Computational Intelligence Laboratory

Lecture 1
Linear Autoencoder

Thomas Hofmann

ETH Zurich – cil.inf.ethz.ch

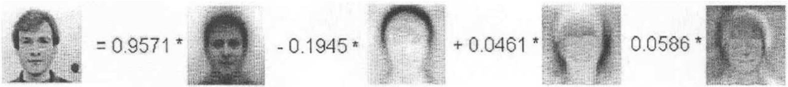
21 February 2020

Section 1

Dimension Reduction

Dimension Reduction

- Dimension reduction
  - given (high-dimensional) data points $\{\mathbf{x}_i \in \mathbb{R}^m\}$, $i = 1, \ldots, n$
  - find **low-dimensional representation** $\{\mathbf{z}_i \in \mathbb{R}^k\}$, $k \ll m$
- Example: face images
  - 2D pixel fields, e.g. $\mathbf{x}_i \in \mathbb{R}^{100 \times 100} \simeq \mathbb{R}^{10000}$ (vectorization)
  - approximate each image by weighted superposition of basis images



(from: Turk and Pentland, Eigenfaces for Recognition, 1991)

  - coefficients $= 4$-dimensional representation

Dimension Reduction: Motivation

- Motivation
  - visualization – e.g. 2D or 3D
  - data compression – fewer coefficients
  - signal recovery – discard irrelevant information (noise)
  - data analysis – discover intrinsic modes of variation
  - feature discovery – learn better representations
  - generative models – latent variables

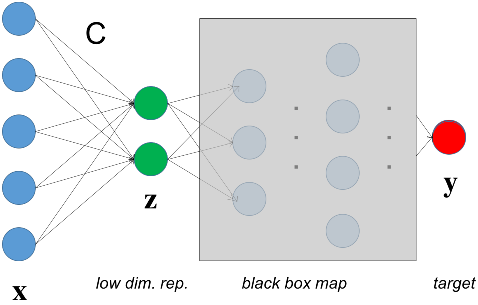
Linear Dimension Reduction

- **Linear** dimension reduction
  - $\mathbf{z}_i = \mathbf{C}\mathbf{x}_i$ for some (fixed) matrix $\mathbf{C} \in \mathbb{R}^{k \times m}$
  - generalizes to new data points: $\mathbf{C}$ represents linear map $\mathbb{R}^m \to \mathbb{R}^k$
  - each **feature** is a linear combination of input variables

$$\mathbf{z} = \mathbf{C}\mathbf{x} \iff z_r = \sum_{s=1}^{m} c_{rs} x_s \ (\forall r), \quad \mathbf{C} = (c_{rs})_{\substack{1 \le r \le k \\ 1 \le s \le m}}$$

  - neural network terminology: each z_r is a **linear unit**
    - computes a linear function of its inputs
    - with weight vector $\mathbf{c}_r = (c_{r1}, \ldots, c_{rm})^\top \in \mathbb{R}^m$ (r-th row of $\mathbf{C}$)

Dimension Reduction: Neural Network View

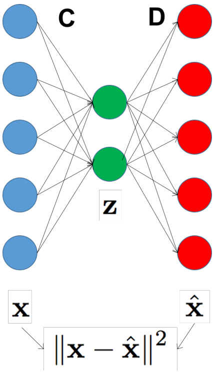


- Can think of this in terms of a (deep) neural network
- Optimize representations w.r.t loss defined over targets $\mathbf{y}$
- Supervised learning $\Longrightarrow$ backpropagation (subsequent lecture)
- Our interest here: **unsupervised learning**

Section 2

Linear Autoencoder

Linear Autoencoder



- Linear reconstruction map $\mathbf{D} \in \mathbb{R}^{m \times k}$
- Parameters $\theta = (\mathbf{C}, \mathbf{D})$ (coder/decoder)
- Use squared reconstruction loss

$$\ell(\mathbf{x}; \theta) = \frac{1}{2}\|\mathbf{x} - \hat{\mathbf{x}}(\theta)\|^2, \quad \hat{\mathbf{x}}(\theta) := \mathbf{DCx}$$

- Sample reconstruction error

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{x}_i; \theta)$$

  - goal: approximately learn identity map
  - only relative to data distribution
  - retrieve intermediate representation
- Fully unsupervised approach: $\mathbf{z}$ acts as a **bottleneck** layer

Low-Rank Approximation

- How can we interpret the linear auto-encoder?
  - it defines a linear map $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^m$ (as a matrix $\mathbf{F} := \mathbf{DC}$)
  - ideally: $\mathbf{F} \approx \mathbf{I}$ (approx. identity), but: bottleneck = **bounded rank**
- **Rank** of a linear map $\mathbf{A} : \mathbb{R}^k \to \mathbb{R}^l$

$$\text{rank}(\mathbf{A}) := \dim\left(\text{im}(\mathbf{A})\right) \leq \min\{k, l\}$$

  - note that for a matrix product (composition of linear maps)

$$\text{rank}(\mathbf{AB}) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\}.$$

  - decomposition rank: $\mathbf{M} = \mathbf{AB}$ with $\mathbf{A} \in \mathbb{R}^{m \times k}$, $\mathbf{B} \in \mathbb{R}^{k \times n}$, if and only if $\text{rank}(\mathbf{M}) \leq k$

Frobenius Norm Objective

- Linear autoencoder performs **low-rank approximation**

$$\text{rank}(\mathbf{F}) \leq \min\{\text{rank}(\mathbf{C}), \text{rank}(\mathbf{D})\} \leq k$$

- **Are there limits on the reconstruction quality achievable?**

- Data matrix $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_n]$ and approximations $\hat{\mathbf{X}} := [\hat{\mathbf{x}}_1 \ldots \hat{\mathbf{x}}_n]$

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} \|\mathbf{x}_i - \hat{\mathbf{x}}_i(\theta)\|^2 = \frac{1}{2n} \|\mathbf{X} - \hat{\mathbf{X}}(\theta)\|_F^2,$$

where $\|\mathbf{A}\|_F := \|\text{vec}(\mathbf{A})\|_2 = \sqrt{\sum_{ij} a_{ij}^2}$ (Frobenius norm)

Eckart-Young Theorem

- **Eckart-Young theorem**: for $k \leq \min\{m, n\}$

$$\underset{\hat{\mathbf{X}}:\text{rank}(\hat{\mathbf{X}})=k}{\arg\min} \|\mathbf{X} - \hat{\mathbf{X}}\|_F^2 = \mathbf{U}\,\boldsymbol{\Sigma}_k\,\mathbf{V}^\top$$

  - $\mathbf{X} = \mathbf{U}\,\boldsymbol{\Sigma}\,\mathbf{V}^\top$ is the Singular Value Decomposition of $\mathbf{X}$
  - $\boldsymbol{\Sigma}_k$ is the truncated diagonal matrix of singular values
  - minimal reconstruction loss $\min_\theta J(\theta) = \sum_{l=k+1}^{\min\{n,m\}} \sigma_l^2$.

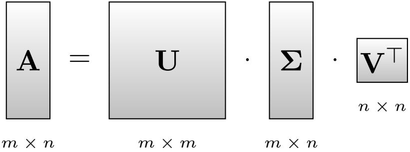
- Optimal rank k approximation: can be obtained via **Singular Value Decomposition** (SVD)
  - C. Eckart, G. Young, The approximation of one matrix by another of lower rank. Psychometrika, Volume 1, 1936

Section 3

Singular Value Decomposition

Singular Value Decomposition

- Any $m \times n$ matrix $\mathbf{A}$ can be decomposed into



  - with $\mathbf{U}$, $\mathbf{V}$ orthogonal, i.e. $\mathbf{U}\mathbf{U}^\top = \mathbf{I}_m$, $\mathbf{V}\mathbf{V}^\top = \mathbf{I}_n$.
  - and with $\mathbf{\Sigma}$ diagonal, $s := \min\{m, n\}$

$$\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_s), \quad \sigma_1 \geq \cdots \geq \sigma_s \geq 0$$

  - “diagonal” $\simeq$ padded w/ zeros to match dimensionality

Singular Vectors and Values

- Columns of $\mathbf{U}$ and $\mathbf{V}$: left/right **singular vectors**
- Entries of $\boldsymbol{\Sigma}$: **singular values**
  - number of distinct singular values $\leq s = \min\{n, m\}$
  - σ_i with two (or more) linearly independent left (or right) singular vectors = **degenerate**
- Uniqueness / ambiguity
  - singular vectors for non-degenerate σ_i: unique up to sign
  - singular vectors for degenerate σ_i: orthonormal basis (non-unique) of span (unique)
- Rank and SVD (exercise)

$$\text{rank}(\mathbf{A}) = r \iff \sigma_r > 0 \wedge \sigma_{r+1} = \sigma_{r+2} = \cdots = 0$$

Section 4

Linear Autoencoder (cont'd)

Optimal Linear Autoencoder via SVD

- Given data $\mathbf{X} \in \mathbb{R}^{m \times n}$ with SVD $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$.
- Define $\mathbf{U}_k := [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_k] \in \mathbb{R}^{m \times k}$ the first k columns of $\mathbf{U}$
- $\mathbf{C}^* = \mathbf{U}_k^\top$ and $\mathbf{D}^* = \mathbf{U}_k$ yields minimal reconstruction error for a linear autoencoder with k hidden units.
  - proof:

$$
\begin{aligned}
\hat{\mathbf{X}} = \mathbf{D}^*\mathbf{C}^*\mathbf{X} = \mathbf{U}_k\mathbf{U}_k^\top\left(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\right) &= \mathbf{U}_k\begin{bmatrix}\mathbf{I}_k & \mathbf{0}\end{bmatrix}\mathbf{\Sigma}\,\mathbf{V}^\top \\
&= \mathbf{U}\begin{bmatrix}\mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{\Sigma}\,\mathbf{V}^\top = \mathbf{U}\mathbf{\Sigma}_k\mathbf{V}^\top = \text{optimal (by EY)}
\end{aligned}
$$

  - for any $\mathbf{A} \in \mathrm{GL}(m)$: $\mathbf{C} = \mathbf{A}\mathbf{U}_k^\top$ and $\mathbf{D} = \mathbf{U}_k\mathbf{A}^{-1}$ are also optimal
  - $\implies$ low-dimensional representation $\mathbf{z}$ has limited interpretability

Weight Sharing

- Corollary: weight sharing $\mathbf{D} = \mathbf{C}^\top$ w/o reducing modeling power
- Reduces ambiguity: $\mathbf{A}^{-1} = \mathbf{A}^\top$, i.e. $\mathbf{A} \in \mathrm{O}(m)$ (orthogonal group)
- $\implies$ mapping $\mathbf{x} \mapsto \mathbf{z}$ uniquely determined up to rotations (permutations, reflections)

Next week: principal component analysis, algorithms