Computer-aided diagnosis

Introduction

Cian Scannell

Eindhoven University of Technology
Department of Biomedical Engineering
(slides from Mitko Veta)

2024

Topic overview

- Introduction to CAD and machine learning
- Linear regression
- Logistic regression, neural networks
- Convolutional neural network
- Deep learning frameworks and applications
- Unsupervised machine learning

Learning goals for this lecture

- Define computer-aided diagnosis (CAD).
- Introduce (deep) machine learning, which at present time dominates CAD.
- Describe an intuitive method for *classification*.
- Introduce the concept of *generalisation* of machine learning models.

Connection with the Registration topic

Knowledge that will be reused in the CAD topic:

- Least-squares fitting
- Probability theory
- Gradient-descent

Definition

Computer-aided diagnosis: image analysis methods that help medical experts in the decision making process.

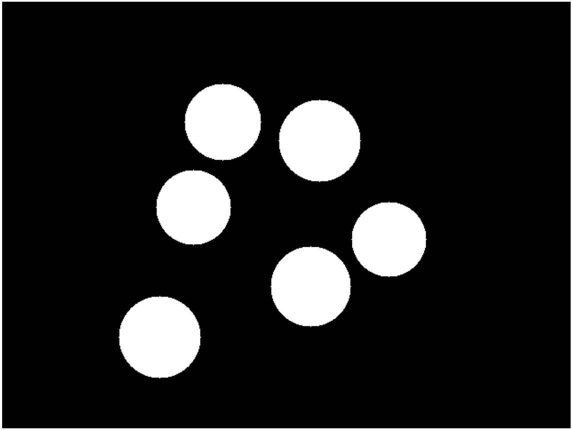
- Automatic evaluation of quantitative measurements.
- Detection and/or segmentation of abnormalities, pathology and objects/structures of interest.
- Characterisation of abnormalities, pathology and objects/structures of interest (e.g. cancer grading).
- Independent second opinion.
- Next level: Automated diagnosis – eliminate the need for medical experts.

An analogy with the automotive industry

- Cruise (speed) control
- Cruise control with lane assist
- Adaptive cruise control
- Tesla's auto-pilot

- Next level: Self-driving cars – eliminate the need for drivers

How would you approach this problem?

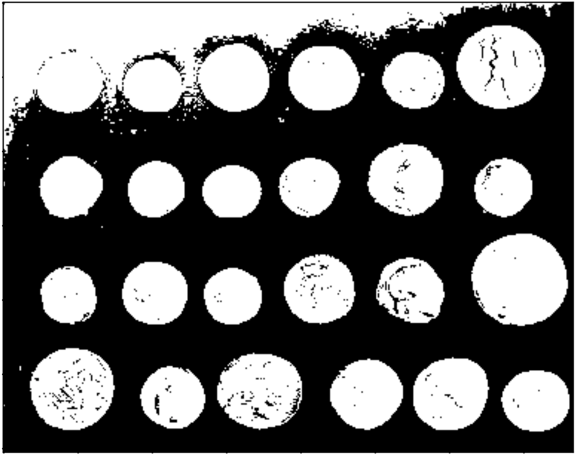


Count and measure all circles in the image.

How would you approach this problem?

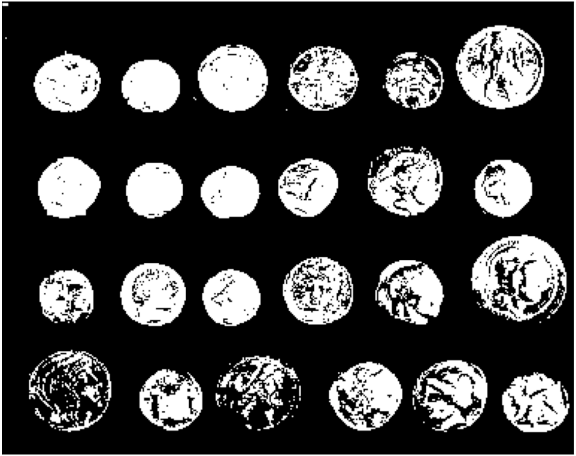
Count and measure all coins in the image.

How would you approach this problem?



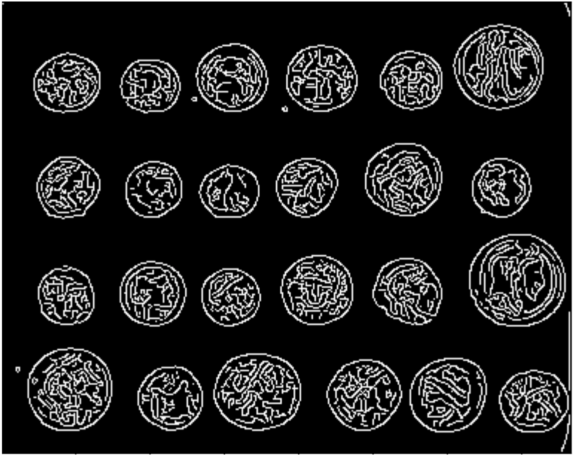
$threshold(image, 100)$

How would you approach this problem?



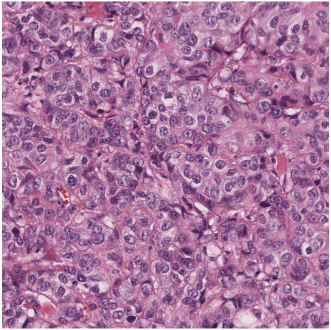
$threshold(image, 140)$

How would you approach this problem?



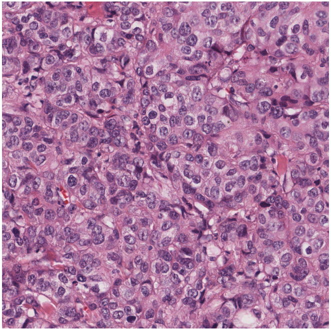
canny_edge_detection(image)

How would you approach this problem?



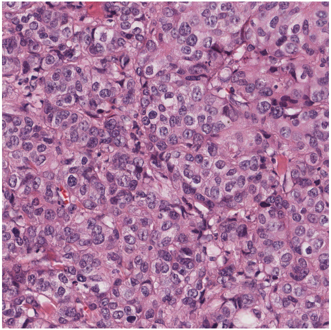
- Detect, segment and classify all cell nuclei.

How would you approach this problem?



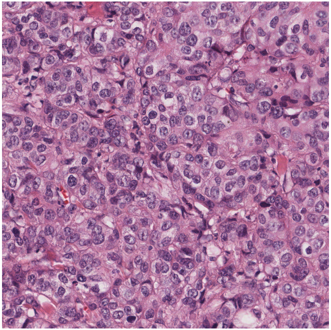
- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.

How would you approach this problem?



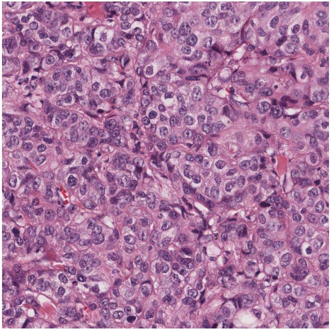
- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.
- Classify as low, intermediate or high grade cancer.

How would you approach this problem?



- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.
- Classify as low, intermediate or high grade cancer.
- Predict the 5-year disease free survival of this patient.

How would you approach this problem?



- Detect, segment and classify all cell nuclei.
- Classify as normal, benign or malignant.
- Classify as low, intermediate or high grade cancer.
- Predict the 5-year disease free survival of this patient.
- Predict if this patient will respond to a specific treatment.

CAD = (Deep) Machine learning?

At present, CAD is dominated by methods that are generally based on (deep) machine learning methodology.

CAD = (Deep) Machine learning?

At present, CAD is dominated by methods that are generally based on (deep) machine learning methodology.

A rough estimate: more than 90% of papers published at flagship medical image analysis conferences (MICCAI, IEEE ISBI, SPIE MI) employ some form of (deep) machine learning methodology.

Historical perspective of machine learning

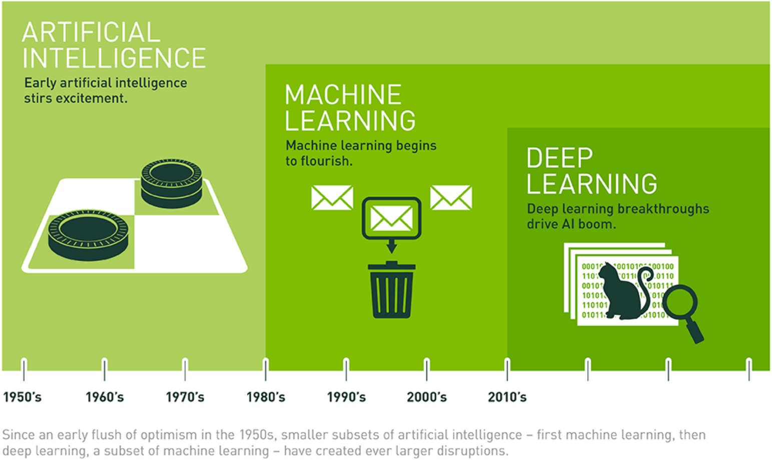
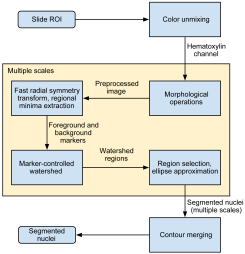
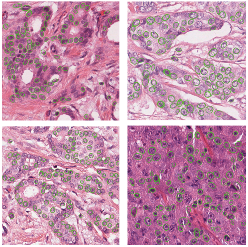


Figure source: nvidia.com

An example from my past work: nuclei area measurement

2010-2011: An image processing pipeline of (mainly) mathematical morphology operators (e.g. the watershed algorithm).

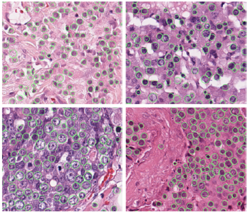


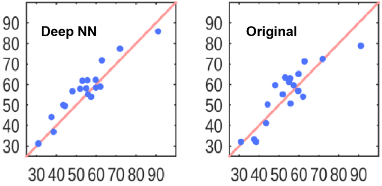


The design and validation of the processing pipeline took the better part of a year.

Figure source: Veta et al. PLOS ONE 2012

2015: A deep neural network for nuclei area measurement.





Manual vs. automatic nuclear area in μm^2

The the training and validation of the deep neural network model took less than a week.

The results were more accurate than the the original method.

Figure source: Veta et al. MICCAI 2016

The central premise of machine learning

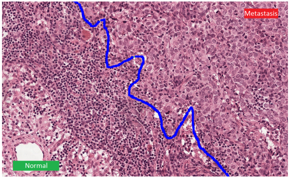
Learn "computer programs" from examples instead of manually writing rules.

The central premise of machine learning

Learn "computer programs" from examples instead of manually writing rules.

Advantage: the same method (e.g. a neural network) can be used to solve a variety of different problems.

Siberian hustky vs. eskimo dog



Normal vs. metastases

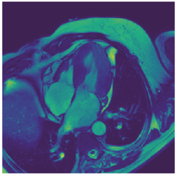
Figures source: (left) Szegedy et al. arXiv 2014, (right) camelyon16.grand-challenge.org

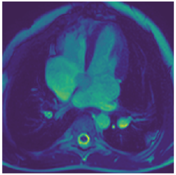
What are the "examples"?

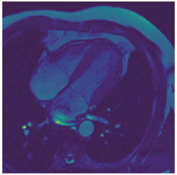
Depends on the particular problem and task.

Dataset: cardiac MRI images.
Task: detect if a specific pathology is present in each image.

In this case, every image is an example and is associated with a binary target: $0 =$ "healthy", $1 =$ "diseased" (i.e. we want to classify each image as "healthy" or "diseases").


diseased

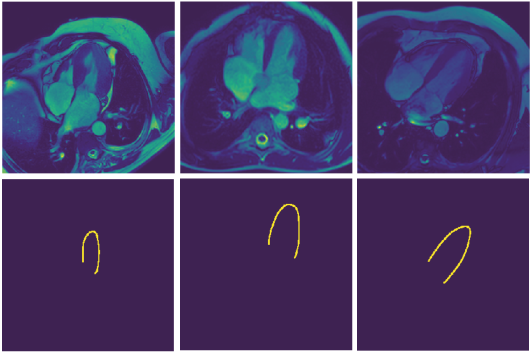

diseased


healthy

What are the "examples"?

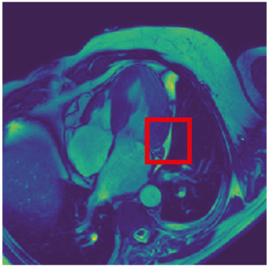
Dataset: cardiac MRI images.
Task: Segment the contours of the left ventricle

In this case, each pixel is an example and is associated with a binary target: 0 = "background", 1 = "contour".



How are the “examples” represented?

Traditionally with feature extraction:



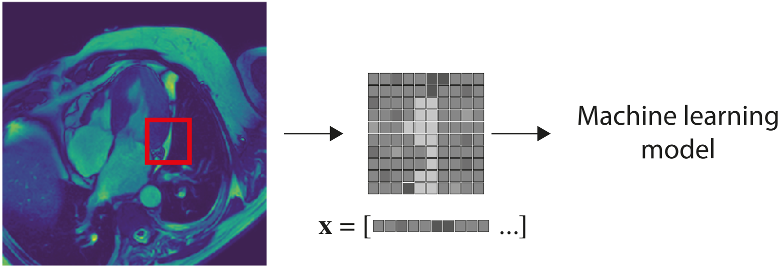
Intensity features
Texture features
Shape features
...

Machine learning model

$$\mathbf{x} = [\mathbf{x}_{\text{intensity}}\ \mathbf{x}_{\text{texture}}\ \mathbf{x}_{\text{shape}} \ldots]$$

How are the “examples” represented?

With raw pixel values (the *de facto* standard for deep learning):



In summary...

In order to design a machine learning algorithm for a specific task we are given a dataset of examples represented by $\mathbf{x}_i$.

Each example is (optionally) associated with a target y_i.

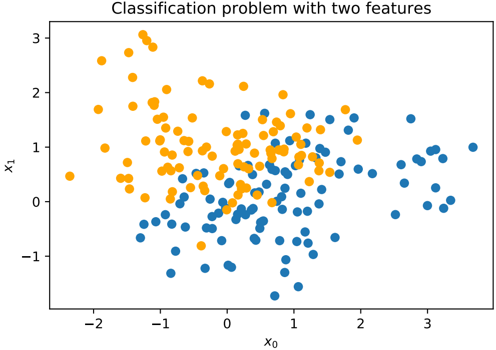
The target can be categorical, such as class membership (e.g. $y_i = \{0, 1\}$), or continuous (e.g. area, volume etc.).

Types of machine learning

- Unsupervised machine learning: given a dataset x_i, find "some interesting properties".
  - Clustering: find groupings of x_i
  - Density estimation: find $p(x_i)$
  - Generative models.
  - ...
- Supervised machine learning: given a training dataset $\{x_i, y_i\}$, predict $\hat{y}_i$ of previously unseen samples.
  - Regression: the target variables y_i are continuous.
  - Classification: the target variables y_i are categorical.
  - ...
- ...

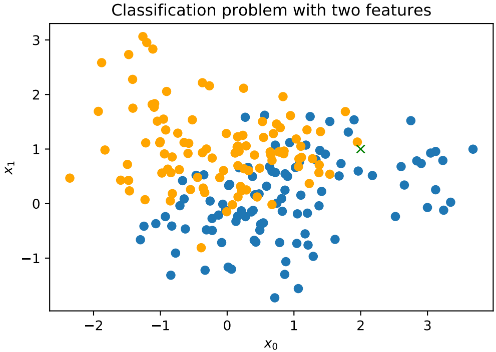
An intuitive model for classification

Assume we have a dataset with two classes (e.g. "benign" and "malignant") and each example in our dataset (e.g. a CT scan) is represented with two features: x_0 and x_1. For the purpose of this discussion, it does not matter which specific features are in question.



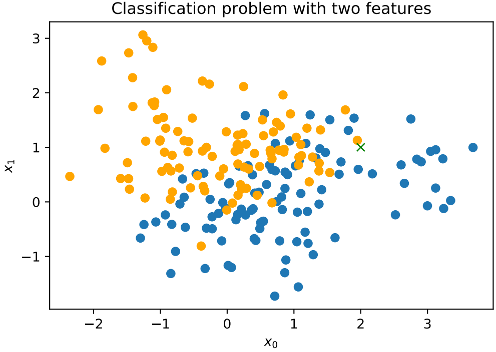
Nearest neighbour classifier

The goal of the machine learning model: when a new data point comes (e.g. a CT scan from a new patient) assign it to one of the two classes (i.e. classify the CT scan as "benign" or "malignant").

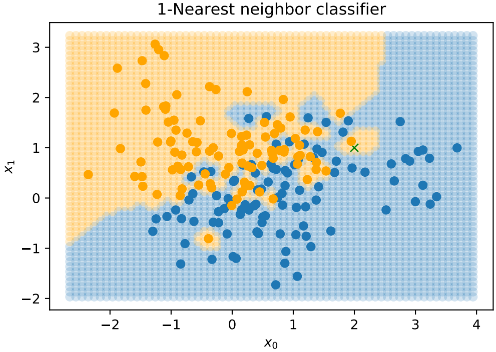


Nearest neighbour classifier

Assign the new data point the class of its nearest neighbour in feature space.

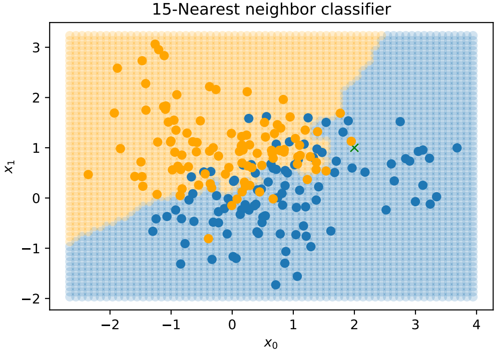


Decision boundary of a NN classifier



Is this a good decision boundary?

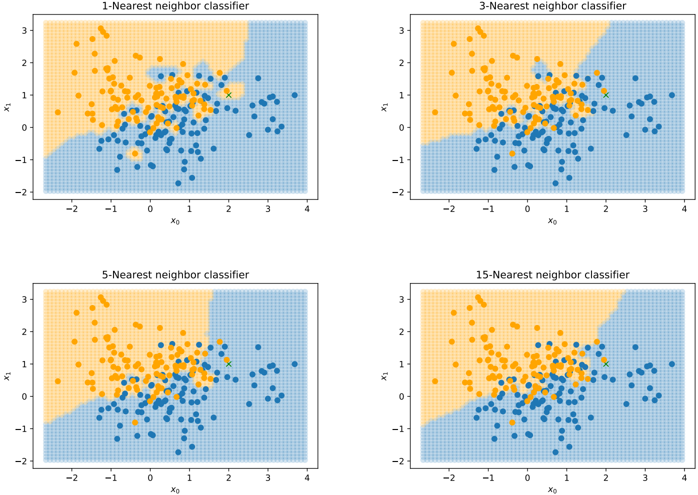
A more general method



Is this a better decision boundary?

k-Nearest neighbours classifier

Assign the new data point the majority class of its k nearest neighbour in feature space.



How is the class determined with k-NN?

For a new example with features $\boldsymbol{x}_{\text{new}} = [x_1, x_2]$, we want to predict the class $\hat{y_{\text{new}}}$.

- Compute the distance to all training samples $\boldsymbol{x}_i$.
  - Most commonly, we use the Euclidean distance:
  - $d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_i) = \sqrt{(x_{\text{new},1} - x_{i,1})^2 + (x_{\text{new},2} - x_{i,2})^2}$
- Sort the training samples based on the distance and pick the k nearest ones to the new example.
- Determine the class of the k nearest training samples.
- Assign to $\boldsymbol{x}_{\text{new}}$ the majority class of its nearest training samples (neighbours).

Some notes on extending k-NN

- Extension to more than two classes is trivial.
- Using k-NN for regression is also possible (e.g. instead of computing the majority class of the nearest neighbours we compute the average target value y).
- Using different distance metric is common, e.g. the L_1-distance:
- $d(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_i) = |x_{\text{new},1} - x_{i,1}| + |x_{\text{new},2} - x_{i,2}|$

Some remaining questions...

Q: We are in principle free to choose the value for k. But how to do this?

Some remaining questions...

Q: We are in principle free to choose the value for k. But how to do this?

A: "Pick the value for k that gives the best performance."

Some remaining questions...

Q: We are in principle free to choose the value for k. But how to do this?

A: "Pick the value for k that gives the best performance."

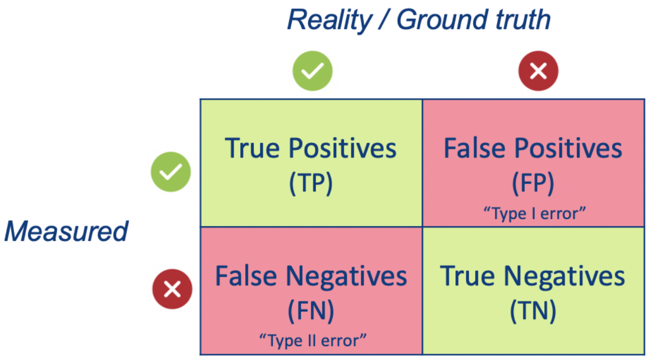
Q: How is the performance of a classification method measured? Which metric do we use and for which dataset do we compute it?

How do we measure the performance of machine learning methods?

The same evaluation metrics that were covered during the registration topic are also used for measuring the performance of classification methoods.

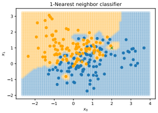
- Classification accuracy and error.
- Sensitivity (recall), specificity.
- Dice overlap (between the ground truth and predicted classes), which is often called the F_1 score in the context of object detection.

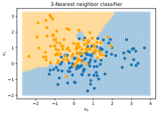
Types of errors in binary classification

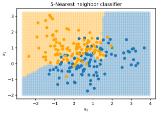


Performance as a function of k

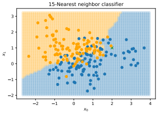
"Pick the value for k that gives the best performance."



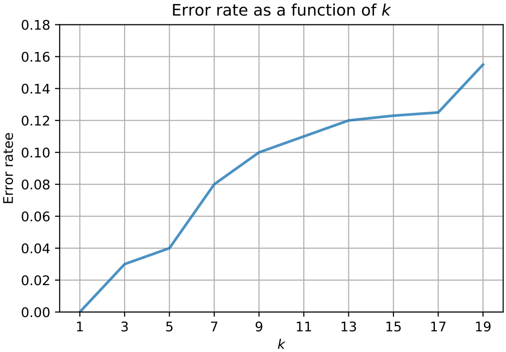




...



...



Choosing the optimal value of k

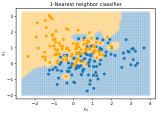
However, if we choose the optimal value of k based on the performance on the training set, we will always select $k = 1$ since in that case the training error is 0.

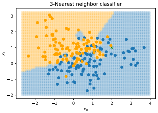
We need to choose k based on the performance on an *independent* test set.

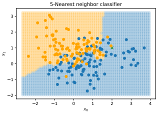
The test set should be independent in the sense that the examples that it contains should in no way be related to the ones in the training set.

Performance as a function of k

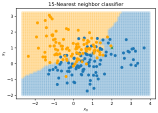
"Pick the value for k that gives the best performance."



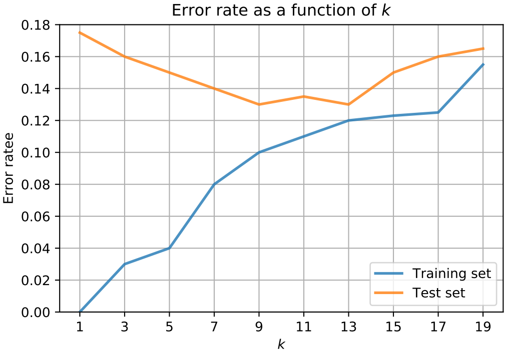




...



...



Generalisation of machine learning methods

The error on the independent test dataset is called the *generalisation* error. It tell us how well we can expect our classifier to generalise its performance on new, previously unseen examples.

In machine learning, we only care about the generalisation error since it is always trivial to design a machine learning model that has perfect performance on the training set (e.g. use 1-NN).

Generalisation and complexity

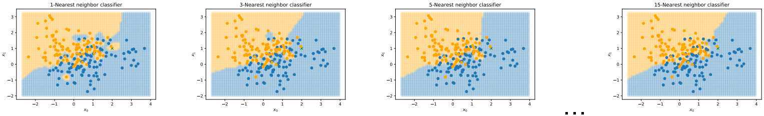
There is a relationship between the complexity of the machine learning model and its generalisation ability.

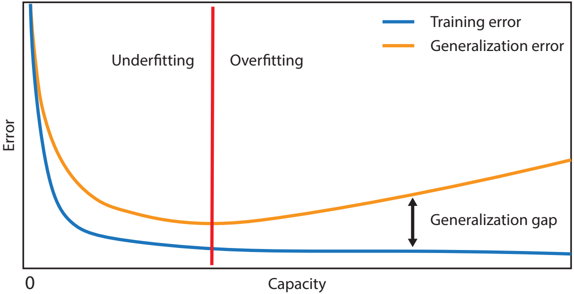
Classifiers that produce **simple** decision boundaries can have higher training errors but usually generalise better to new samples.

Classifiers that produce **complex** decision boundaries can have lower training errors but usually generalise worse to new samples.

Generalisation and complexity

Note that complexity **decreases** with k.





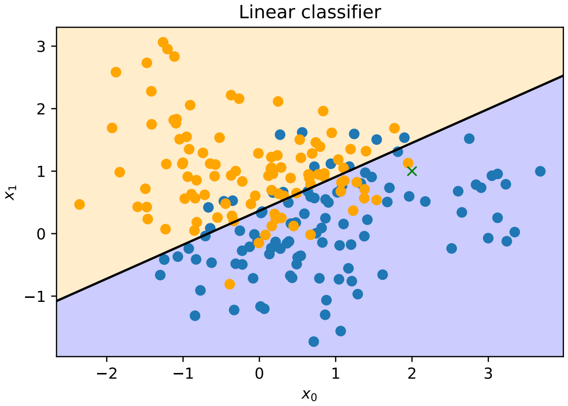
Types of machine learning

- Unsupervised machine learning: given a dataset x_i, find "some interesting properties".
  - Clustering: find groupings of x_i
  - Density estimation: find $p(x_i)$
  - Generative models.
  - ...
- Supervised machine learning: given a training dataset $\{x_i, y_i\}$, predict $\hat{y}_i$ of previously unseen samples.
  - Regression: the target variables y_i are continuous.
  - Classification: the target variables y_i are continuous.
  - ...
- ...

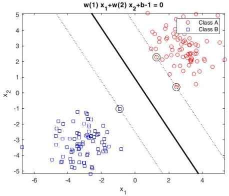
Types of machine learning models

- Parametric models
  - The number of parameters is fixed, i.e. it does not grow with the number of training samples
  - Once the model is trained (the parameters of the model are determined), we can "throw away" the training dataset.
- Non-parametric models
  - The number of parameters is not fixed, and it grows with the number of parameters.
  - k-NN is an example of a non-parametric machine learning model.

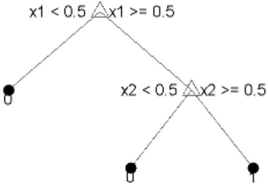
Linear model for classification



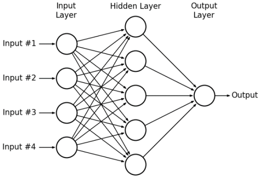
Other methods for classification and regression



Support vector machines



Decision trees

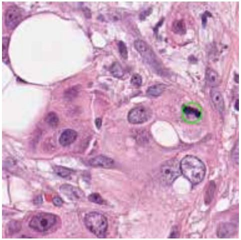


Neural networks

Figure source: mathworks.com

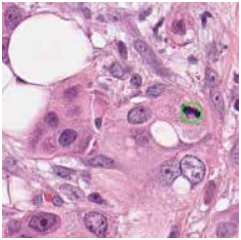
Discussion point 1

How can you apply machine learning for an object detection problem?



Discussion point 1

How can you apply machine learning for an object detection problem?



What are the "examples" being classified? What are the features representing the examples?

Discussion point 2

Suppose that you have a training dataset with 100,000 training samples. Which classifier will be computationally more efficient: 3-NN or 11-NN?

Discussion point 3

How can you apply a machine learning model for affine image registration?

Discussion point 4

Can you relate the concept of generalisation to the evaluation of image registration methods?