Jaime S. Cardoso
Full Professor
jaime.cardoso@inesctec.pt
jaime.cardoso@fe.up.pt
http://www.fe.up.pt/~jsc/

INESC TEC and Faculty of Engineering of University of Porto
Portugal

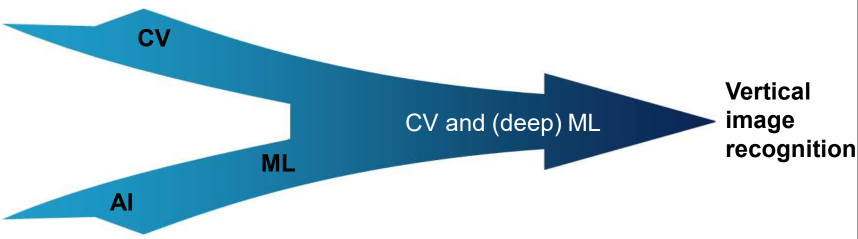
# Basics II
# (Shallow) Machine Learning
# (for Computer Vision)

July 02nd, 2022, Porto, Portugal

1

## Roadmap

- **A brief history of Computer Vision**
  - Convergence of Machine Learning and Signal Processing and Computer Vision




- **The main components in ML**
- Deep learning and Vertical Image Recognition

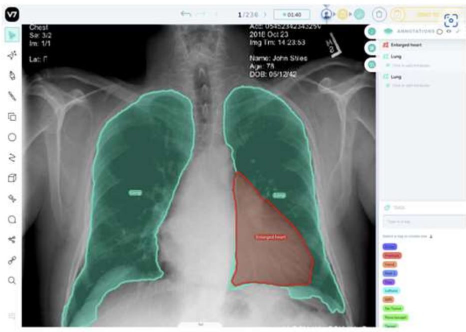


2

# Applications

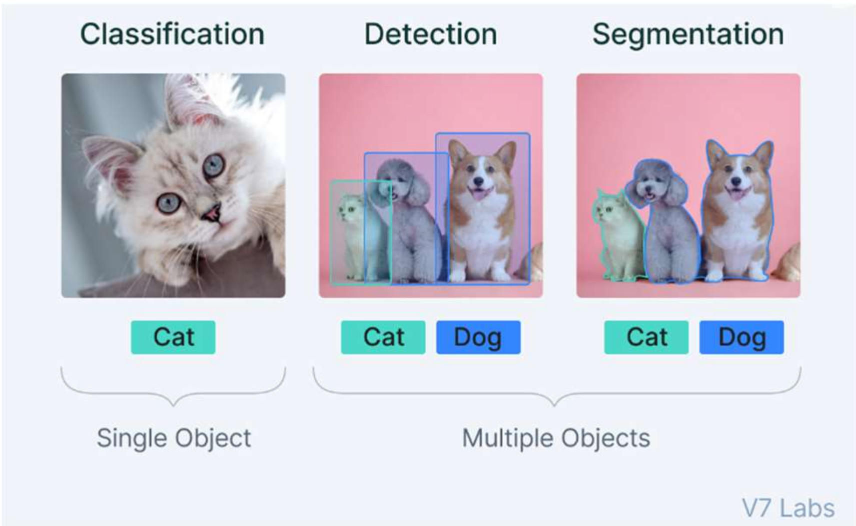
**Autonomous Vehicles**


**Medical Image Analysis**



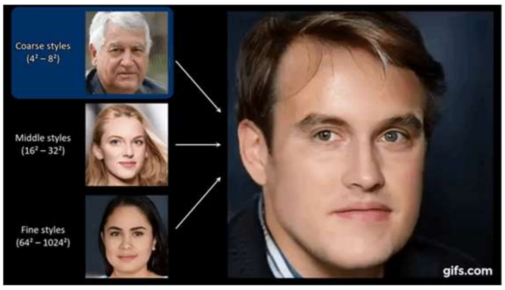
3

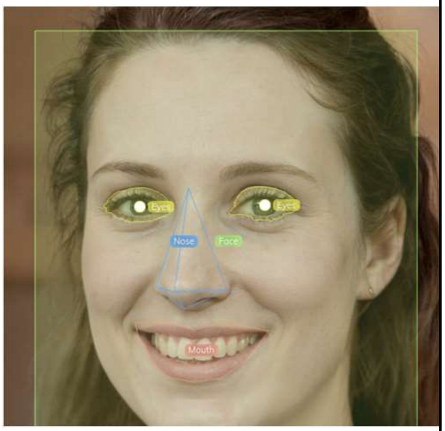
# Common CV Tasks




4

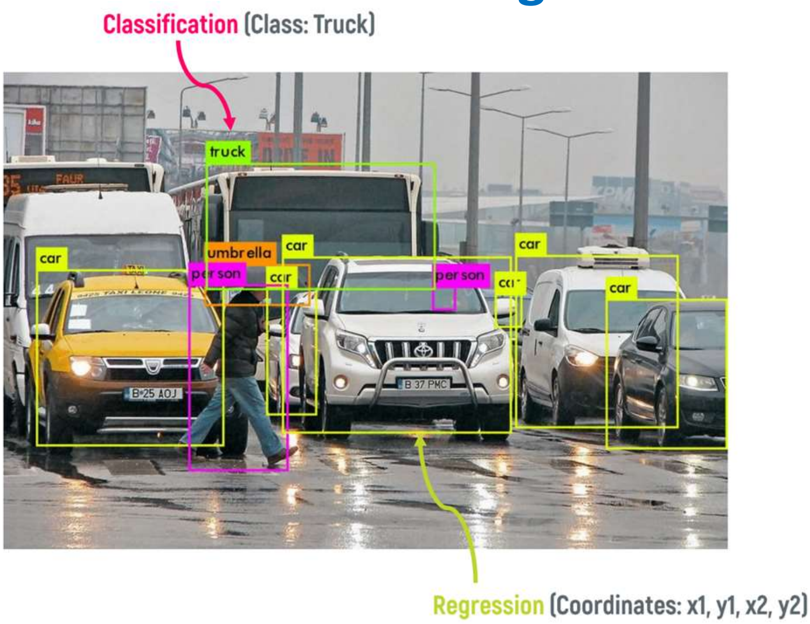
## More common CV Tasks




5

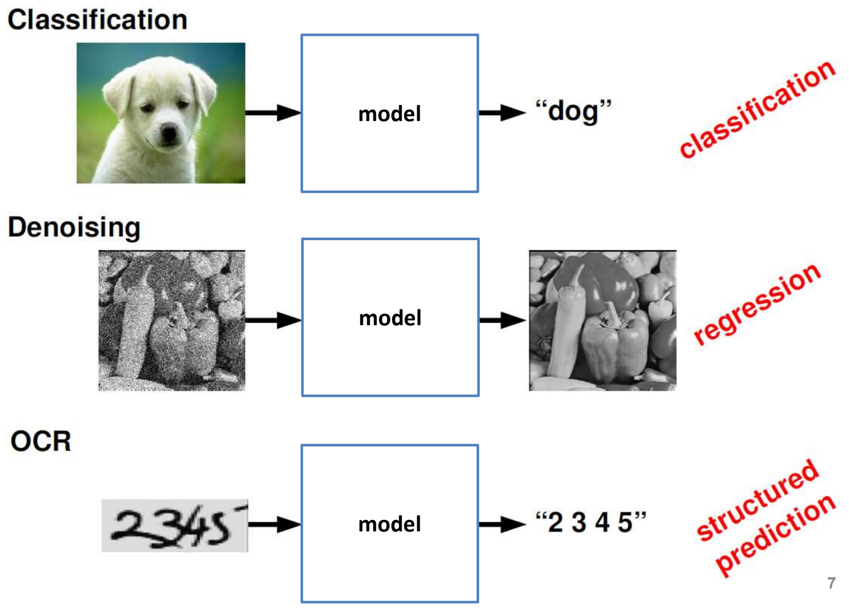
## Classification+Regression




6

6

# Supervised Learning: Examples




7

7

# Taxonomy of the Learning Settings

Goals and available data dictate the type of learning problem

- Supervised Learning
  - Classification
    - Binary
    - Multiclass
      - Nominal
      - Ordinal
  - Regression
  - Ranking
  - Counting
- Semi-supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- etc.

8

8

# Developing a Model

- As in any other computer tasks, modelling requires a “program” providing **detailed instructions**
- These instructions are typically mathematical equations, which characterize the relationship between inputs and outputs
- Formulating these equations is the central problem in modelling

9

9

# Developing a Model – Types of Models

**Models**
**How can we define these instructions and equations?**

- **Fixed Models (model driven)**
- **Parametric models (model + data driven)**
- **Non-parametric models (data driven)**

10

10

# Developing Fixed Models

- Closed-form equations that define how the outputs are derived from the inputs
- Being **all the characteristics fixed** when the equations are derived we refer them as fixed models
- Suitable for **simple and fully understood problems**

***Example:***

Compute how much it takes an apple to hit the ground on Earth:

$$t = \sqrt{\frac{2h}{9.8}}$$

**Most problems are too complex and / or not sufficiently understood for us to use fixed models.**

11

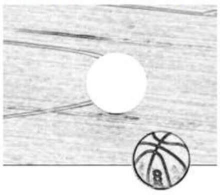
11

Input image

This image is CC0 1.0 public domain

Edge image

2 ½-D sketch



3-D model

This image is CC0 1.0 public domain

| Input Image | Primal Sketch | 2 ½-D Sketch | 3-D Model Representation |
|---|---|---|---|
| Perceived intensities | Zero crossings, blobs, edges, bars, ends, virtual lines, groups, curves boundaries | Local surface orientation and discontinuities in depth and in surface orientation | 3-D models hierarchically organized in terms of surface and volumetric primitives |

Stages of Visual Representation, David Marr, 1970s

Fei-Fei Li et al.

12

# Artificial Intelligence (AI)

- " [...automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning..." (Bellman, 1978)
- " The branch of computer science that is concerned with the automation of intelligent behaviour." (Luger and Stubblefield, 1993)
- "The ultimate goal of AI is to create technology that allows computational machines to function in a highly intelligent manner. (Li Deng 2018)

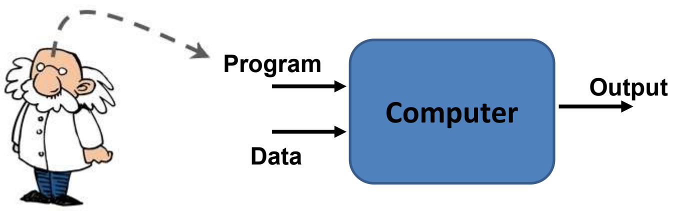
13

13

# AI: three generations

1st wave of AI: **the sixties**

- emulates the decision-making process of a human expert




14

14

## AI: three generations

1st wave of AI: **the sixties**

- Based on expert knowledge
  - "if-then-else"
- Effective in narrow-domain problems
- Focus on the head or most important parameters (identified in advance), leaving the "tail" parameters and cases untouched.

- Transparent and interpretable
- Difficulty in generalizing to new situations and domains
- Cannot handle uncertainty
- Lack the ability to learn algorithmically from data

15

15

## History of ideas in CV (recognition)

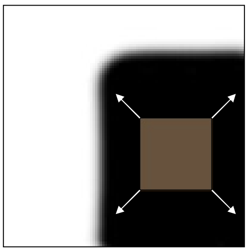
- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features

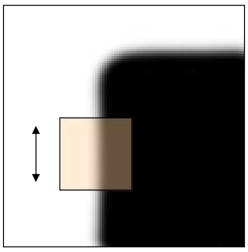
Svetlana Lazebnik

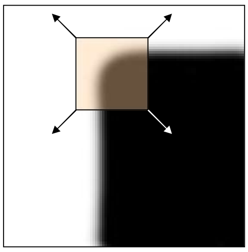
16

## Corners

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity







"**flat**" region: no change in all directions

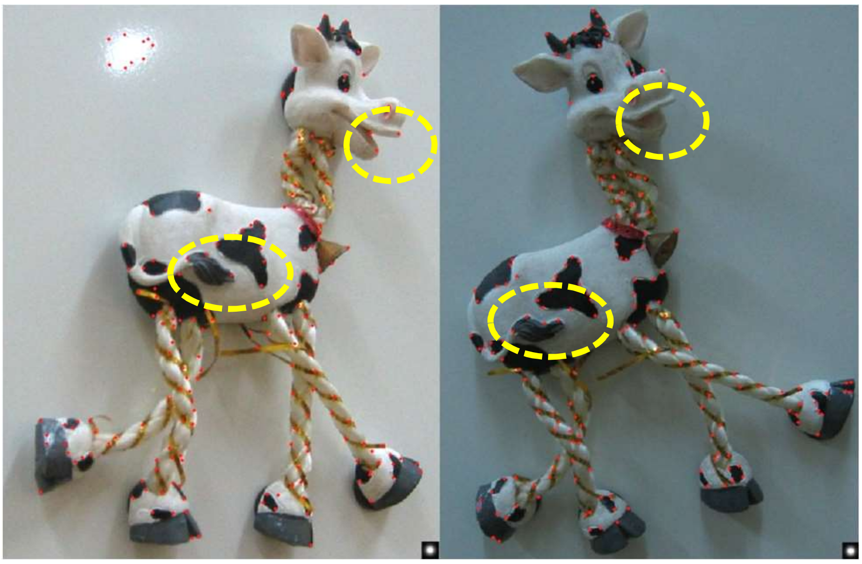
"**edge**": no change along the edge direction

"**corner**": significant change in all directions

Alyosha Efros

17

## Harris corner detector

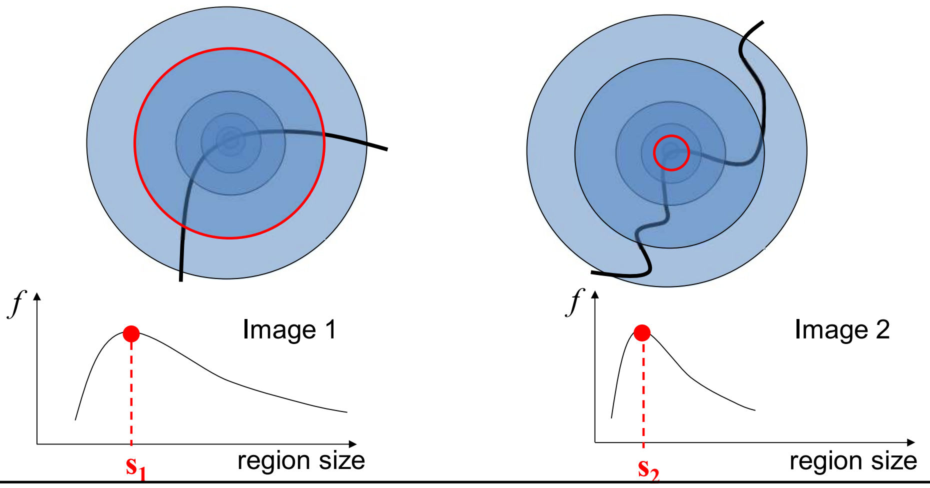


Darya Frolova, Denis Simakov

18

# Scale invariant detection

**Intuition** - Find scale that gives local maxima of some signature function $f$ in **both position and scale**.

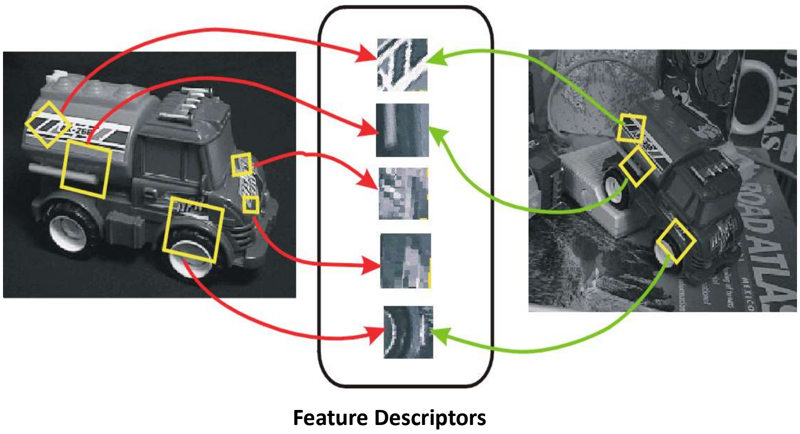



19

# Invariant local features

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, …




20

# Local descriptors

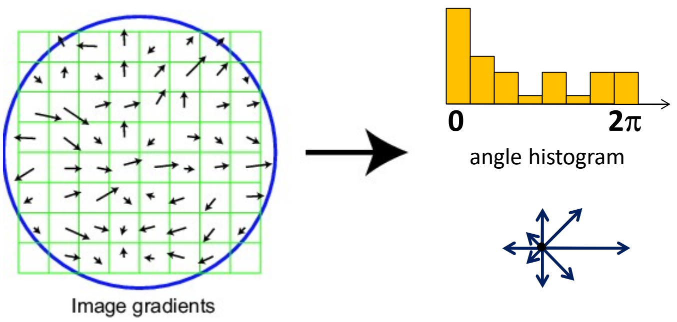
- In each detected feature (point), a descriptor is then extracted
- Histogram-based descriptors
  - Based on the histogram of oriented gradient
  - SIFT, SURF, GLOH and HOG
- Compact descriptors
  - Based on binary strings obtained comparing pairs of image intensities
  - BRIEF, ORB, BRISK and FREAK

21

# SIFT descriptor

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations




**Distinctive image features from scale-invariant keypoints**. David G. Lowe. *IJCV* 60 (2), pp. 91-110, 2004.

22

## History of ideas in recognition

- 1960s – early 1990s: the geometric era
- 1990s: appearance-based models
- Mid-1990s: sliding window approaches
- Late 1990s: local features

Sensor → Pixels → Filtering → Features → Decisions

Svetlana Lazebnik

23

## Developing a Model – Types of Models

**Models**
**How can we define these instructions and equations?**

**Fixed Models (model driven)**

**Parametric models (model + data driven)**

**Non-parametric models (data driven)**

24

24

# Developing Parametric Models

**You need to estimate the parameter g!**

Go to Mars and collect some data:

| Height (h) | Falling time (t) |
|---|---|
| 0.5 | 0.2 |
| 1.3 | 0.4 |
| 2.8 | 0.46 |
| 4 | 0.68 |
| 7.3 | 0.7 |

Inputs
(in this case the input vector has only one component)

Outputs

25

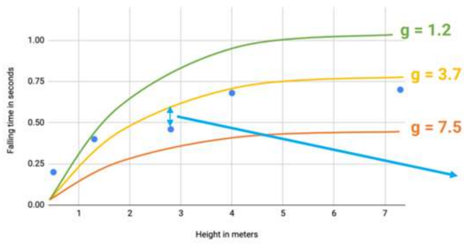


# Developing Parametric Models

A value should be selected for g so the model produces estimates close to the measured times when presented with the corresponding heights as input.

**Search for the parameter that leads to the predictions that are closest to the observed data:**

Seems to be the best model




Due to inaccuracies in the structure of the model or measurement errors

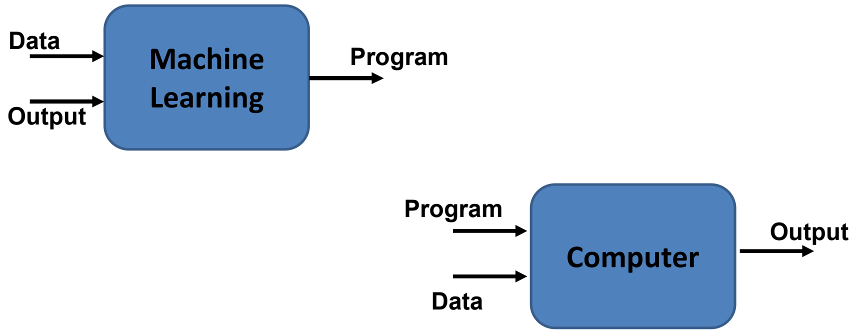
In this case is easy to define a good value for g but this is not generally possible in most problems which involve complex relationships and multiple variables

26

# AI: three generations

2<sup></sup>nd wave of AI: **the eighties**

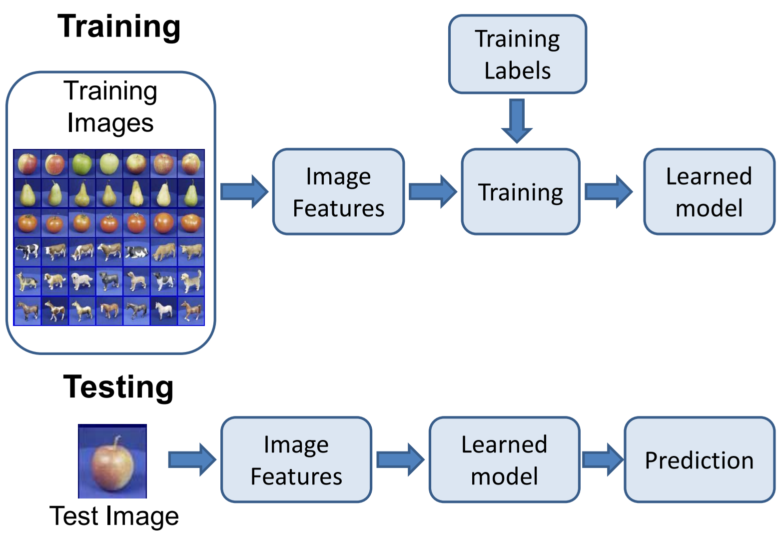
- Based on (shallow) machine learning




27

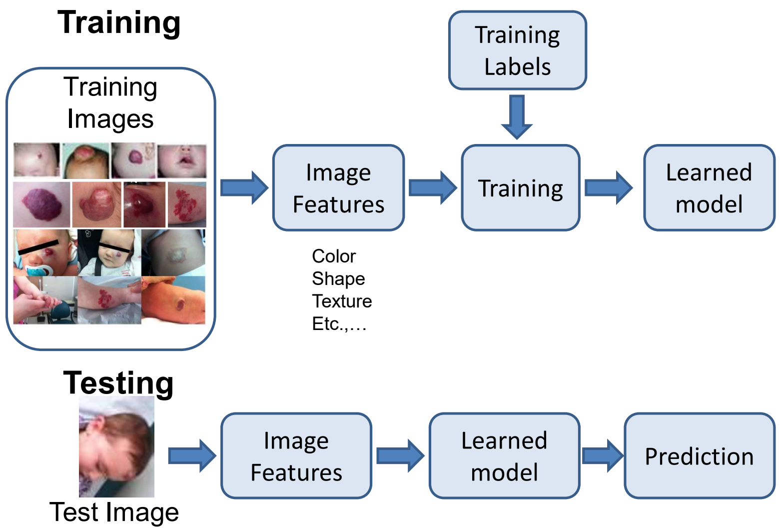
**27**

# Classification in computer vision




**28**

# Classification in computer vision



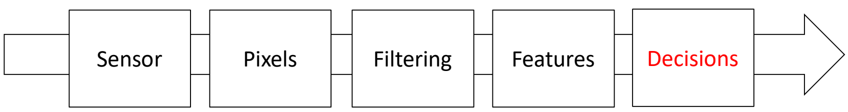

Features are designed by humans, requiring significant expertise

29

29

# An example: our system

- **Sensor**
  – The camera captures a 2D image
- **Preprocessing**
  – Adjustments for average intensity levels
  – Segmentation to separate object from background
- **Feature Extraction**
  – Assume a specialist told us that length and color help on the classification task.




30

30

## An example: multiple features

- We can use two features in our decision:
  - lightness: $x_1$
  - length: $x_2$
- Each lesion image is now represented as a point (feature vector)

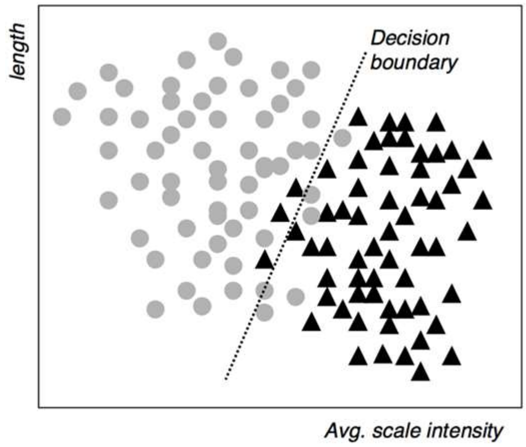
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

in a two-dimensional **feature space**.

31

31

## An example: multiple features




Scatter plot of lightness and length features for training samples. We can compute a **decision boundary** to divide the feature space into two regions with a classification rate of 95.7%.

32

32

## The problem of overfitting

**Models rely on training data to learn**

If we allow too much complexity, the model will "memorize" the training data, instead of extracting useful relationships

**OVERFITTING**

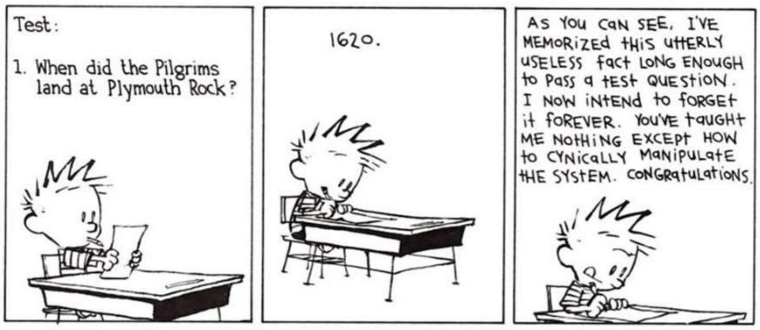




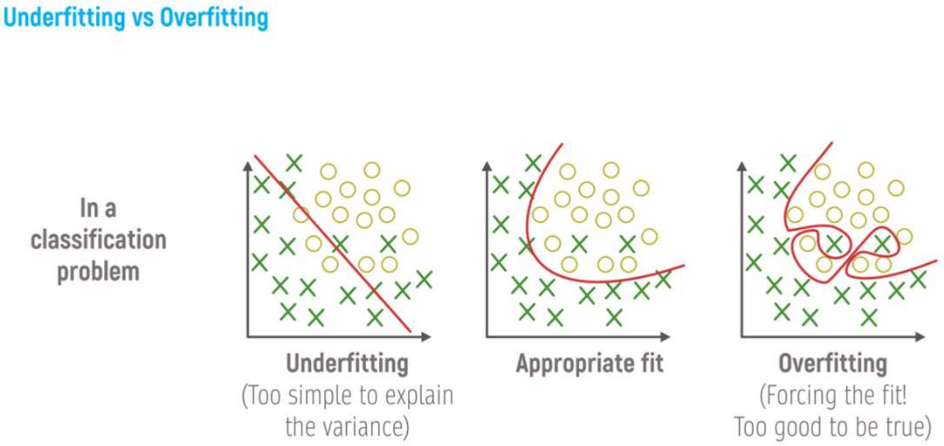
## The problem of overfitting

**Memorizing vs Understanding**

- Overfitting is like when someone memorizes things to pass an exam
  - He'll be too biased on the exercises he saw in classes
  - If he gets a slightly different question in the exam, he won't know how to answer



# The problem of overfitting




35

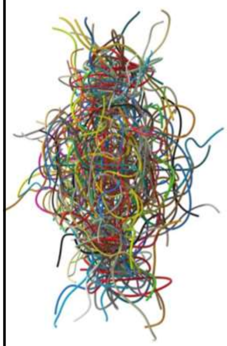
35

# The problem of overfitting


36

36

# Developing Non-Parametric Models



- Models that **rely heavily on data**, instead of human expertise, can be called data-driven models
- Used in **complex problems** where the relationship between inputs and outputs is not known
- Requires **large amounts of data**

37

37

# Parametric vs Non-Parametric Models

Parametric model | **assume known distributions in the data**

**Simplicity** – Easier to understand and interpret
**Training Speed** – Faster to train and learn from data
**Less training data** – Does not require a huge quantity of data for training and work well even if the fit to the data is not perfect

**Constrained** – Limited by the functional form chosen
**Limited complexity** – Better suited to less complex problems
**Poor fit**– In practice the methods are unlikely to match the underlying mapping function – do not offer the best fit to data
**In pre-processing -** the analyst often spends considerable time transforming data so it stands with some specific distribution (for example normal distribution)

Linear Regression
Logistic Regression
Perceptron
Naïve Bayes
...

38

38

## Parametric vs Non-Parametric Models

**Nonparametric model |** **do not assume distributions in the data**

**Flexibility** – Capable of fitting a large number of functional forms
**Power** – No assumptions (or weak ones) about the underlying function. Learn from data.
**Performance** – Can result in higher accuracy since they offer better fit
**In pre-processing –** there is no distribution assumptions and time can be saved in preprocessing steps (e.g. no need to transform the data to normal distributions)

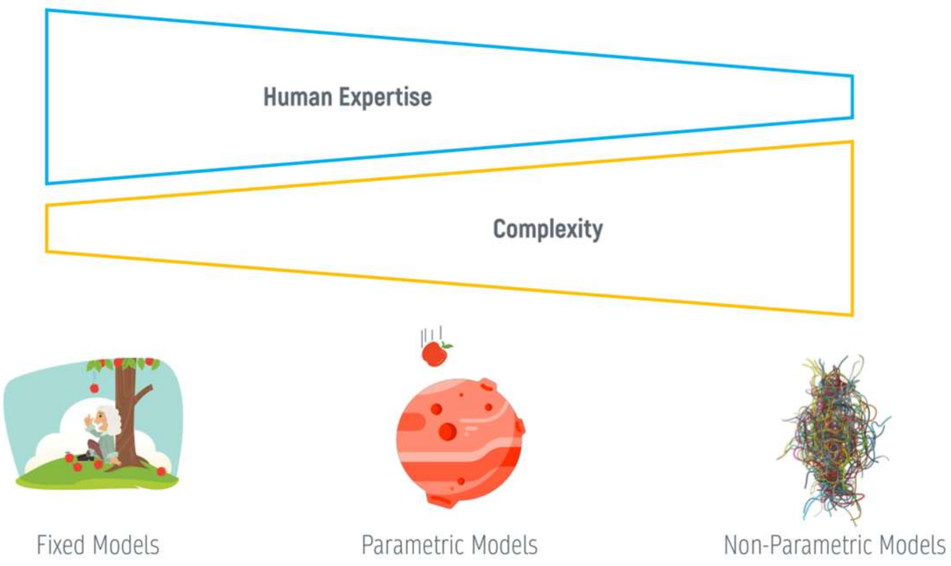
**Training data** – Require more data than the parametric models
**Slower** – Slower due to the several parameters needed to train
**Overfitting** – Higher risk of overfitting the training data
**Lack of interpretability –** Harder to explain why specific predictions are made in some of the algorithms

- KNN
- Decision Trees
- Non-linear SVM
- ...

39

39

## Model Recap




40

40

... but with common traits

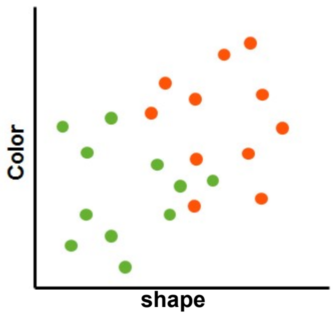
# FOR THE SAME PROBLEM, DIFFERENT SOLUTIONS

41

41

# Data Driven Design

- When to use?
  - Difficult to reason about a generic rule that solves the problem
  - Easy to collect examples (with the solution)

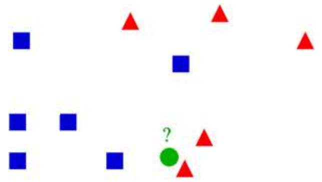



42

42

# k-Nearest neighbour classifier

- For a new point, find the $k$ closest points from training data
- Labels of the $k$ points "vote" to classify



k = 1

If the query lands here, the 1NN consist of 1 positive, so we classify it as positive.

Blue = negative
Red = positive

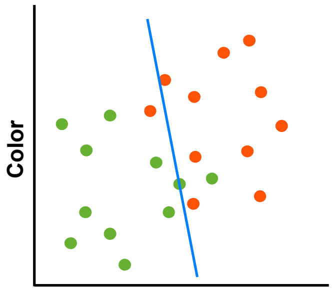




# kNN as a classifier

- **Advantages**:
  - Simple to implement
  - Flexible to feature / distance choices
  - Naturally handles multi-class cases
  - Can do well in practice with enough representative data
- **Disadvantages:**
  - Large search problem to find nearest neighbors → Highly susceptible to the **curse of dimensionality**
  - Storage of data
  - Must have a meaningful distance function

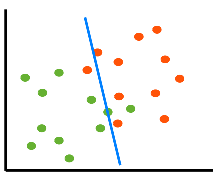
# Design of a Classifier

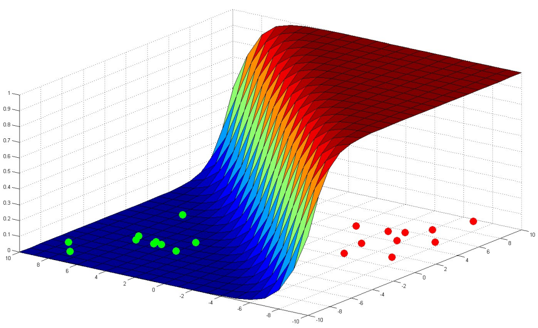



45

45

# Design of a Classifier

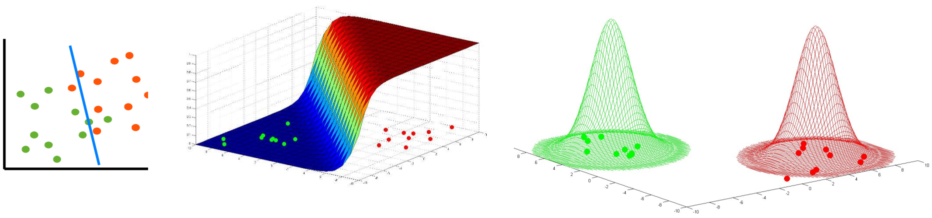




46

46

## Design of a Classifier



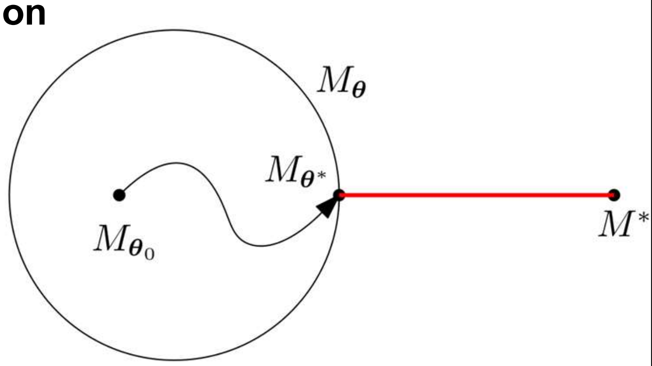




**DIFFERENT SOLUTIONS BUT WITH COMMON INGREDIENTS**

# Common steps

- The learning of a model from the data entails:
  - **Model representation**
  - **Goal Function (Loss/Cost or Fitness)**
  - **Optimization**

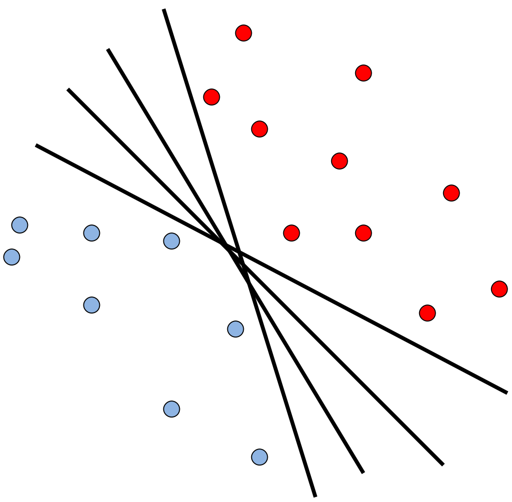







# Linear classifiers

Find linear function to separate positive and negative examples

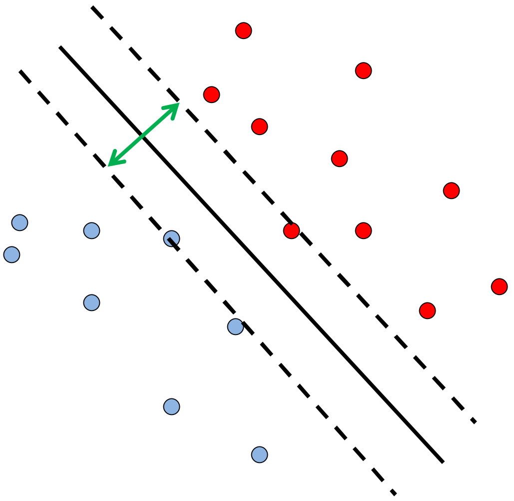


$\mathbf{x}_i$ positive : $\mathbf{x}_i \cdot \mathbf{w} + b \geq 0$

$\mathbf{x}_i$ negative : $\mathbf{x}_i \cdot \mathbf{w} + b < 0$

Which line
is best?

# Support Vector Machines



Classifier based on *optimal separating line (for 2D case)*

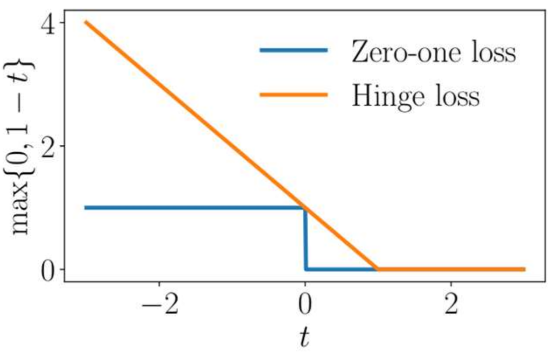
Maximize the ***margin*** between the boundary and the positive and negative training examples

51

51

# Loss Function

- Hinge Loss

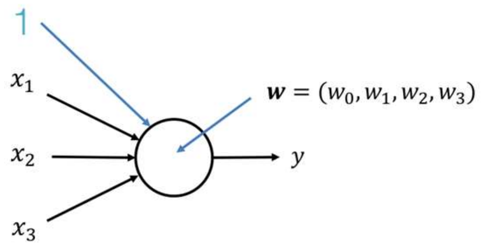



52

52

# Perceptron

- Weights: $\boldsymbol{w}$
  - Strength of the link from input $\boldsymbol{i}$
  - Input signals $x_i$ weighted by $w_i$ and linearly combined: $a = \sum_i w_i x_i + w_0$
- Activation function: $\boldsymbol{h}$
  - Numerical signal produced: $y = h(a)$

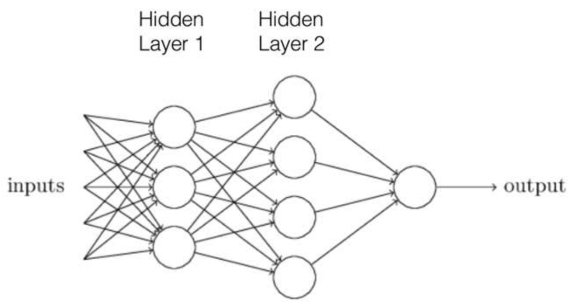



adapted from Pascal Poupart

53

53

# Multi-layer perceptron (MLP)




Sets of layers and the connections (weights) between them define the network architecture.

Each layer receives its inputs from the previous layer and forwards its outputs to the next layer

Explanation in 3Blue1Brown about the multi-layer perceptron
https://www.youtube.com/watch?v=aircAruvnKk

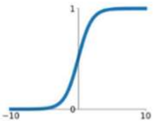
adapted from Nielsen

54

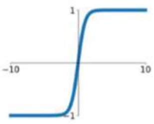
54

# Activation function

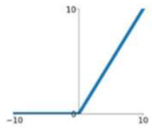
**Sigmoid**
$\sigma(x) = \frac{1}{1+e^{-x}}$



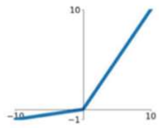
**tanh**
$\tanh(x)$



**ReLU**
$\max(0, x)$



**Leaky ReLU**
$\max(0.1x, x)$



**Maxout**
$\max(w_1^T x + b_1, w_2^T x + b_2)$

**ELU**
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

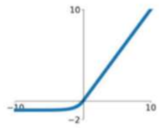


Image source: https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092 55

55

# Typical Loss functions

- Regression
  - Mean Squared Error (MSE) / L2
  - Mean Absolute Error (MAE) / L1
- Binary Classification
  - Binary Cross-Entropy (BCE)
  - Hinge Loss
- Multi-Class Classification
  - Multi-Class Cross-Entropy (CE)

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - x_i|}{n}$$

$$\text{BCE} = -\frac{1}{m}\sum_{i=1}^{m}(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i))$$

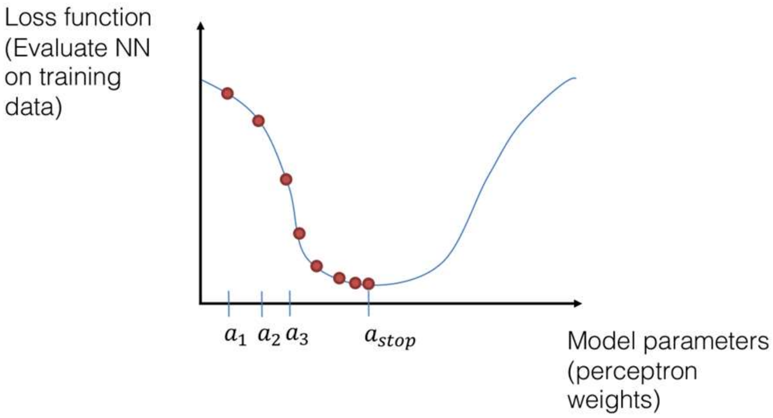
$$\text{CE} = -\frac{1}{m}\sum_{i=1}^{m} y_i \cdot \log(\hat{y}_i)$$

- The output of the last layer must be coupled with the loss function:
  - Regression → linear activation
  - Binary classification → sigmoid
  - Multiclass classification → softmax

56

56

# Train NN with Gradient Descent




James Tompkin

57

**57**

# Sequential gradient descent

$$w_{ji} \leftarrow w_{ji} - \eta \frac{\partial \, L}{\partial w_{ji}}$$

$\partial$ L ⟶ Error or Loss

$\eta$ ⟶ Learning rate or step length

- In practice, the training is typically done using **sequential gradient descent**, i.e. in each iteration (step), calculate the error and update the weights
- A complete pass over the training set is called an <u>epoch</u>
- How can we compute the gradient efficiently given an arbitrary network structure?
  - backpropagation algorithm

Explanation in 3Blue1Brown about the training process
https://www.youtube.com/watch?v=IHZwWFHWa-w

58

**58**

Hyper parameters / user defined parameters

# AVOIDING OVERFITTING AND DATA MEMORIZATION

59

59

## Regularization

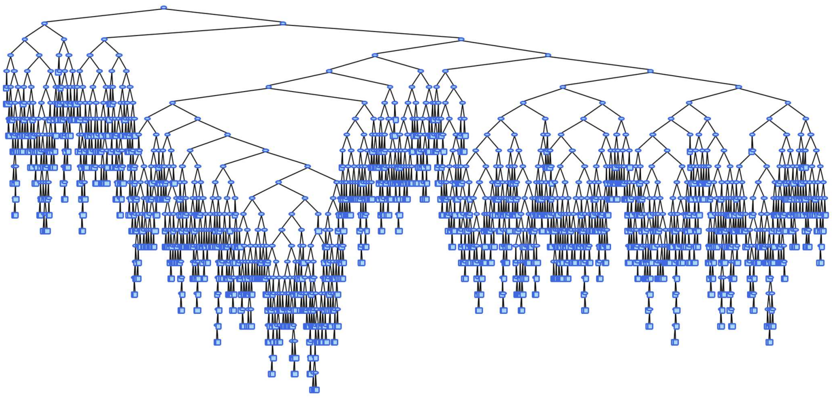
- To build a machine learning algorithm we specify model family, a cost function and optimization procedure
- Regularization is any modification we make to a learning algorithm that is intended to reduce its generalization error but not its training error
  - There are many regularization strategies
- Regularization works by trading increased bias for reduced variance. An effective regularizer is one that makes a profitable trade, reducing variance significantly while not overly increasing the bias.

60

60

## Decision Tree

- Overfitting in decision trees

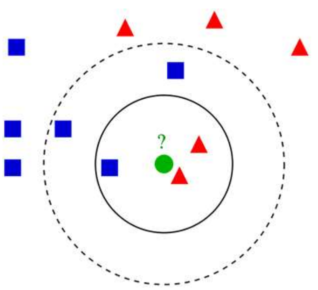


61

61

## k-Nearest neighbour classifier

- For a new point, find the $k$ closest points from training data
- Labels of the $k$ points “vote” to classify




k = 5

If the query lands here, the 5 NN consist of 3 negatives and 2 positives, so we classify it as negative.

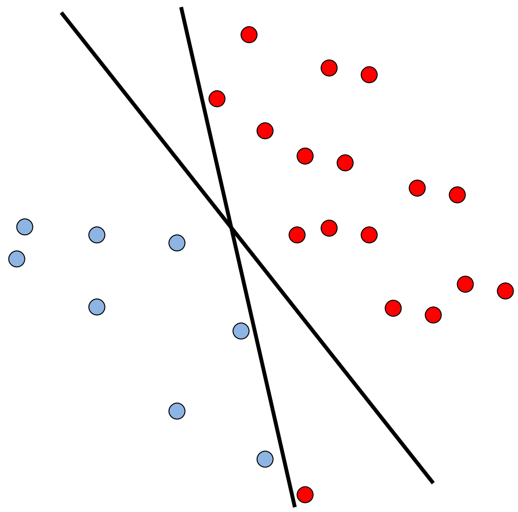
Blue = negative
Red = positive

62

62

# Linear classifiers

Find linear function to separate positive and negative examples



Which line is best?

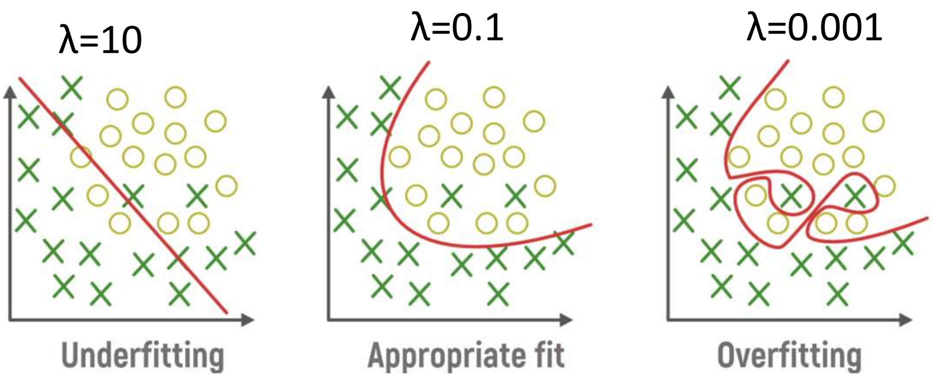
63

**63**

# Regularization

Cost Function

– Minimize (error in data) + λ (model complexity)

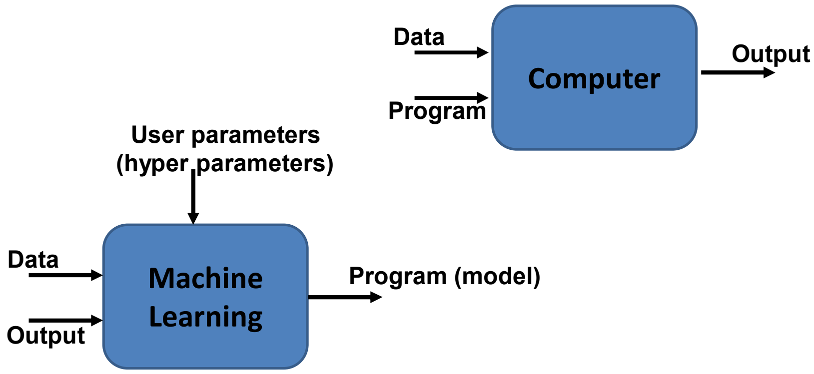



64

**64**

# What is Machine Learning?

- Automating the Automation




65

65

THERE ARE SO MANY OPTION TO DESIGN A CLASSIFIER…

# A FAIR JUDGEMENT OF YOUR ALGORITHM

66

66

## Model assessment, selection

- How to Compare Models?
- How can we select the right complexity model ?

67

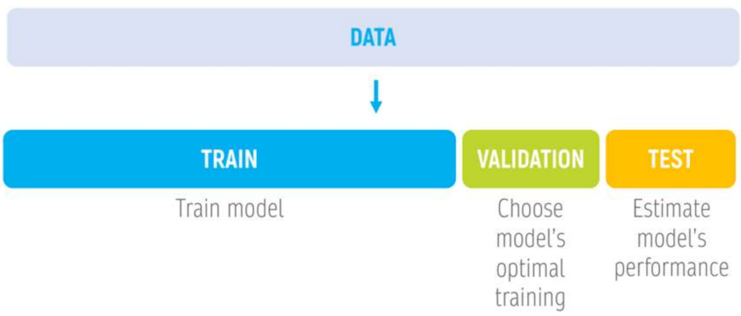
67

## Training - general strategy

**How to avoid overfitting?**

How to prepare for the unknown?

- Keep some data aside!




68

68

## Training - general strategy

For this technique to work, you need to make sure both parts are **representative** of your data. A *good practice* is to **shuffle** the order of the dataset before *splitting*.


69

69

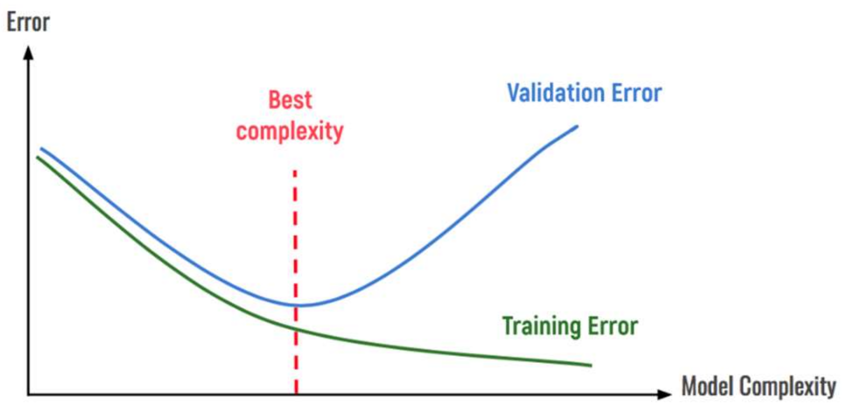
## The problem of overfitting

| | Low *Training* Error | High *Training* Error |
|---|---|---|
| Low *Testing* Error | The model is learning! | Probably some error in your code. Or you've created a *psychic* AI. |
| High *Testing* Error | OVERFITTING | The model is not learning. |

70

70

# Training - general strategy

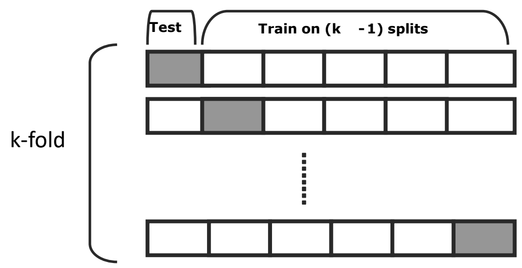



71

71

# Hold out / test set method

- It is simple, however
  - We waste some portion of the data
  - If we do not have much data, we may be lucky or unlucky with our test data
- With **cross-validation** we reuse the data

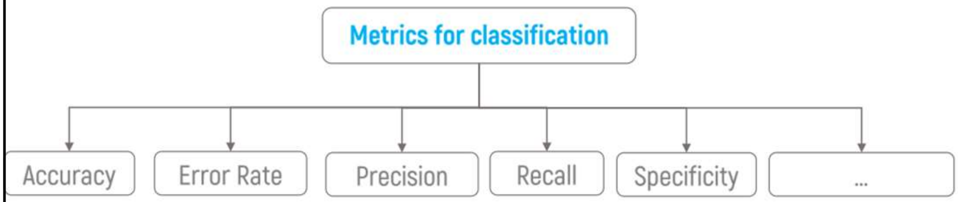



72

72

# Evaluation Metrics

An evaluation metric quantifies the performance of a predictive model




73

73

# Evaluation Metrics

An evaluation metric quantifies the performance of a predictive model

It all starts with ...

| | | True Class | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted Class | Positive | TP | FP |
| | Negative | FN | TN |

Type I Error

Type II Error

In classification, predictions are either correct or wrong.

We can encode this in a **confusion matrix**.

74

74

# References


- Richard Szeliski, *Computer Vision: Algorithms and Applications*, 2010 - http://szeliski.org/Book/
- David Forsyth and Jean Ponce, *Computer Vision: A Modern Approach* 2nd Edition, 2012
- Simon J. D. Prince, *Computer Vision: Models, Learning, and Inference*, 2012
- Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing* 3rd Edition, 2007
- Richard Hartley and Andrew Zisserman, *Multiple View Geometry* 2nd Edition, 2004
- Ian Goodfellow , Yoshua Bengio, Aaron Courville and Francis Bach, *Deep Learning*, 2016


75

# References


- Fei-Fei Li et al. (Stanford University) - CS 131 Computer Vision: Foundations and Applications
  - http://vision.stanford.edu/teaching/cs131_fall1617/index.html
- James Tompkin et al. (Brown University) - CSCI 1430: Introduction to Computer Vision
  - https://cs.brown.edu/courses/csci1430/
- Kristen Grauman et al. (University of Texas at Austin) - CS 376: Computer Vision
  - http://vision.cs.utexas.edu/376-spring2018/
- Rob Fergus et al. (New York University) - CSCI-GA.2271-001: Computer Vision
  - https://cs.nyu.edu/~fergus/teaching/vision/index.html


76

# References


- Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
- Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, John Wiley & Sons, 2001
- Thomas Mitchell, Machine Learning, McGraw-Hill, 1997.
- P. Domingos, "A few useful things to know about machine learning," CACM, 2012
- Andrew Moore, Support Vector Machines Tutorial, http://www.autonlab.org/tutorials/svm.html
- Supervised learning models Model Evaluation and Comparison, Evaluation Metrics, Carina Albuquerque


77

77

# References


- Selim Aksoy, Introduction to Pattern Recognition, Part I, http://retina.cs.bilkent.edu.tr/papers/patrec_tutorial1.pdf
- Ricardo Gutierrez-Osuna, Introduction to Pattern Recognition, http://research.cs.tamu.edu/prism/lectures/pr/pr_l1.pdf
- Pedro Domingos, Machine Learning, http://courses.cs.washington.edu/courses/cse446/14wi/
- Kristen Grauman, Discriminative classifiers for image recognition, http://www.cs.utexas.edu/~grauman/courses/spring2011/slides/lecture22_classifiers.pdf
- Victor Lavrenko and Nigel Goddard, Introductory Applied Machine Learning, http://www.inf.ed.ac.uk/teaching/courses/iaml/


78

78

# References


- Recognizing and Learning Object Categories
  http://people.csail.mit.edu/torralba/shortCourseRLOC/index.html
- Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes, (K. Murphy, A. Torralba, W. Freeman), NIPS 2003
- Max-Margin Markov Networks , (B. taskar, C. Guestrin, D. Koller), NIPS 2004
- Large Margin Methods for Structured and Interdependent Output Variables, (I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun), JMLR, vol 6, 2005
- Learning Spatial Context: Using Stuff to Find Things, (G. heitz, D. Koller), ECCV 2008,
  http://ai.stanford.edu/~gaheitz/Research/TAS/
- An Empirical Study of Context in Object Detection, (S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, M. Hebert), CVPR 2009
  http://www.cs.cmu.edu/~santosh/projects/context.html
- Generative Models for Visual Objects and Object Recognition via Bayesian Inference, L. Fei-Fei, 2006
- Modeling Mutual Context of Object and Human Pose in Human-Object Interaction Activities, (B. Yao, L. Fei-Fei), CVPR 2010
  http://videolectures.net/cvpr2010_fei_fei_mmco/
- No Hype, All Hallelujah: Structured Models in Computer Vision, (S. Nowozin), NIPS 2010


79

79

# References


- Graphical Models for Time Series, (D. Barker, A. T. Cemgil), IEEE Signal Processing Magazine, vol 27, 2010
- Dynamic Graphical Models, (J. Bilmes), IEEE Signal Processing Magazine, vol 27, 2010
- A Martingale Framework for Detecting Changes in Data Streams by Testing Exchangeability, (S. Ho, H. Wechsler), TPAMI 2010
- Introduction to Statistical Relational Learning, (L. Getoor, B. Taskar), The MIT Press 2007
- Combining Video and Sequential Statistical Relational Techniques to Monitor Card Games, (L. Antanas, B. Gutmann, I. Thon, K. Kersting, L. De Raedt), ICML 2010
- Relational Learning for Collective Classification of Entities in Images, (A. Chechetka, D. Dash, M. Philipose), AAAI 2010
- Grouplet: A Structured Image Representation for Recognizing Human and Object Interactions, (B. Yao, L. Fei-Fei), CVPR 2010


**Thank You for Your Attention!**

80

80