Autoencoders

Peida Wu

IMS

December 30, 2025

Neural Networks: Concept and Structure

- **What are Neural Networks?**
  - Computational models inspired by the brain.
  - Learn patterns from data (e.g., classification, feature extraction).
- **Structure**:
  - Layers: Input $\rightarrow$ Hidden $\rightarrow$ Output.
  - Nodes connected by weights.
- **Example**:
  - Input pixels of pictures, the neural network can make classification(e.g. cat or dog)
  - Input some time series, the neural network can make forecast()

Mathematical Foundation

- **Forward Propagation**:
  - Hidden layer: $\mathbf{h} = f(W\mathbf{x} + \mathbf{b})$
  - W: weight matrix, $\mathbf{b}$: bias, f: activation (e.g. ReLU $f(z) = \max(0, z)$, Sigmoid)
- **Optimization**:
  - Loss function: e.g. $L = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$, $L = \sum \hat{y}_i \log \hat{y}_i$
  - Minimize via (stochastic) gradient descent.

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta),$$

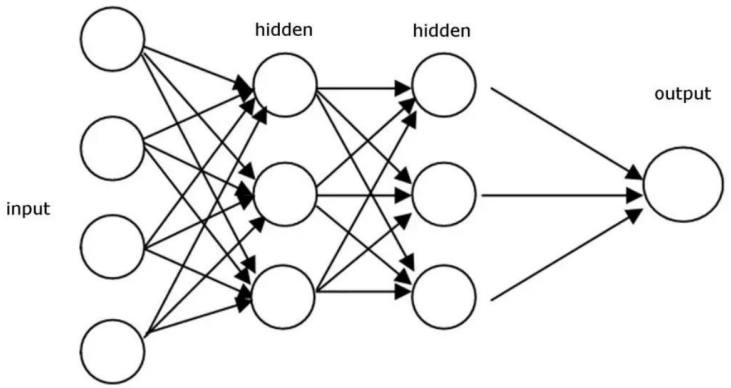


Figure 1: Neural network

Motivation of Autoencoder

Why Autoencoders?

- **High-dimensional data** are everywhere: images, time series, functional data.
- Direct computation on raw data can be inefficient, noisy, or redundant.
- Goal: learn a **compact latent representation** z that captures the essential information.
- Provides a nonlinear generalization of classical dimensionality reduction (e.g., PCA).

Key Idea

- Use a neural network to **compress** data (encoder) and then **reconstruct** it (decoder).
- If reconstruction is accurate, z preserves the most important structure of x.

Autoencoder Structure & Loss

Basic Structure

- **Encoder:** $z = f_\theta(x)$
- **Latent space:** low-dimensional representation
- **Decoder:** $\hat{x} = g_\phi(z)$

Optimization Objective

$$\min_{\theta,\phi} \; L(\theta,\phi) = \frac{1}{n}\sum_{i=1}^{n} \left\| x_i - g_\phi(f_\theta(x_i)) \right\|^2$$

(Mean squared reconstruction error)

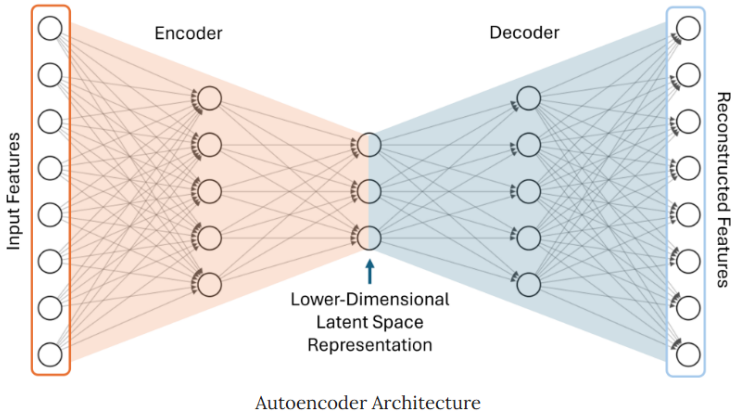


Figure 2: Encoder → Latent → Decoder

Autoencoder vs. PCA

Connection:

- A linear autoencoder with one hidden layer and mean squared error loss learns the same subspace as PCA.
- Both minimize reconstruction error:

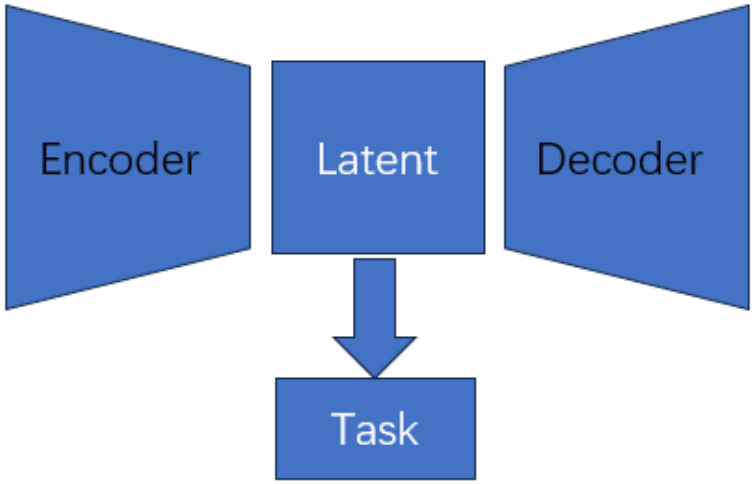
$$\min_{W,V} \|X - VWX\|_F^2 \quad \Rightarrow \quad W \text{ spans the top-}k \text{ principal components.}$$

Comparison:

	PCA	**Autoencoder**
Model	Linear projection	Neural network
Solution	Closed-form via SVD	Optimized via GD
Flexibility	Only linear	Nonlinear, deep
Objective	Maximize variance / minimize reconstruction error	Minimize reconstruction error (can add extra terms)
Output	Principal components	Latent code

Practical Uses of Autoencoders

General Idea: Learn a latent representation z that is useful for many downstream tasks.



Examples:

- **Dimensionality reduction**: project to 2D/3D latent space (e.g. MNIST).
- **Clustering**: DEC learns better clusters from latent codes.
- **Anomaly detection**: reconstruct normal patterns; large reconstruction error $\Rightarrow$ anomaly.

Variants of Autoencoders

1. Denoising Autoencoder (DAE) Add noise to input $\tilde{x}$, train to reconstruct clean x:

$$L = \|x - g_\phi(f_\theta(\tilde{x}))\|^2$$

Learns robust features, denoises data, captures manifold structure.

2. Sparse Autoencoder Encourage sparse activations via regularization:

$$L = L_{\text{recon}} + \beta \sum_j \mathrm{KL}(\rho \,\|\, \hat{\rho}_j)$$

Each hidden unit specializes, useful for feature learning and classification.

3. Variational Autoencoder (VAE) Learn distribution $q_\phi(z|x)$, maximize ELBO:

$$\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - KL(q_\phi(z|x) \,\|\, p(z))$$

Smooth latent space, supports sampling and generative modeling.

Some Limitations of Autoencoders

1. Focus on Pixel-wise Reconstruction.
2. Risk of Learning Identity Mapping.
3. Unstructured Latent Space.
4. Sensitivity to Outliers and Noise.
5. Difficulty with High-Dimensional and Complex Data.

Motivation and Intuition

Motivation: Why not plain Autoencoders?

- Autoencoders learn compressed latent codes, but:
  - No clear **probabilistic interpretation**.
  - Latent space is **irregular**, hard to sample new data.
  - Cannot generate realistic new samples beyond training data.
- We want:
  - A **probabilistic model** of data.
  - A smooth latent space where sampling is meaningful.

Intuition of VAE

- Treat latent variable z as drawn from a prior $p(z)$ (e.g. $\mathcal{N}(0, I)$).
- Encoder learns a distribution $q_\phi(z|x)$, not just a point.
- Decoder generates x from z via $p_\theta(x|z)$.
- Training: make $q_\phi(z|x)$ close to true posterior $p(z|x) \Rightarrow$ optimize Evidence Lower Bound (ELBO).

Visual Explanation

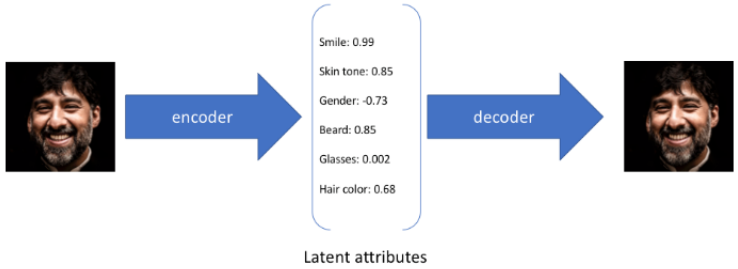


Figure 3: AE Face

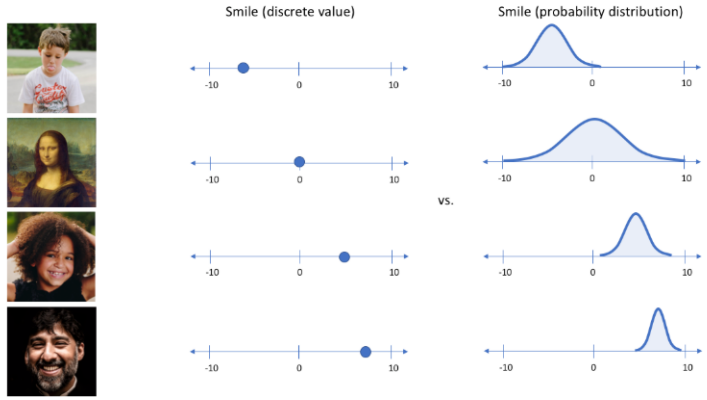


Figure 5: AE vs VAE

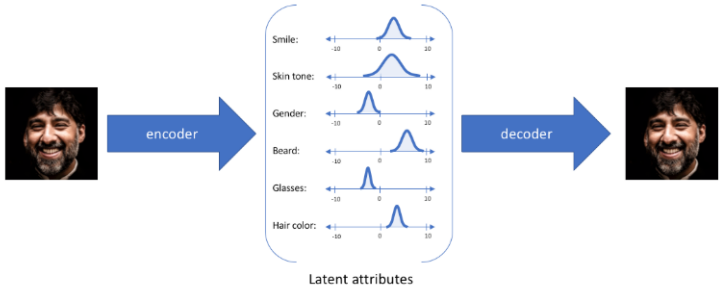


Figure 4: VAE Face

Variational Inference: Why and What

Problem. Posterior is intractable:

$$p(z \mid x) = \frac{p(x,z)}{p(x)}, \qquad p(x) = \int p(x,z)\, dz \text{ (often intractable).}$$

Idea. Introduce a tractable family $q_\phi(z)$ (or $q_\phi(z \mid x)$) to approximate $p(z \mid x)$ by minimizing

$$D_{\mathrm{KL}}\big(q_\phi(z) \,\|\, p(z \mid x)\big).$$

Key identity.

$$\log p(x) = \underbrace{\mathbb{E}_{q_\phi}\big[\log p(x,z) - \log q_\phi(z)\big]}_{\text{ELBO}} + D_{\mathrm{KL}}(q_\phi(z) \,\|\, p(z \mid x)).$$

Since $D_{\mathrm{KL}} \geq 0$,

$$\log p(x) \;\geq\; \text{ELBO} \;=\; \mathbb{E}_{q_\phi}\big[\log p(x,z)\big] - \mathbb{E}_{q_\phi}\big[\log q_\phi(z)\big].$$

Optimize ELBO $\Rightarrow$ q_ϕ approaches $p(z \mid x)$ while avoiding $p(x)$.

How to connect VI to VAE

ELBO Derivation Steps

$$\mathcal{L}_q = E_{z\sim q_\varphi}[\log p_\theta(x|z) + \log p(z) - \log q_\varphi(z|x)]$$
$$= E_{z\sim q_\varphi}[\log p_\theta(x|z)] - E_{z\sim q_\varphi}\left[\log \frac{q_\varphi(z|x)}{p(z)}\right]$$
$$= E_{z\sim q_\varphi}[\log p_\theta(x|z)] - (q_\varphi(z|x)\|p(z))$$

Interpretation of the Two Terms

- **Term 1:** $E_{z\sim q_\varphi}[\log p_\theta(x|z)]$
  - **Reconstruction Likelihood** (Maximize this)
- **Term 2:** $(q_\varphi(z|x)\|p(z))$
  - **Latent Prior Similarity** (Minimize this)
  - Common prior: $p(z) = \mathcal{N}(0, I)$

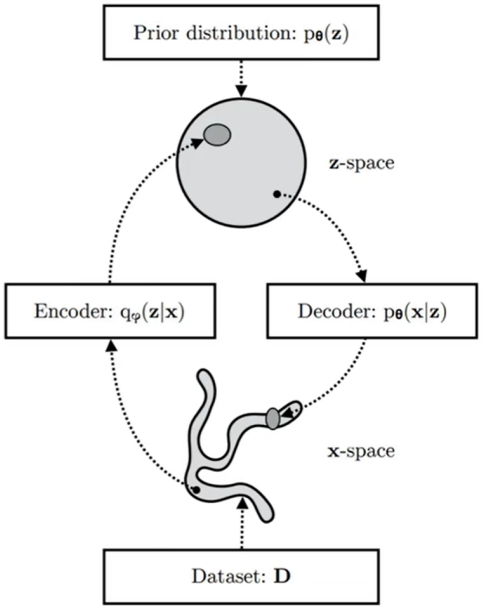


Figure 6: Progress

KL divergence between Gaussians

Setup.

$$q_\varphi(z \mid x) = \mathcal{N}(z; \mu_\varphi, \Sigma_\varphi), \quad p(z) = \mathcal{N}(z; 0, I), \quad k = \dim(z).$$

KL definition.

$$D(q_0 \| q_1) = \mathbb{E}_{q_0}[\log q_0(z) - \log q_1(z)].$$

Key identities.

$$\mathbb{E}_{q_0}[(z-\mu_0)(z-\mu_0)^\top] = \Sigma_0,$$
$$\mathbb{E}_{q_0}[(z-\mu_1)(z-\mu_1)^\top] = \Sigma_0 + (\mu_0-\mu_1)(\mu_0-\mu_1)^\top,$$
$$\mathbb{E}[x^\top A x] = \mathrm{tr}(A\,\mathbb{E}[xx^\top]).$$

General closed form.

$$D(\mathcal{N}(\mu_0,\Sigma_0) \,\|\, \mathcal{N}(\mu_1,\Sigma_1)) = \frac{1}{2}\Big(\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) + (\mu_1-\mu_0)^\top \Sigma_1^{-1}(\mu_1-\mu_0)$$

$$- \mathrm{k} + \log \frac{\det \Sigma_1}{\det \Sigma_0} \Big).$$

Reconstruction Loss by Gaussians

$$p_\theta(x|z) \sim N(x; \mu_\theta(z), \sigma^2 I)$$

$$\log p_\theta(x|z) = -c||x - \mu_\theta(z)||^2 + d$$

That's why the reconstruction likelihood seems like MSE.
Monte Carlo Sampling is used to estimate the $E_{z\sim q}(\log p_\theta(x|z))$, but it'll bring uncertainty, leading to fialure to backpropagate.

Reparameterization

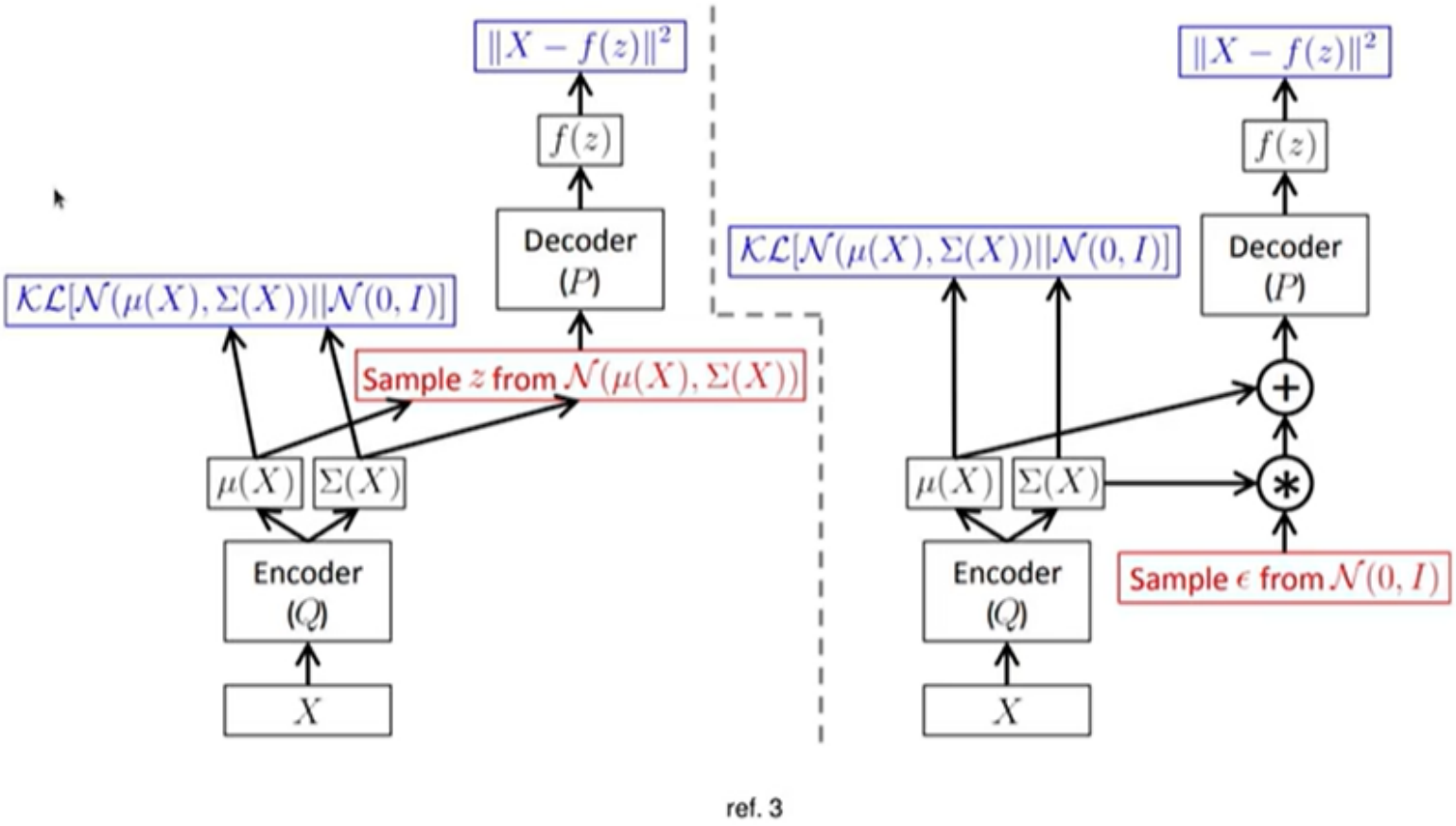


Figure 7: Reparameterization

Reference

Doersch, C., 2016. Tutorial on variational autoencoders. arXiv preprint arXiv:1606.05908.
Kingma, D.P. and Welling, M., 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Thanks!