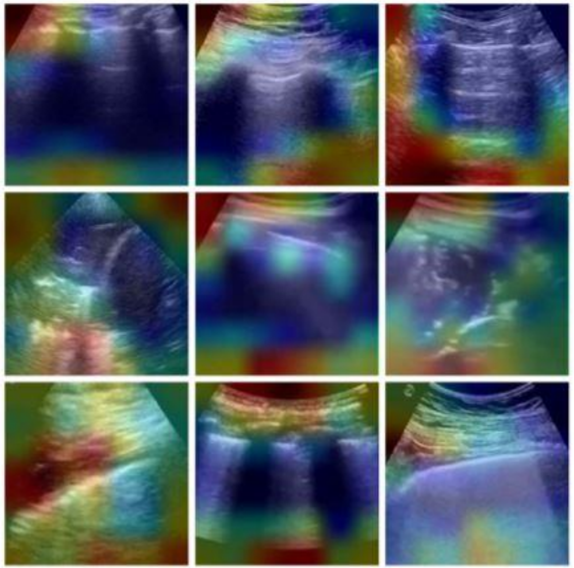
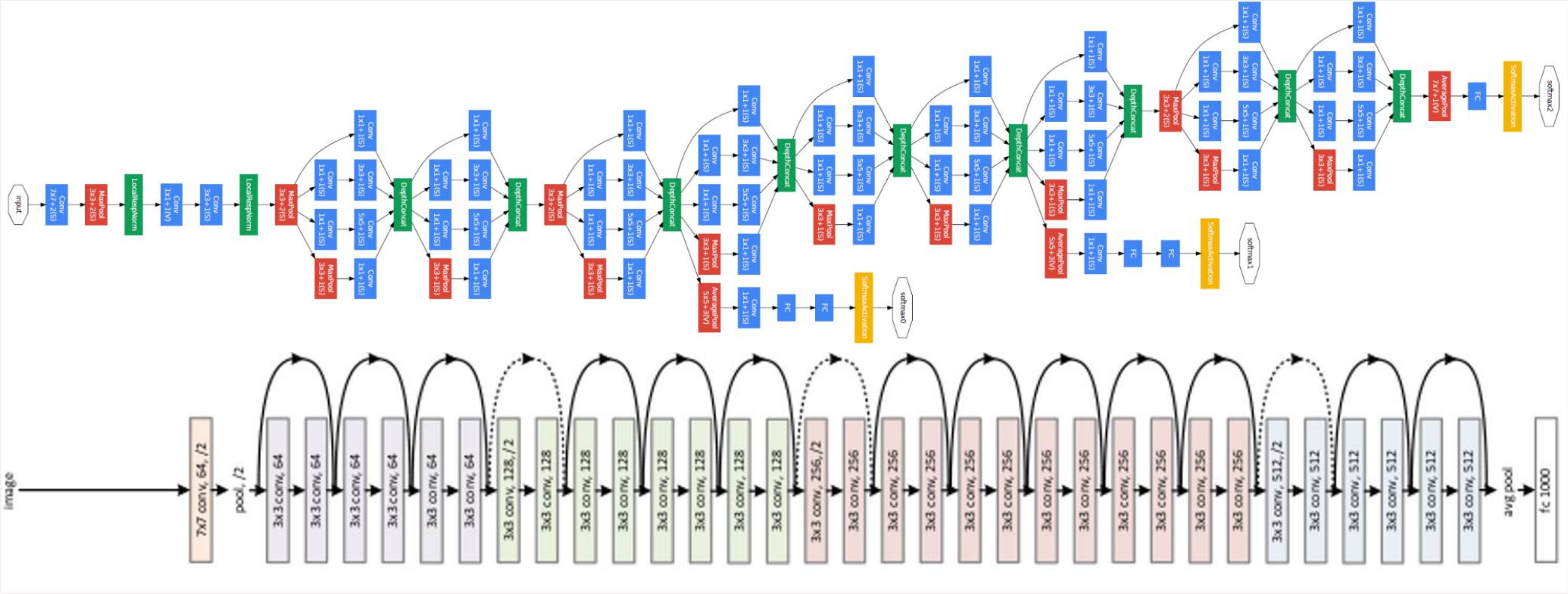


Unsupervised machine learning (8DC00)

Navchetan Awasthi

Previous lecture: deep learning models



Previous lecture: frameworks

Keras

PYTORCH

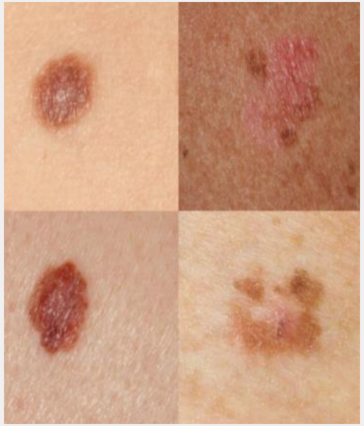
GitHub

TensorFlow

Sonnet

Previous lecture: applications

Classification

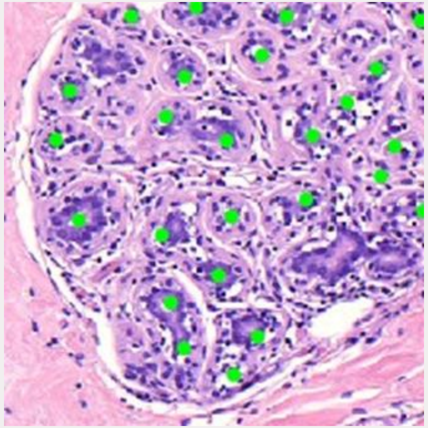


Esteva et al.,
Nature 2017

Regression

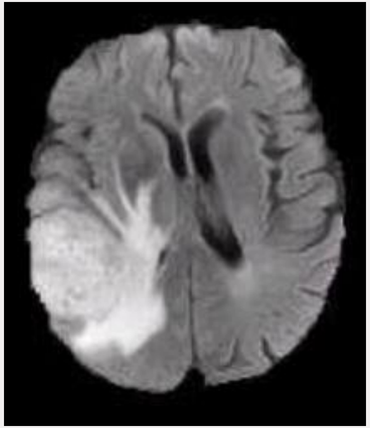
Heslinga et al.,
SPIE-MI 2019

Detection



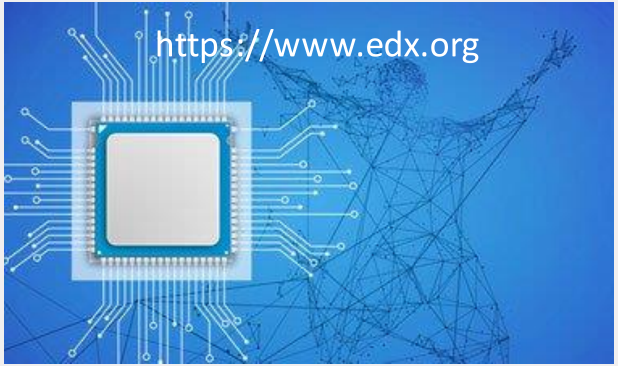
Wetstein et al.,
SPIE-MI 2019

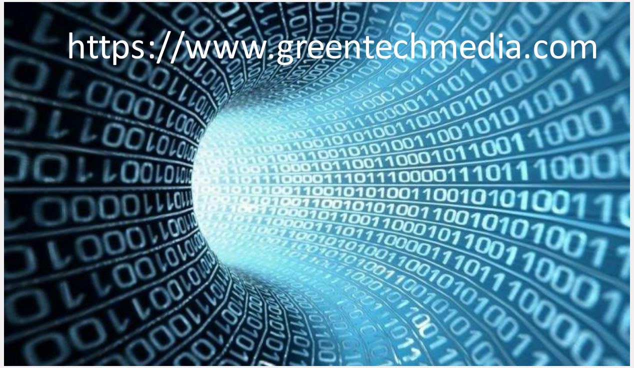
Segmentation



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

Reinforcement Learning

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.

Lecture outline

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

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

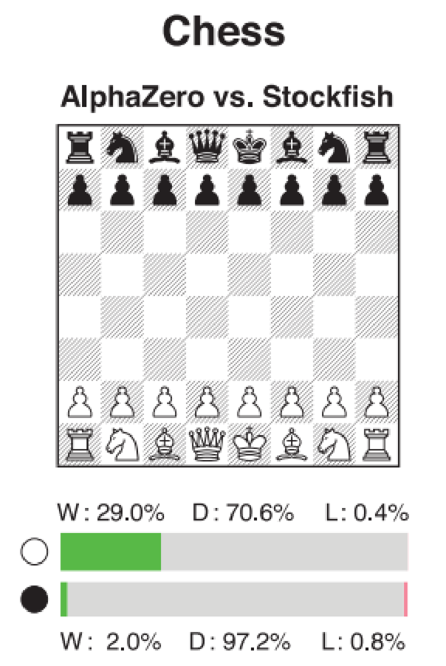
COMPUTER SCIENCE

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

David Silver[1,2]*†, Thomas Hubert[1]*, Julian Schrittwieser[1]*, Ioannis Antonoglou[1], Matthew Lai[1], Arthur Guez[1], Marc Lanctot[1], Laurent Sifre[1], Dharshan Kumaran[1], Thore Graepel[1], Timothy Lillicrap[1], Karen Simonyan[1], 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 wi
alpha-beta search that expa
by using a large number of
domain-specific adaptations
these augmentations, focu
Chess Engine Championsh
world champion Stockfish (
programs, including Deep
architectures (*1*, *12*).
In terms of game tree c
substantially harder game
is played on a larger board
pieces; any captured opp
sides and may subsequently
on the board. The strongest
as the 2017 Computer Sho
world champion Elmo, ha
feated human champions
use an algorithm similar t
puter chess programs, aga
optimized alpha-beta sear
domain-specific adaptatio
AlphaZero replaces the
edge and domain-specific



Silver, Hubert, Schrittweiser 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

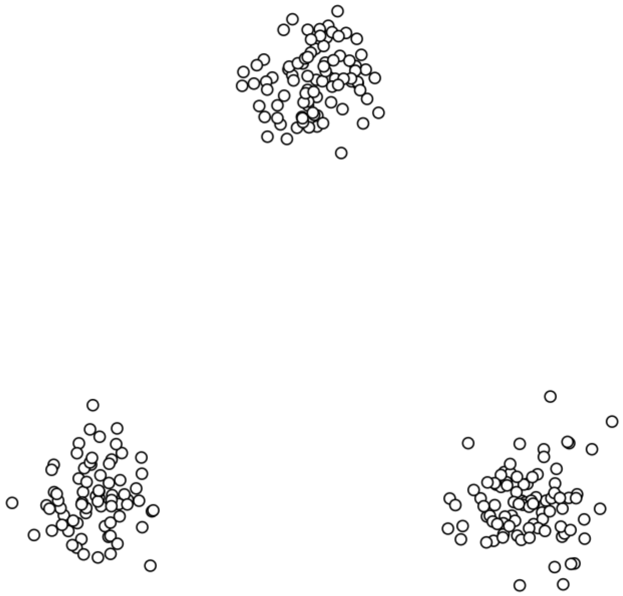
Imran Ahmed
Cambridge University
United Kingdom
ia311@cam.ac.uk

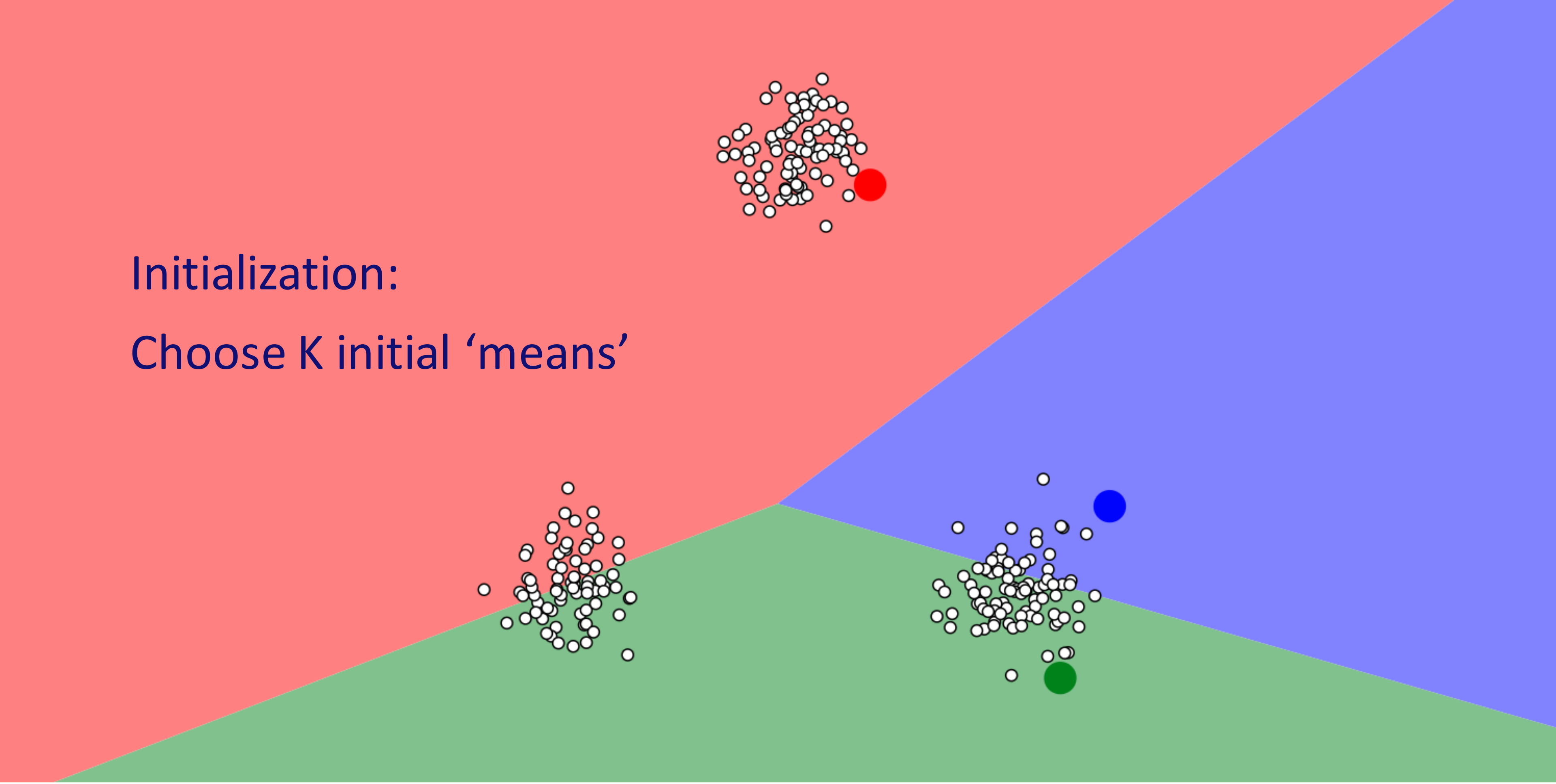
However, not often used in healthcare. Why?

How could we employ reinforcement learning for a surgery robot?

Finding clusters using K-means

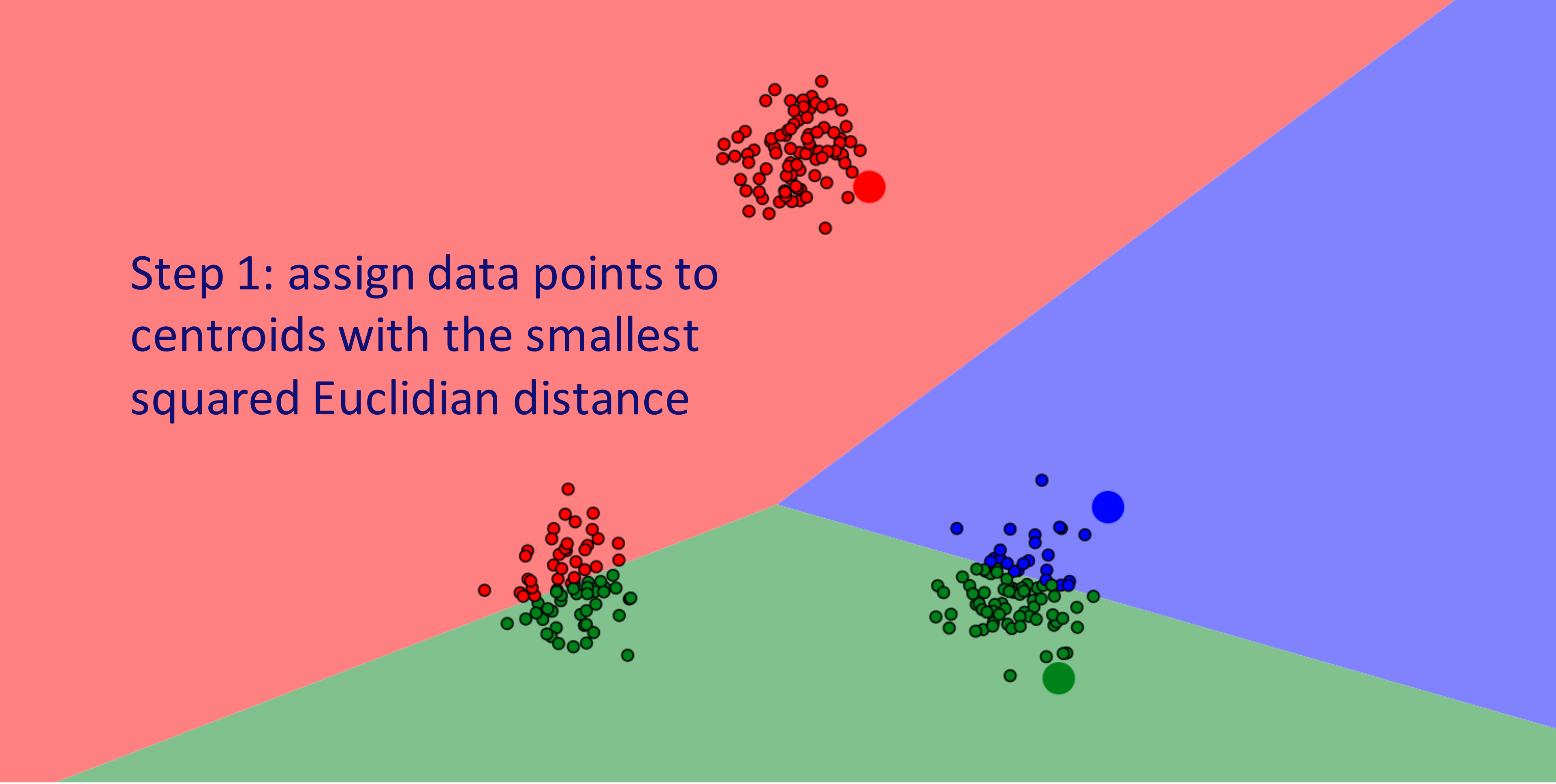
https://www.naftaliharris.com/blog/visualizing-k-means-clustering/





Initialization:

Choose K initial ‘means’



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

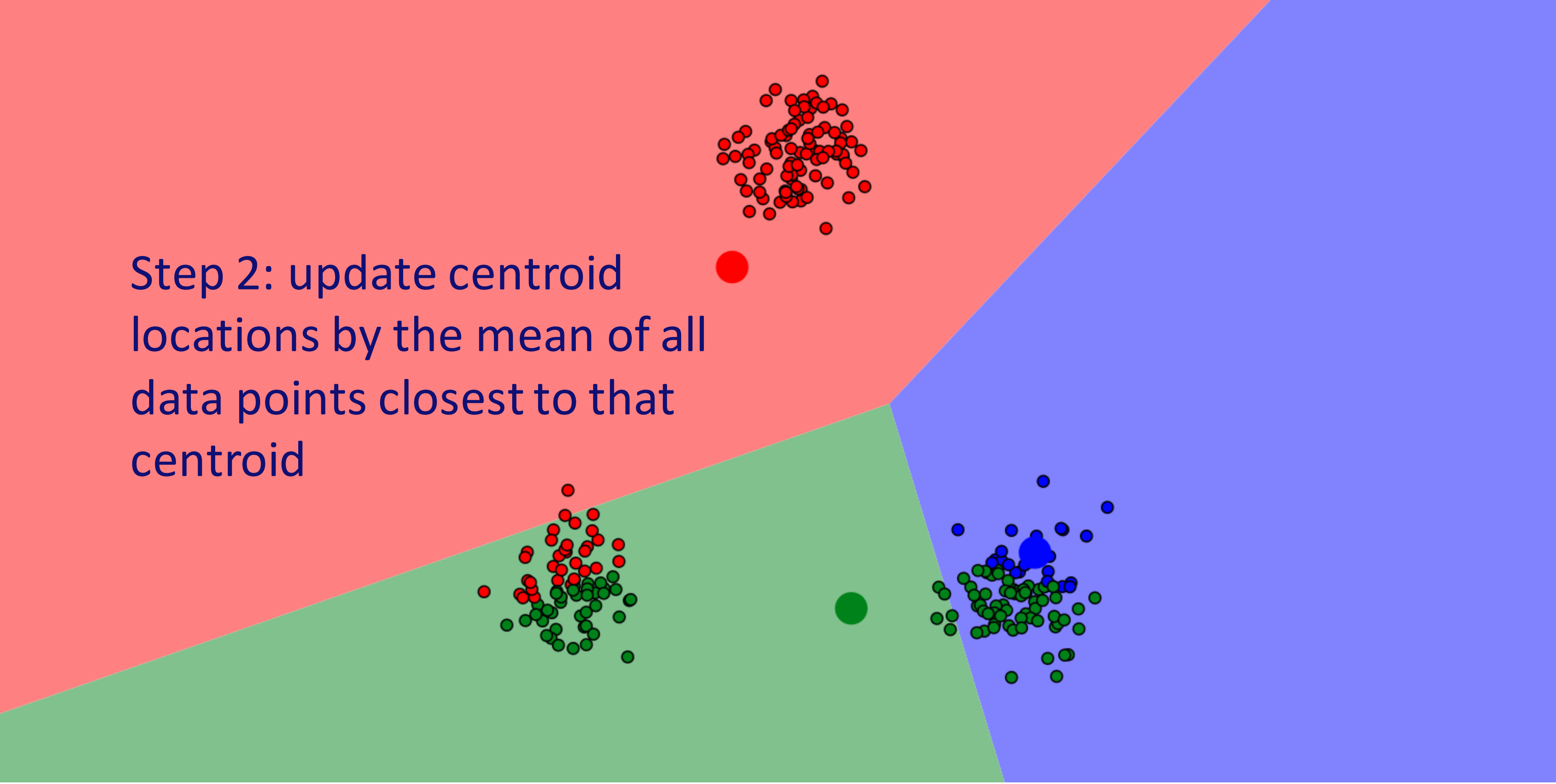
K-means – evaluate clustering performance

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

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

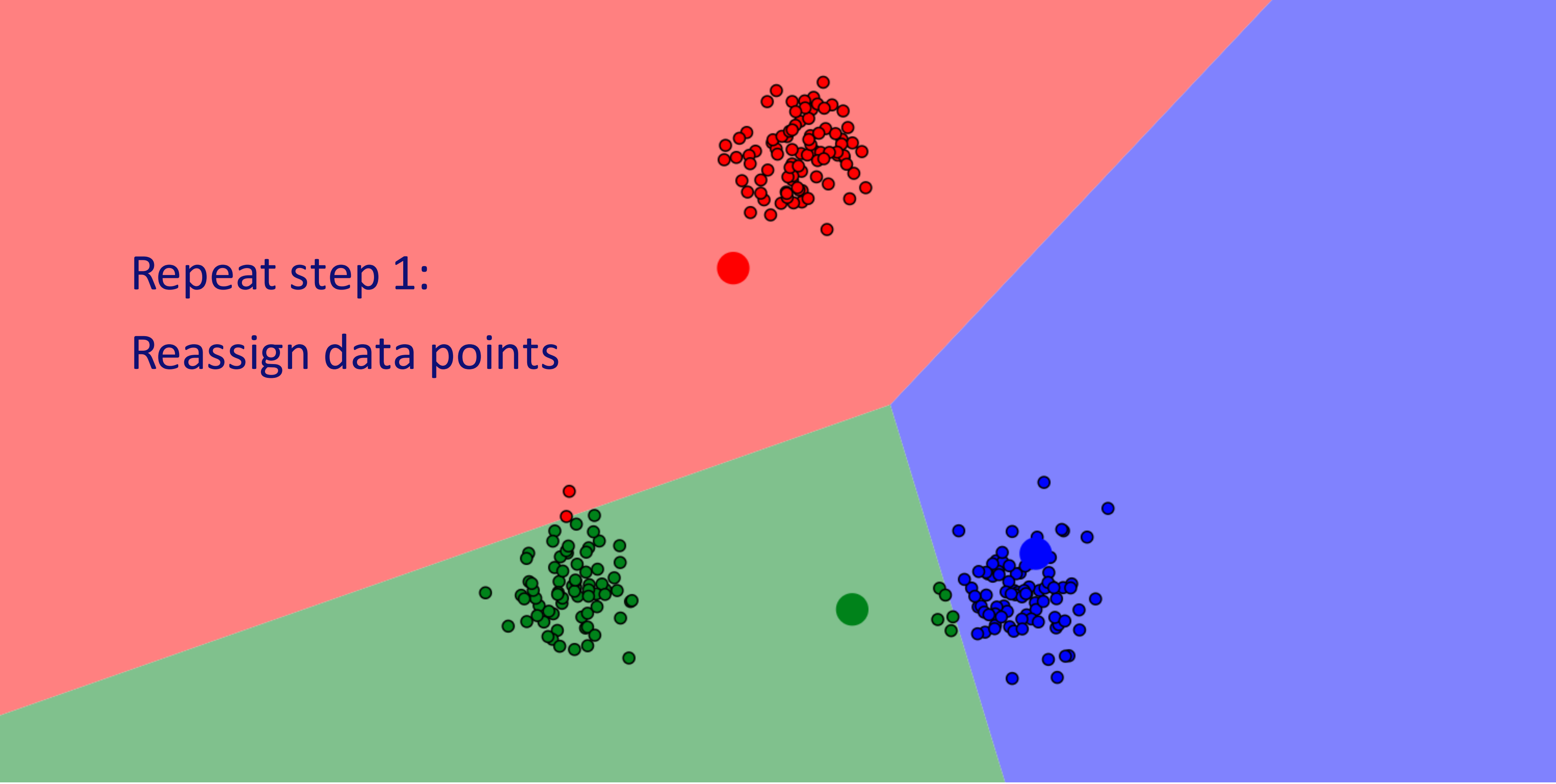
x_i are the points, W are the cluster centroids

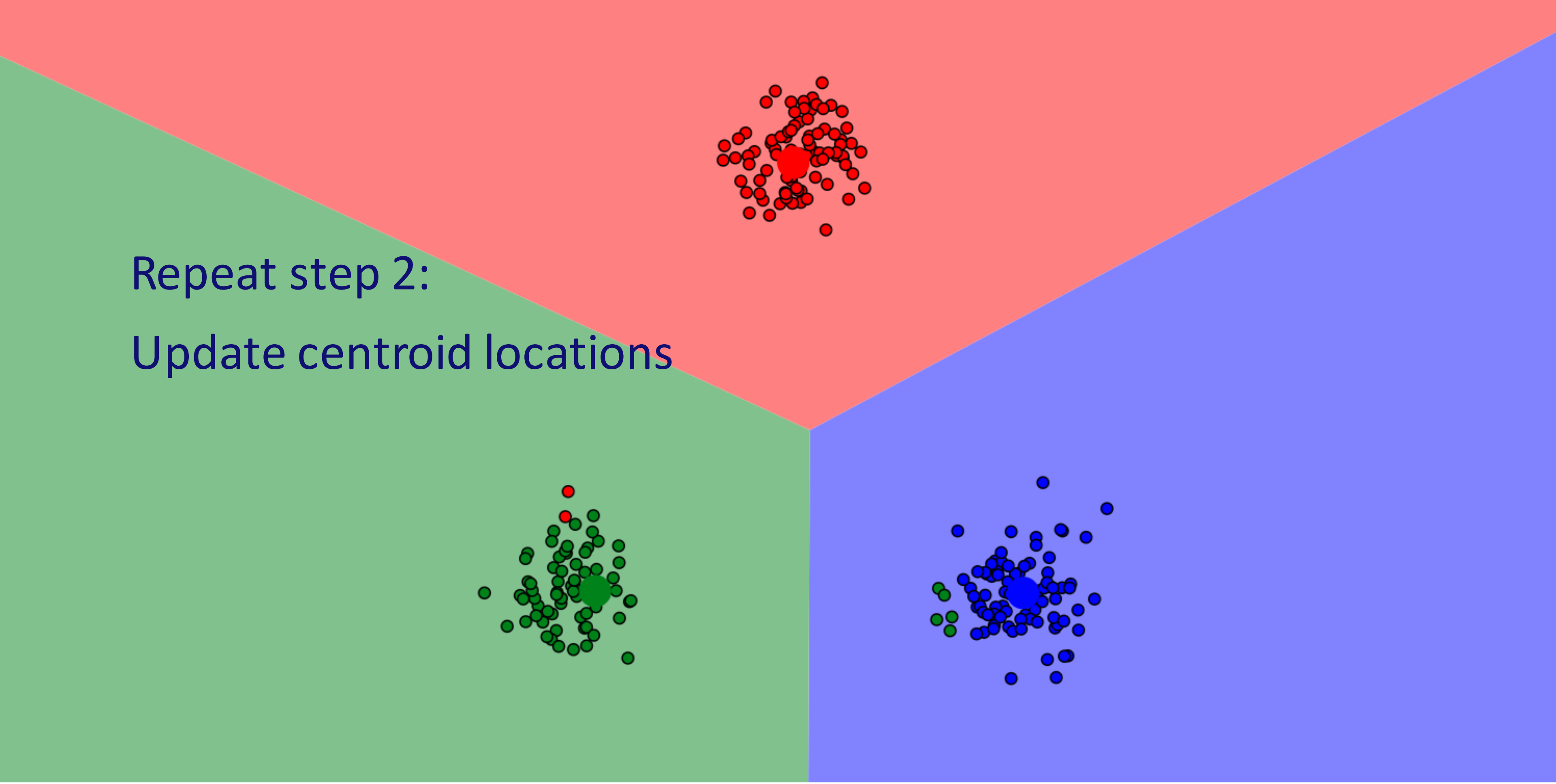
Step 2: update centroid locations by the mean of all data points closest to that centroid



Repeat step 1:

Reassign data points



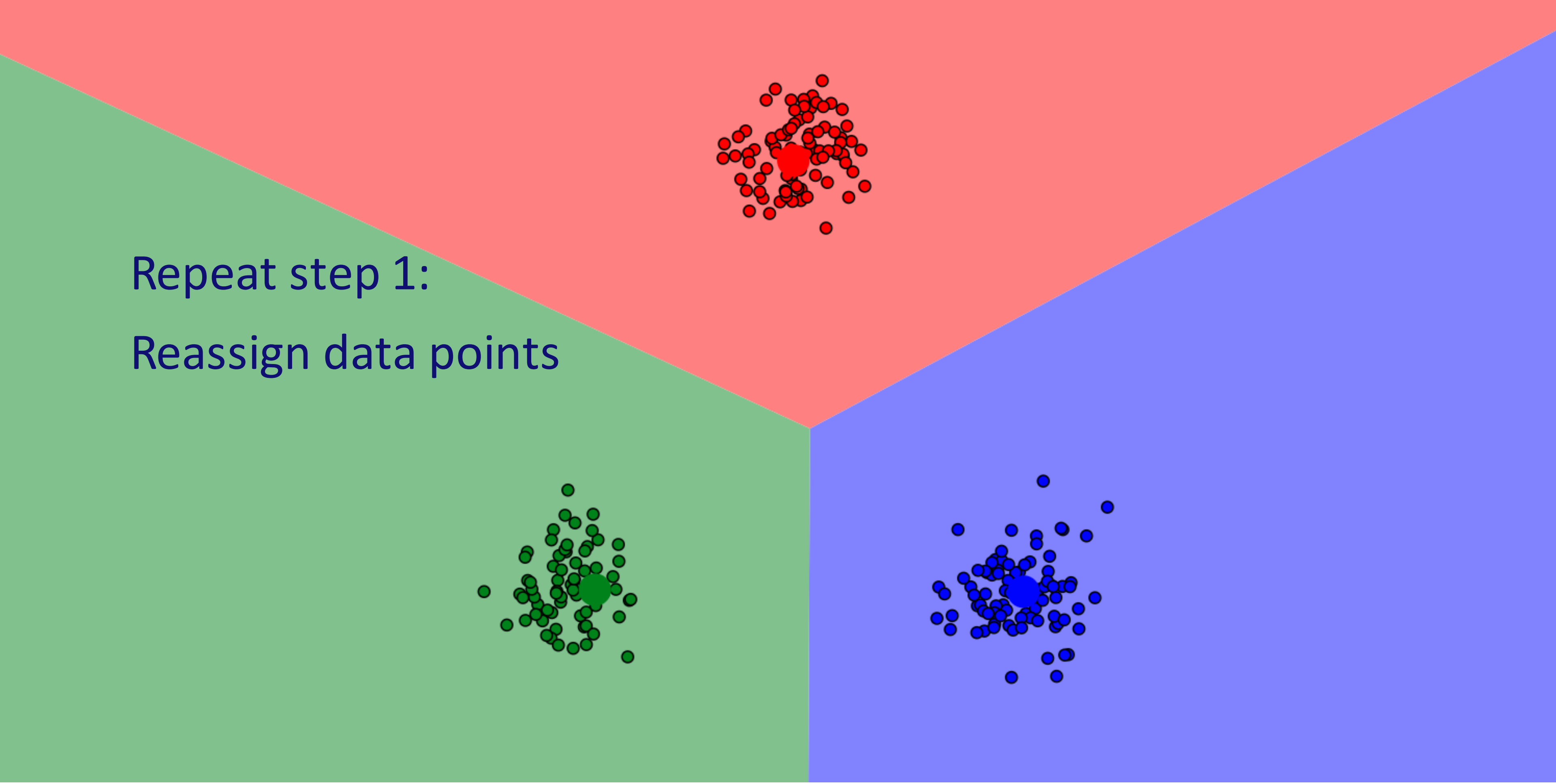


Repeat step 2:

Update centroid locations

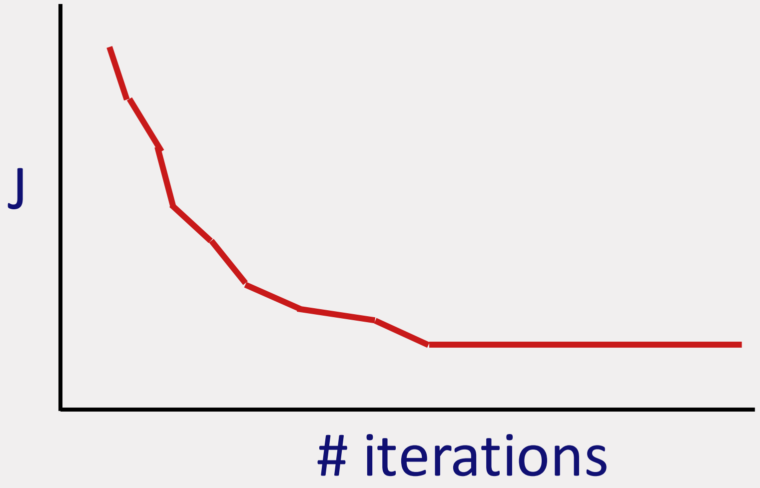
Repeat step 1:

Reassign data points



When do we stop?

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

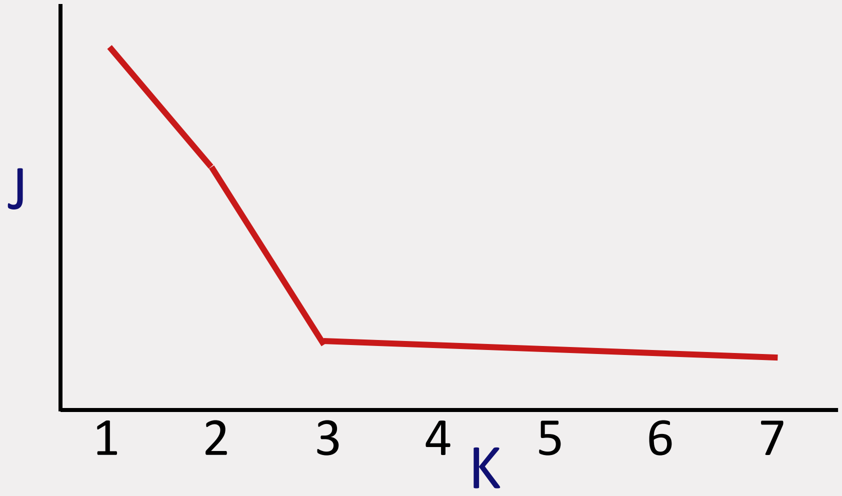


How do we choose initial centroid locations?

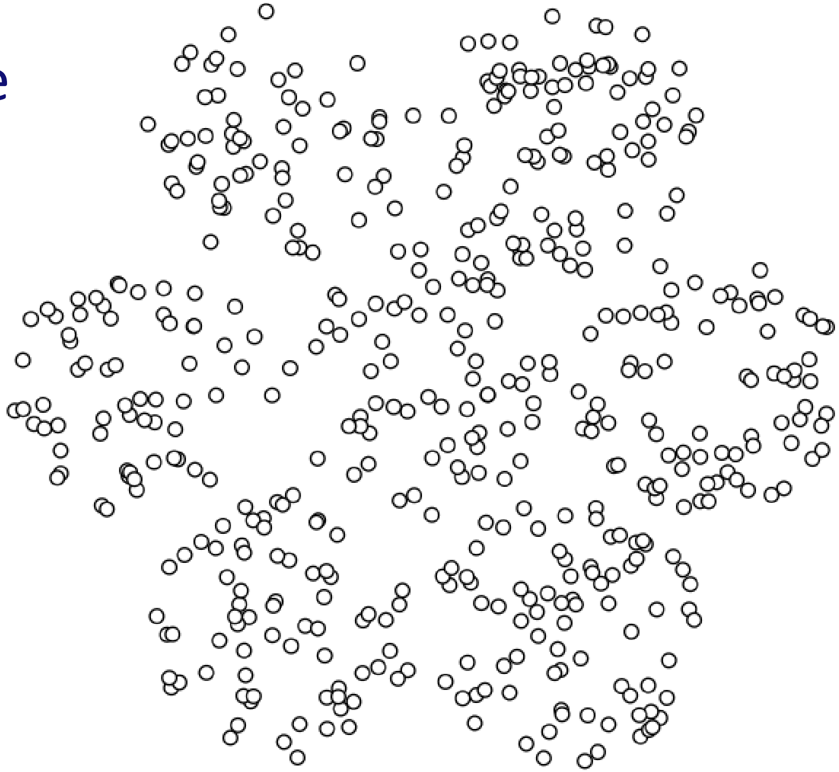
- Random
- Farthest points
- Manual?
  - Supervised
  - Difficult for high-dimensional data

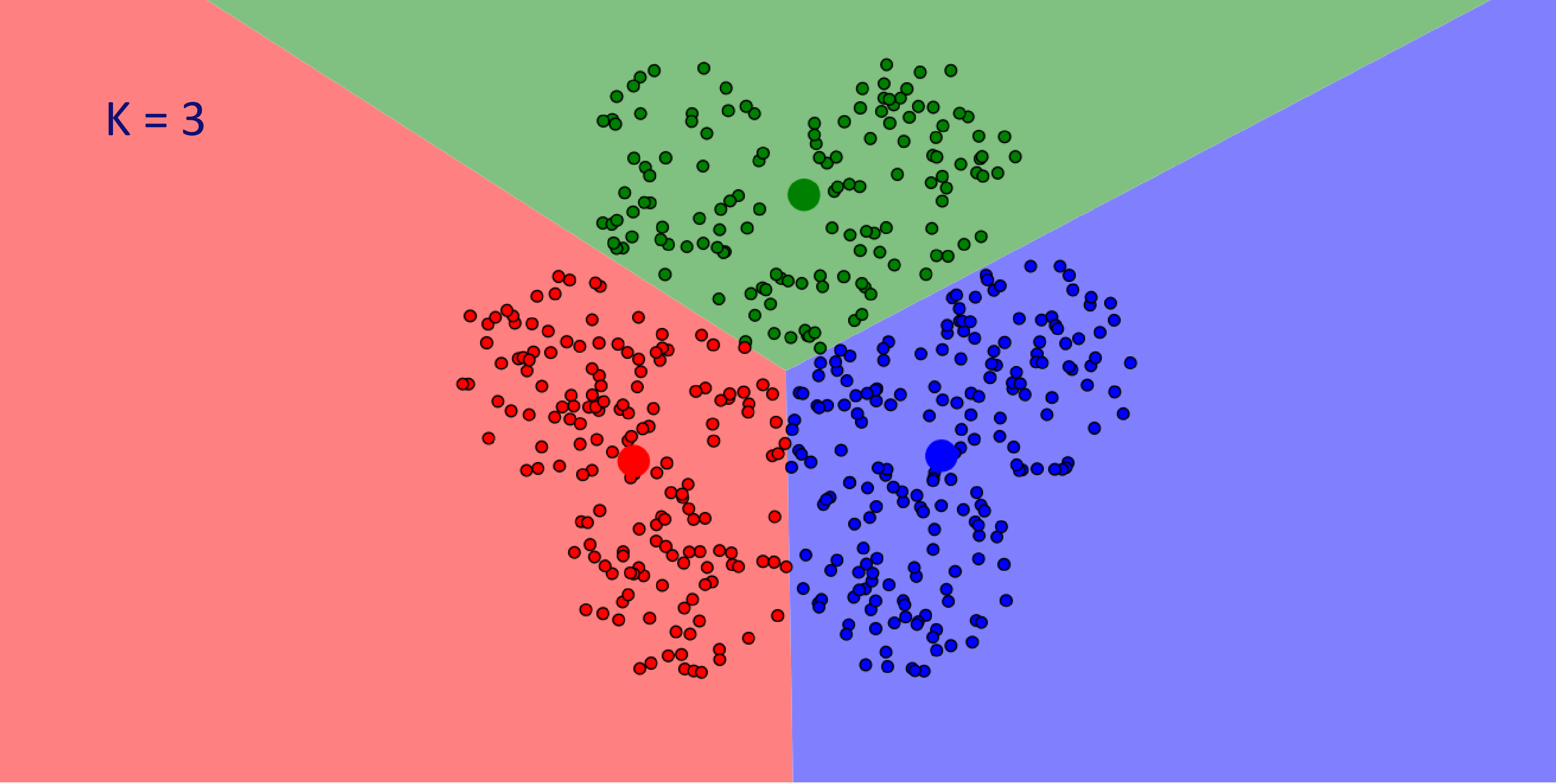
How do we choose K?

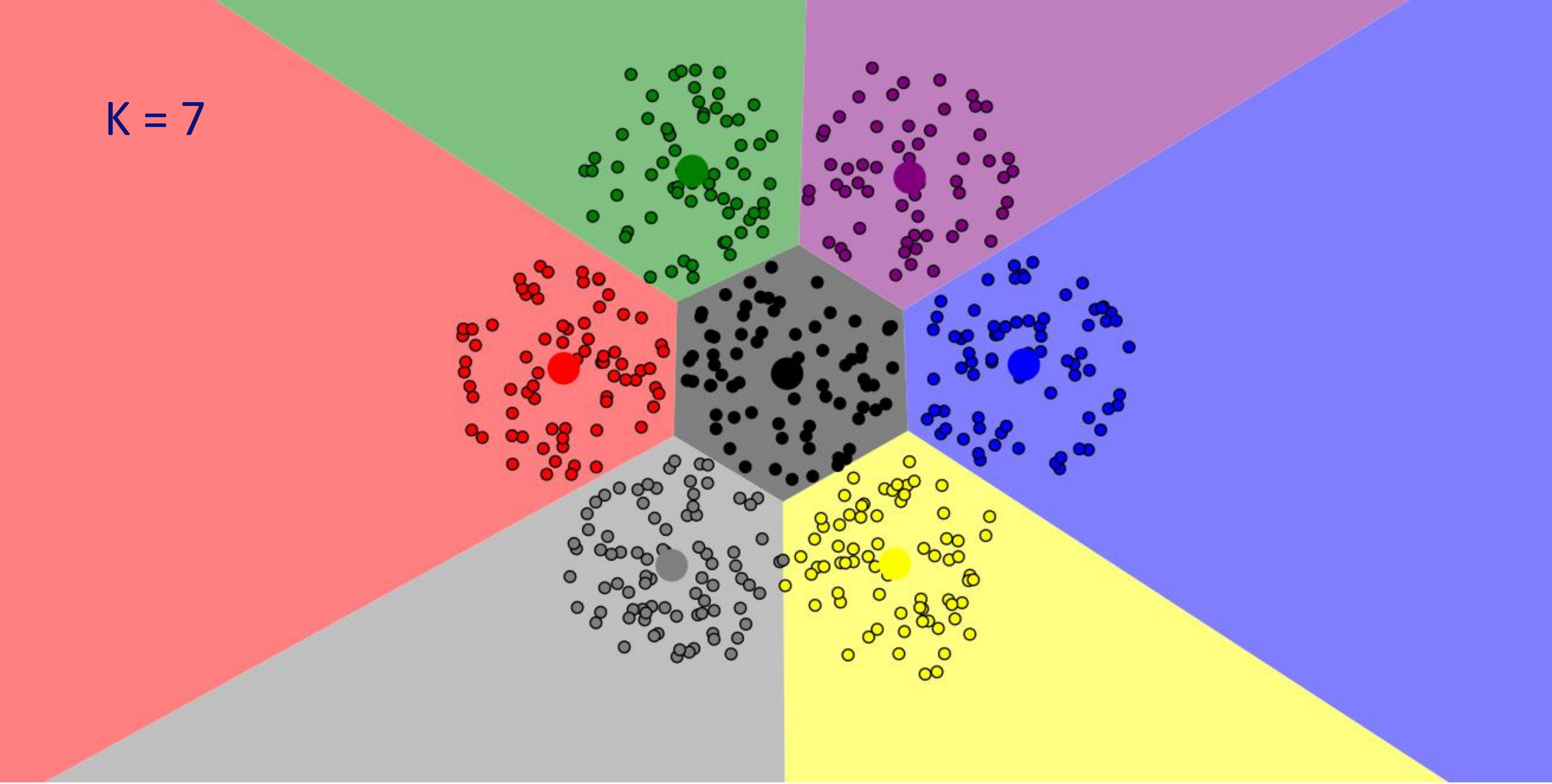
- K (=number of means) is a hyperparameter

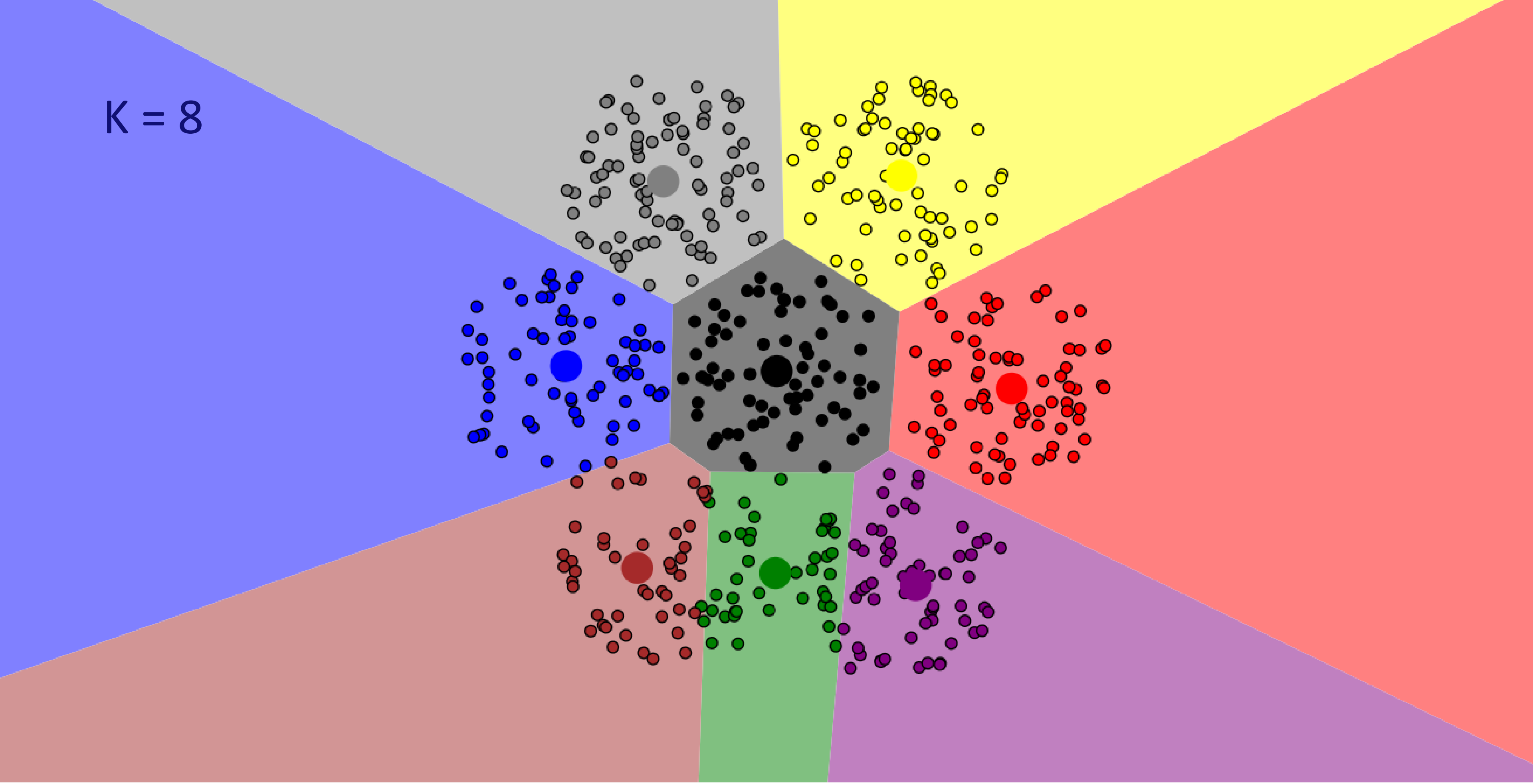


Another example




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!

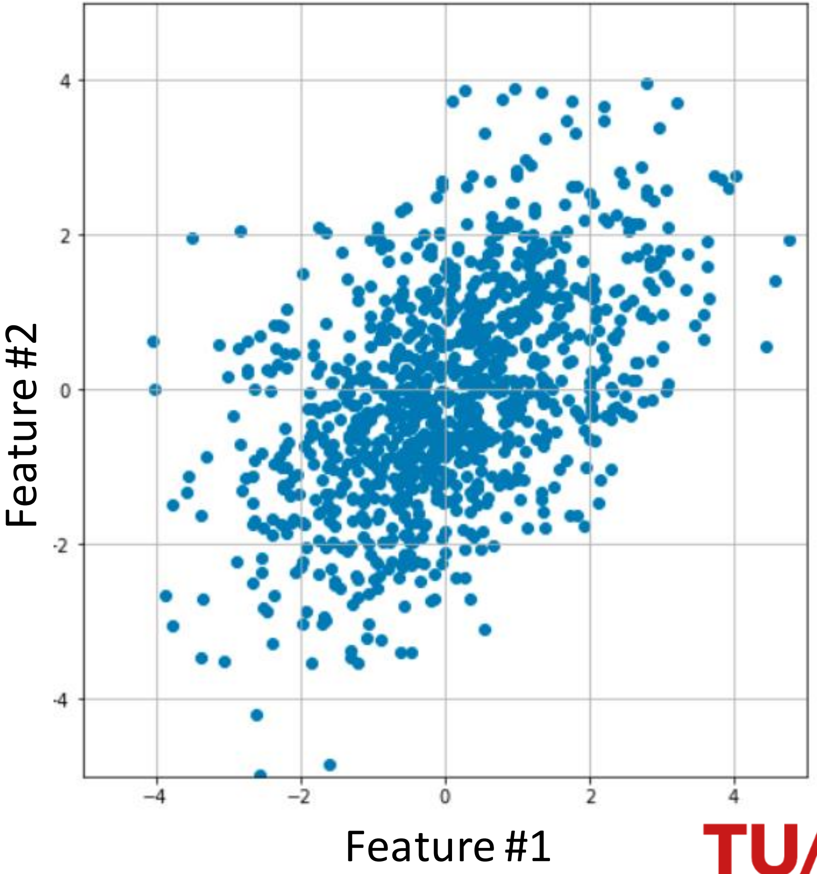
Principal Component Analysis

Example data set

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

Sampled from 2D Gaussian distribution

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



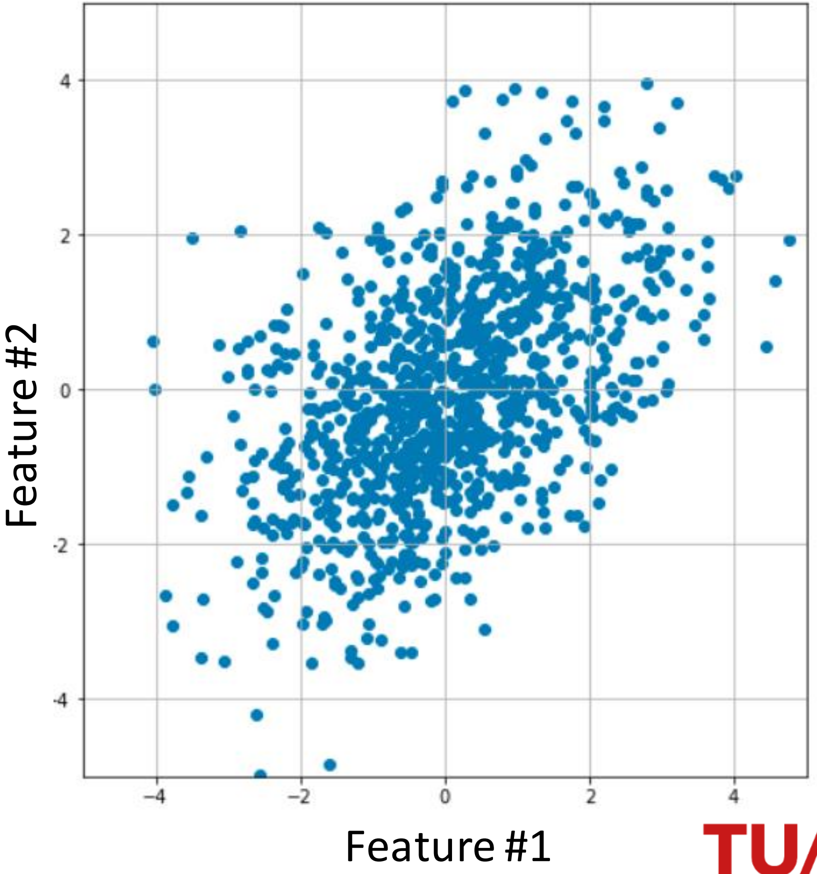
Principal Component Analysis

Example data set

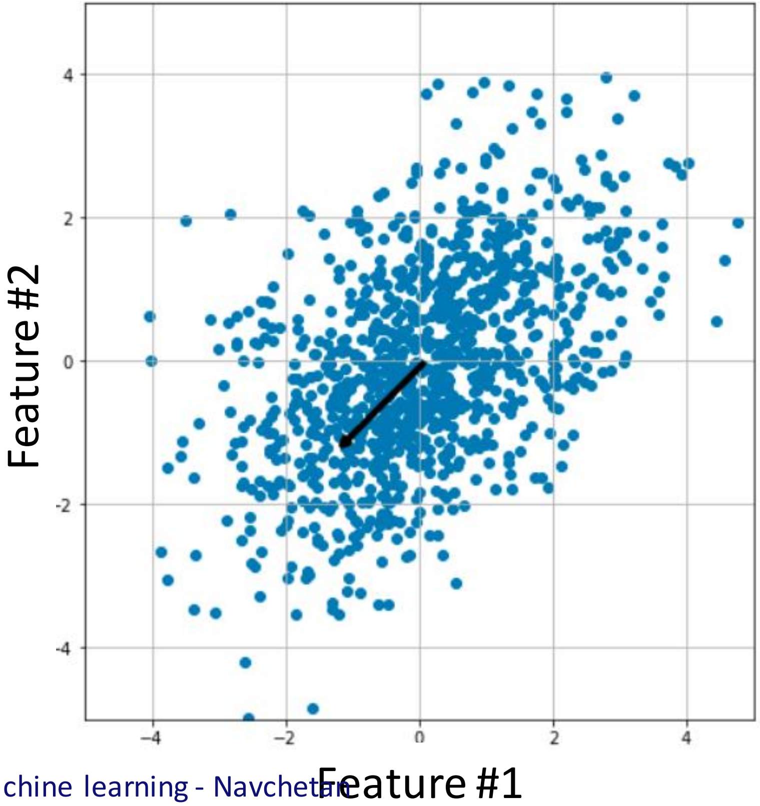
M-by-2 matrix $\boldsymbol{X}$ containing M points

Sampled from 2D Gaussian distribution

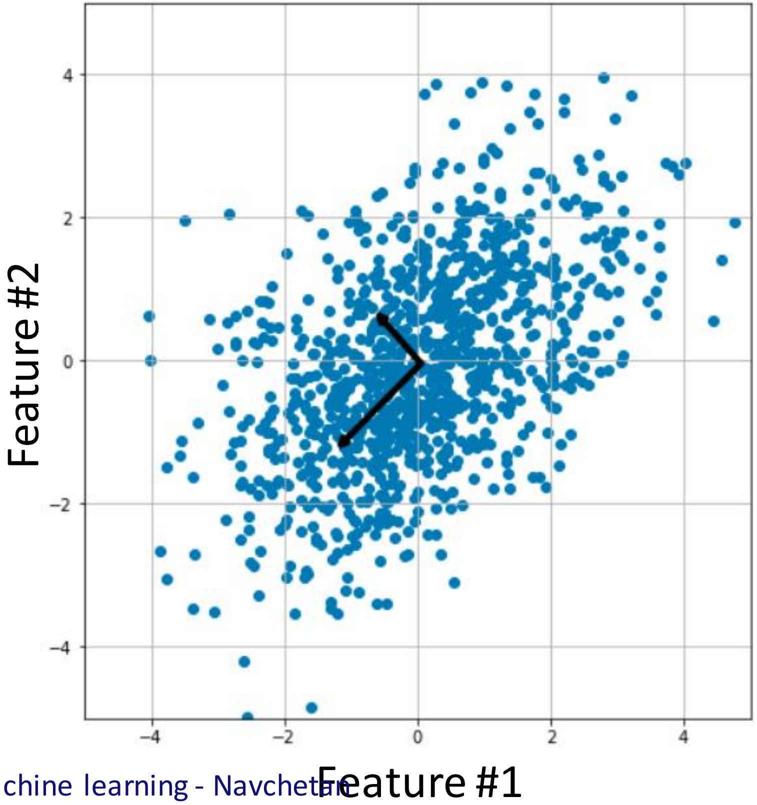
$$\mu_1 = 0$$
$$\mu_2 = 0$$
$$\sum = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$$



Principal Component 1



Principal component are orthogonal to one another



Finding the principal components

- Center data by subtracting mean of each variable

$$\widehat{\boldsymbol{X}} = \boldsymbol{X} - \overline{\boldsymbol{X}}$$

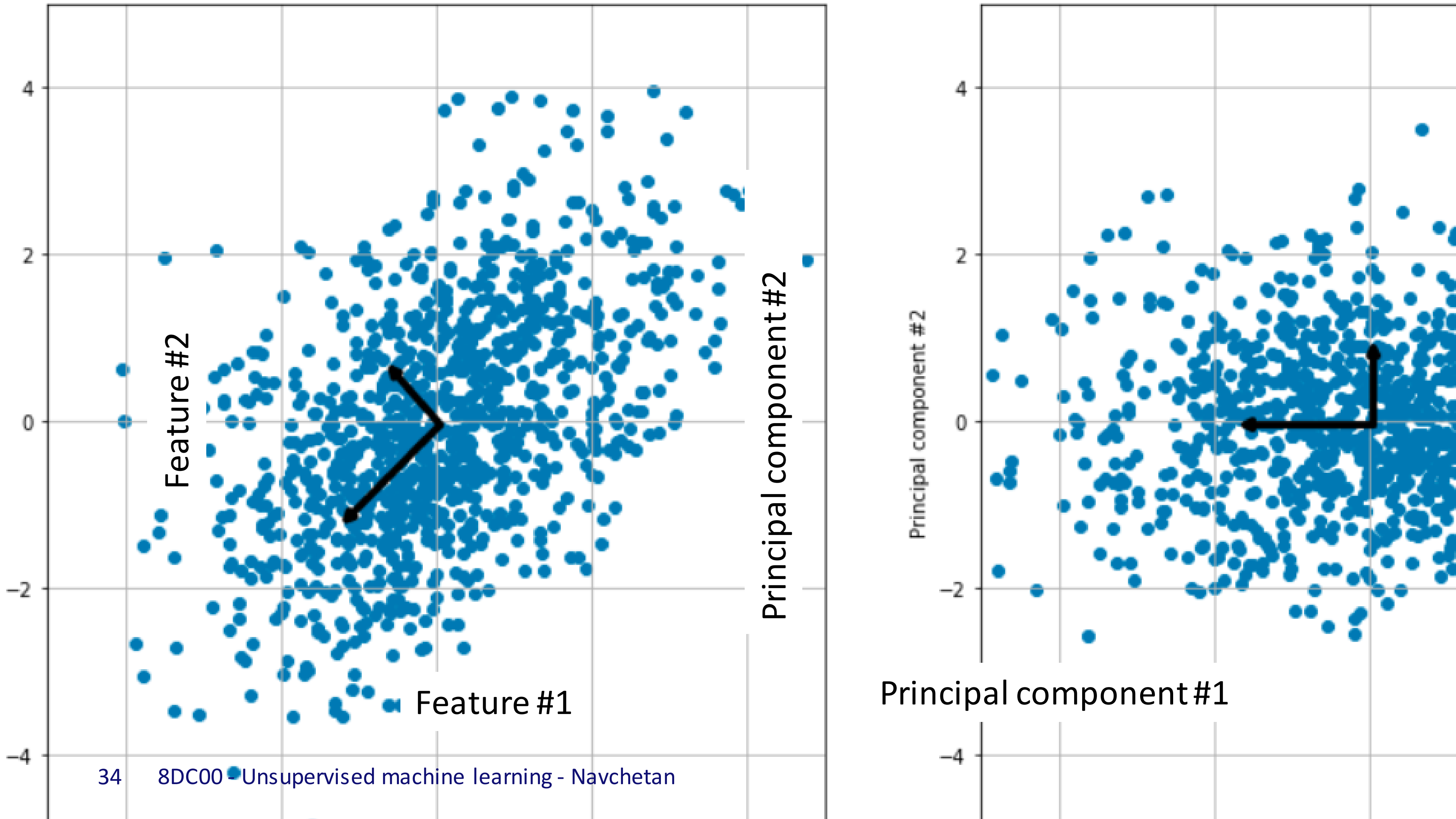
- Calculate covariance matrix

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

- 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 $\boldsymbol{X_{pca}}$

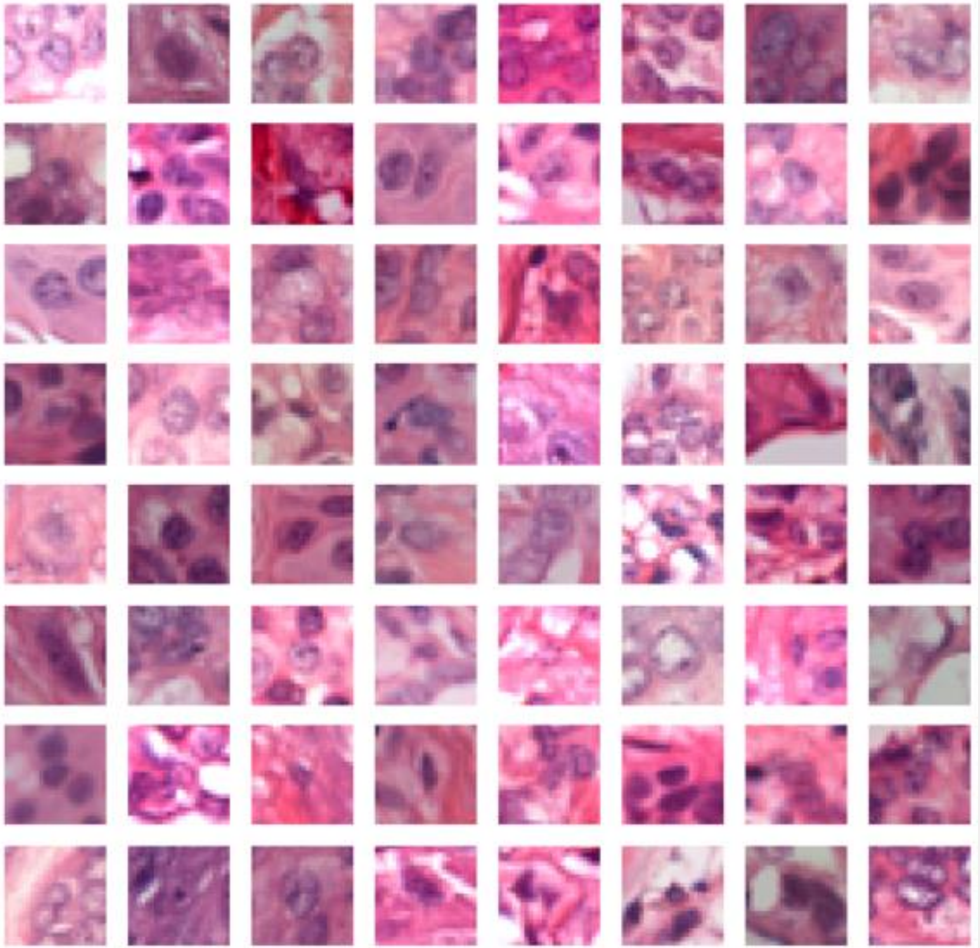

4
2
0
−2
−4
Feature #2
Feature #1
Principal component #2
Principal component #1
Principal component #2

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!



Questions so far?

PCA Example:

https://setosa.io/ev/principal-component-analysis/

Break!

Emperor Penguin (Antarctica)

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

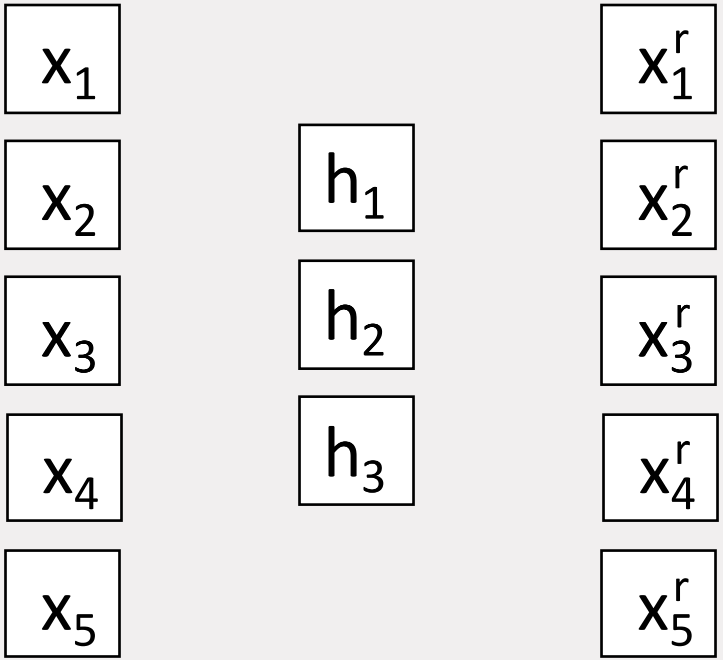
Lecture outline

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

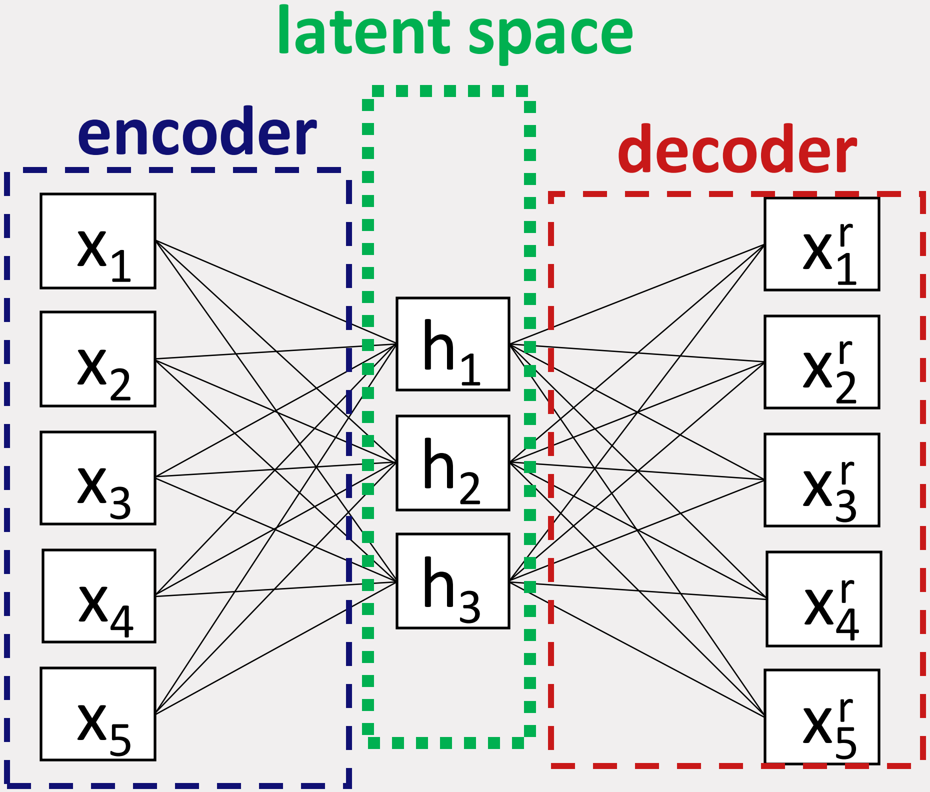
Awesome explanation

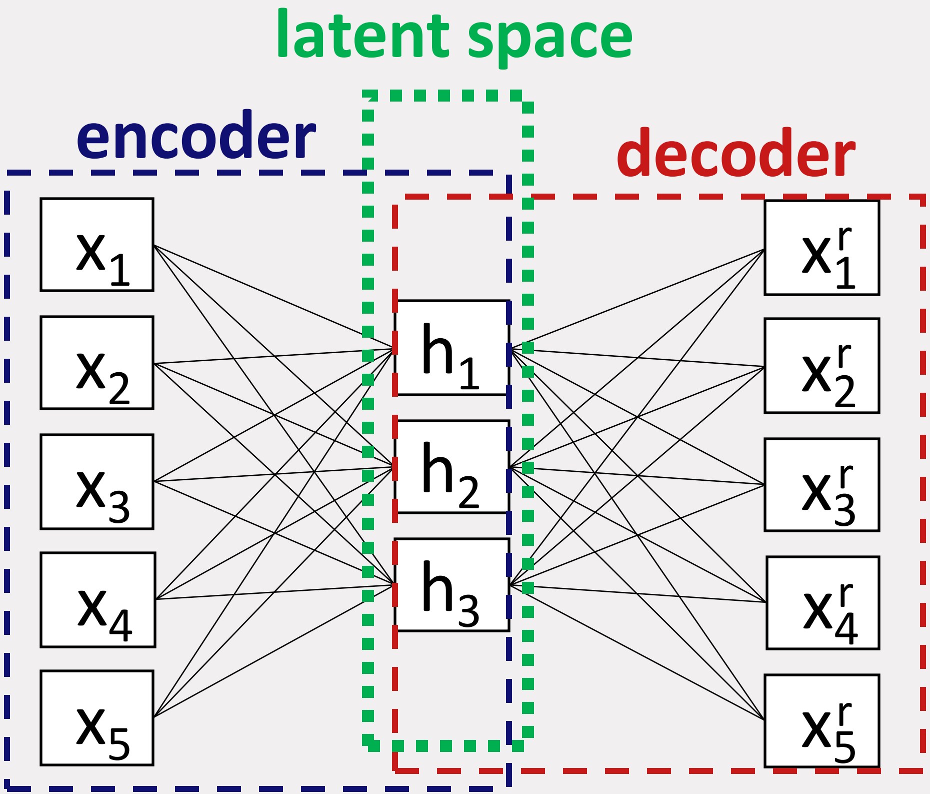
https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

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:

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

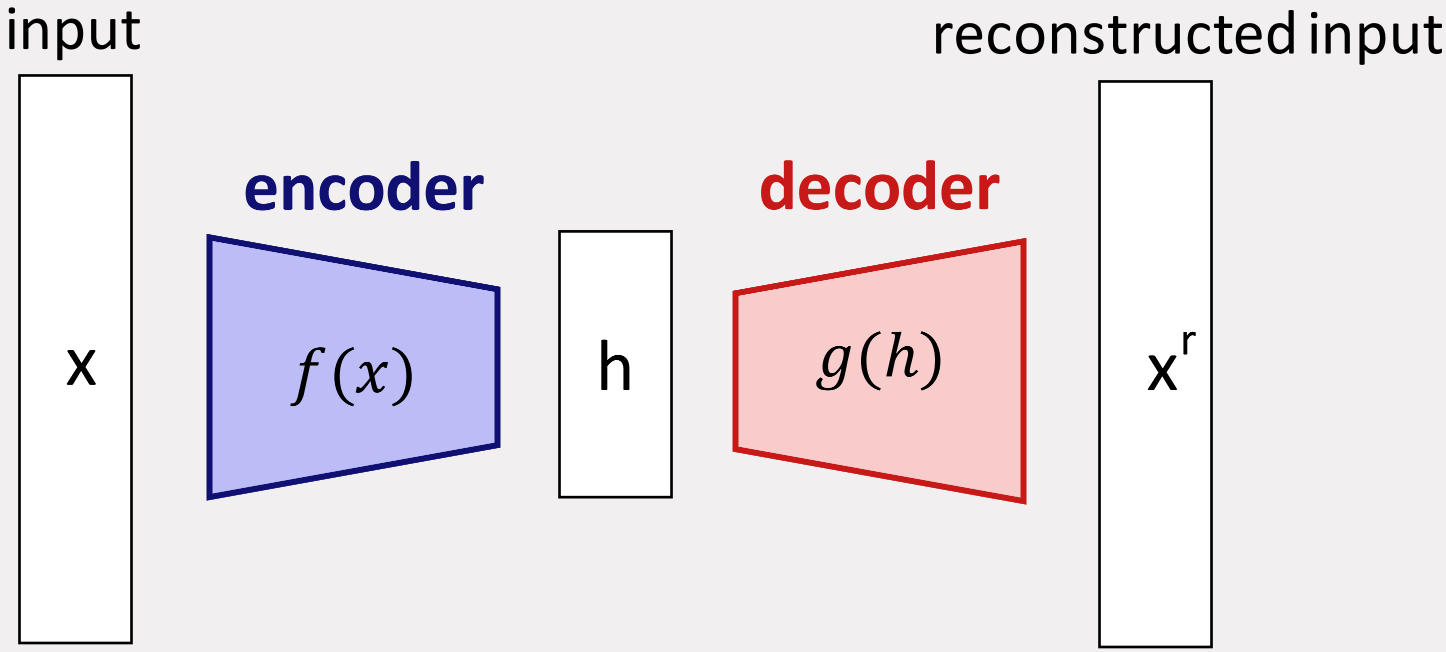
Decoder:

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

Penalize dissimmilarity

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

Autoencoder – a more general representation



Autoencoder

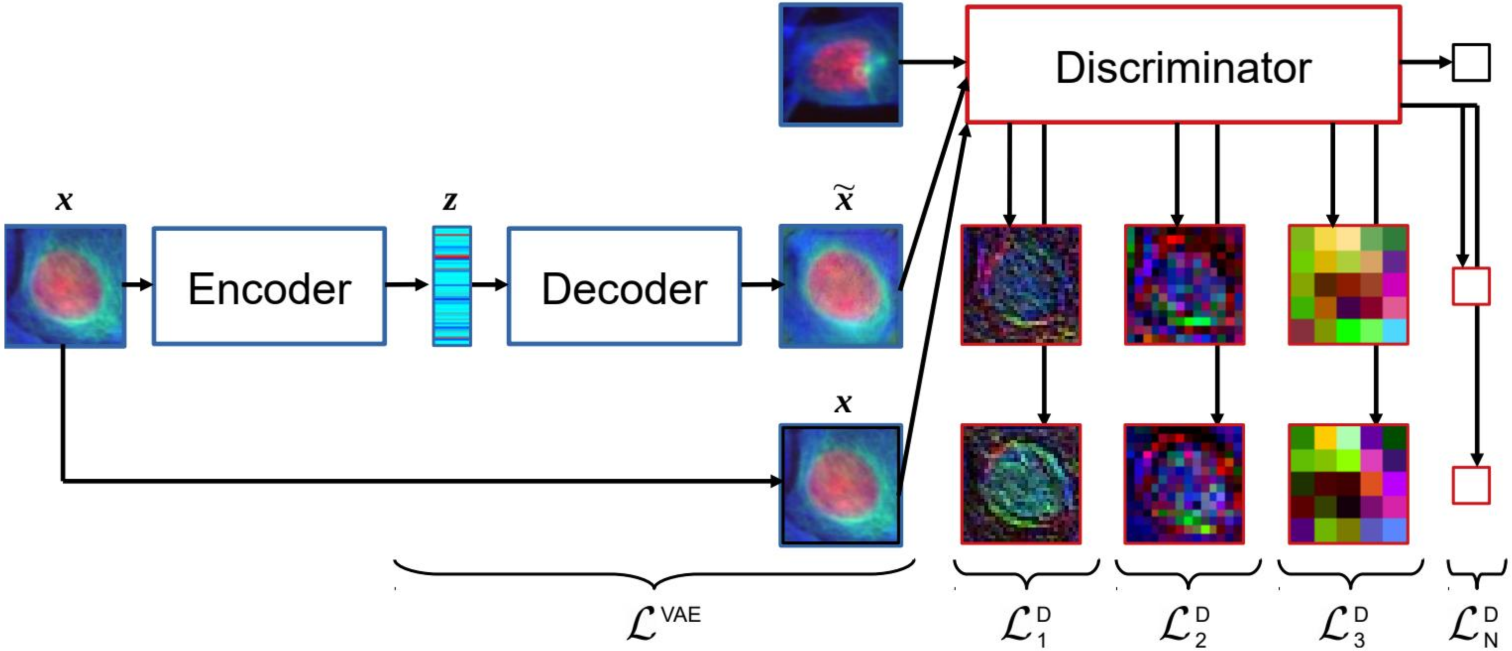
- Encoder/decoder can me simple or complex
- For example: a deep convolutional neural network

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 $\boldsymbol{z}$) used to predict compounds with 1-nearest-neighbors

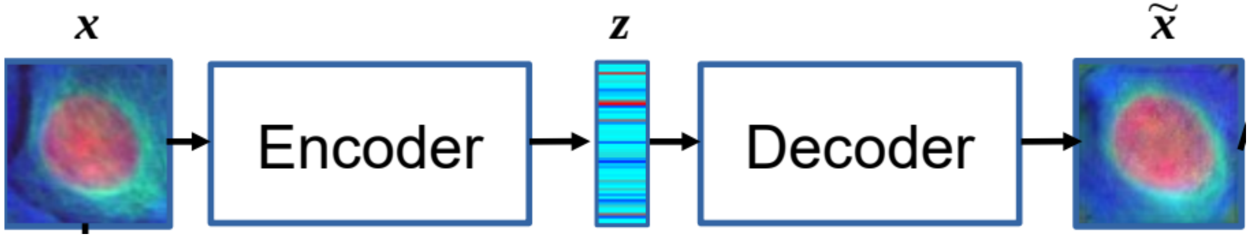


Image Denoising Example

we will train the autoencoder to map noisy digits images to clean digits images.

https://blog.keras.io/building-autoencoders-in-keras.html

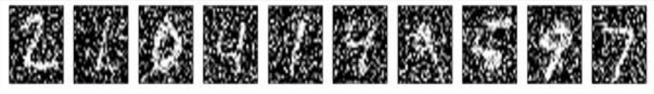
Image Denoising Example

```python
from keras.datasets import mnist
import numpy as np

(x_train, _), (x_test, _) = mnist.load_data()

x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
x_train = np.reshape(x_train, (len(x_train), 28, 28, 1))
x_test = np.reshape(x_test, (len(x_test), 28, 28, 1))

noise_factor = 0.5
x_train_noisy = x_train + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_train.shape)
x_test_noisy = x_test + noise_factor * np.random.normal(loc=0.0, scale=1.0, size=x_test.shape)

x_train_noisy = np.clip(x_train_noisy, 0., 1.)
x_test_noisy = np.clip(x_test_noisy, 0., 1.)
```

https://blog.keras.io/building-autoencoders-in-keras.html

Image Denoising Example

```
n = 10
plt.figure(figsize=(20, 2))
for i in range(1, n + 1):
    ax = plt.subplot(1, n, i)
    plt.imshow(x_test_noisy[i].reshape(28, 28))
    plt.gray()
    ax.get_xaxis().set_visible(False)
    ax.get_yaxis().set_visible(False)
plt.show()
```



Can autoencoder learn to recover the original digits?

https://blog.keras.io/building-autoencoders-in-keras.html

Image Denoising Example

```
input_img = keras.Input(shape=(28, 28, 1))

x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(input_img)
x = layers.MaxPooling2D((2, 2), padding='same')(x)
x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(x)
encoded = layers.MaxPooling2D((2, 2), padding='same')(x)

# At this point the representation is (7, 7, 32)

x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(encoded)
x = layers.UpSampling2D((2, 2))(x)
x = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(x)
x = layers.UpSampling2D((2, 2))(x)
decoded = layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

autoencoder = keras.Model(input_img, decoded)
autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
```

https://blog.keras.io/building-autoencoders-in-keras.html

Image Denoising Example

```
autoencoder.fit(x_train_noisy, x_train,
                epochs=100,
                batch_size=128,
                shuffle=True,
                validation_data=(x_test_noisy, x_test),
                callbacks=[TensorBoard(log_dir='/tmp/tb', histogram_freq=0, write_graph=False)])
```

https://blog.keras.io/building-autoencoders-in-keras.html

Image Denoising Example

https://blog.keras.io/building-autoencoders-in-keras.html

‘Supervised’ learning terminology

Supervised methods

Weakly Supervised

Semi-supervised

Unsupervised methods

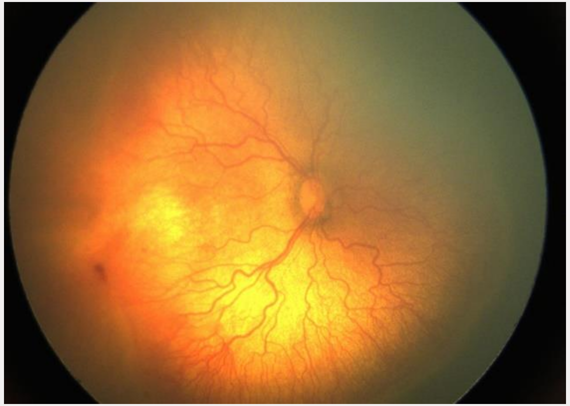
Self-supervised

Semi-supervised

Global labels

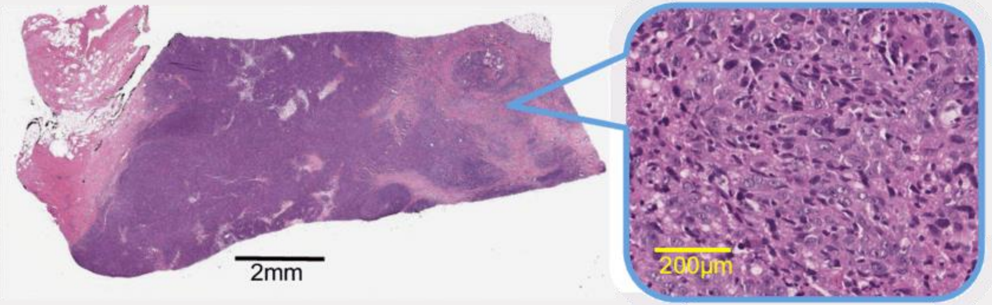
E.g. A single diagnosis is based on a series of retinal fundus images

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