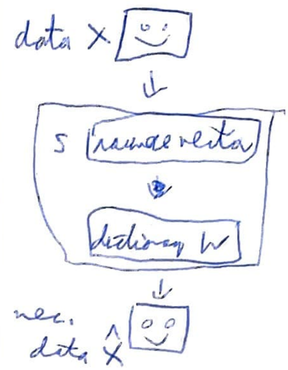
Lecture 4 – Autoencoder

~~Advantages~~ Sparse Coding

→ Vector with many 0-elements

→ reconstruction with dictionary W



Advantages Sparse Coding

low storage cost: $S = (12, 0, 7, 0, 0) \rightarrow S = \{0{:}12, 2{:}7\}$

Interpretability: empirically, data representation usually meaningful

Linear Sparse Coding

- dataset: $x_1, \dots, x_N \in \mathbb{R}^d$
- representation: $S_1, \dots, S_N \in \mathbb{R}^h$
- dictionary: $W \in \mathbb{R}^{d \times h}$

Objective:

$$\min_{W, S_1, \dots, S_N} \frac{1}{N} \sum_{i=1}^{N} \Big[\underbrace{\|x_i - W s_i\|^2}_{\text{reconstruction}} + \underbrace{\lambda \|S_i\|_0}_{\text{sparsity}} \Big]$$

$$\|\cdot\|_0 = \Big[\sum_i |x_i|^p\Big]^{1/p} = \sum_i |x_i|^0 \leftarrow \text{counts non-zero elements}$$

Problem: $\|S_i\|_0$ is not differentiable and nonconvex

Better Objective

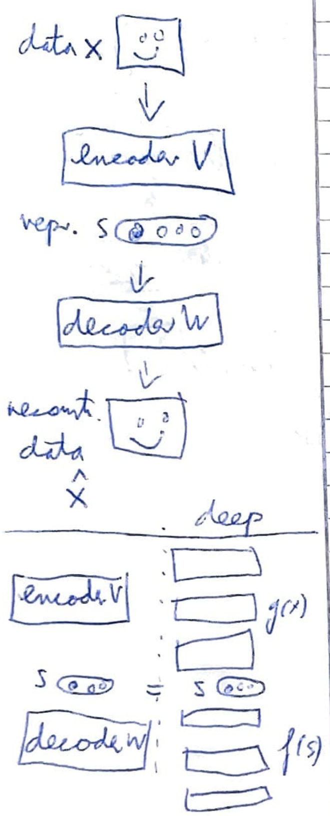
$$\min_{W, S_1, \dots, S_N} \frac{1}{N} \sum_{i=1}^{N} \Big[\|x_i - W s_i\|^2 + \underbrace{\lambda \|S_i\|_1}_{\text{penalizes non-zero rep.}} \Big] + \underbrace{\eta \|W\|_F^2}_{\text{penalizes large weights}}$$

$$\|W\|_F = \sqrt{\sum_{ij} |W_{ij}|^2}$$

$$\|W\|_F^2 = \sum_{ij} |W_{ij}|^2$$

Advantage: $\|\cdot\|_1$ is convex and almost differentiable

$\|\cdot\|_1$ does not yield maximally sparse solution as $\|\cdot\|_0$ but still very sparse solution



Autoencoders

Learn a function (matrix) V that sparsely encodes the data x_i: $s_i = V x_i$

New objective:

$$\min_{W,V} \frac{1}{N}\sum_{i=1}^{N} \left[\|x_i - WVx_i\|^2 + \lambda \|Vx_i\|_1 \right] + \eta \|W\|_F^2$$

W and V matrices can be learned via (stochastic) Gradient Descent

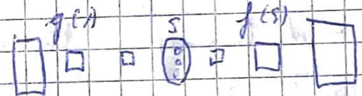
Deep Autoencoder

Same as autoencoder but more complex coding and decoding functions.

$WVx_i \to f \circ g(x_i)$

Denoising Autoencoder: apply noise to data → deep autoencoder learns noise-robust representation

Bottleneck Autoencoder:



Middle layer ~~same~~ small → only learns most essential elements

Self-supervised Learning: train autoencoder on colorizing images, then use it to identify objects

Transfer Learning: Train model on difficult ~~task~~ task (many classes) and use model for another simpler task

Summary

- as PCA, CCA, ICA, sparse coding and autoencoders learn representations from unsupervised data
- can use deep learning features: convolution, pooling, etc.
- purposes: produce compact (sparse) representations, invariant representations, abstract representations, data transformation