Tuesday, March 31

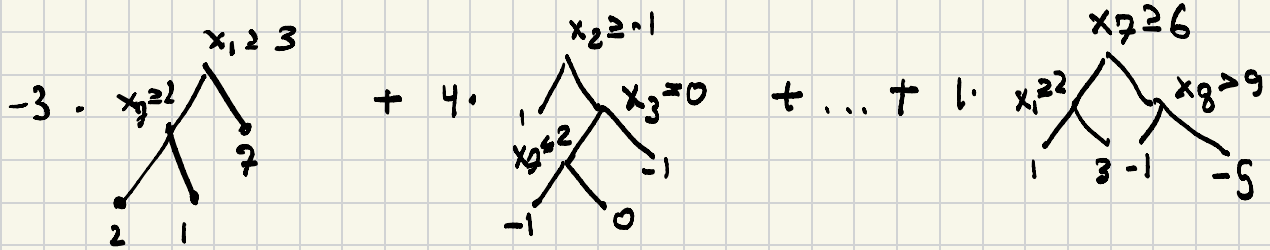
- Midterm 4/23
- Exam/project prep 4/21

- James Enouen (USC)
  - ↳ Estimating Shapley values (my research interest)
  - ↳ Thursday 12pm in Roberts North 15

Today

- Autoencoders!! :)

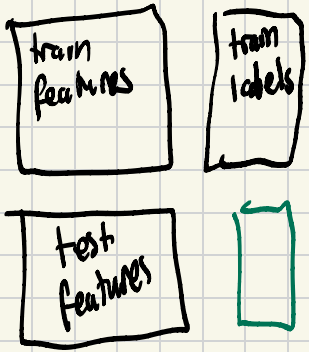
Gradient Boosting: $F_t = \alpha_1 f_1 + \alpha_2 f_2 + \dots + \alpha_t f_t$



Gradient boosting was SOTA on tabular data from 2014-2025

TabPFN and friends (Transformer for tabular data) is now SOTA

↳ in context learning



Learning

↳ Supervised
- classification
- regression

↳ Unsupervised
- autoencoders
- clustering

↳ Semi-supervised
- reinforcement learning

Question: When our data doesn't have labels, what can we do with it?
- observations
- images
- text

Answer: Use the data as its own label

Autoencoder

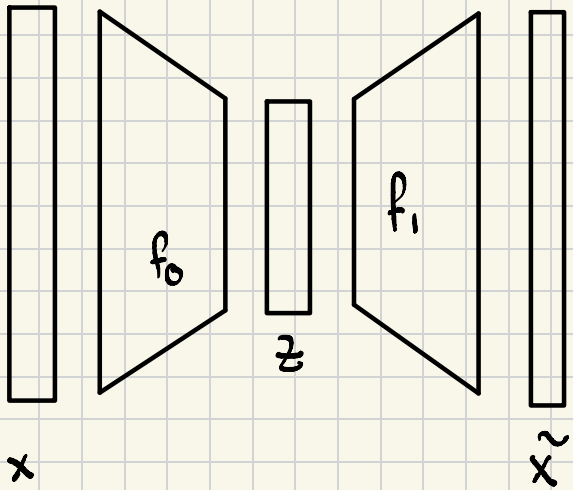
Encoder: $f_0 : \mathbb{R}^d \to \mathbb{R}^k$

Decoder: $f_1 : \mathbb{R}^k \to \mathbb{R}^d$

Input: $x \in \mathbb{R}^d$

Embedding: $z = f_0(x)$

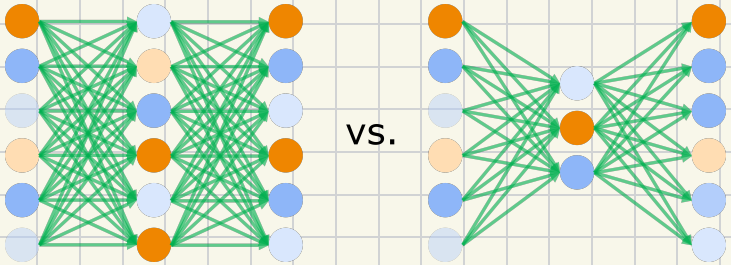
Output: $\tilde{x} = f_1(z) = f_1(f_0(x))$



Reconstruction Loss:

$$\mathcal{L}_{recon} = \frac{1}{n} \sum_{i=1}^{n} \| x^{(i)} - \tilde{x}^{(i)} \|_2^2$$

Q: How should we choose k?



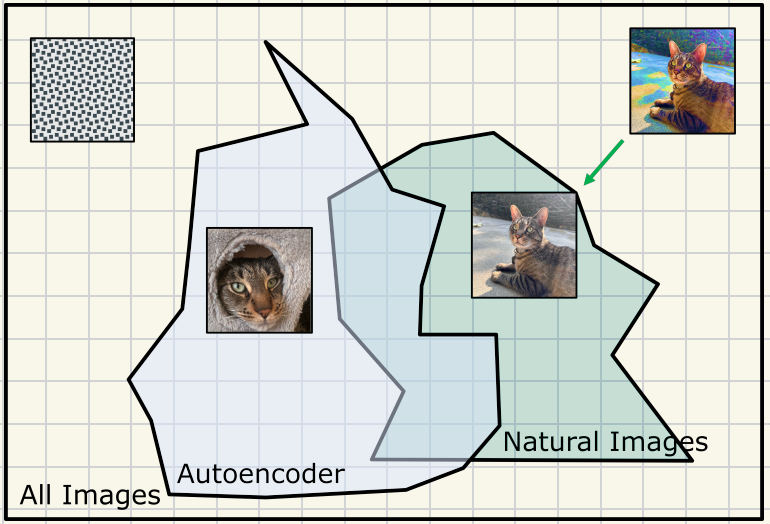
Autoencoder applications:

- data compression
- denoising
- inpainting
- representation learning

↑ depend on k rather than d

Manifold Perspective

Since $k \leq d$, we lose info.
But this is ok, because we're only interested in a manifold (subset) of $\mathbb{R}^d$



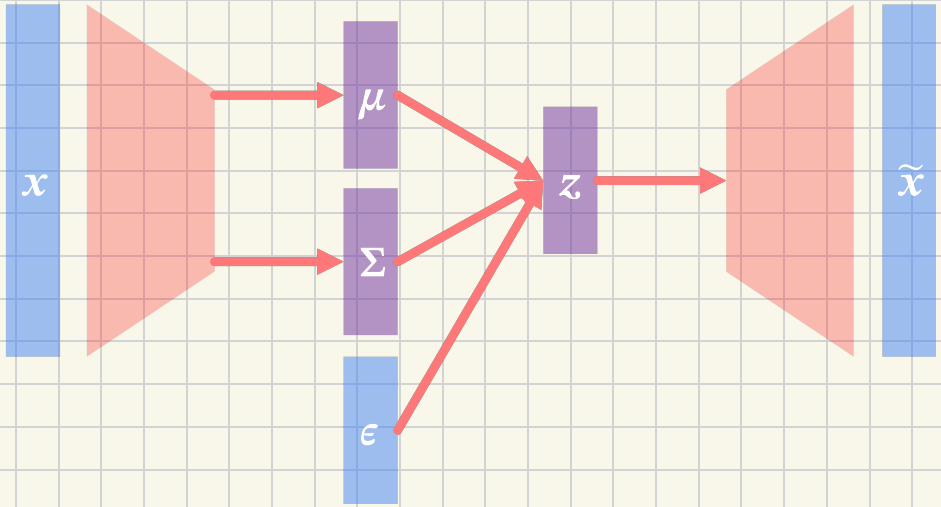
Variational Autoencoders

Goal: Evenly distribute embeddings $z = f_\theta(x) \in \mathbb{R}^k$, i.e.,

$z \sim \mathcal{N}(\mu_x, \Sigma_x)$ where

mean $\mu_x \in \mathbb{R}^k$ depends on x

variance $\Sigma_x \in \mathbb{R}^{k \times k}$ depends on x



$f_\theta : \mathbb{R}^d \to \mathbb{R}^k \times \mathbb{R}^{k \times k}$

$\epsilon \sim \mathcal{N}(0, I) \in \mathbb{R}^k$

$\mu_x, \Sigma_x = f_\theta(x)$

$z = \mu_x + \Sigma_x^{1/2} \epsilon$

Variational Loss

tension!

reconstruction error + distribution distance

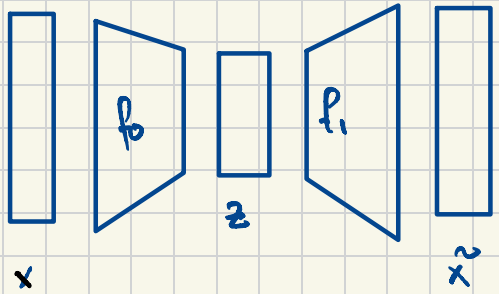
$$\frac{1}{n}\sum_{i=1}^{n} \| x^{(i)} - \tilde{x}^{(i)} \|_2^2 + \alpha \frac{1}{n} \sum_{i=1}^{n} D_{KL}\left(\mathcal{N}(\mu_{x^{(i)}}, \Sigma_{x^{(i)}}) \,\|\, \mathcal{N}(0, I) \right)$$

Kullback-Leibler divergence measures "distance" between distributions

$$D_{KL}(P, Q) = \mathbb{E}_{z \sim P} \log \frac{P(z)}{Q(z)}$$

Thursday, April 2

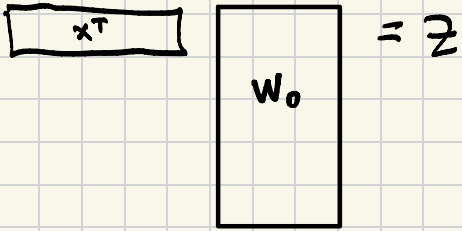
Autoencoder

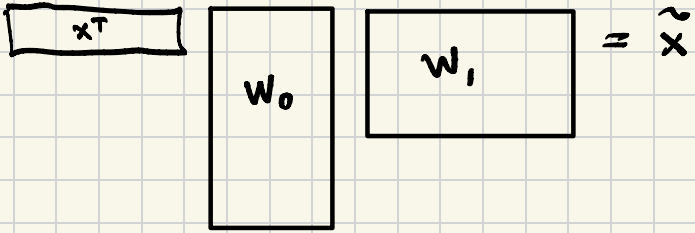


Principal Component Analysis

"linear regression" of autoencoders

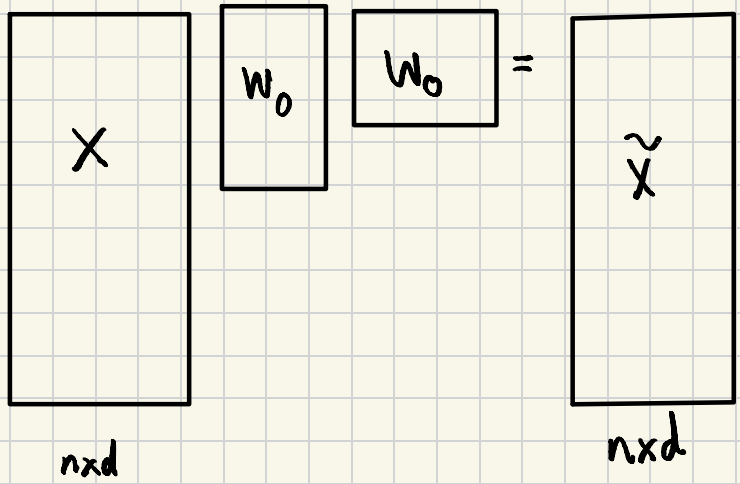
$f_0 = W_0 \in \mathbb{R}^{d \times k}$ $\quad f_1 = W_1 \in \mathbb{R}^{k \times d}$





Goal: $\min \sum_{i=1}^{n} \| x^{(i)} - \tilde{x}^{(i)} \|_2^2$

$= \min \sum_{i=1}^{n} \| x^{(i)} - x^{(i)T} W_0 W_1 \|_2^2$

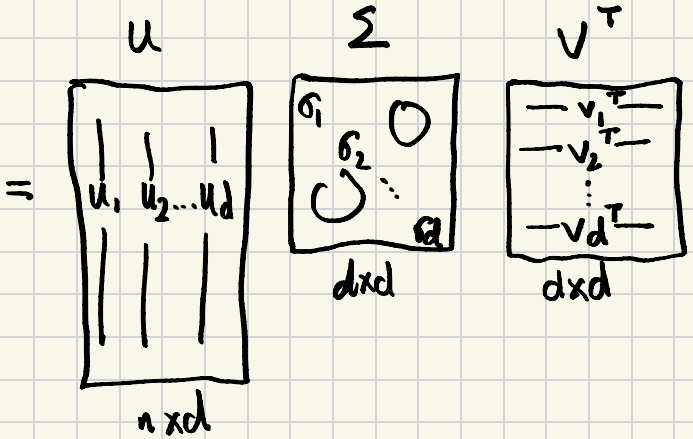


$= \min \| X - \tilde{X} \|_F^2$ ← sum of squared entries

Singular Value Decomposition

$X \in \mathbb{R}^{n \times d}$ $\quad n \geq d$ WLOG

$X = U \Sigma V^T$



outer product view

$$= \sum_{i=1}^{d} \sigma_i u_i v_i^T$$

left singular vectors $u_1, \ldots, u_d \in \mathbb{R}^n$ orthonormal

right singular vectors $v_1, \ldots, v_d \in \mathbb{R}^d$ orthonormal

singular values $\sigma_1 \geq \ldots \geq \sigma_d \geq 0$

rank k matrix iff

$$k = |\{\sigma_i : \sigma_i > 0\}|$$

rank k version of X is

$$X_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Eckart-Young-Mirsky

$$X_k = \underset{\text{rank-}k\ \tilde{X}}{\operatorname{argmin}} \|X - \tilde{X}\|_F^2$$

Tools

(A) Frobenius norm and trace: $\|X\|_F^2 = \operatorname{tr}(X^T X)$

sum of squared entries

sum of diagonal entries

$$\|X\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{d} [X]_{i,j}^2 = \sum_{i=1}^{n} \|[X]_{i,:}\|_2^2 = \operatorname{tr}(X^T X)$$

(B) Cyclic property: $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ where $A \in \mathbb{R}^{n \times d}$, $B \in \mathbb{R}^{d \times n}$

$$\operatorname{tr}(AB) = \sum_{i=1}^{n} [AB]_{i,i} = \sum_{i=1}^{n} \sum_{j=1}^{d} [A]_{i,j} [B]_{j,i}$$

$$= \sum_{j=1}^{d} \sum_{i=1}^{n} [B]_{j,i} [A]_{i,j} = \sum_{j=1}^{d} [BA]_{j,j} = \operatorname{tr}(BA)$$