

Unsupervised machine learning (8DC00)

Friso G. Heslinga

Previous lecture: frameworks



PYTORCH



Previous lecture: applications

Classification



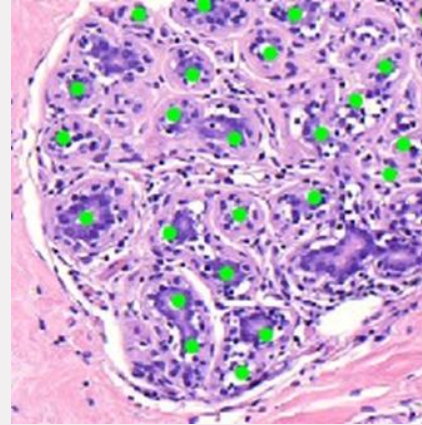
Esteva et al.,
Nature 2017

Regression



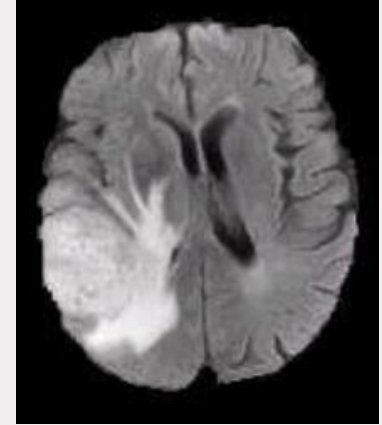
Heslinga et al.,
SPIE-MI 2019

Detection



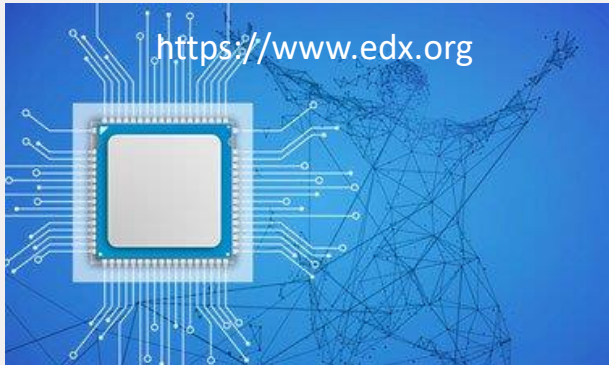
Wetstein et al.,
SPIE-MI 2019

Segmentation



Dong et al.,
MIAU 2017

Previous lecture: deep learning enabled



- Hardware improvements
- Parallel computing (GPU)



- Digital data
 - More (image) data
- = also true for medical imaging

Previous examples required labeled data

learning from labeled data = **supervised training**

Alternatives:

Unsupervised

Semi-supervised

Self-supervised

Reinforcement Learning

Learning outcomes – update!

- Students can describe the difference between **supervised** and **unsupervised learning** and name advantages of both methods
- Students can apply K-means to find motivate why they **would choose a deep learning model** for a specific research application
- Students can name **two key developments** that have enabled deep learning

Lecture outline

- Supervised vs unsupervised
- K-means
- Principal component analysis
- *Break (15 mins)*
- Auto-encoders
- Semi-supervised

- Supervised deep learning project example

Learning strategies

Supervised

Learning from examples (=training data) that are labeled with their desired outputs. The goal is to learn general rules that maps inputs to outputs.

Unsupervised

Learning from examples without labels. The goals are:

- Learning the entire probability distribution that generated a dataset
- Finding structure in data
- Reducing dimensionality → feature learning

Another learning strategy (no exam material)

Reinforcement Learning

Learning by interacting with a dynamic environment to achieve a certain goal (such as driving a vehicle or playing a game against an opponent). The system is provided feedback in terms of rewards and punishments as it navigates its problem space.

Reinforcement learning

RESEARCH

COMPUTER SCIENCE

A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play

David Silver^{1,2*,†}, Thomas Hubert^{1,2}, Julian Schrittwieser^{1,2}, Ioannis Antonoglou¹, Matthew Lai¹, Arthur Guez¹, Marc Lanctot¹, Laurent Sifre¹, Dhharshan Kumaran¹, Thore Graepel¹, Timothy Lillicrap¹, Karen Simonyan¹, Demis Hassabis^{1,†}

The game of chess is the longest-studied domain in the history of artificial intelligence. The strongest programs are based on a combination of sophisticated search techniques, domain-specific adaptations, and handcrafted evaluation functions that have been refined by human experts over several decades. By contrast, the AlphaGo Zero program recently achieved superhuman performance in the game of Go by reinforcement learning from self-play. In this paper, we generalize this approach into a single AlphaZero algorithm that can achieve superhuman performance in many challenging games. Starting from random play and given no domain knowledge except the game rules, AlphaZero convincingly defeated a world champion program in the games of chess and shogi (Japanese chess), as well as Go.

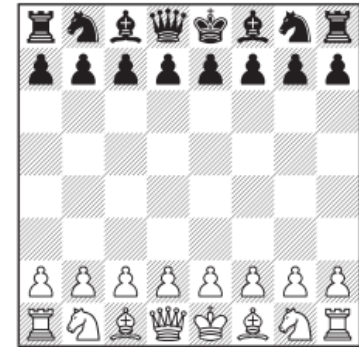
programmers, combined with alpha-beta search that expands by using a large number of domain-specific adaptations to these augmentations, focus on Chess Engine Championships world champion Stockfish (1, 12).

In terms of game tree complexity, chess is a substantially harder game than Go, which is played on a larger board with more pieces; any captured pieces are removed from the board and may subsequently be recaptured. The strongest chess programs, including the 2017 Computer Chess world champion Elmo, have defeated human champions by using an algorithm similar to the optimized alpha-beta search used in traditional chess engines.

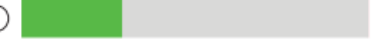
AlphaZero replaces the search and domain-specific adaptations with a single, general-purpose reinforcement learning algorithm.

Chess

AlphaZero vs. Stockfish



W: 29.0% D: 70.6% L: 0.4%



W: 2.0% D: 97.2% L: 0.8%

Silver, Hubert, Schrittwieser et al., Science (2018)

Reinforcement learning in healthcare

Deep Reinforcement Learning for Sepsis Treatment

Aniruddh Raghu
Cambridge University
United Kingdom
ar753@cam.ac.uk

Matthieu Komorowski
Imperial College London
United Kingdom
m.komorowski14@imperial.ac.uk

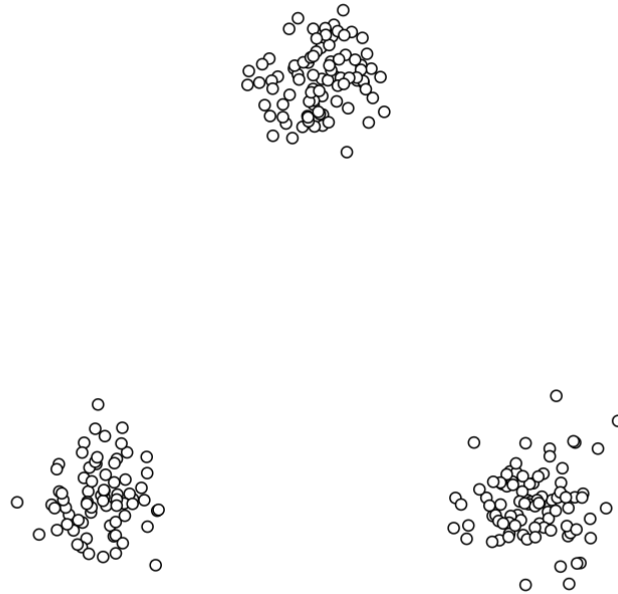
Imran Ahmed
Cambridge University
United Kingdom
ia311@cam.ac.uk

However, not often used in healthcare. Why?

How could we employ reinforcement learning for a surgery robot?

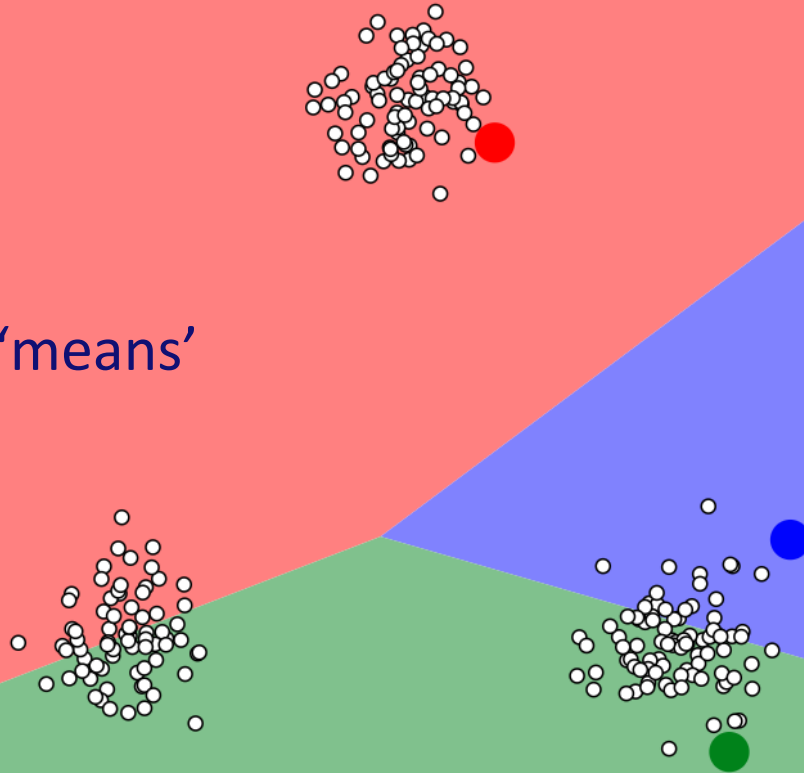
Finding clusters using K-means

<https://www.naftaliharris.com/blog/visualizing-k-means-clustering/>

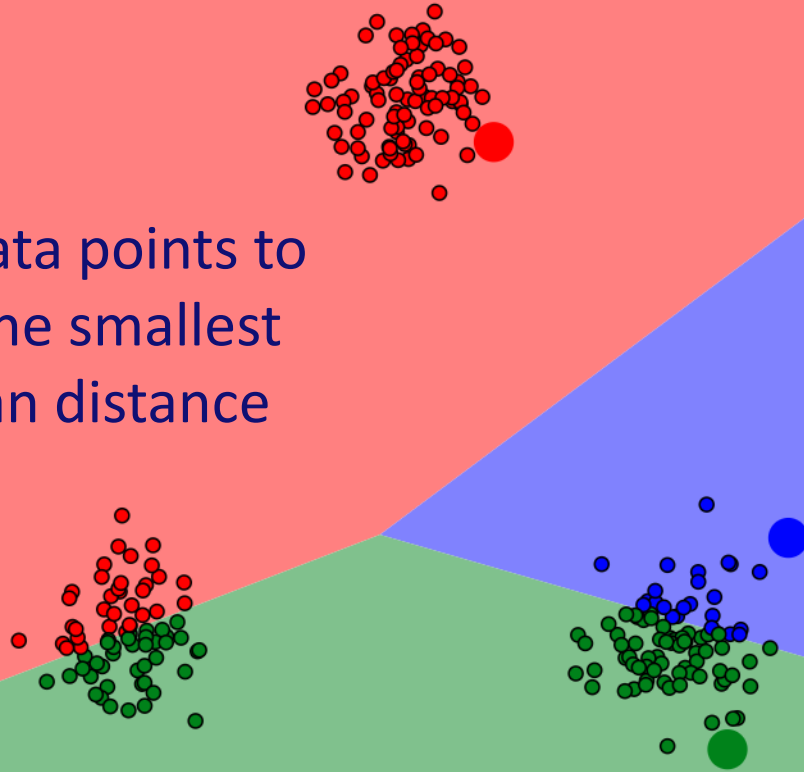


Initialization:

Choose K initial 'means'



Step 1: assign data points to centroids with the smallest squared Euclidian distance



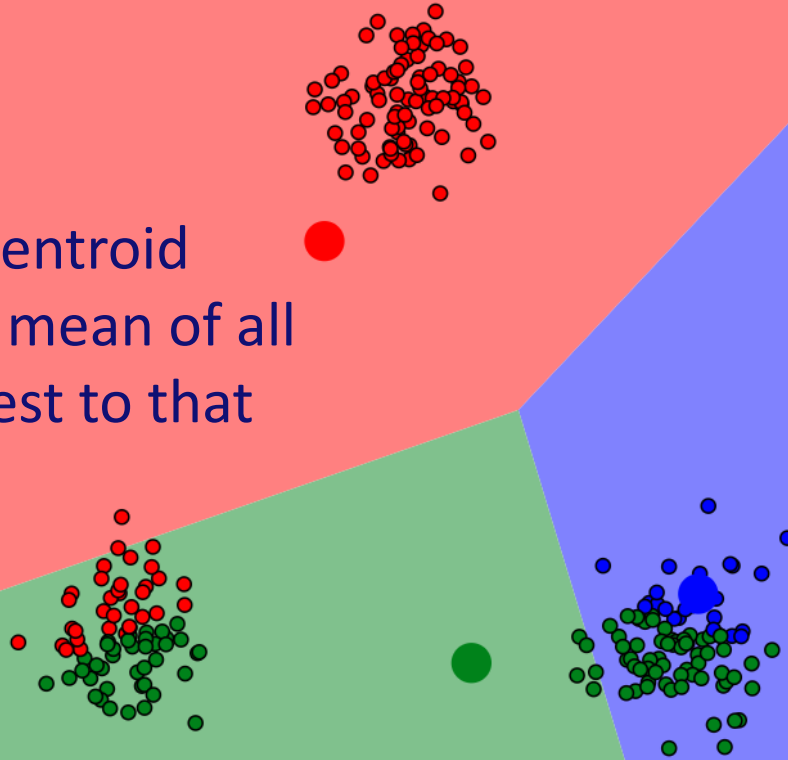
K-means – evaluate clustering performance

Average squared Euclidean distance between each point and the closest cluster:

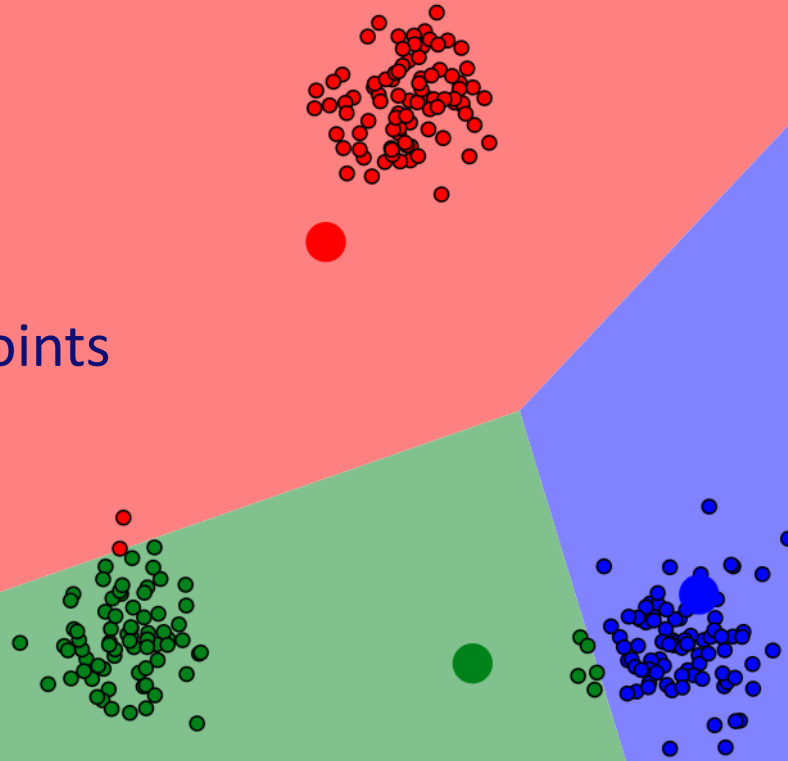
$$J(W) = \frac{1}{N} \sum_i ||\min_k (W_k - x_i) ||_2^2$$

x_i are the points, W are the cluster centroids

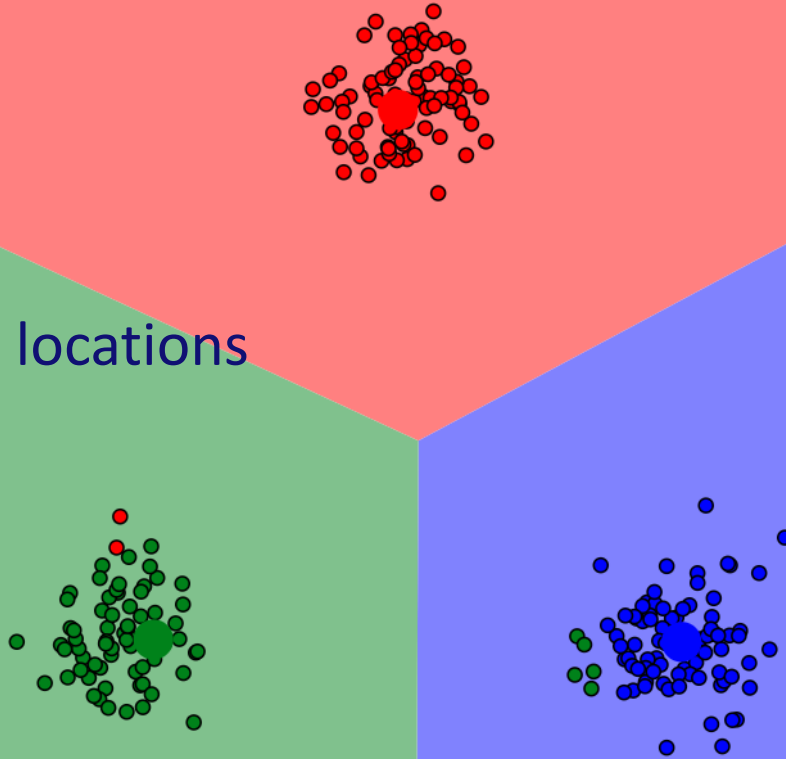
Step 2: update centroid locations by the mean of all data points closest to that centroid



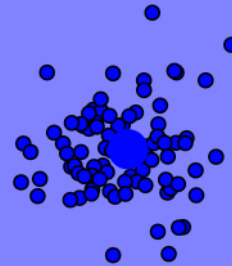
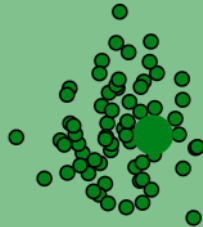
Repeat step 1:
Reassign data points



Repeat step 2:
Update centroid locations

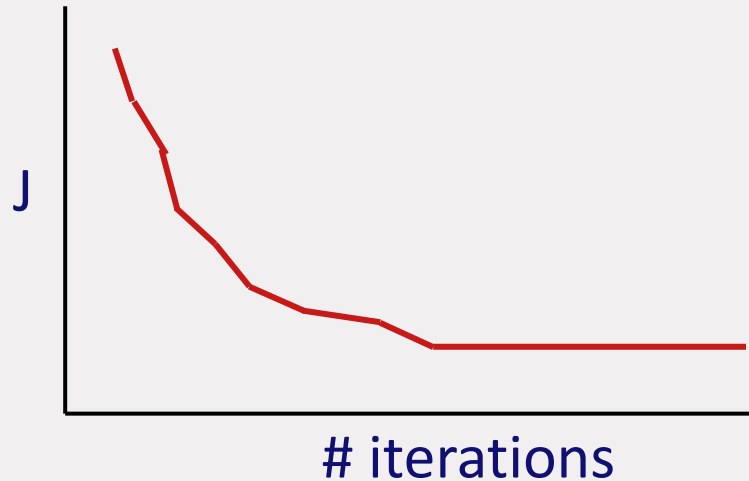


Repeat step 1:
Reassign data points



When do we stop?

- When the error J does not decrease anymore
- After n iterations

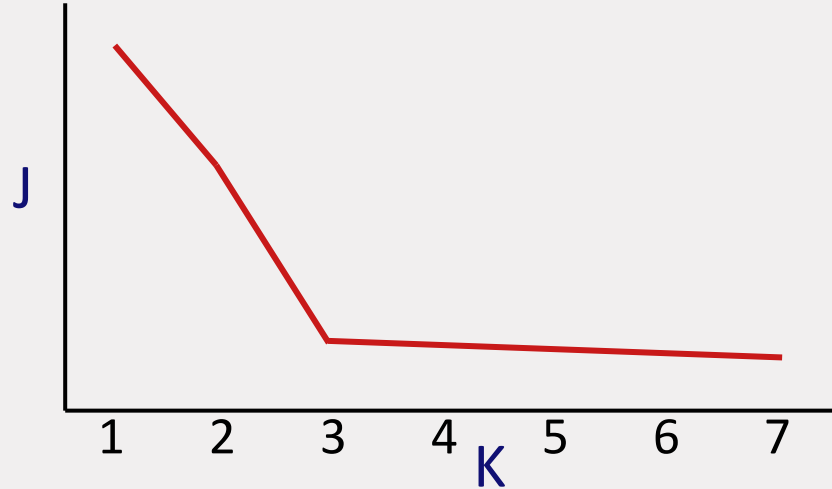


How do we choose initial centroid locations?

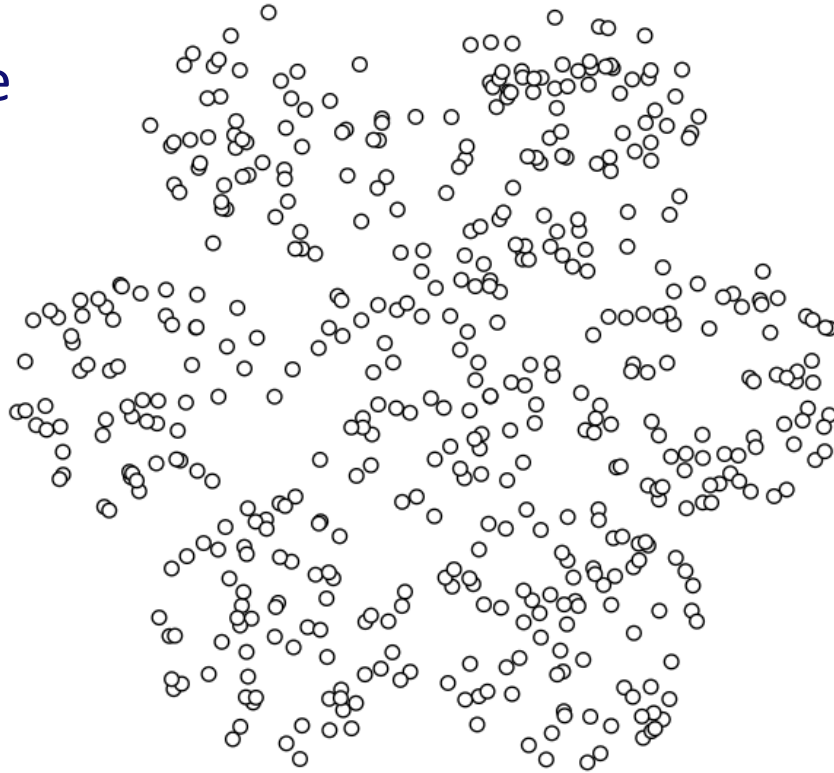
- Random
- Farthest points
- Manual?
 - Supervised
 - Difficult for high-dimensional data

How do we choose K?

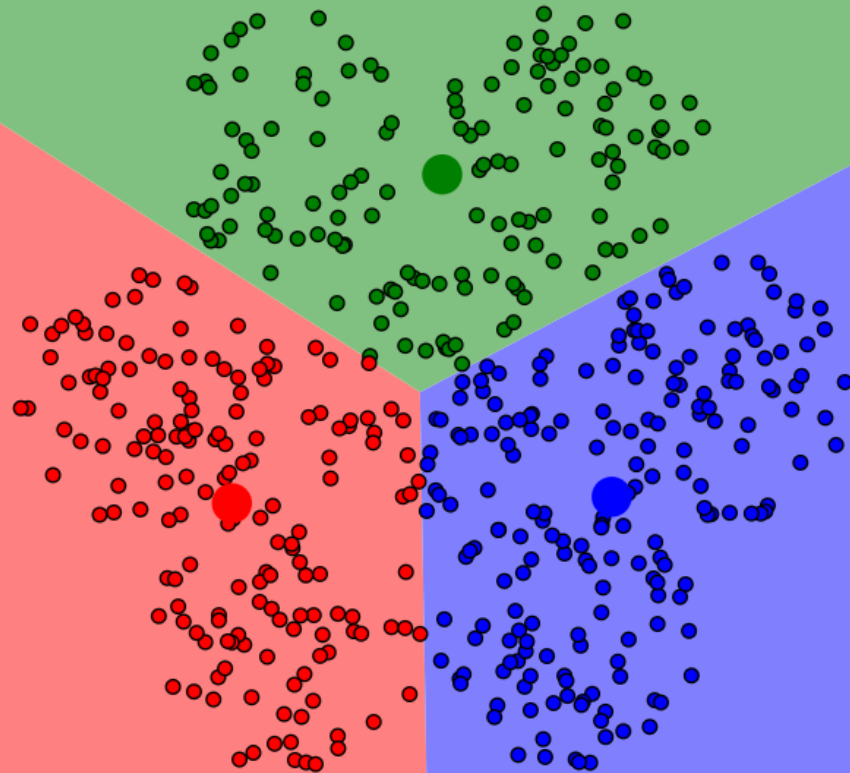
- K (=number of means) is a hyperparameter



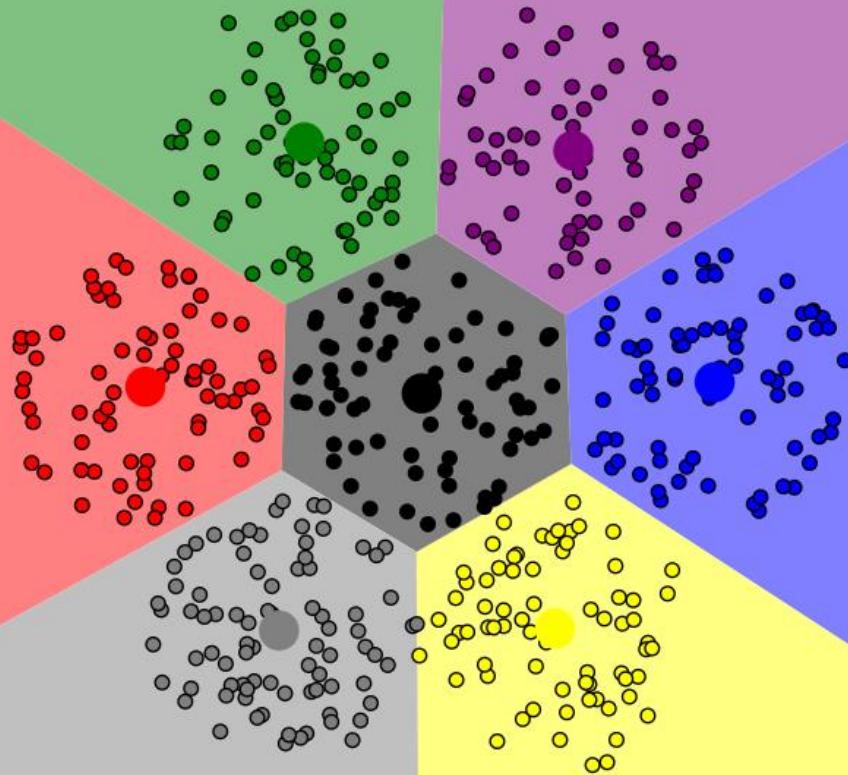
Another example



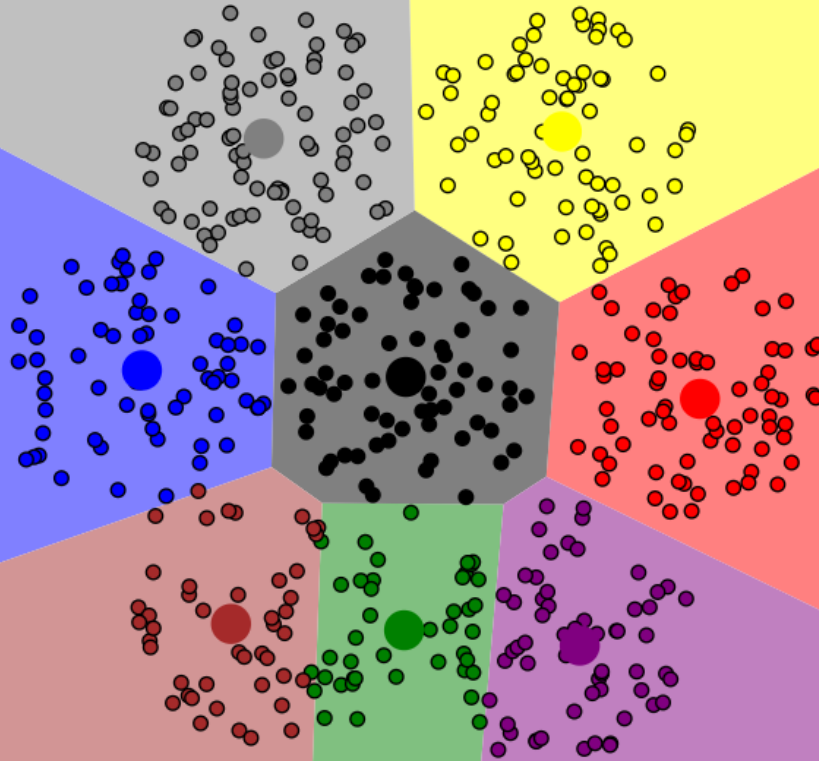
$K = 3$



$K = 7$



$K = 8$



Principal Component Analysis (PCA)

Goal: Finding the principle components that describe our data.

= finding the directions in which the data shows most variation

Useful for dimensionality reduction

- E.g. find low-dimensional classification boundaries

Results in better generalization!

Principal Component Analysis

Example data set

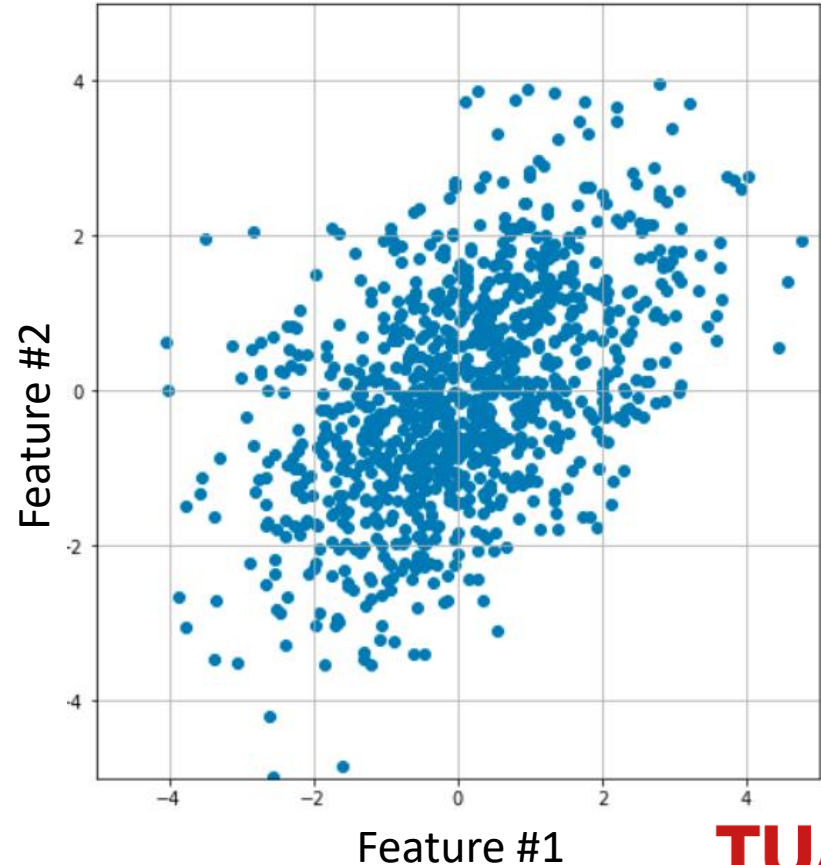
M -by-2 matrix \mathbf{X} containing M points

Sampled from 2D Gaussian distribution

$$\mu_1 = 0$$

$$\mu_2 = 0$$

$$\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$



Principal Component Analysis

Example data set

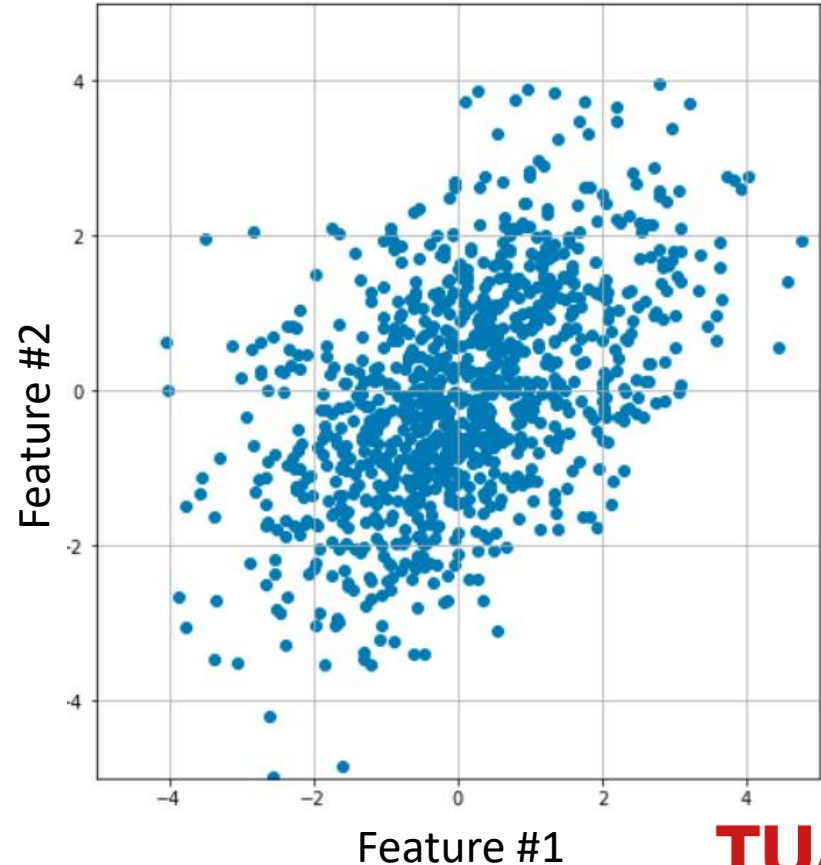
M -by-2 matrix \mathbf{X} containing M points

Sampled from 2D Gaussian distribution

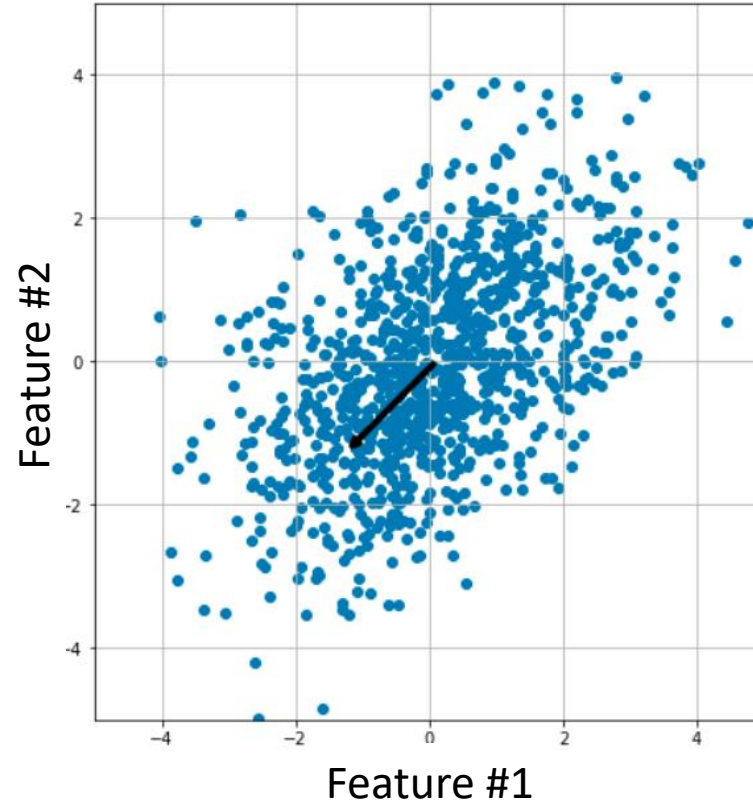
$$\mu_1 = 0$$

$$\mu_2 = 0$$

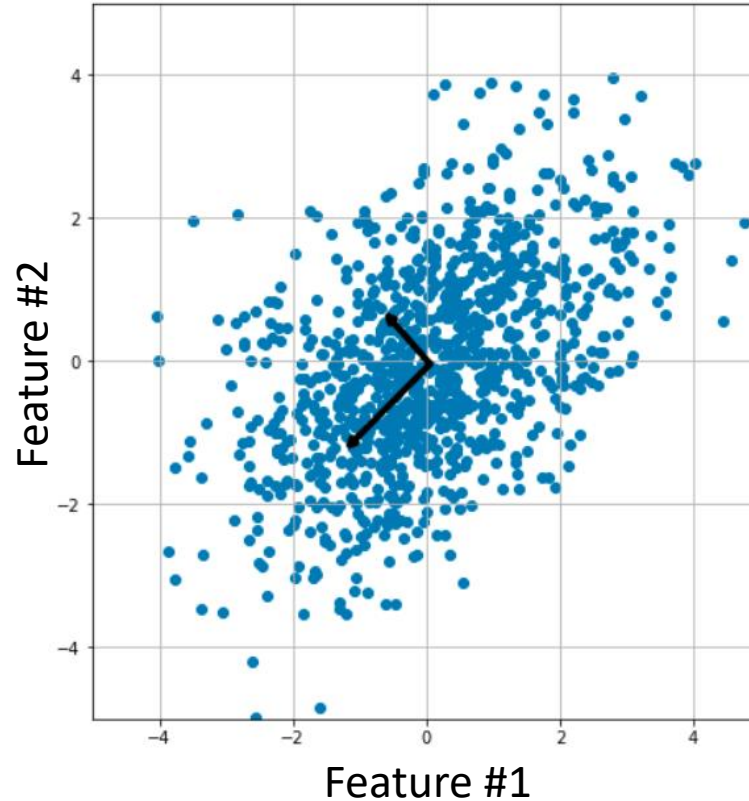
$$\Sigma = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$



Principal Component 1



Principal component are orthogonal to one another



Finding the principal components

- Center data by subtracting mean of each variable

$$\hat{X} = X - \bar{X}$$

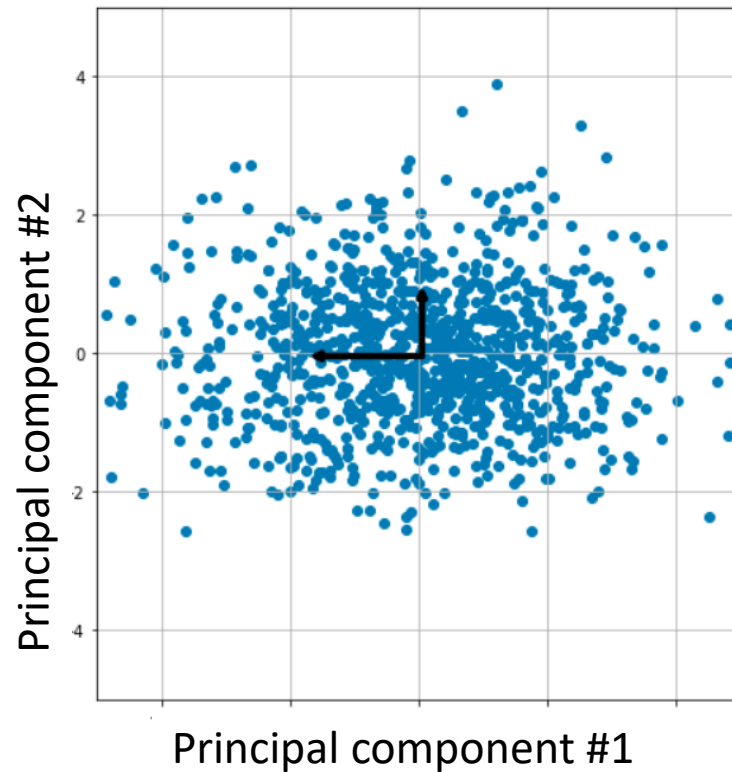
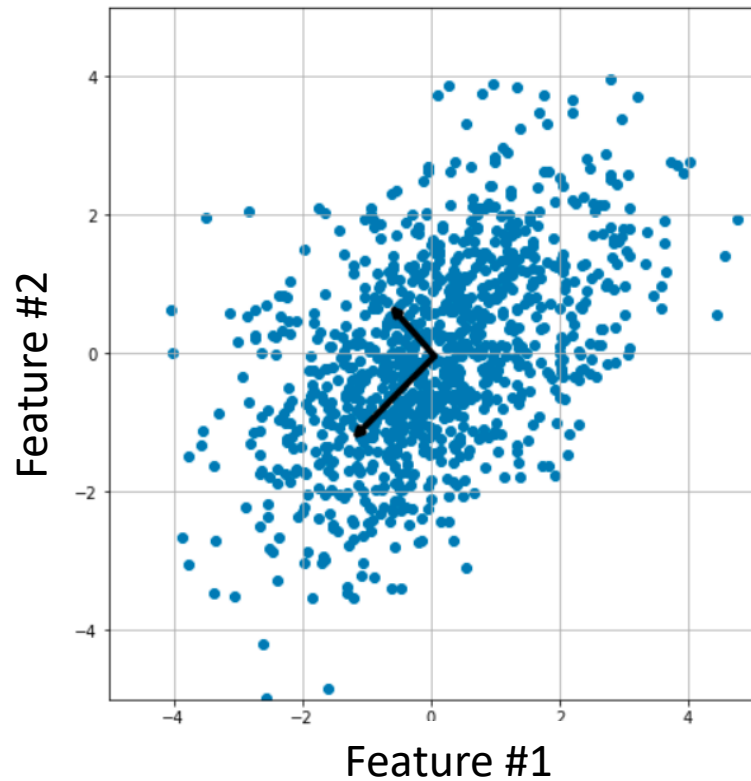
- Calculate covariance matrix

$$\Sigma = \frac{1}{M-1} X^T X$$

- Singular value decomposition (SVD) to find a matrix U that contains eigenvectors, ordered by largest to smallest variance

→ **principal components**

- Multiply \hat{X} with U to obtain X_{pca}

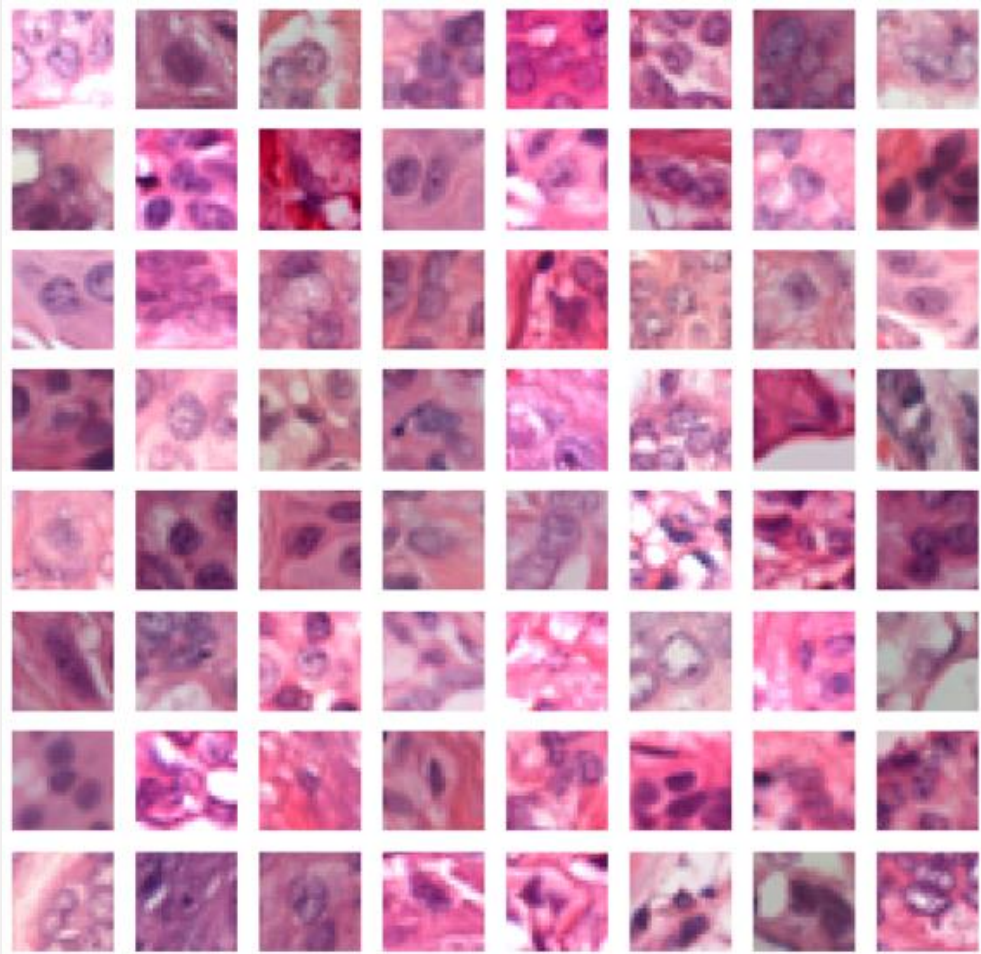


Dimensionality reduction

Instead of using all eigenvectors from \mathbf{U} we can select a set of n principal components.

For example, we can select the eigenvectors that contain 95% of the variance.

More info → PCA demo!



Questions so far?

Break!

Emperor Penguin (Antarctica)

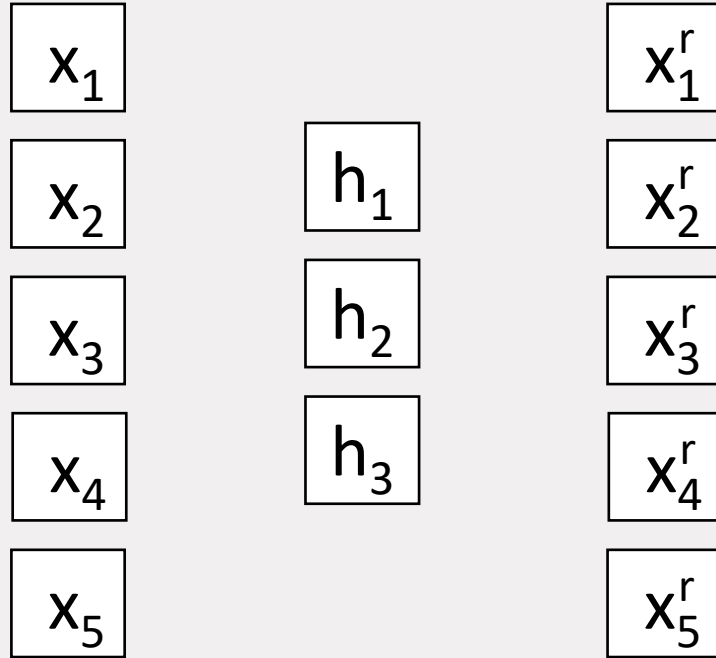
<https://www.independent.co.uk/news/science/>



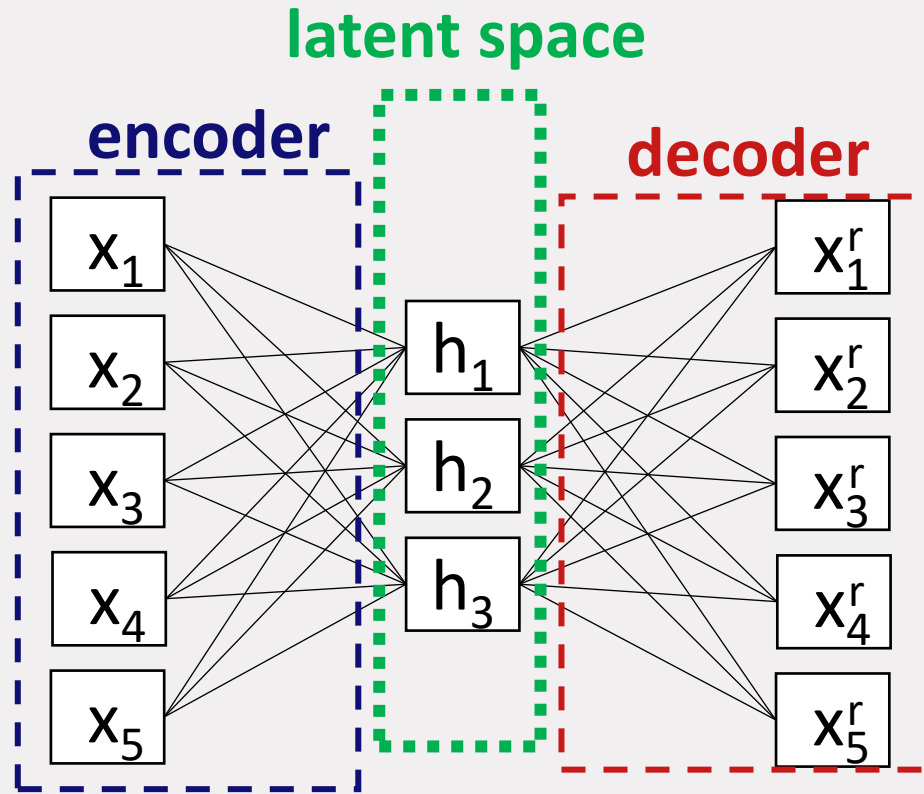
Lecture outline

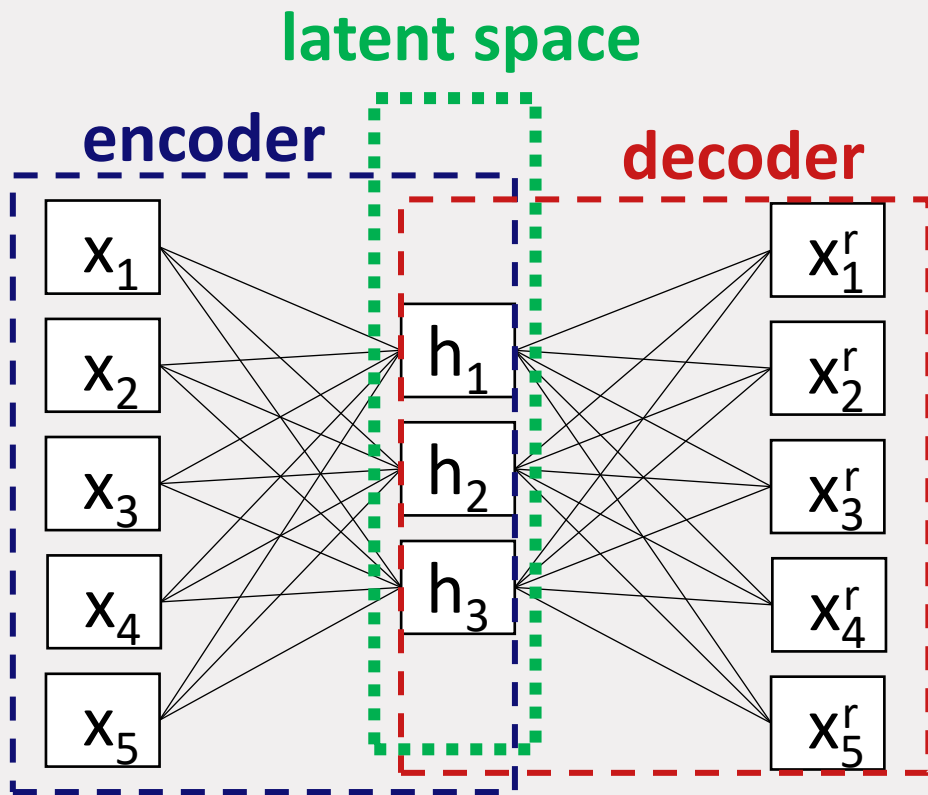
- Supervised vs unsupervised
- K-means
- Principal component analysis
- *Break (15 mins)*
- Auto-encoders
- Semi-supervised
- Self-supervised

Autoencoder



Goal = reconstruct input x_i ,
using a restricted number of
latent variables h_i





Encoder:

$$h = f(x)$$

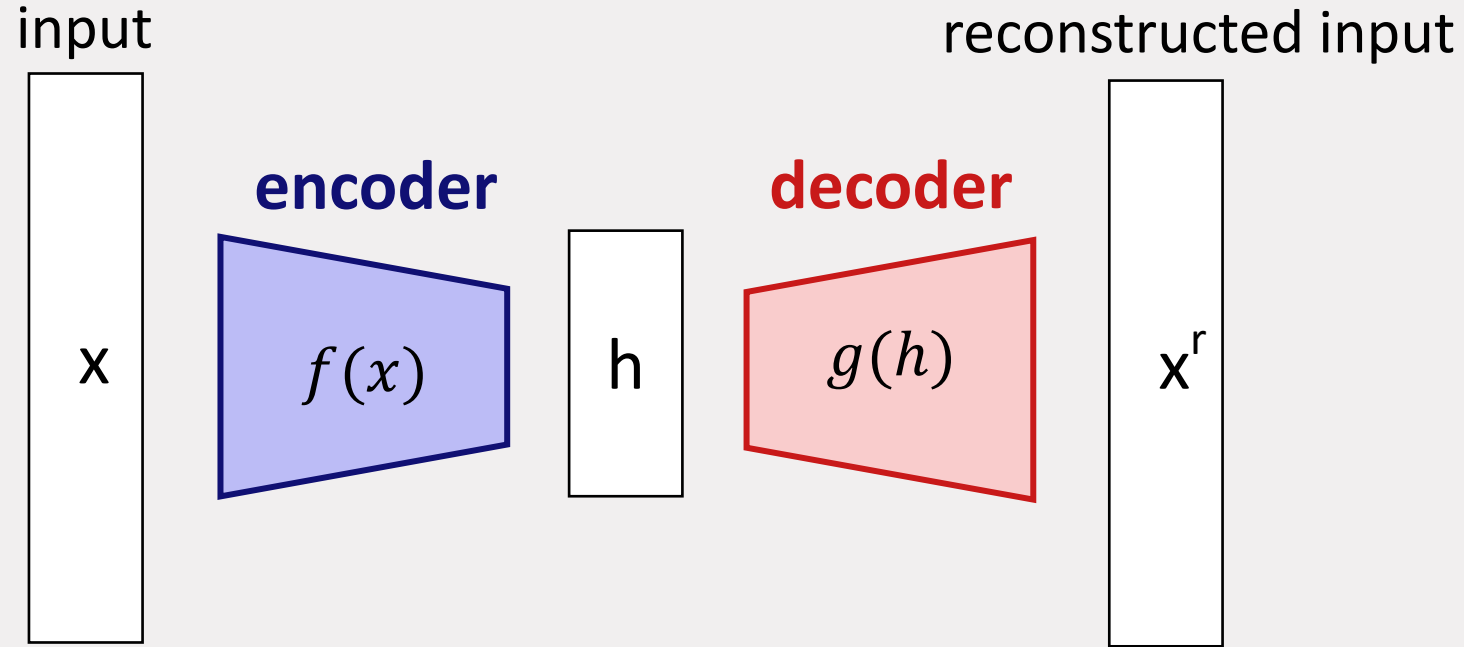
Decoder:

$$x^r = g(h)$$

Penalize dissimilarity

$$L(x, g(f(x)))$$

Autoencoder – a more general representation



Autoencoder

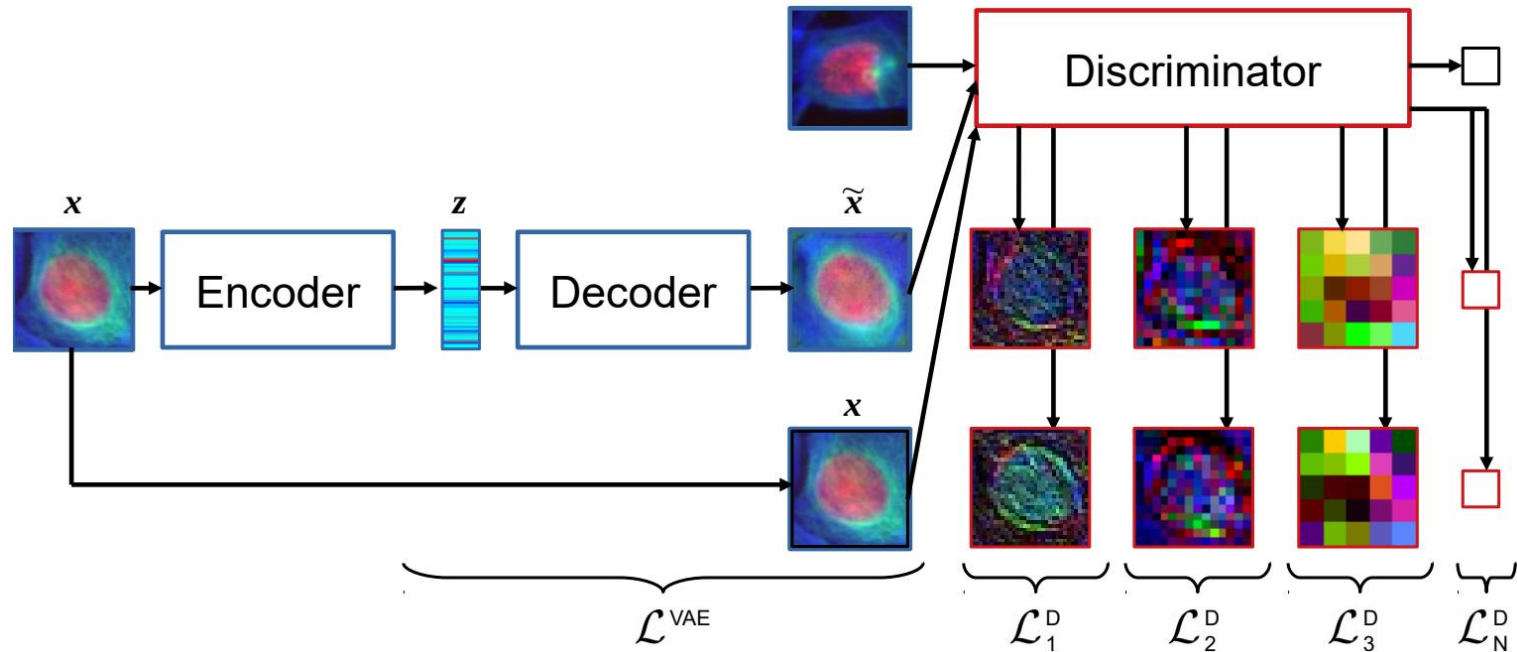
- Encoder/decoder can be simple or complex
- For example: a deep convolutional neural network

Applications

- Dimension reduction!
- Latent variables can be used for secondary objective, e.g. classification
- Denoising (by adding noise to the input and reconstructing the original)
- Generative models – generating new (image) data

Example: Unsupervised representation learning to capture single-cell phenotypic variation

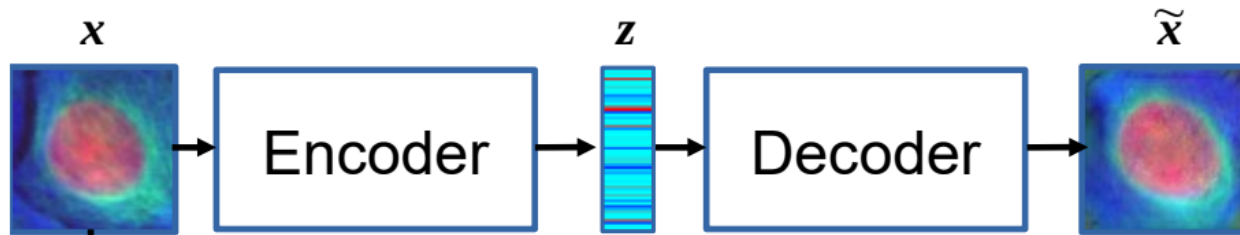
Lafarge et al., MIDL 2019



Example: Unsupervised representation learning to capture single-cell phenotypic variation

Lafarge et al., MIDL 2019

- Human MCF7 cells
- Treated with different compounds
- Latent space representation (here called \mathbf{z}) used to predict compounds with 1-nearest-neighbors



'Supervised' learning terminology

**Supervised
methods**

*Weakly
Supervised*

*Semi-
supervised*

**Unsupervised
methods**

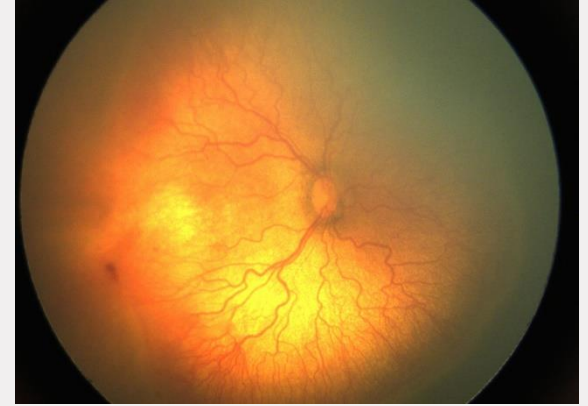
*Self-
supervised*

Semi-supervised

Global labels

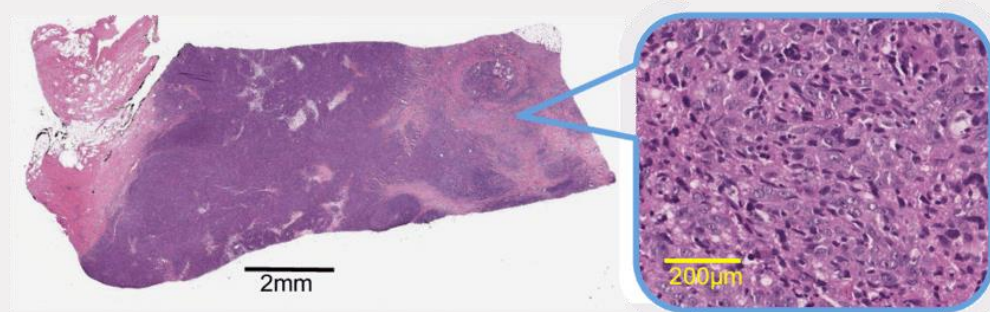
E.g. A single diagnosis is based on a series of retinal fundus images

NIH National Eye Institute



Partially labeled data

E.g. Only cells in part of a whole slide histopathology image are segmented



Analysis of Histopathology, Jimenez-del-Toro

Some remarks on semi-supervised learning

- Fewer labeled data needed
- For many medical applications data is still limited, e.g. because disease is rare
- Use knowledge from a related task
- Humans also learn in a semi-supervised fashion

Summary

- Supervised versus unsupervised
- Finding structures (e.g. K-means)
- Dimension reduction (e.g. PCA, autoencoders)
- Semi-supervised learning

