = Encoder (compresses input) + Decoder (reconstructs input).
- A **Variational Autoencoder** = Adds probability → instead of encoding an image into a single vector, it encodes it into a **distribution (mean + variance)**.
- Then samples from this distribution to decode.
- Result: You don't just reconstruct images → You can **generate new ones** by sampling from latent space.

2. Structure of a VAE:

- **Encoder**: Takes input image → outputs latent mean (μ) and variance (σ).
- **Latent Sampling**: Sample $z \sim N(\mu, \sigma^2)$ using **reparameterization trick**.
- **Decoder**: Takes z → reconstructs an image.

Loss Function =

1. **Reconstruction Loss** (how close generated image is to original).
2. **KL Divergence** (regularizes latent space to look like Gaussian distribution).



3. VAE vs GAN:

Feature	VAE	GAN
Training	Stable, easy	Tricky, unstable
Output Quality	Blurry	Sharp, realistic
Latent Space	Structured (useful for interpolation)	Hard to interpret
Use Cases	Data compression, representation learning, anomaly detection	Image synthesis, art, deepfakes

4. Practical Uses:

- **Image generation** (like GANs, but more structured).
- **Anomaly detection** (e.g., fraud, medical images).
- **Data interpolation** (e.g., morphing between two faces).
- **Representation learning** for downstream tasks.

5. How VAEs Work:

VAEs combine ideas from traditional autoencoders and variational inference:

1. Encoder:

The encoder maps the input data (e.g., an image) to a continuous probability distribution, typically represented by a mean and a variance, rather than a single fixed point. This probabilistic approach creates a flexible latent space where similar data points are clustered.

2. Latent Space:

A random sample is taken from this learned distribution in the latent space. This randomness is key to the VAE's ability to generate novel data.

3. Decoder:

The decoder takes the sampled latent vector and reconstructs it back into the original data space, aiming to produce an output that is similar to the input.

4. Probabilistic Loss:

The VAE's loss function includes two parts:

- A reconstruction loss that measures how well the decoder reconstructs the original input.
- A KL divergence loss that forces the learned distribution in the latent space to be close to a standard prior distribution (like a Gaussian distribution), ensuring it has a useful structure for generation.