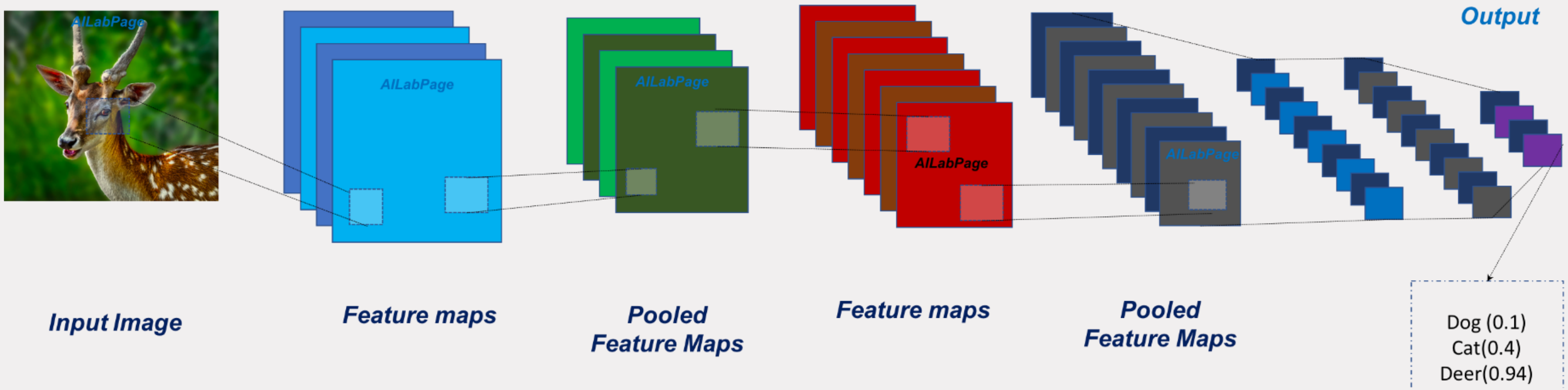
Unsupervised machine learning (8DC00)

Cian Scannell

Previous lecture

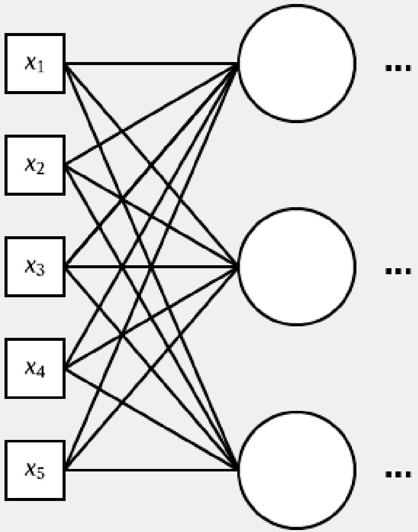
- Convolutional neural networks (CNN)



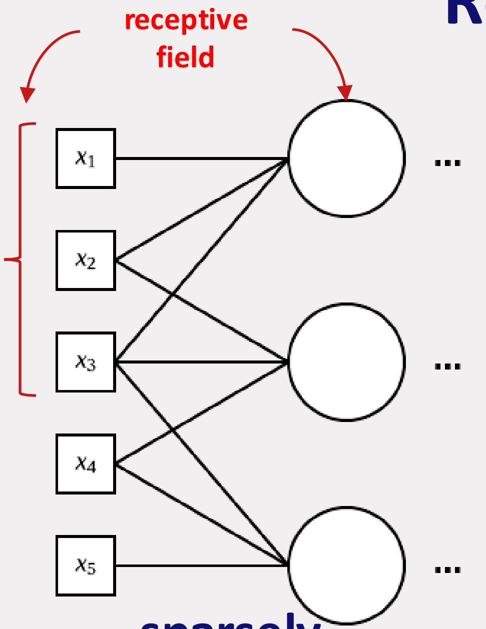
- By now: students can design a (simple) CNN

Image from: https://vinodsblog.com/

Previous lecture

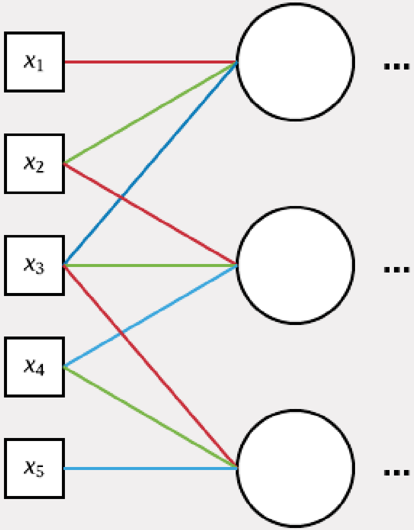


"regular" NN
15 weights



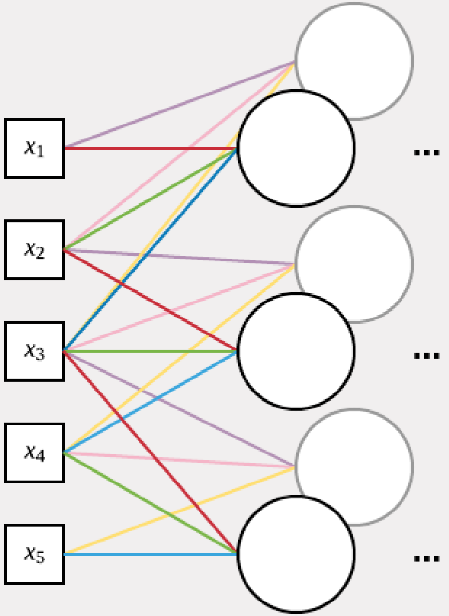
sparsely connected NN
9 weights

Reducing # of weights



shared weights
3 weights

Previous lecture



two sets shared weights

6 weights

$$[a_{1,1} \quad a_{1,2} \quad a_{1,3}] = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] * [w_{1,1} \quad w_{1,2} \quad w_{1,3}]$$

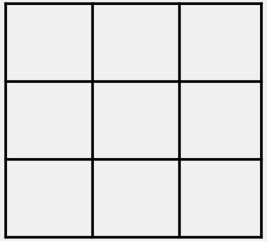
$$[a_{2,1} \quad a_{2,2} \quad a_{2,3}] = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5] * [w_{2,1} \quad w_{2,2} \quad w_{2,3}]$$

$[w_{1,1} \quad w_{1,2} \quad w_{1,3}]$, and $[w_{2,1} \quad w_{2,2} \quad w_{2,3}]$ are **convolution kernels**. They extract features.

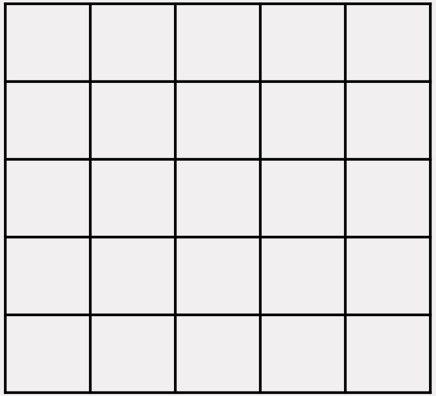
Previous lecture: Kernel size

1 x 1

3 x 3

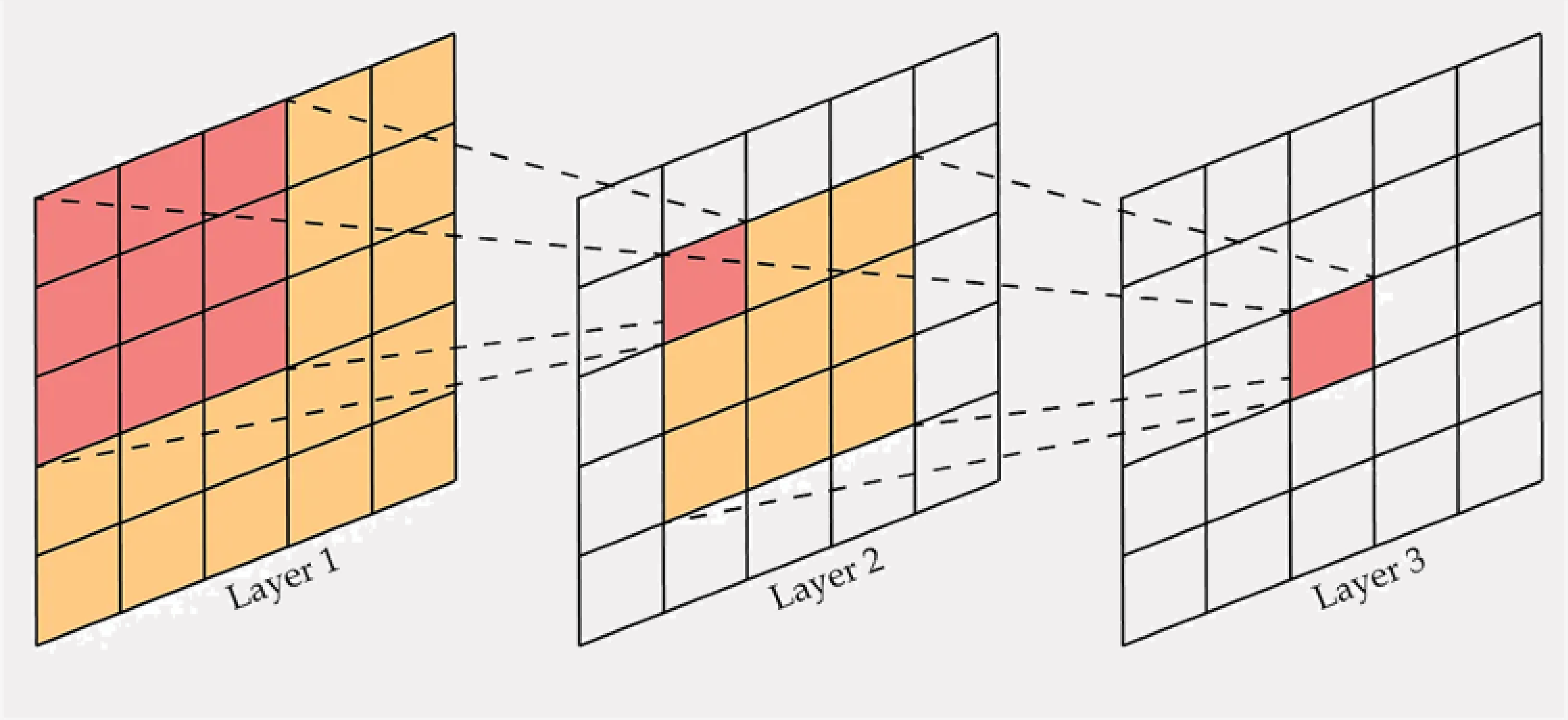


5 x 5



- More weights → more information
- More computations / memory
- Receptive field

Receptive field



Previous lecture: Max-pooling

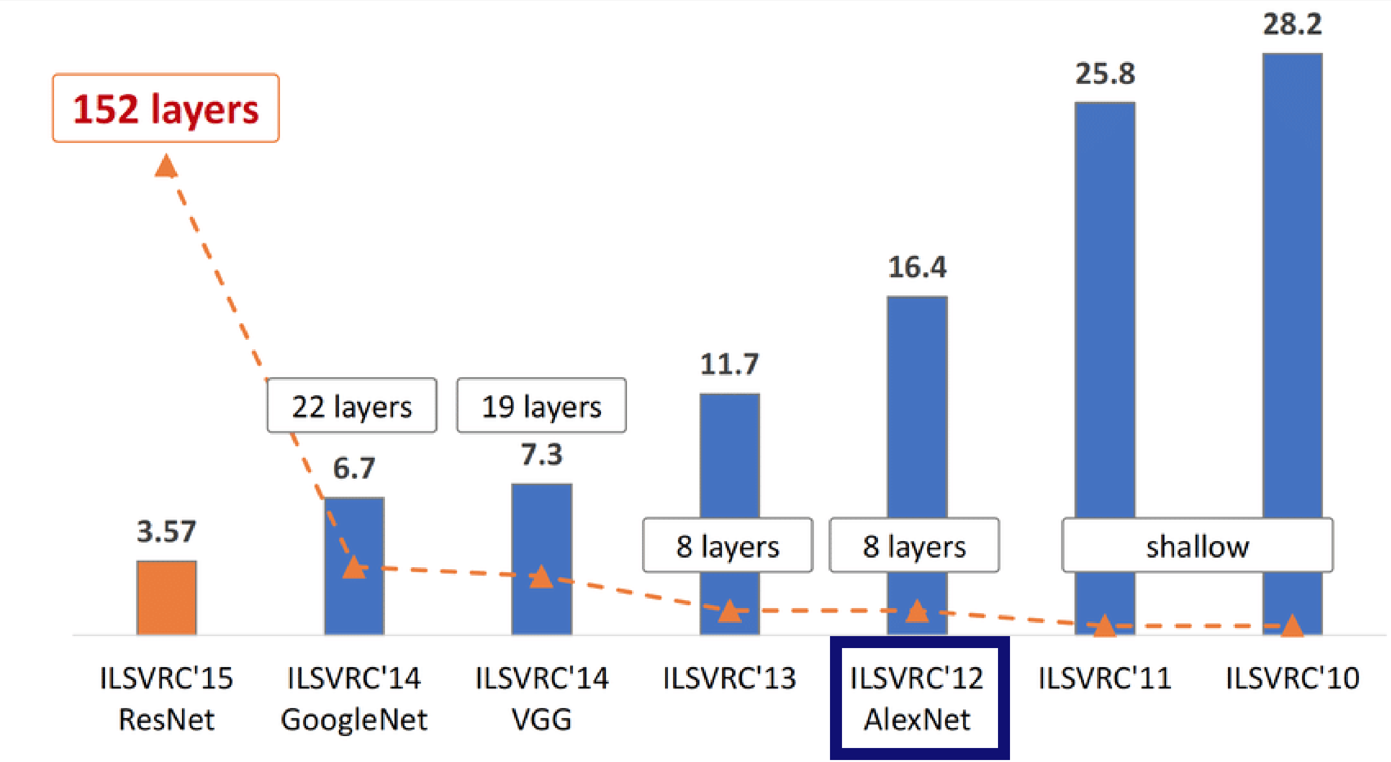
- Reduce size of feature space
- Maximum of features
- Typical kernel size = 2 x 2

Addition to last lecture:

Stride = step size of the sliding window

- Typically 1 for convolutions
- Typically equal to kernel size for pooling operations

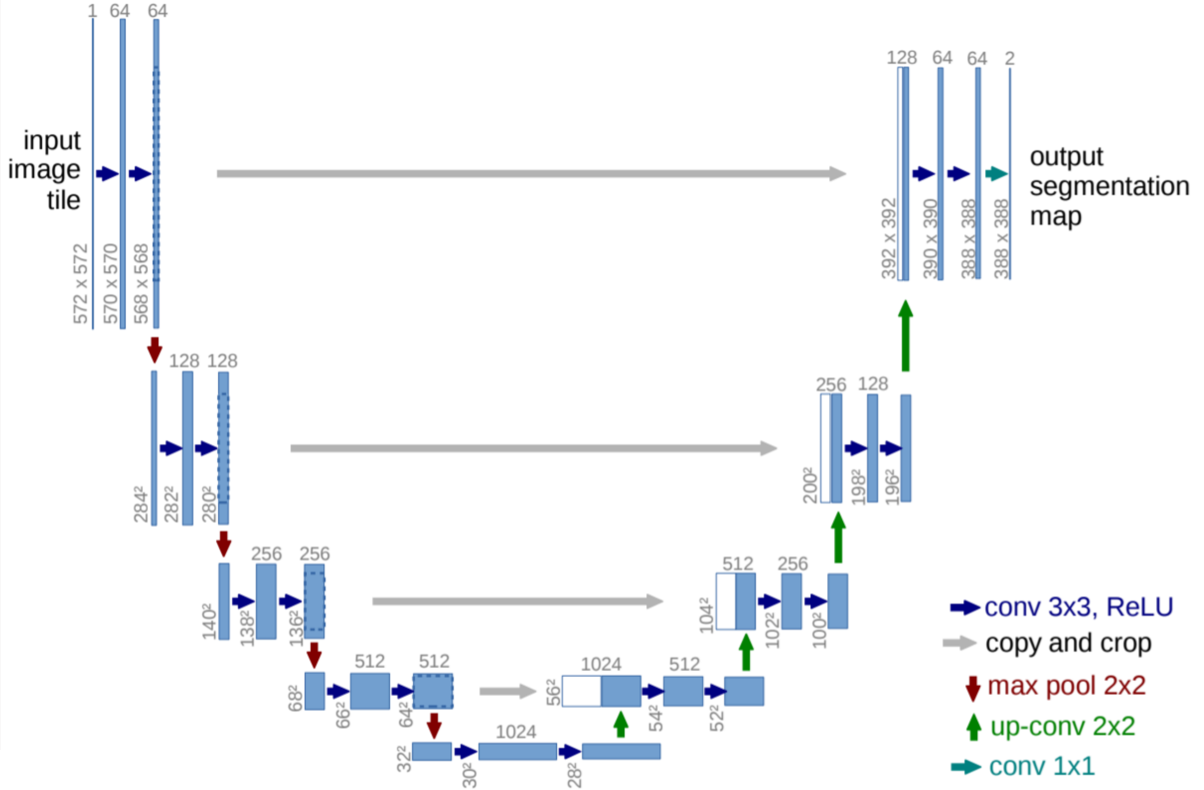
Error rates ImageNet (top 5 accuracy)



TU/e

U-Net: Segmentation

Ronneberger et al., MICCAI 2015



Solving machine learning problems

- Type of problem?
- Preprocessing steps?
- What kind of ‘labels’ do you need?
- What should the output look like?
- What should be the activation function in the final layer?
- What model do you choose?
- How do you evaluate performance?
- Other considerations?

Previous examples required labeled data

learning from labeled data = **supervised training**

Learning outcomes

- Student can describe the difference between **supervised** and **unsupervised learning** and name advantages of both methods
- Student can apply **K-means** to find **clusters** in data
- Student can explain **Principal Component Analysis** and motivate **dimensionality reduction**
- Student can explain the concept of an **Autoencoder** and motivate why abstract features (latent variables) can be used for a secondary task.

Learning strategies

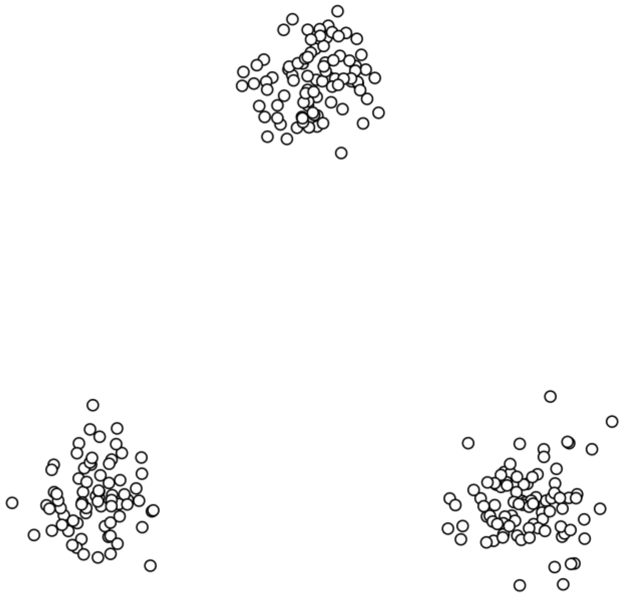
Supervised
Learning from examples (=training data) that are labeled with their desired outputs. The goal is to learn general rules that maps inputs to outputs.

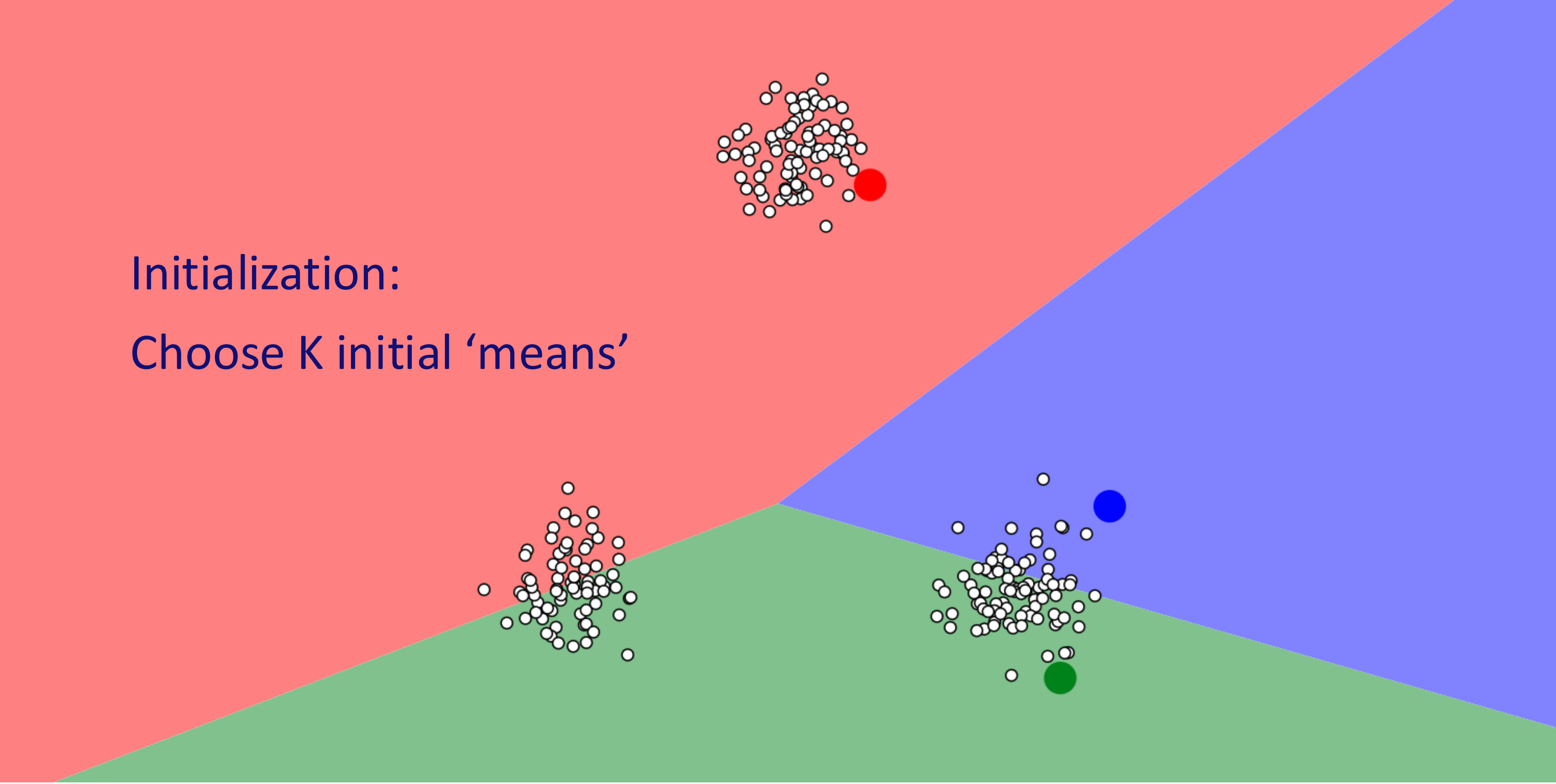
Unsupervised
Learning from examples without labels. The goals are:

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

Finding clusters using K-means

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





Initialization:

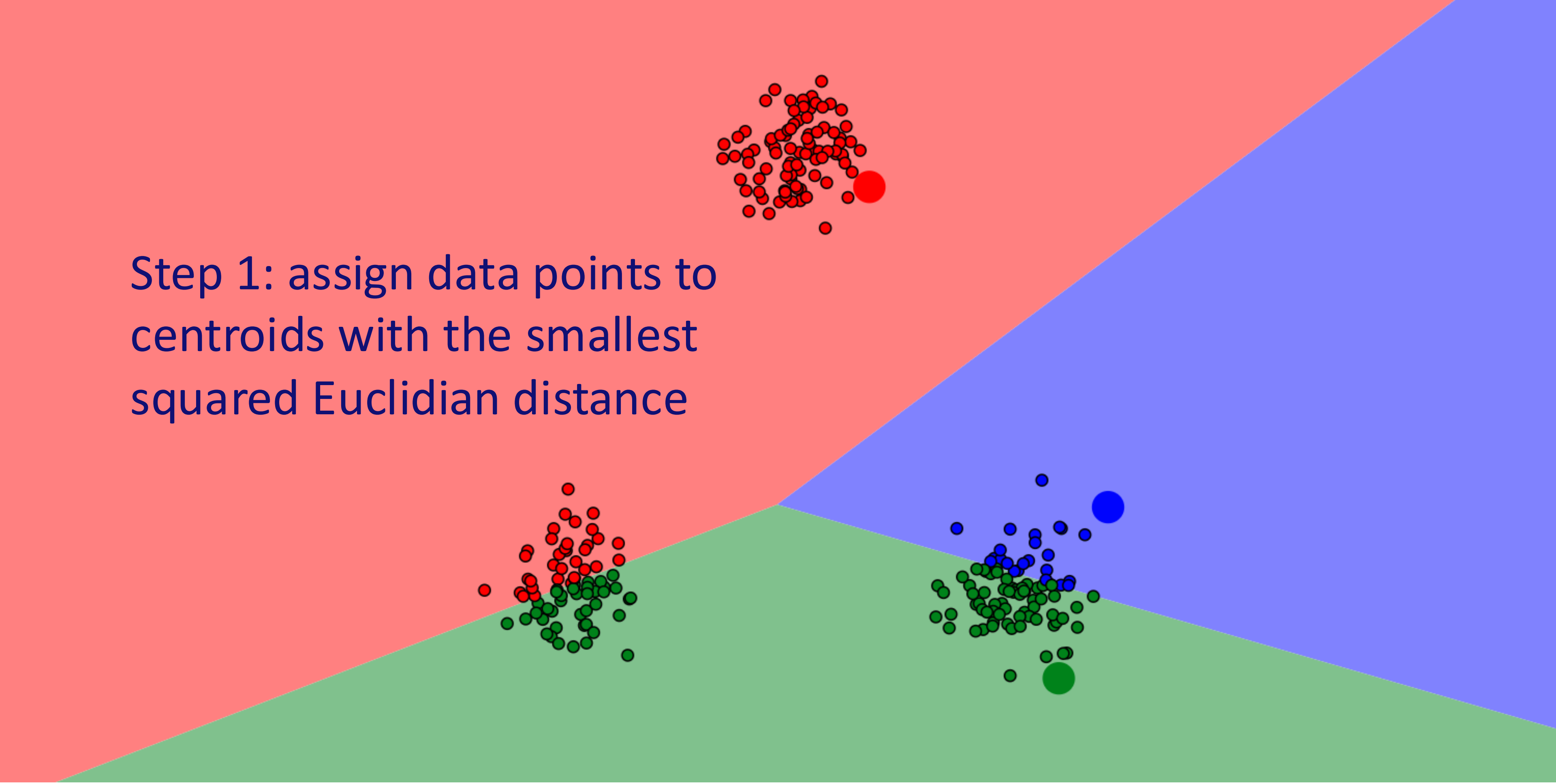
Choose K initial ‘means’

K-means – evaluate clustering performance

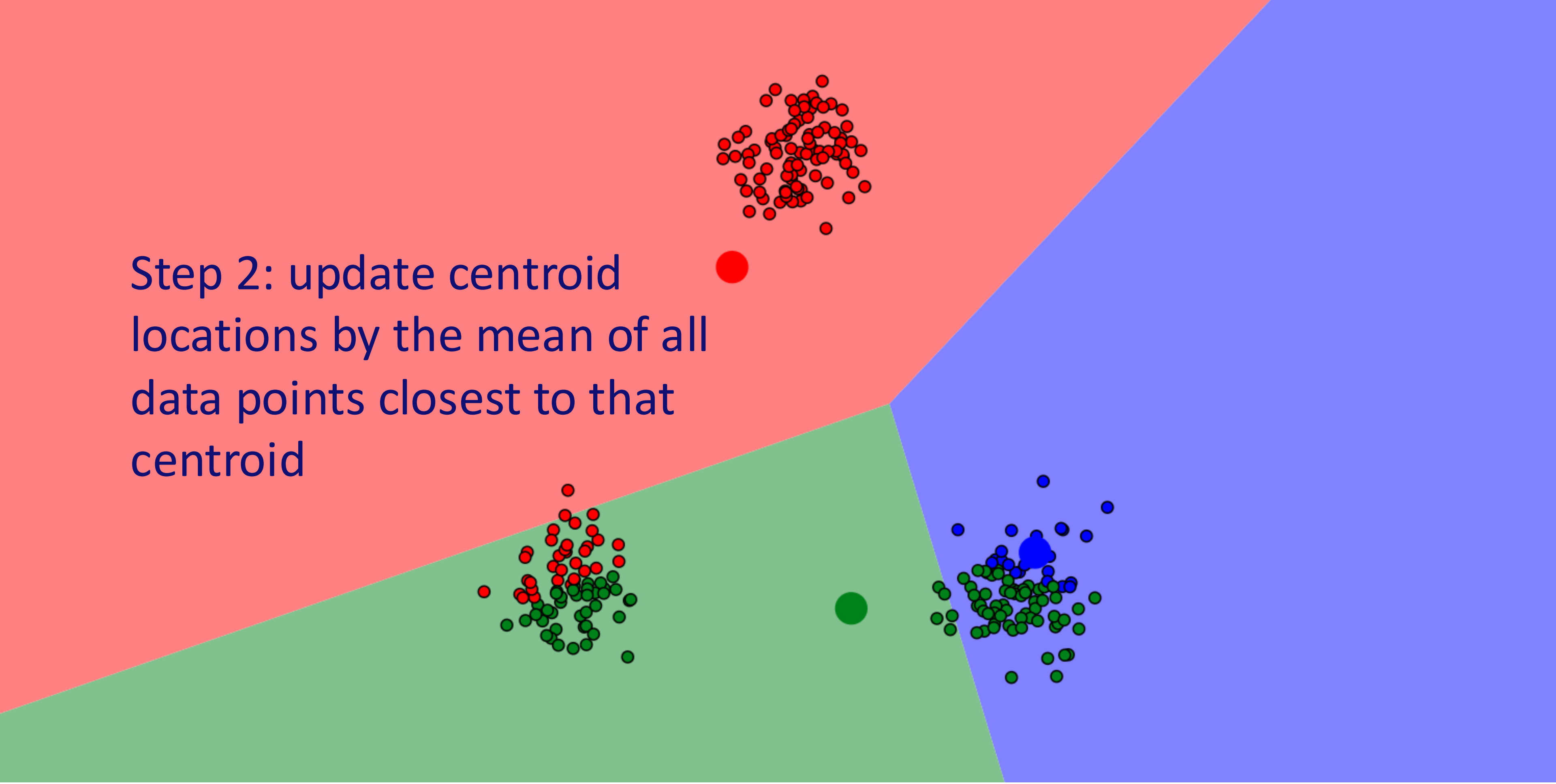
Average squared Euclidean distance between each point and the closest cluster:

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

x_i are the points, W are the cluster centroids



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

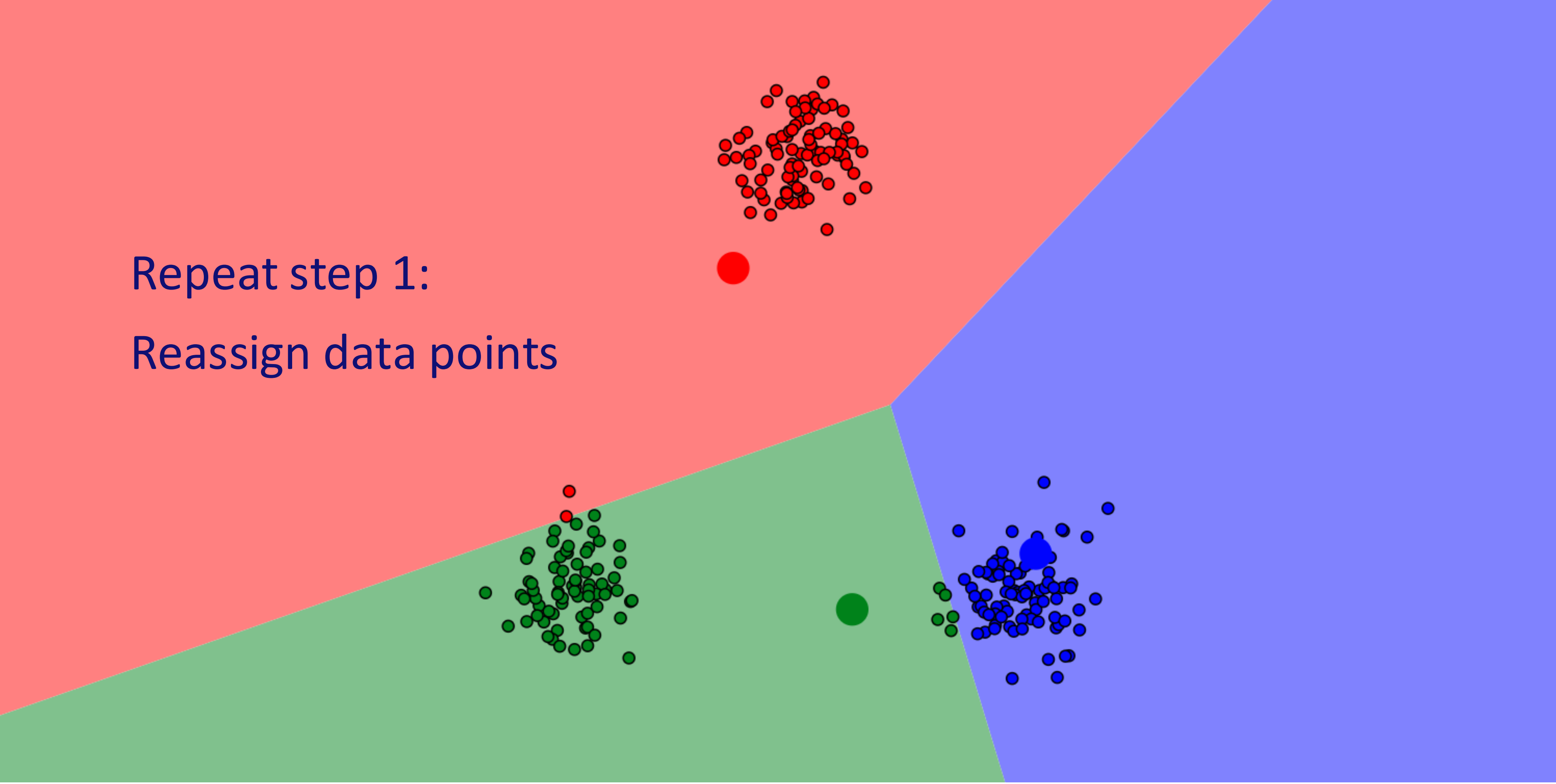


Step 2: update centroid locations by the mean of all data points closest to that centroid

TU/e

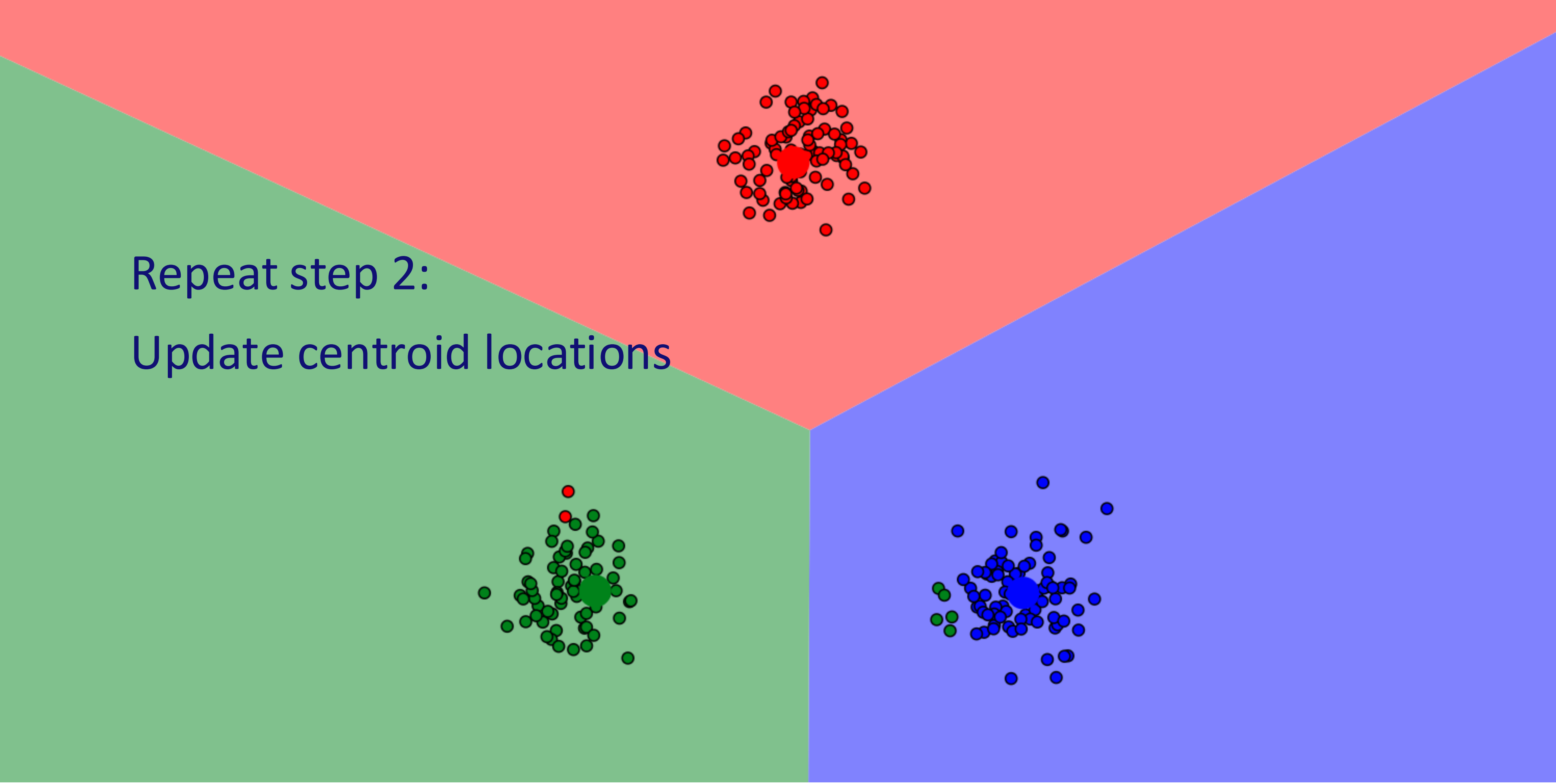
Repeat step 1:

Reassign data points



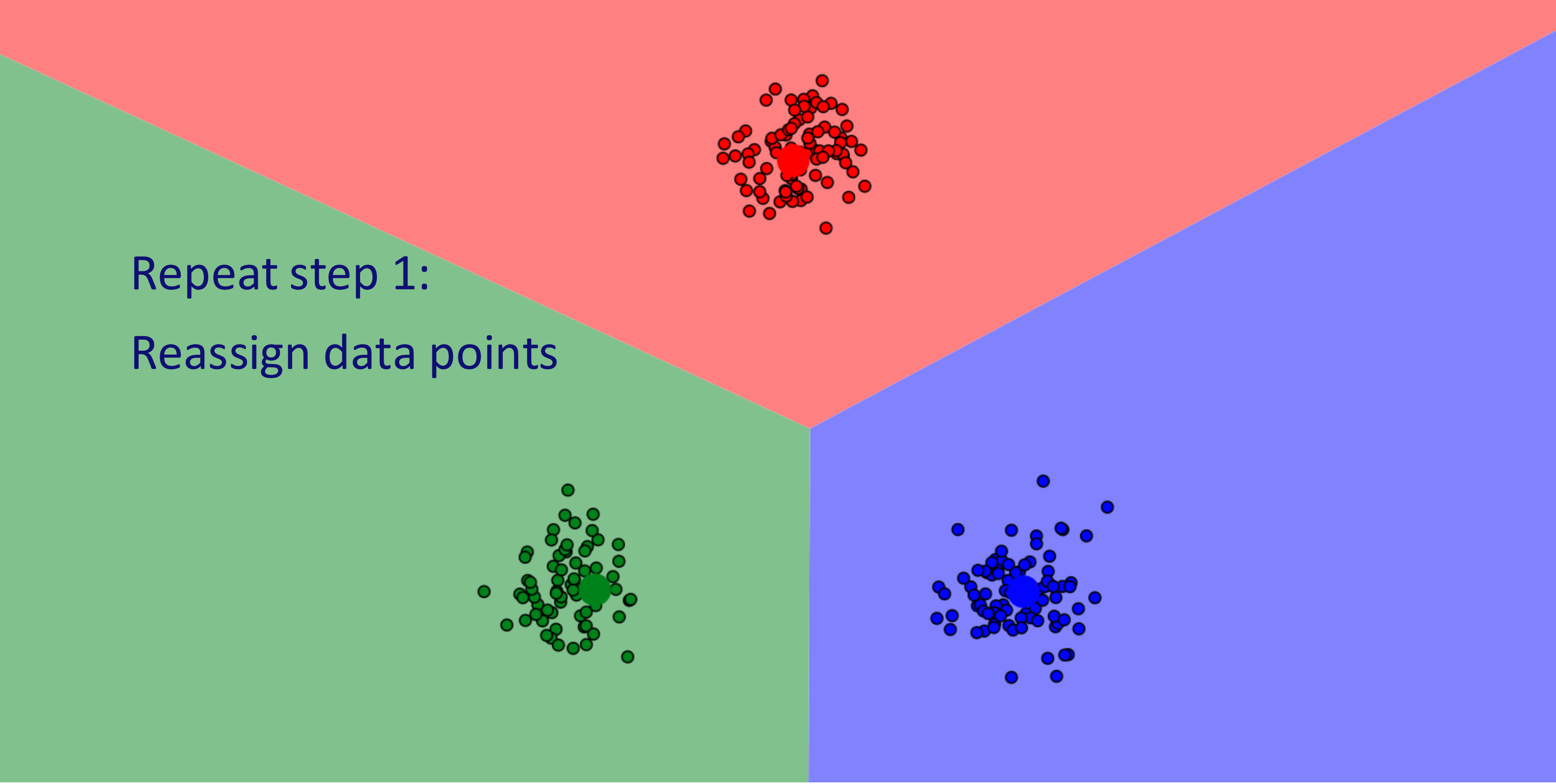
Repeat step 2:

Update centroid locations



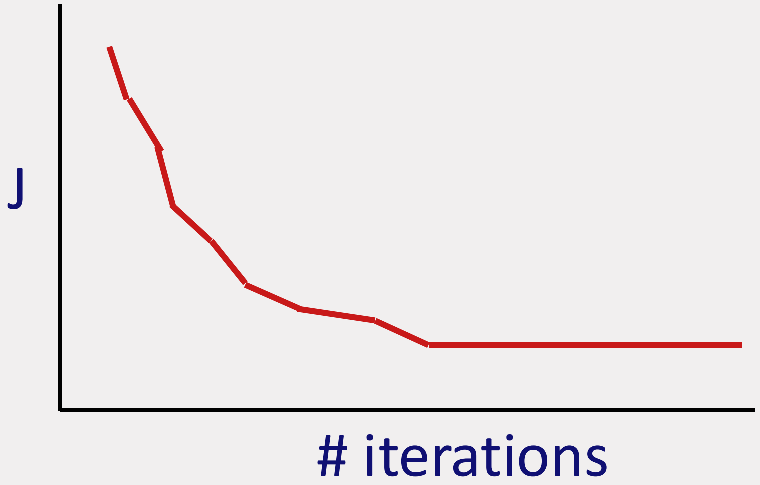
Repeat step 1:

Reassign data points



When do we stop?

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

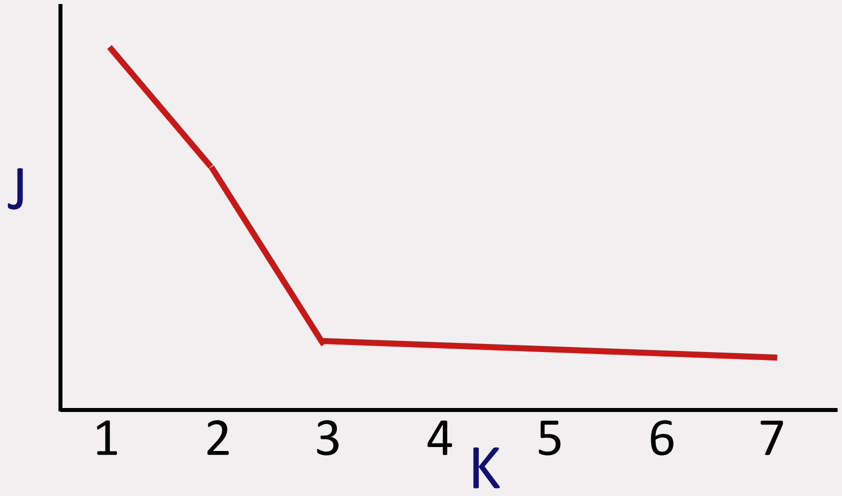


How do we choose initial centroid locations?

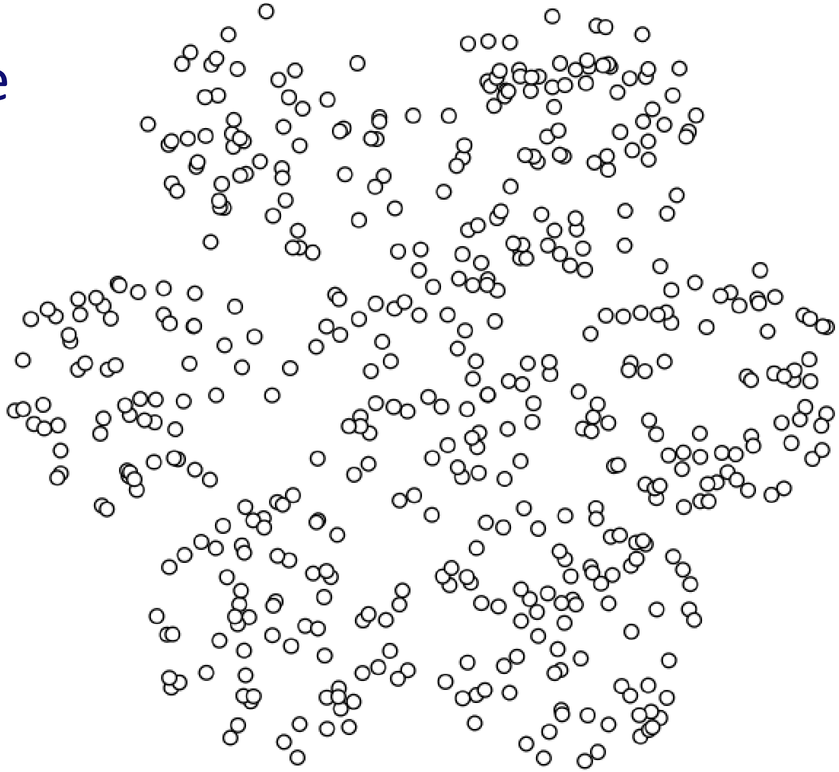
- Random
- Farthest points
- Manual?
  - Supervised
  - Difficult for high-dimensional data

How do we choose K?

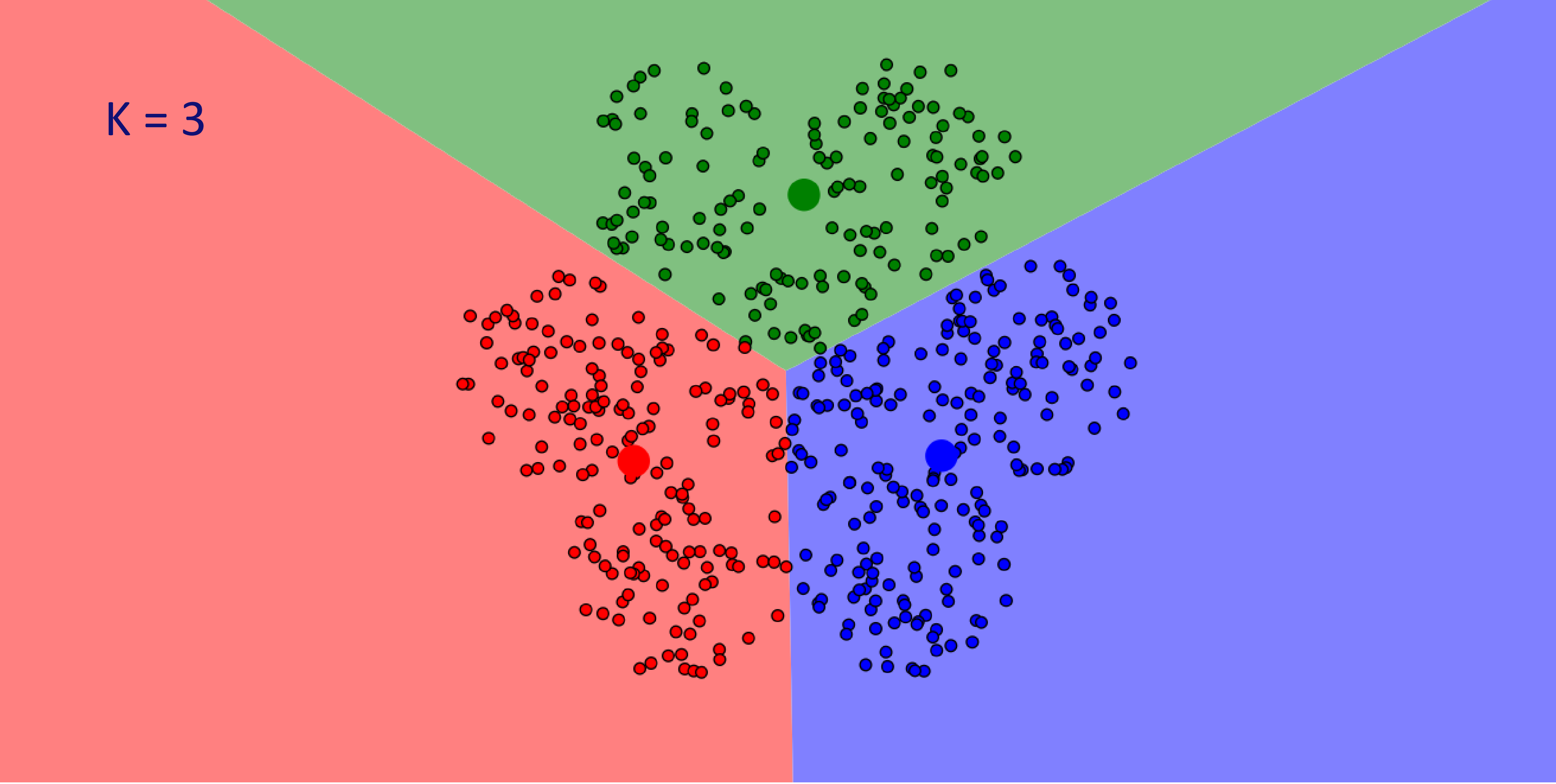
- K (=number of means) is a hyperparameter



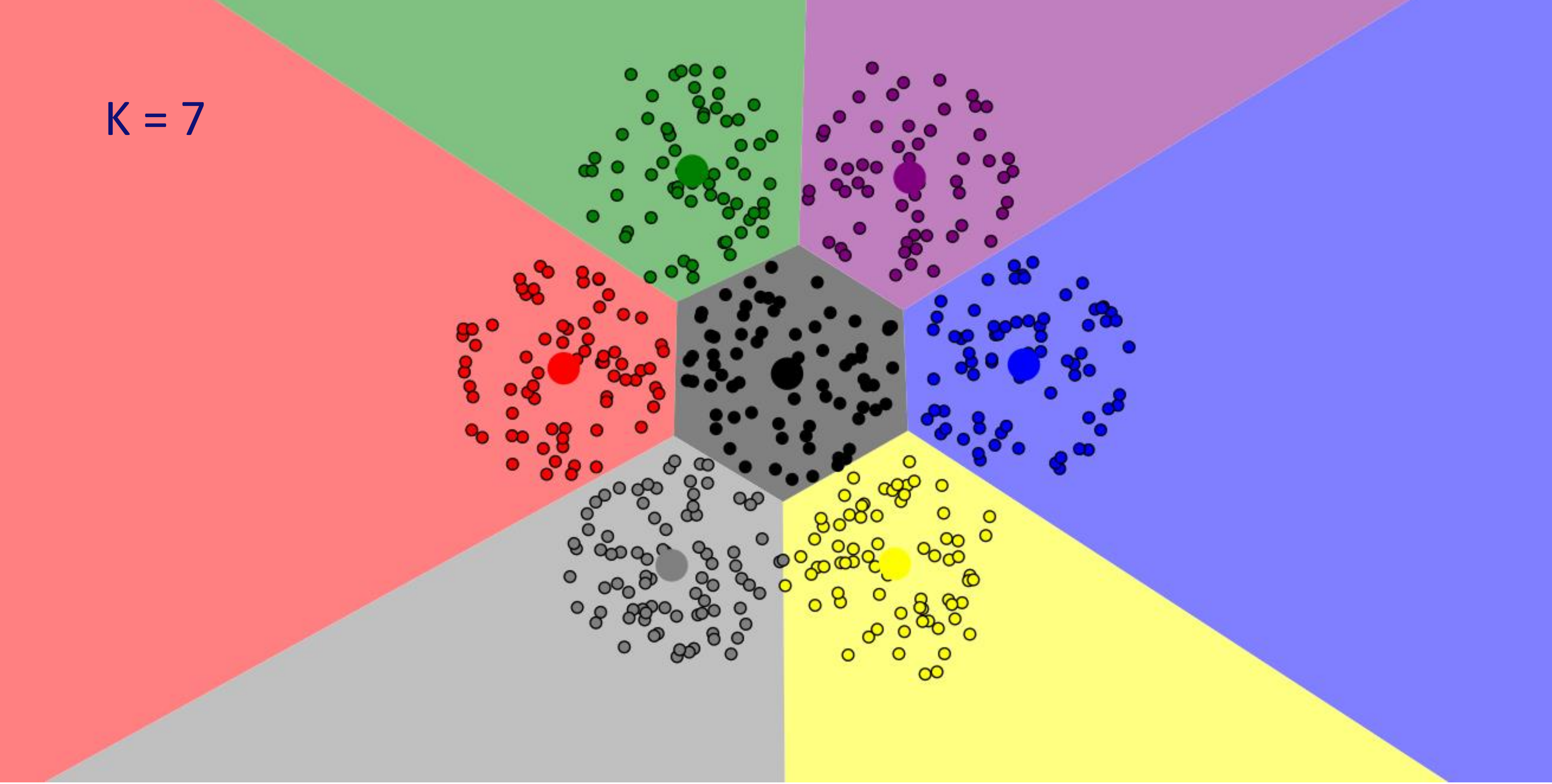
Another example

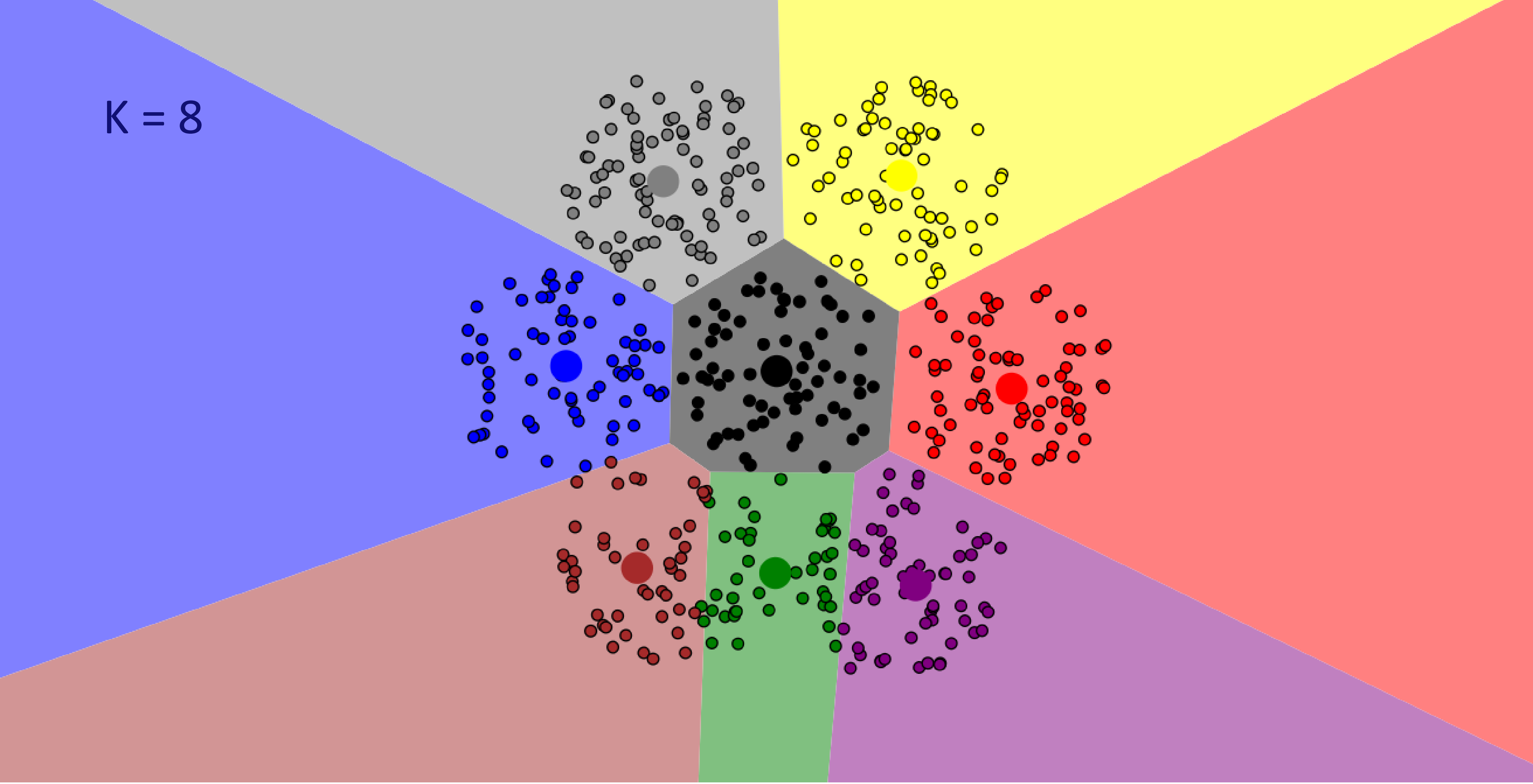


K = 3



K = 7




K = 8

Questions so far?

Principal Component Analysis (PCA)

Goal: Finding the principal 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!

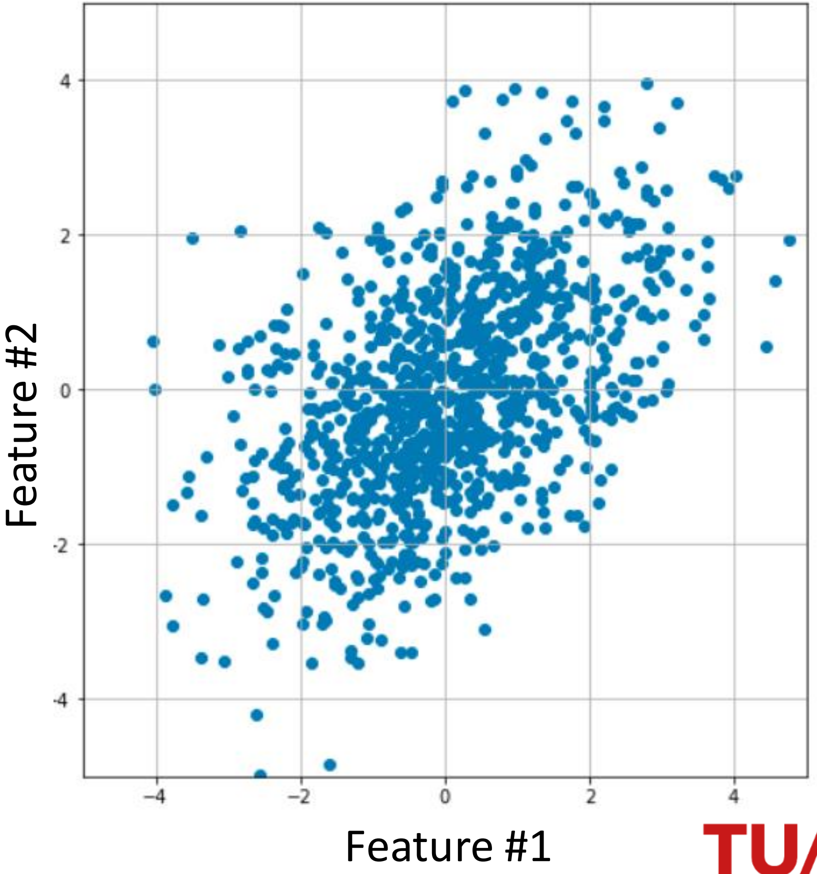
Principal Component Analysis

Example data set

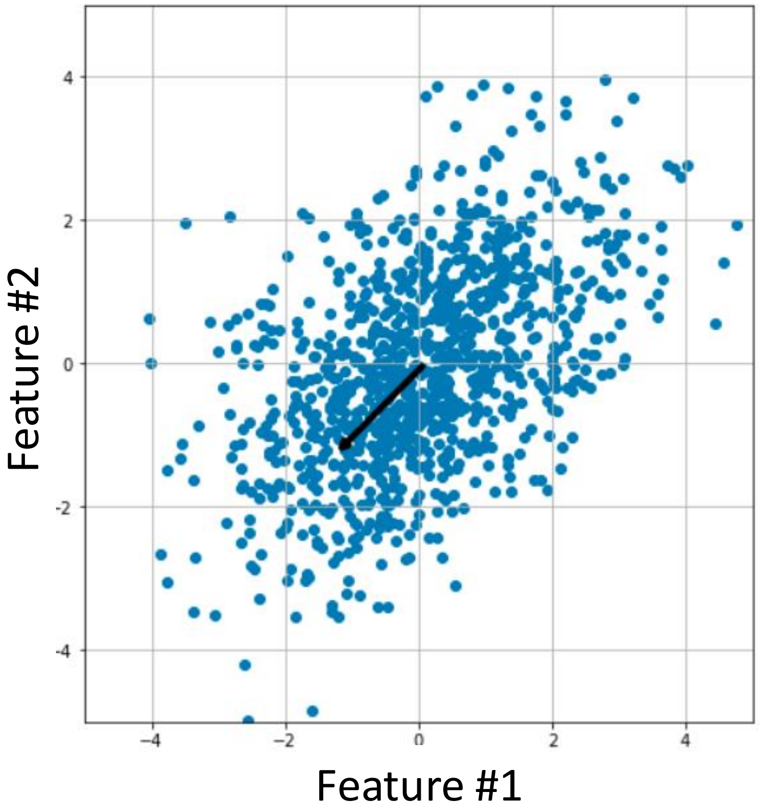
M-by-2 matrix $\boldsymbol{X}$ containing M points

Sampled from 2D Gaussian distribution

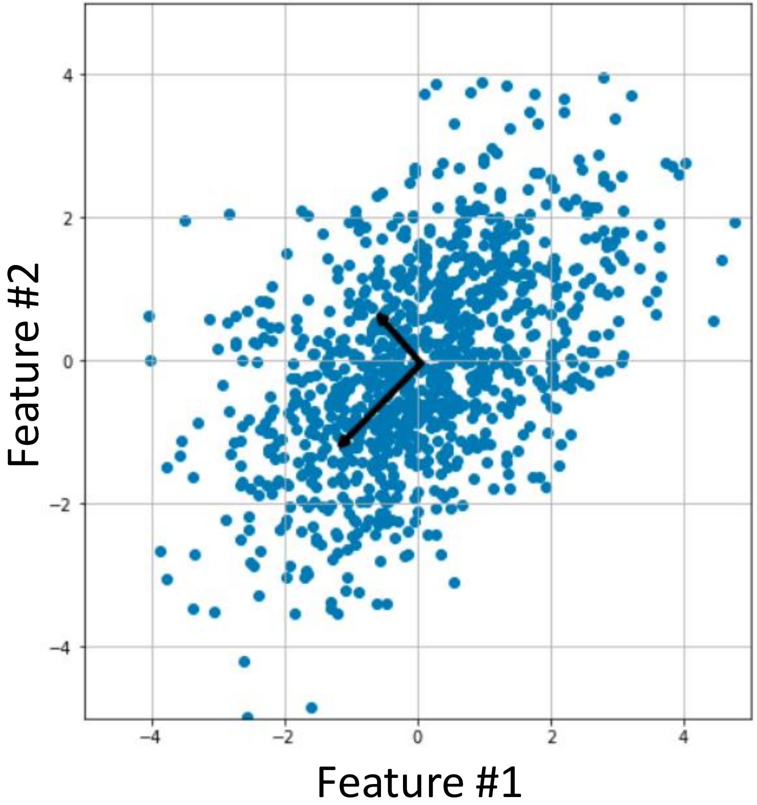
$$\mu_1 = 0$$
$$\mu_2 = 0$$
$$\sum = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$



Principal Component 1



Principal component are orthogonal to one another



Finding the principal components

- Center data by subtracting mean of each variable

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

- Calculate covariance matrix

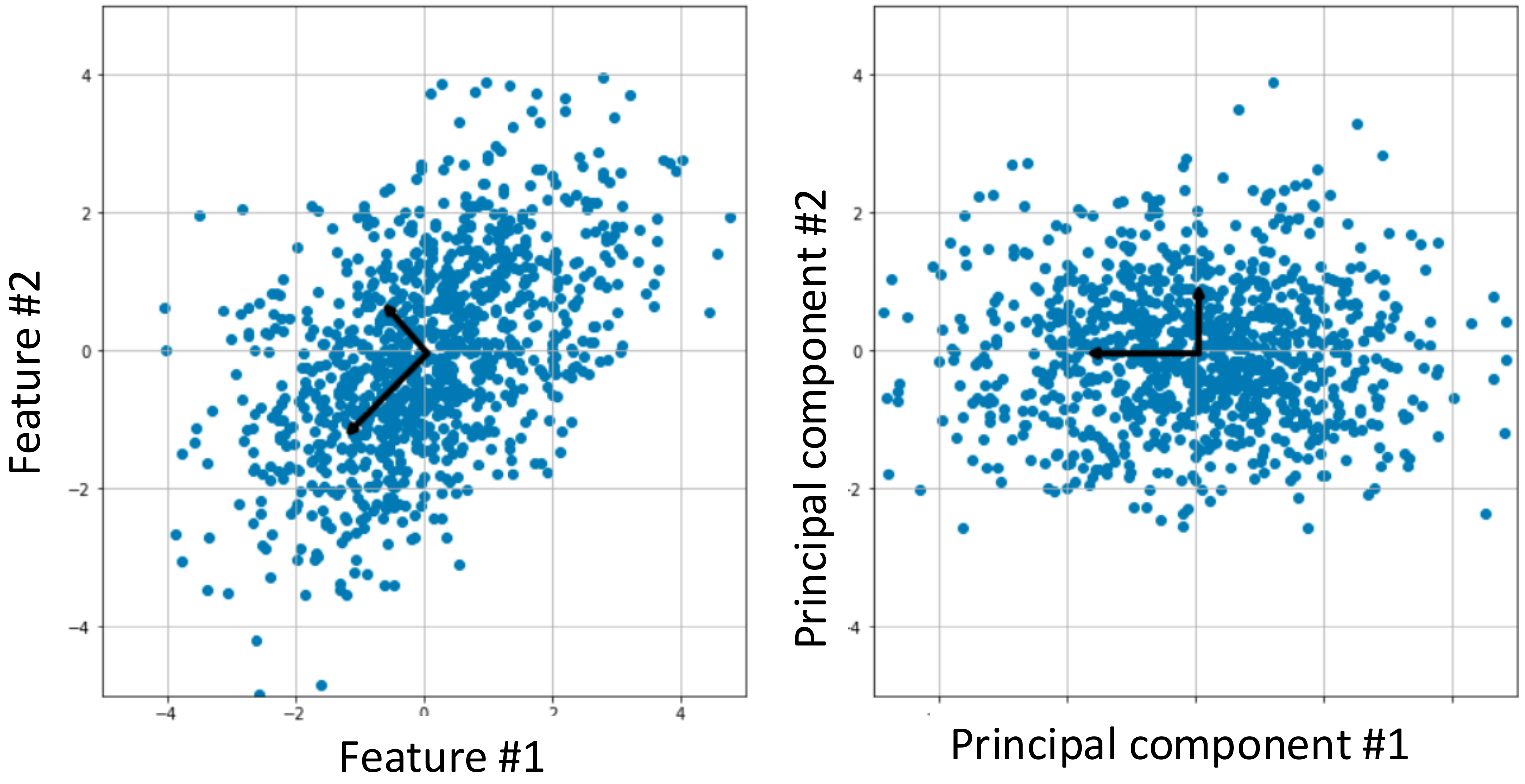
$$\Sigma = \frac{1}{M-1}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$$

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

→ **principal components**

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

TU/e



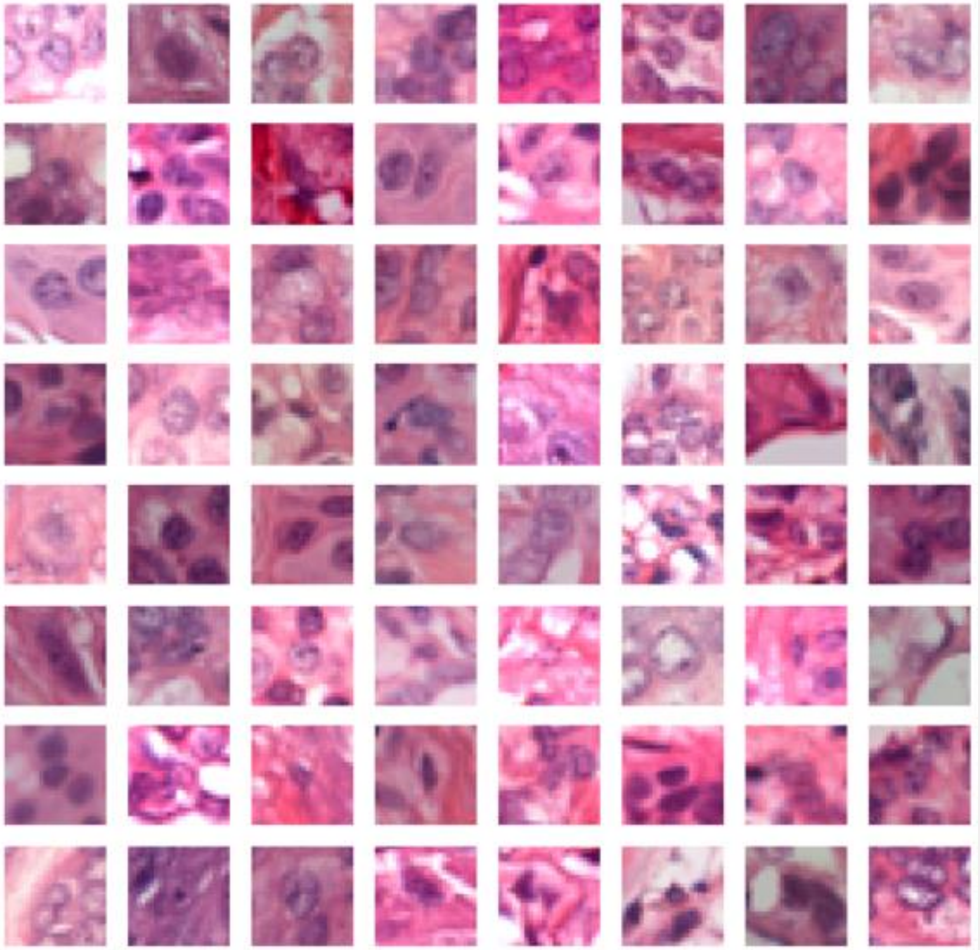
Feature #2
4
2
0
−2
−4
−4
−2
0
2
4
Feature #1
Principal component #2
4
2
0
-2
-4
Principal component #1

Dimensionality reduction

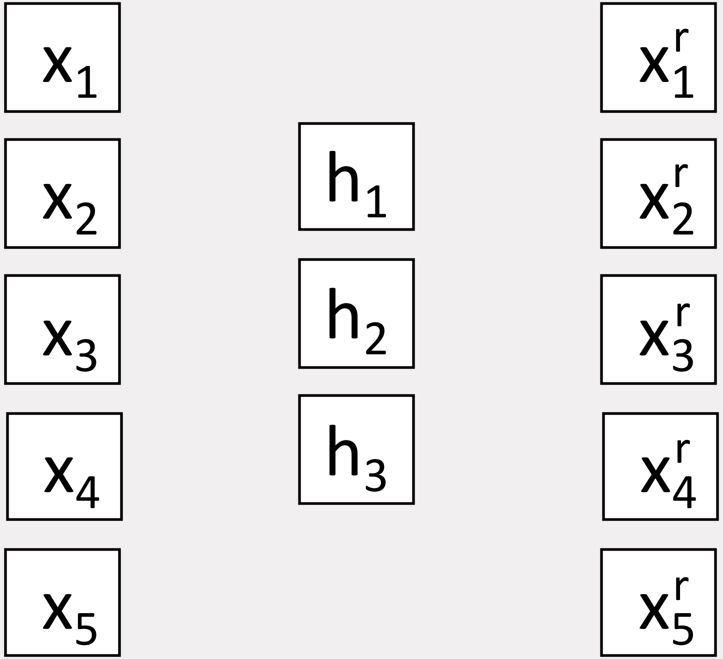
Instead of using all eigenvectors from $\boldsymbol{U}$ we can select a set of n principal components.

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

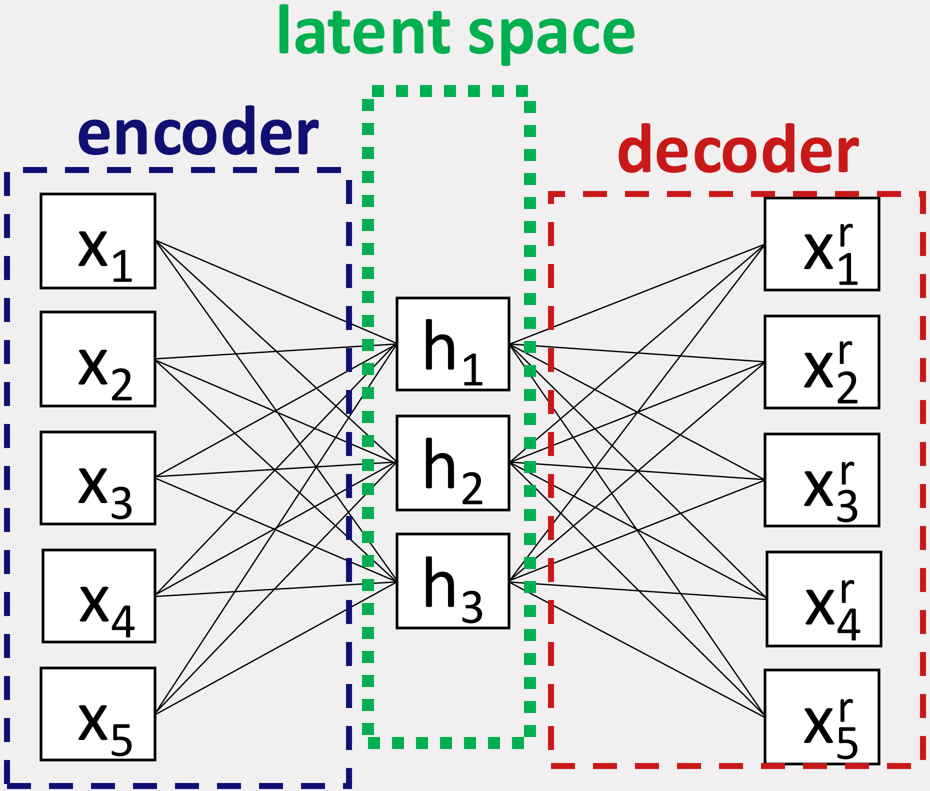
More info → PCA demo!

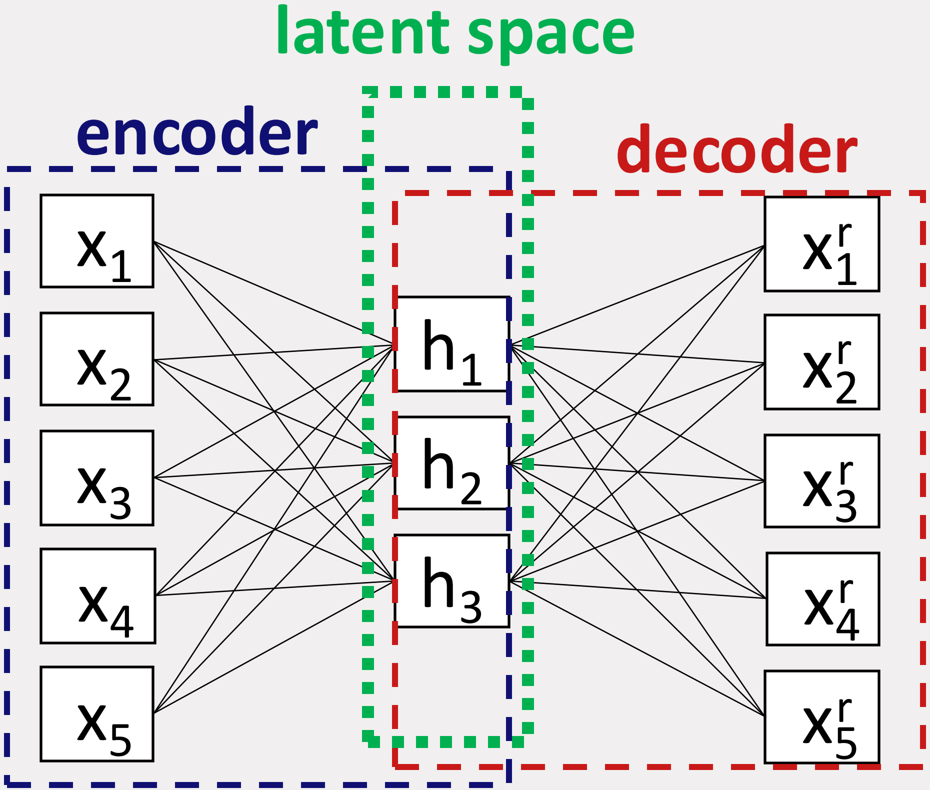


Autoencoder



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


latent space
encoder
decoder
x_1
x_2
x_3
x_4
x_5
h_1
h_2
h_3
x_1^r
x_2^r
x_3^r
x_4^r
x_5^r



Encoder:

$$h = f(\boldsymbol{x})$$

Decoder:

$$x^r = g(\boldsymbol{h})$$

Penalize dissimmilarity

$$L(\boldsymbol{x}, g(f(\boldsymbol{x})))$$

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

‘Supervised’ learning terminology

Supervised methods

Weakly Supervised

Semi-supervised

Unsupervised methods

Self-supervised

TU/e

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