Lecture 11

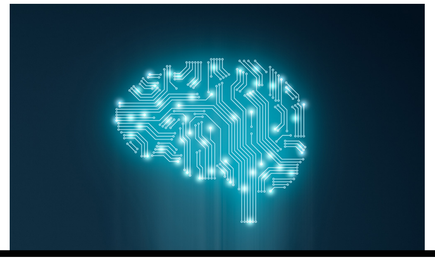
01.12.2025

Today's plan and announcements

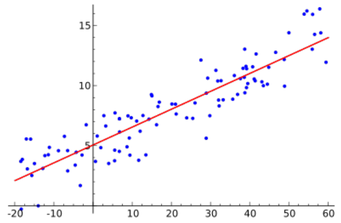
- Unsupervised learning
  - Dimensionality reduction through principal component analysis (PCA)

- This week
  - Python exercise for kNN
  - Python exercise for PCA

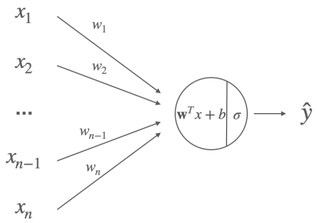
Introduction



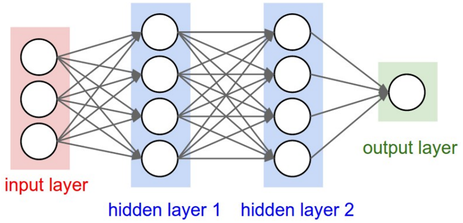
Linear regression



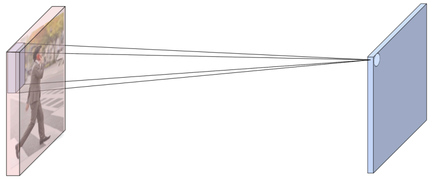
Logistic regression



Neural networks

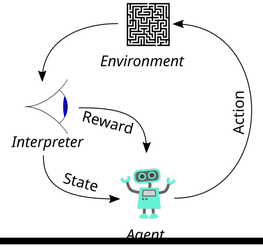


Convolutional neural networks

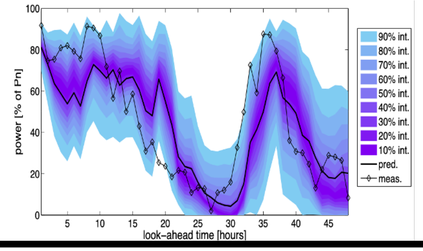


AI & sustainability

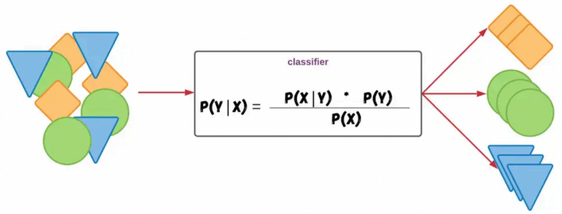
Reinforcement learning



Recurrent neural networks

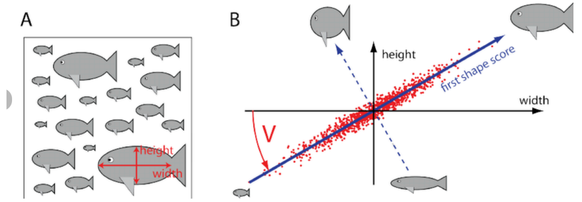


Naive Bayes

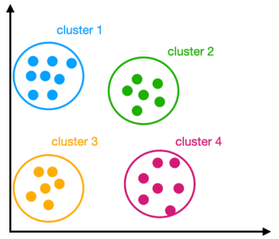


KNN

Dimensionality reduction



Clustering

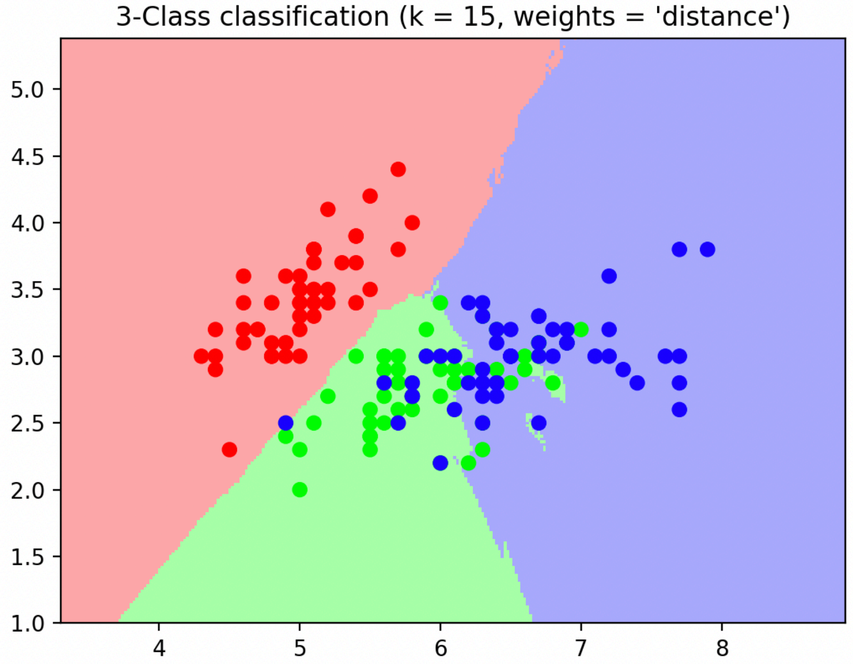


Review - k-NN for classification and regression

- k-Nearest Neighbors (k-NN) classifier assumes that data points of similar classes exist in close proximity
  - Similar inputs have similar outputs

- It assigns label to a query point according to label of its k closest neighbors
  - Neighbors depend on the distance metric
  - Hyper parameters: k and the distance metric
  - No parameter to train/learn
  - Inference requires using the entire dataset

- Actively used in similarly search systems (recommendation, image similarity, semantic search in language processing)
  - Often after embedding features appropriately



https://stackoverflow.com/questions/45075638/graph-k-nn-decision-boundaries-in-matplotlib

Review of data statistics through quiz 1 grades

$\{x^i \in \mathbb{R}\}_{i=1}^{269}$ i^{th} student's grade of quiz 1

Mean: 10.18, Standard deviation: 1.14, Mode: 11

μ σ

Quartiles:

1st: 9.5, 25% of data points is below this number: 9.5

2nd (median): 11, 50% of data points is below this number

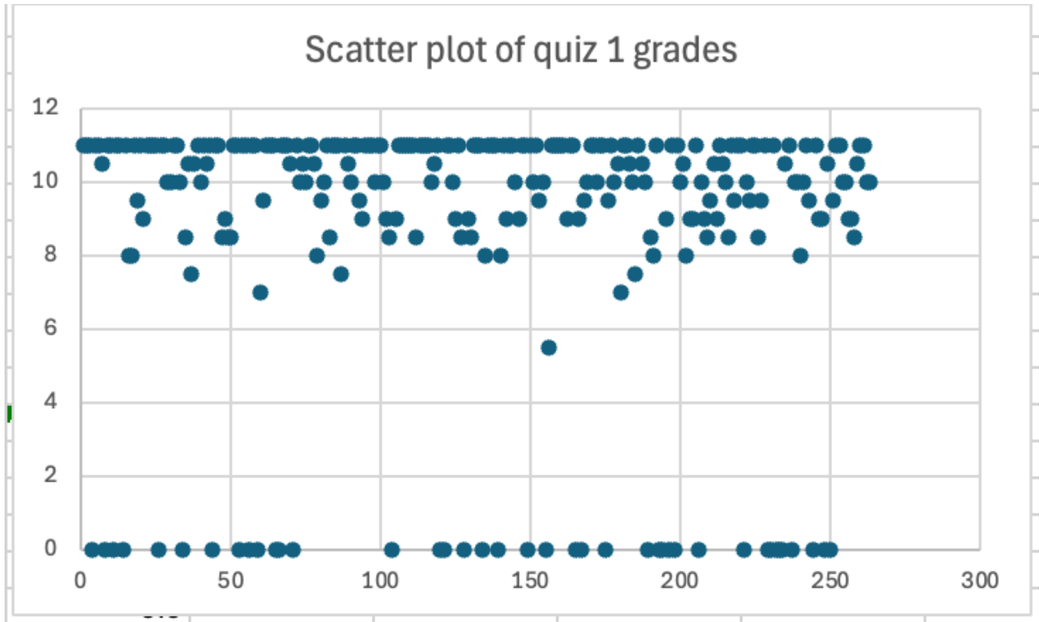
3rd: 11, 75% of grades is below this number

Conclusion: the quiz was way too easy!

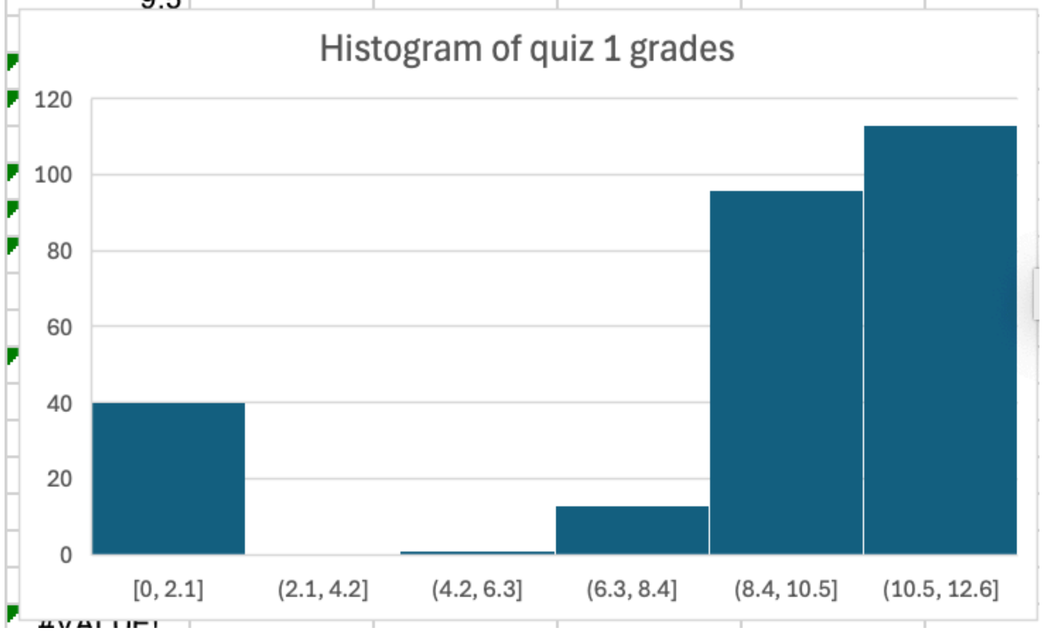
z-score

How would we "standardize" the data?

$\tilde{x}^i = \frac{x^i - \mu}{\sigma}$ $\Rightarrow$ $\{\tilde{x}^i\}_{i=1}^{N}$ has mean 0 & $\tilde{\sigma} = 1$.



9.5



Introduction
Unsupervised learning

Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a **data set with no pre-existing labels** and with a minimum of human supervision.

in unsupervised learning, we don't use labels anymore
Note: the objective is vague but we will consider 2 concrete instances

- dimensionality reduction ... this lecture
- clustering ... next lecture

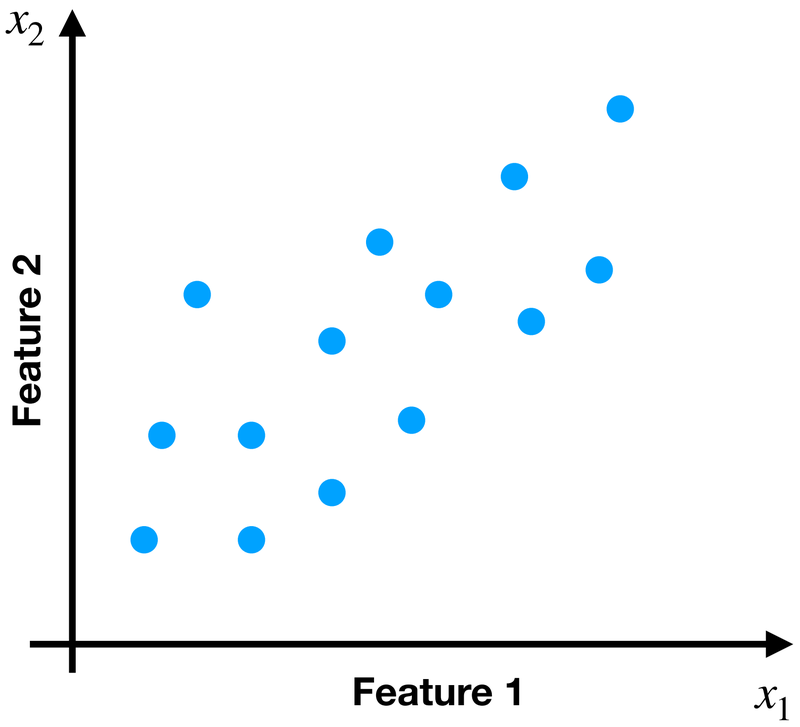
Dimensionality reduction

Through

principal component analysis

Motivation
Intuition

We have samples described by a number of features (in this picture 2 features)



We want to find a smaller set of new features that explain our sample because:

Less features is easier to visualize

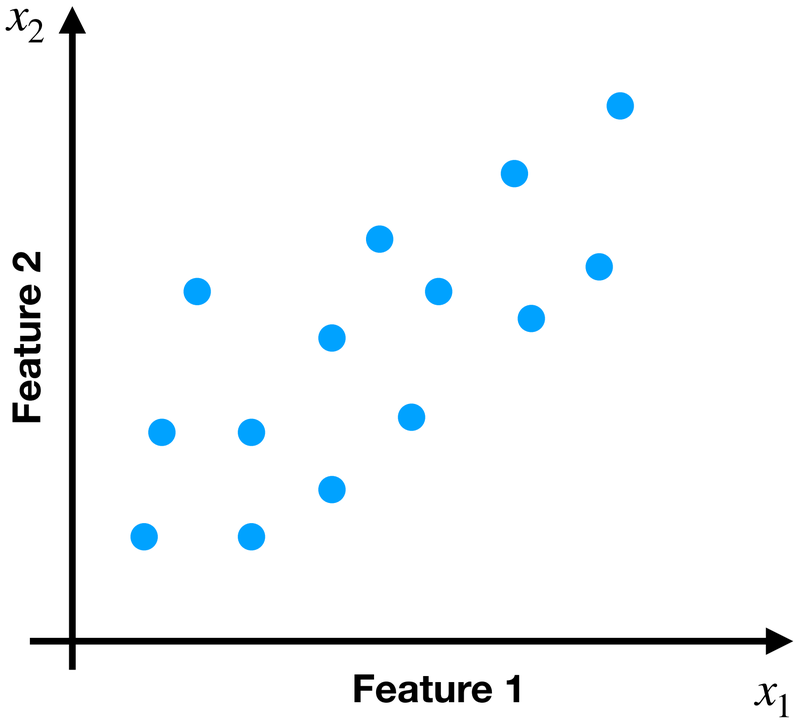
Some of the current feature can be redundant

Some of the current features are not very useful to describe our samples

Principal component analysis (PCA)

Approach to dimensionality reduction

How to find this smaller set of new features ?

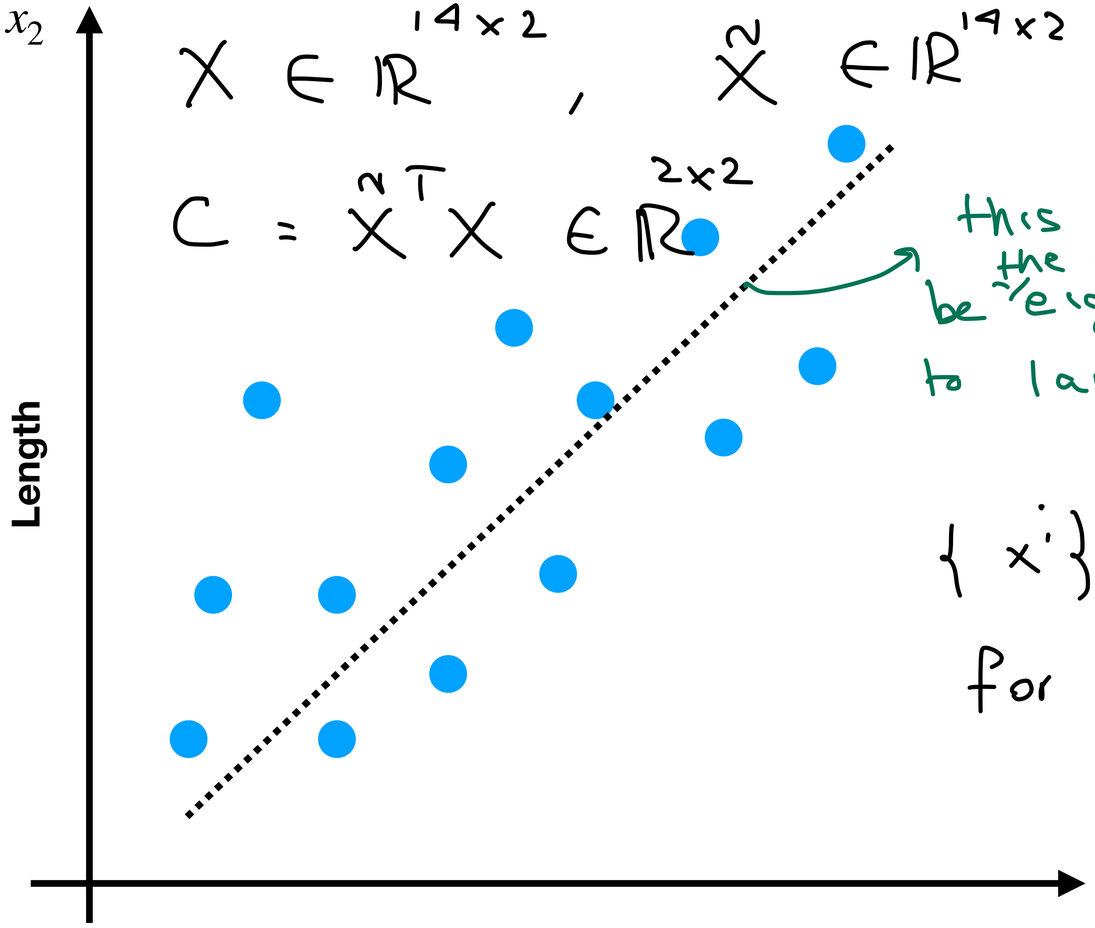


PCA : Find the best **linear combination** of features to create new reduced number of features that explain our samples ~~better~~ well

PCA

Projection of points onto a lower dimensional subspace

How to find this smaller set of new features ?



$X \in \mathbb{R}^{14 \times 2}$, $\tilde{X} \in \mathbb{R}^{14 \times 2}$ is standardized X

$C = \tilde{X}^T \tilde{X} \in \mathbb{R}^{2 \times 2}$

this line would be the span of the eigenvector corresponds to largest eigenvalue of C.

$w_1 x_1 + w_2 x_2$

The new feature

A mix of length and weight that describe our samples better

$\{x^i\}_{i=1}^N$, $x^i \in \mathbb{R}^2$

for every x^i, we compute

$a^i = w_1 x_1^i + w_2 x_2^i$

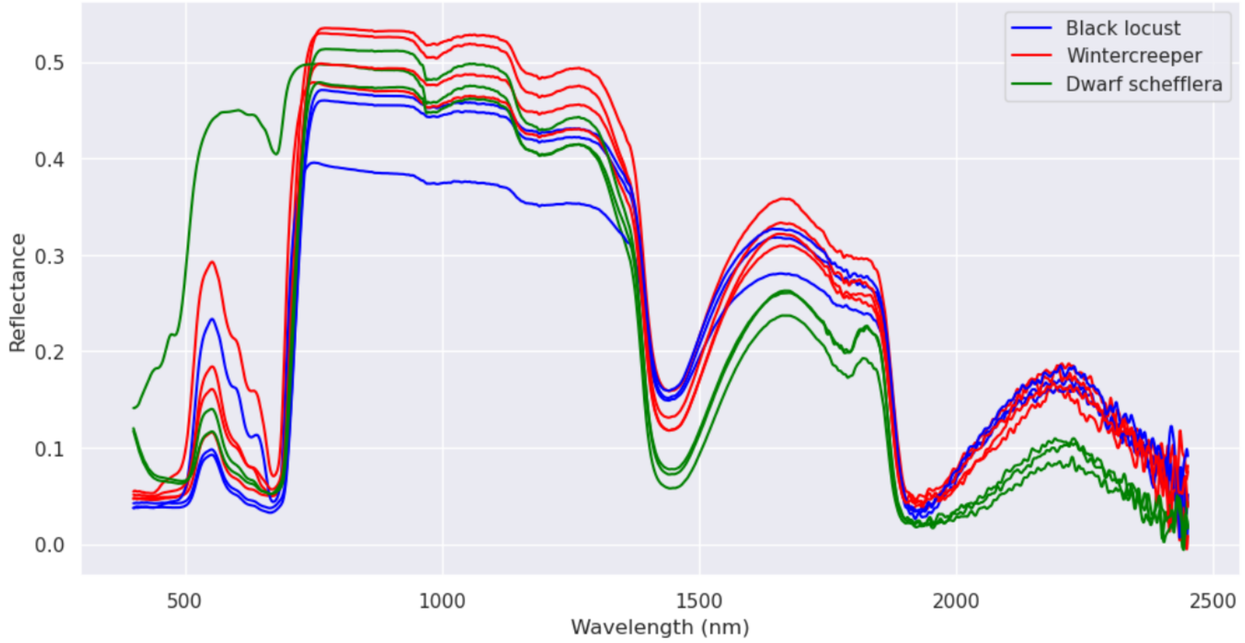
$\{a^i\}_{i=1}^N$, $a^i \in \mathbb{R}$

PCA - example in python and further applications

- **Your Python PCA example**
  - Dataset: Leaves' optical reflectance measured at each nm in the range of 450-2500 nm Wavelength, see more details about the experiment here
  - Goal: can we reduce the dimensionality of feature vector and still capture the distinguishing features of each leaf

$N = 18$

$\{ x^i \in \mathbb{R}^{2051} \}_{i=1}^{18}, \quad C = X^T X \in \mathbb{R}^{2051 \times 2051}$



$a^i \in \mathbb{R}^3$

3 principal components are the eigenvectors corresponds to the first 3 largest eigenvalues of $C \in \mathbb{R}^{2051 \times 2051}$

- **Other applications of PCA:** dynamical system model order reduction, audio compression for recommendation algorithms, text processing for news recommendation

Finding distance of a point to a subspace

- Subspace $S \subset \mathbb{R}^d$: let $\theta_i \in \mathbb{R}^d, a_i \in \mathbb{R}, i = 1,2,\ldots,r$

a linear combination of $\{\theta_i\}_{i=1}^r$ is given by $\sum_{i=1}^r a_i \theta_i$

subspace S is spanned by $\{\theta_i\}_{i=1}^r$ $\quad S = \left\{ \sum_{i=1}^r a_i \theta_i \mid a_i \in \mathbb{R}, \theta_i \in \mathbb{R}^d, i = 1,\ldots,r \right\}$

- Distance of a point $x \in \mathbb{R}^d$ to a subspace $S \subset \mathbb{R}^d$:

$$\text{dist}(x, S) = \min_{s \in S} \|x - s\|_2 = \min_{a \in \mathbb{R}^r} \|x - \Theta a\|_2, \text{ where } \Theta = \begin{bmatrix} \theta_1 & \theta_2 & \ldots & \theta_r \end{bmatrix} \in \mathbb{R}^{d \times r}$$

- The minimizer of the above is equivalent to the minimizer of $\min_{a \in \mathbb{R}^r} \|x - \Theta a\|^2$, so we square the objective

to be able to differentiate it and find the optimizer (verify that the objective function is convex in $a \in \mathbb{R}^r$

$$\|x - \Theta a\|_2^2 = \langle x - \Theta a, x - \Theta a \rangle = \|x\|_2^2 - 2\langle x, \Theta a \rangle + \|\Theta a\|^2$$

$$= \|x\|_2^2 - 2 a^T \Theta^T x + a^T \Theta^T \Theta a$$

Projection of a point onto a subspace

- To compute the optimizer, $\frac{\partial J}{\partial a} = 0$ in squared distance: $J(a) = \|x\|^2 - 2a^T\Theta^T x + a^T\Theta^T\Theta a$

$$-\Theta^T x + \Theta^T\Theta a = 0 \Rightarrow a^* = (\Theta^T\Theta)^{-1}\Theta^T x$$ invertible

(we are assuming $\{\theta_i\}_{i=1}^r$ are linearly independent, so $\Theta = [\theta_1 \cdots \theta_r]$ is

- Choosing orthonormal vectors spanning the subspace $\Longrightarrow \Theta^T\Theta = I_{r\times r}$ identity matrix

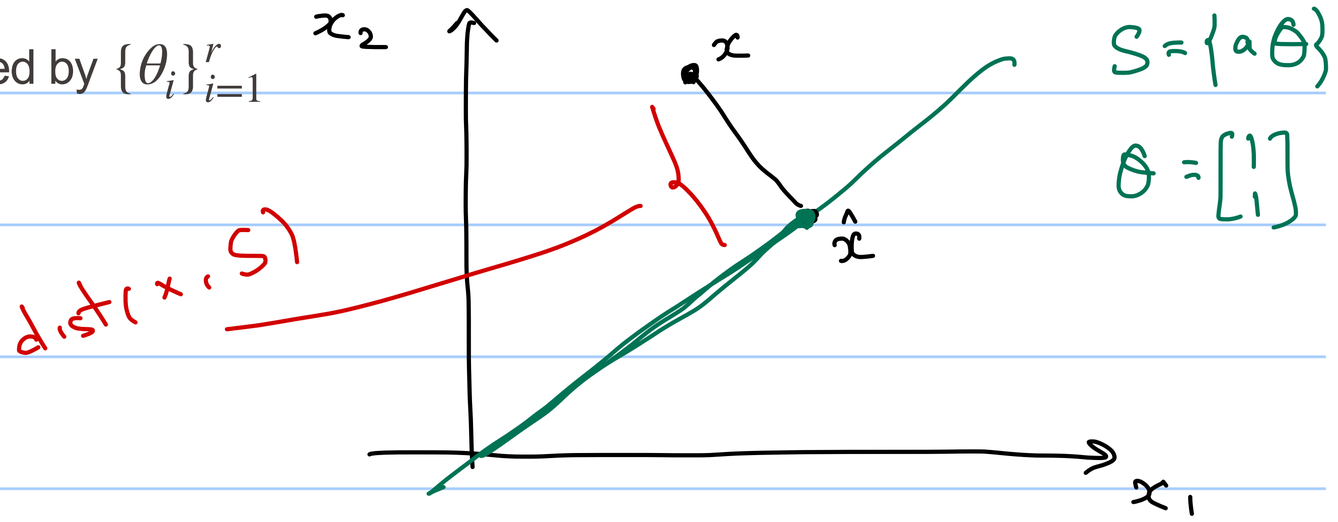
$$\begin{cases} \theta_i^T\theta_i = 1 \\ \theta_i^T\theta_j = 0 \quad j \neq i \end{cases}$$

$$\Rightarrow a^* = \Theta^T x \Rightarrow \text{dist}(x, S)^2 = \|x - \Theta a^*\|_2^2$$

- Projection of point $x \in \mathbb{R}^d$ onto the subspace S spanned by $\{\theta_i\}_{i=1}^r$

$$\hat{x} = \Theta\Theta^T x \in \mathbb{R}^2$$

$$a^* = \Theta^T x$$



PCA - finding a subspace to minimize average distance of data to it

- Data set $\{x^i\}_{i=1}^N$ $\quad x^i \in \mathbb{R}^d$: d feature vectors

- Given a subspace $S \subset \mathbb{R}^d$, spanned by orthonormal vectors $\{\theta_i\}_{i=1}^r$, $\quad r << d$
  - sum of distances of all data points to the subspace is

$$\frac{1}{N} \sum_{i=1}^{N} \text{dist}(x^i, S)$$

average

Recall $S = \left\{ \sum_{i=1}^{r} a_i \theta_i \;\middle|\; a_i \in \mathbb{R},\ \theta_i \in \mathbb{R}^d \right\}$.

- PCA (principal component analysis) objective is to find $\Theta = [\theta_1 \;\; \theta_2 \;\; \dots \;\; \theta_r] \in \mathbb{R}^{d \times r}$ minimizing the above sum of distances
  - In other words, choosing the subspace S

$$\min_{\Theta \in \mathbb{R}^{d \times r}} \frac{1}{N} \sum_{i=1}^{N} \text{dist}(x^i, S)^2$$

subject $\Theta^T \Theta = I_{r \times r}$

Formulation of PCA objective using the data matrix

First, standardize the data (z-score standardization),

let $X = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix} \in \mathbb{R}^{N \times d}$ be the standardize data matrix

- For a given matrix, it's Frobenius norm is given by $\|X\|_F = \sqrt{\text{trace}(X^T X)}$
- You can verify that the PCA objective can be written as $\min_{\Theta \in \mathbb{R}^{d \times r}} \|X - X\Theta\Theta^T\|_F^2$ subject to $\Theta\Theta^T = I_{r \times r}$
- Next, we will see how to find the optimizer above

0) Recall: $\hat{x} = \Theta\Theta^T x$

1) first row of $X\Theta\Theta^T$ is

$(\hat{x}^1)^T = [x_1^1 \; x_2^1 \; \cdots \; x_d^1]\Theta\Theta^T$

.... $\Rightarrow$

$X - X\Theta\Theta^T = \begin{bmatrix} (x^1)^T - (\hat{x}^1)^T \\ (x^2)^T - (\hat{x}^2)^T \\ \\ (x^N)^T - (\hat{x}^N)^T \end{bmatrix}$

Eigenvalue decomposition of a symmetric matrix

- Given the standardized data matrix $X \in \mathbb{R}^{N \times d}$, Let $C = X^T X \in \mathbb{R}^{d \times d}$
  - Observe that C is symmetric, it is also positive semi-definite
    $\Rightarrow$ eigenvalues of C are non-negative & real-valued.
- Fact: a symmetric matrix $C \in \mathbb{R}^{d \times d}$ has an eigenvalue decomposition as follows:

$$C = V D V^T, \quad D = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{bmatrix}, \quad \lambda_i \in \mathbb{R} \text{ are eigenvalues of } C$$

$$V = \begin{bmatrix} v_1 & v_2 & \cdots & v_d \end{bmatrix}, \quad v_i \in \mathbb{R}^d \text{ is eigenvector corresponding to eigenvalue } \lambda_i$$

A symmetric matrix is diagonalizable.

Solution to PCA using eigenvalue decomposition

- Recall PCA objective $\frac{1}{N}\min_{\Theta \in \mathbb{R}^{d\times r}} \|X - X\Theta\Theta^T\|_F$ with $\Theta = \begin{bmatrix}\theta_1 & \theta_2 & \dots & \theta_r\end{bmatrix} \in \mathbb{R}^{d\times r}$, $\Theta\Theta^T = I_{r\times r}$

Finding the r-dimensional subspace ($r < d$) spanned by $\{\theta_i\}_{i=1}^r$ that minimizes average distance of data points to it

- Do eigenvalue decomposition of $C = X^T X$
  - Order the eigenvalues from largest to smallest $\lambda_1 \geq \lambda_2 \geq \dots \lambda_d$
  - Choose the eigenvectors corresponding to $\{\lambda_i\}_{i=1}^r$, namely, $\{v_i\}_{i=1}^r$
  - Then $\Theta^* = [v_1 \quad v_2 \quad \dots \quad v_r]$

$\{v_i\}_{i=1}^r$ are called the r principal components of data matrix X

Projection of the data onto principal components

- Given the subspace $S \subset \mathbb{R}^d$ spanned by columns of Θ^*, namely, the r eigenvectors corresponding to the r largest eigenvalues of $C = X^T X$,
  - The feature matrix in reduced dimension (compressed data) is $A = X\Theta^* \in \mathbb{R}^{N \times r}$
  - each datapoint $x^i \in \mathbb{R}^d$ is reduced in dimension and represented by $a^i = \Theta^{*T} x^i$

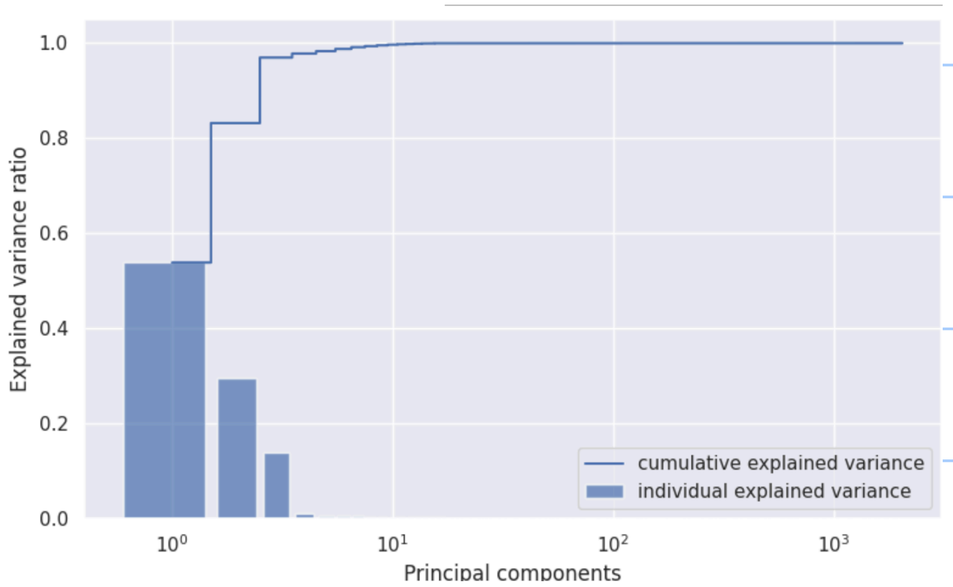
- The projection of data matrix on S is $\hat{X} = X\Theta^*\Theta^{*T}$

- We can work with $A \in \mathbb{R}^{N \times r}$ instead of $X \in \mathbb{R}^{N \times d}$

PCA - how to determine the subspace dimension

- To fix a dimension r, one approach is through using explained variance
  - Each principal component explains a portion of data variability
  - Explained variance for r principal components: $\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$, where λ_i are the ordered eigenvalues of $C = X^T X$
  - Pick smallest number of principal components that explains ~ 90-95% of the variance

$$\min_r \text{ such that } \frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq \sim \frac{90\text{-}95}{100}$$



PCA example - for text processing

N documents

- Consider a corpus of documents having d unique words
- Create features using the so-called TFIDF: term frequency inverse document frequency
  - For a document i, term frequency of word j:

$$f_{term}(i,j) = \frac{\#\text{ of times word } j \text{ appears in document } i}{\#\text{ of all words in document } i}$$

  - For the corpus of documents, document frequency of word j:

$$f_{doc}(j) = \frac{\#\text{ of times word } j \text{ appears in any document}}{\#\text{ of all document}}$$

- TFIDF embedding

$$x_j^i = f_{term}(i,j) \times \log \frac{1}{f_{doc}(j)}, \qquad \begin{array}{l} i = 1, \dots, N \\ j = 1, \dots, d. \end{array}$$

PCA example applied to two texts

Let's go through the example in "Principal Component Analysis" lecture of Stanford EE104 https://ee104.stanford.edu/lectures/pca.pdf

We will use PCA to reduce the dimensionality of the features from $X \in \mathbb{R}^{100\times2262}$ to $A \in \mathbb{R}^{100\times2}$.

With the above reduction, we can still distinguish the two texts

1) *The Critique of Pure Reason* by Immanuel Kant

2) *The Problems of Philosophy* by Bertrand Russell

Autoencoder for dimensionality reduction

- PCA assumes the data is approximately on a lower dimensional subspace.
- Generally, the data might be approximately on a lower dimensional surface (not a subspace). In this case, other dimensionality reduction techniques such as autoencoder can be used
- An autoencoder is a type of neural network used for dimensionality reduction
- Training: minimizing the reconstruction error / loss $\sum_{i=1}^{N} L\left(x^i, \hat{x}^i\right)$

Autoencoder

Autoencoder vs. PCA

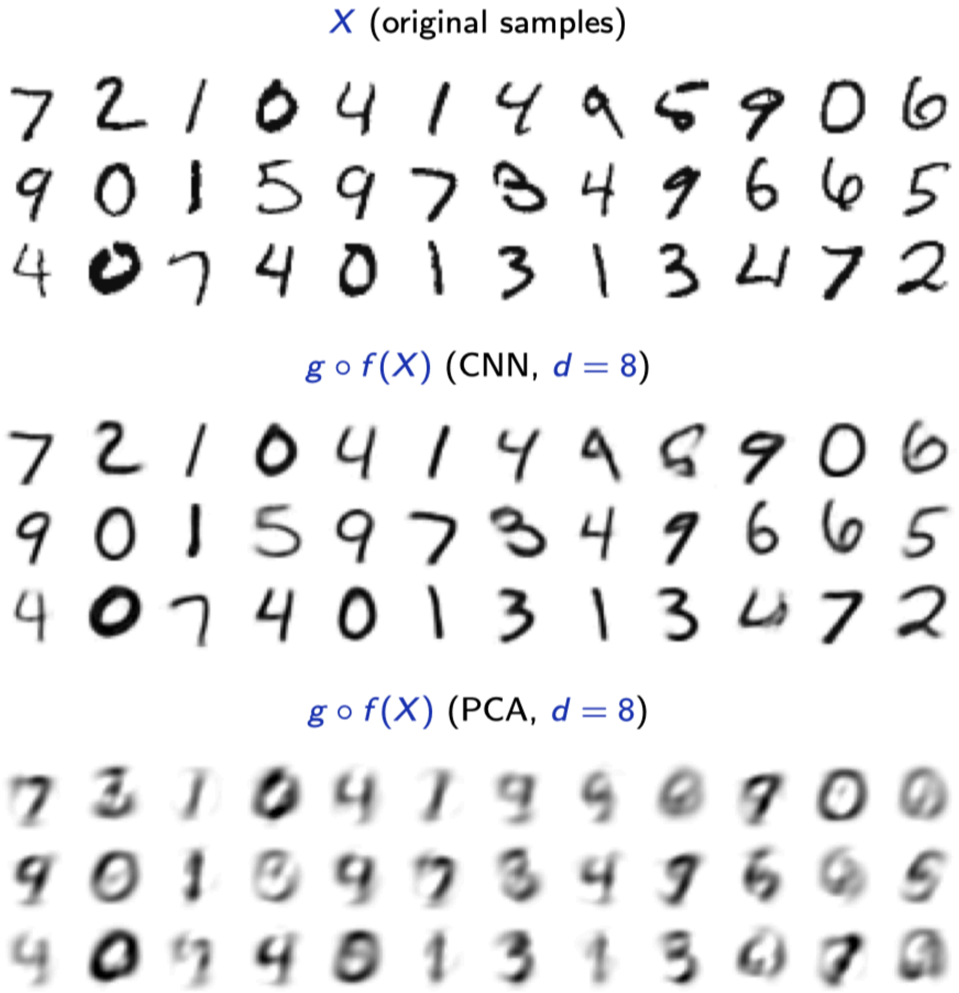


Image credit: F. Fleuret, Deep Learning (EPFL)

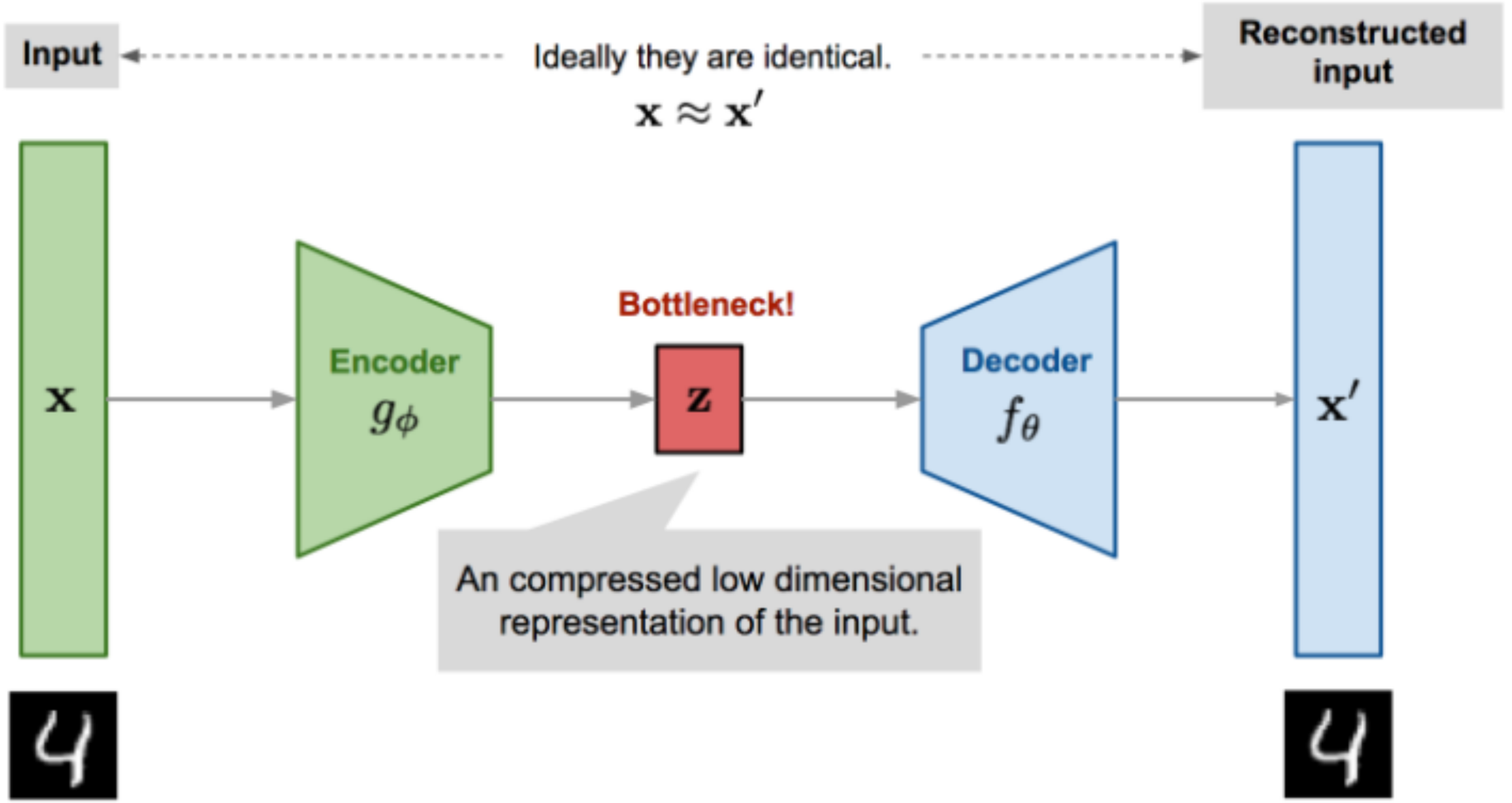
Top: Some examples of the original MNIST test samples

Middle: Reconstructed output from an auto-encoder with a latent space of 8 dimensions

This auto-encoder uses convolutional layers, and was trained on the MNIST training set

Bottom: Reconstructed output from PCA with 8 reduced dimensions

Autoencoder
Introduction



Summary

- Dimensionality reduction
  - Exploratory data analysis
  - Visualizing data
  - With reduced dimensional data
    - Reduce memory/computation requirements for other ML tasks
    - Avoids overfitting

- PCA
  - Assumes data lies approximately on a lower dimensional subspace
  - Easy to compute
  - Connection to singular value decomposition (you will see in next problem set)