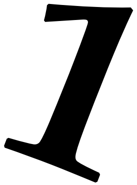
Machine Learning: Introduction

Battista Biggio

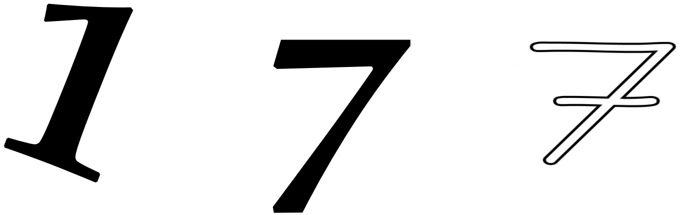
What Number Is This?

7

What Number Is This?



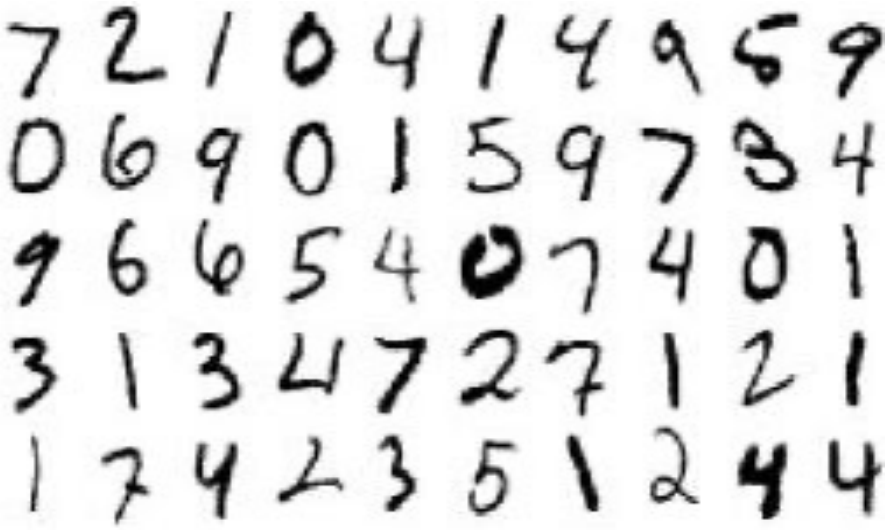
What Number Is This?



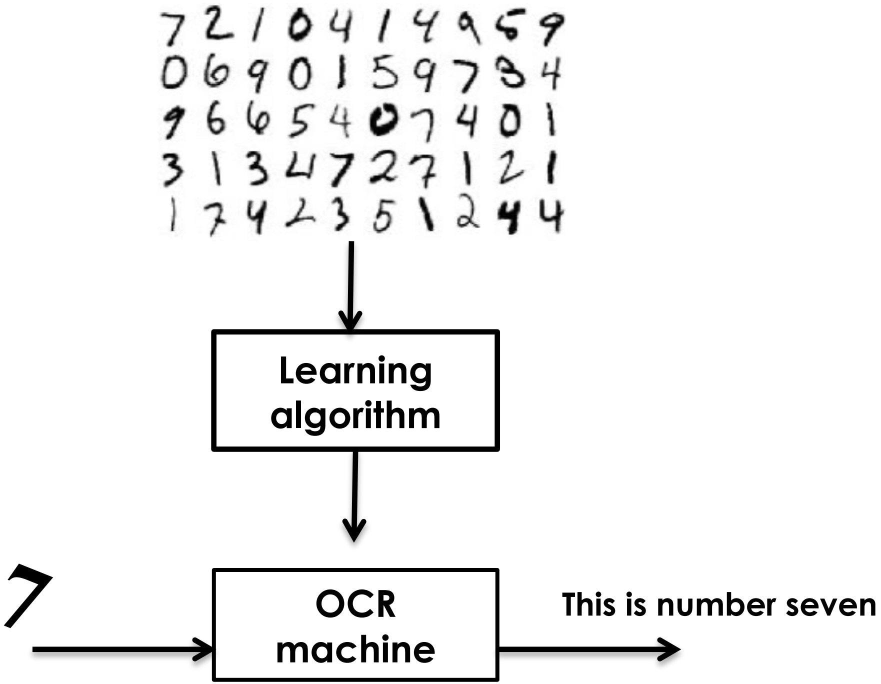
- Are you able to write in Python (or any other language) the **exact algorithm** (step after step) that you use to **recognize** the above numbers?

Writing a **deterministic** algorithm to recognize numbers from images is very difficult…

But we can collect easily many example images…



If We Could Design a Machine that Learns from Examples…



Learning algorithm

OCR machine

This is number seven

So, What Is Machine Learning?

*Machine learning is the technology that we use to solve a problem by **learning** the solution **through examples***

"The goal of machine learning is to build computer systems that automatically improve with experience"

Tom M. Mitchell, The discipline of Machine Learning, 2006

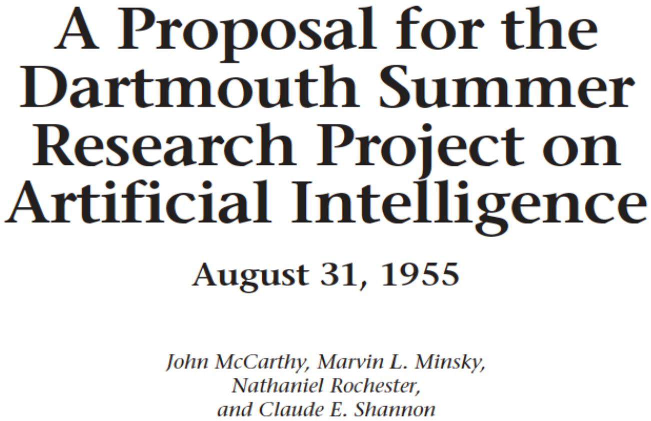
Take-Home Messages

1. Machine learning is very useful when **no algorithmic solution** is known. It also avoids a detailed algorithm to overfit known cases, reducing errors

2. When you are able to devise algorithmic solutions (*step after step through every possible corner case*) that work 100% of the time, **you should not use machine learning**!

When Did It Start?

It All Started In 1955

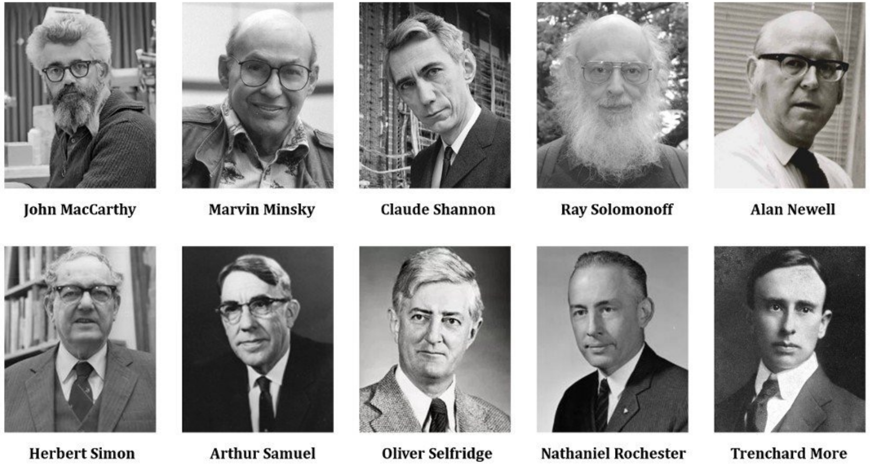


A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence

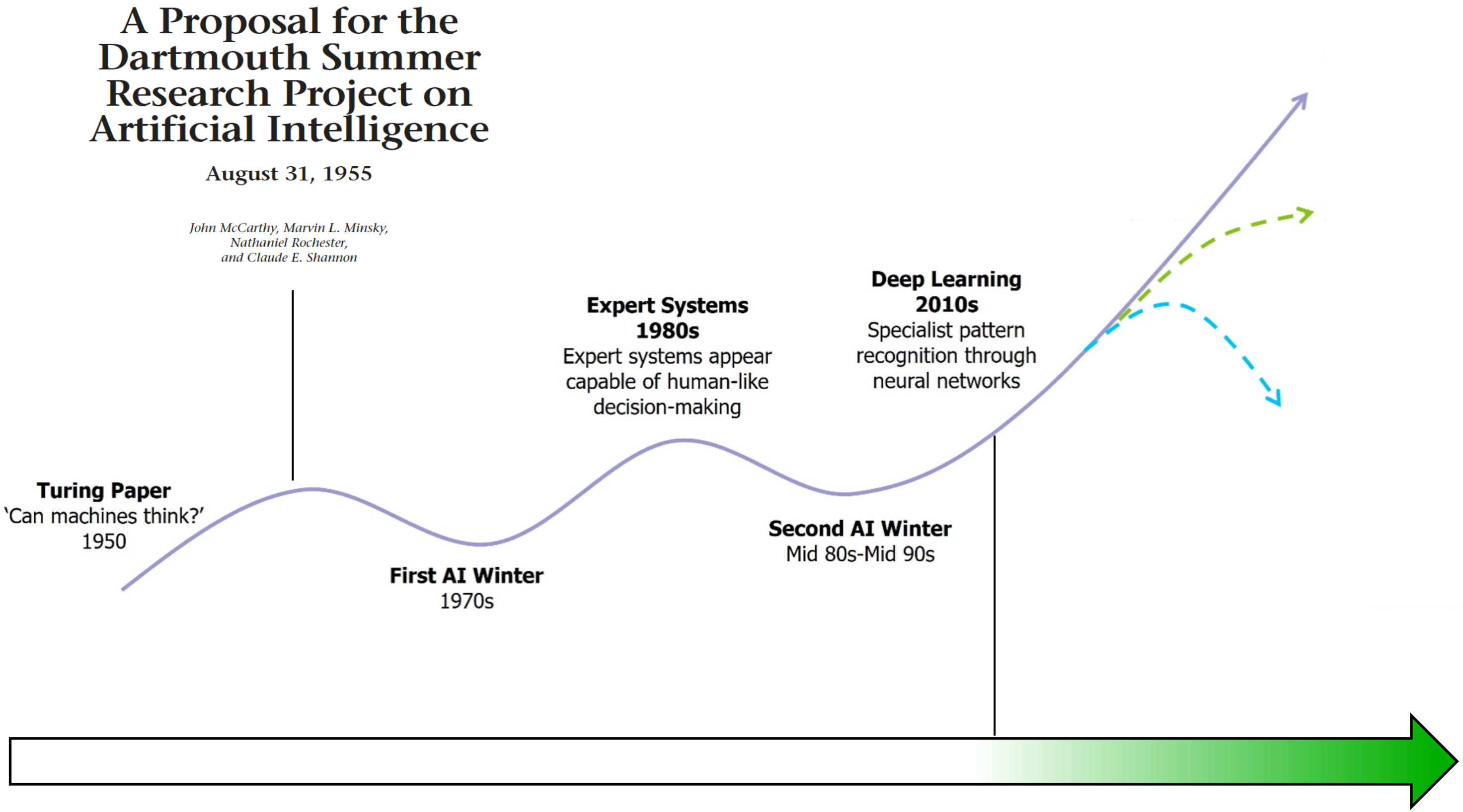
August 31, 1955

John McCarthy, Marvin L. Minsky,
Nathaniel Rochester,
and Claude E. Shannon

1956 Dartmouth Conference:
The Founding Fathers of AI



http://www.aaai.org/ojs/index.php/aimagazine/article/view/1904



A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence

August 31, 1955

John McCarthy, Marvin L. Minsky,
Nathaniel Rochester,
and Claude E. Shannon

Turning Paper
'Can machines think?'
1950

Expert Systems
1980s
Expert systems appear capable of human-like decision-making

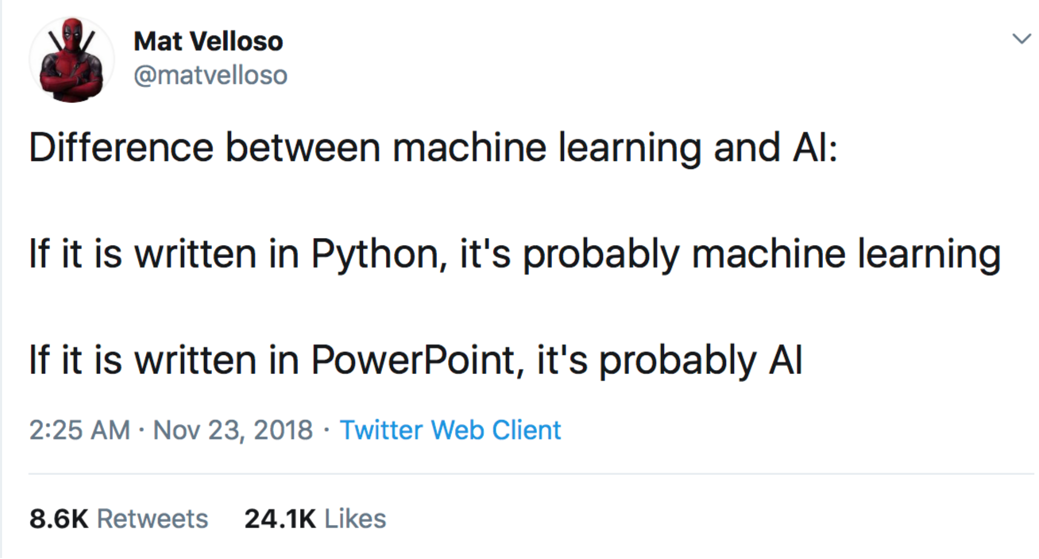
Deep Learning
2010s
Specialist pattern recognition through neural networks

First AI Winter
1970s

Second AI Winter
Mid 80s-Mid 90s

… from the idea of mimicking human reasoning to learning from examples (*machine/deep learning*)

But… What's the Difference between AI/ML?



Mat Velloso
@matvelloso

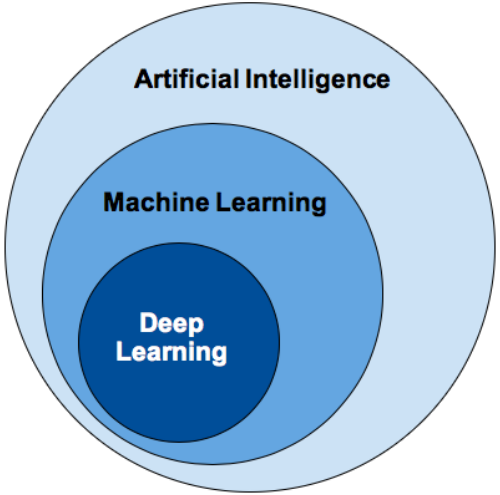
Difference between machine learning and AI:

If it is written in Python, it's probably machine learning

If it is written in PowerPoint, it's probably AI

2:25 AM · Nov 23, 2018 · Twitter Web Client

8.6K Retweets 24.1K Likes



Data-Driven AI/ML (1990-now)

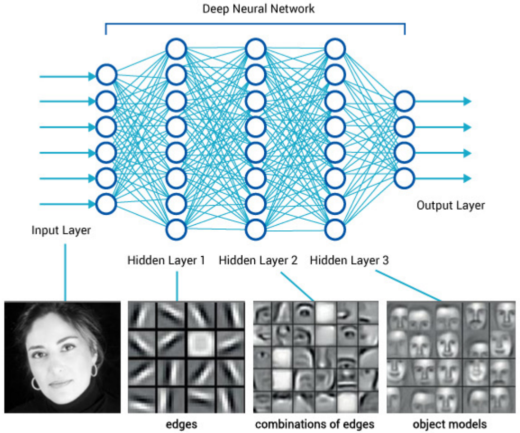
Big Data

Facebook 350 millions of images per day

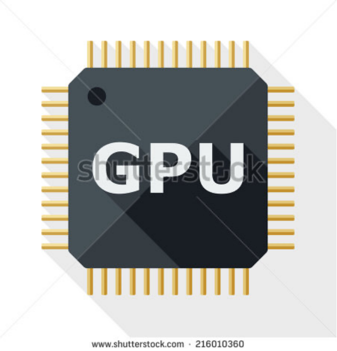
Walmart 2.5 Petabytes customer data hourly

YouTube 300 hours of videos per minute

Deep Learning



GPU

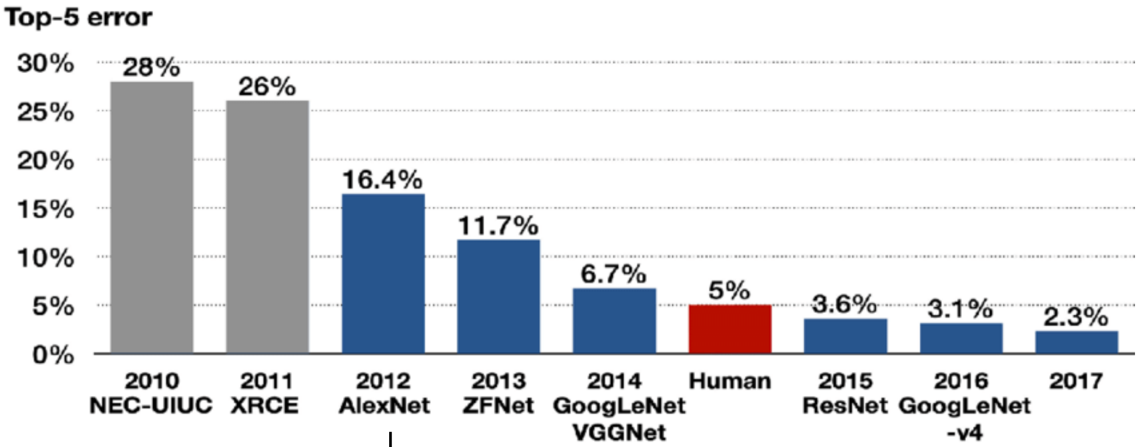


ImageNet Large Scale Visual Recognition Challenge

IMAGENET

1,2M training images
100K test images
1,000 classes

Fei-Fei Li



Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton

Artificial Intelligence and Machine Learning Today

AI is going to transform industry and business as electricity did about a century ago

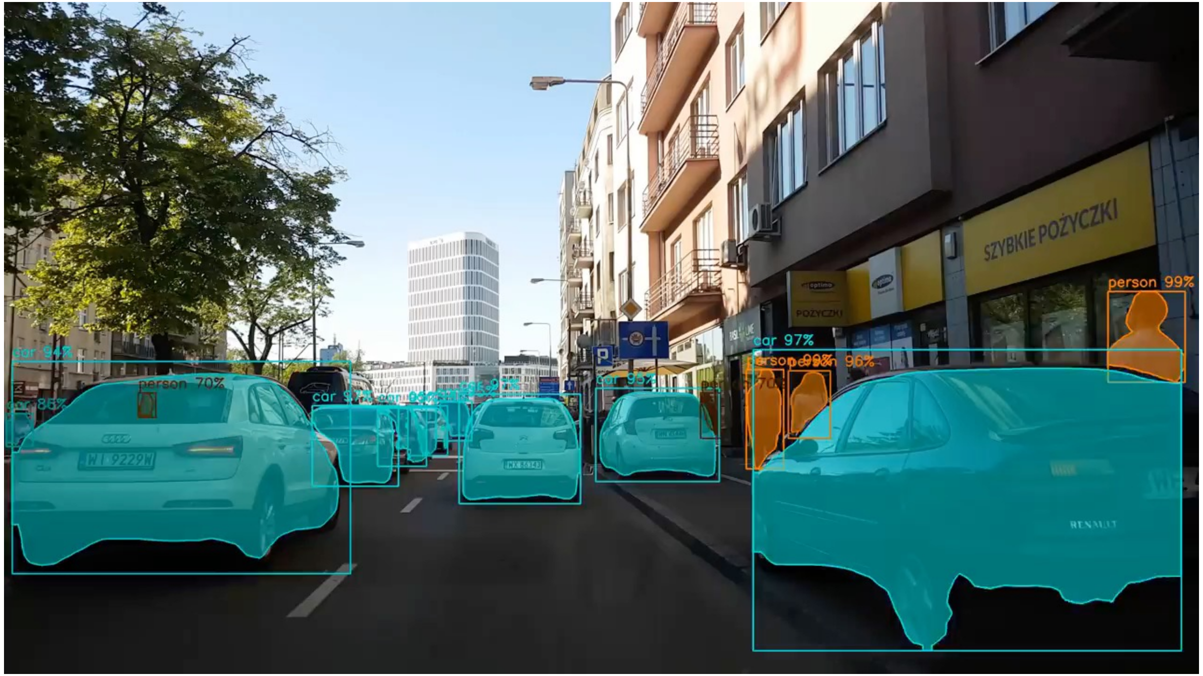
Andrew Ng, Jan. 2017

Applications:

- Cybersecurity
- Robotics
- Healthcare
- Speech recognition
- Virtual assistants
- …

Computer Vision for Self-Driving Cars

He et al., *Mask R-CNN*, ICCV ’17

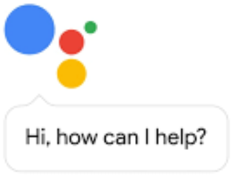


Speech Recognition for Virtual Assistants

Amazon Alexa

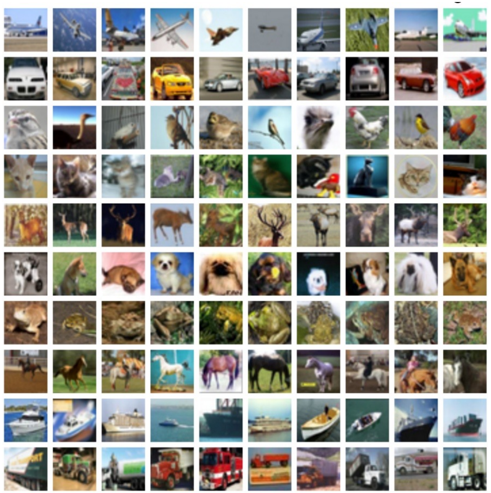
Apple Siri

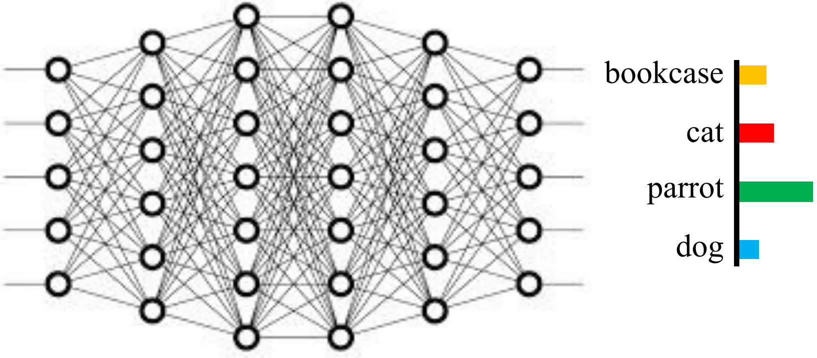
Microsoft Cortana



Google Assistant

Modern AI is Numerical Optimization + Big Data





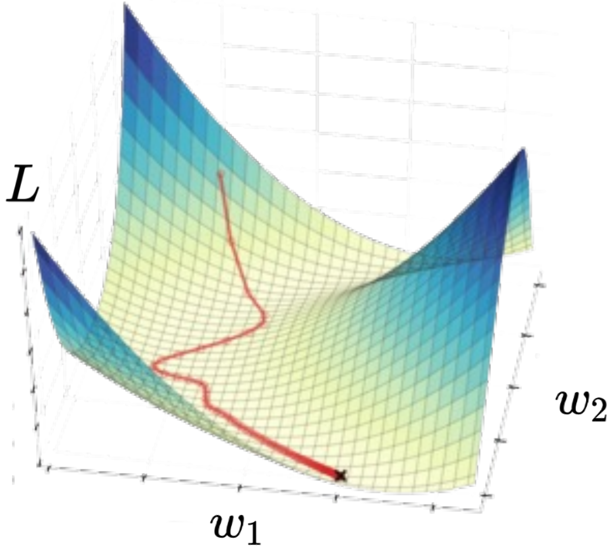
$$\min_{\boldsymbol{w}} L(D; \boldsymbol{w})$$

The goal is to minimize the fraction of *classification errors*

... by iteratively updating the classifier parameters $\boldsymbol{w}$ along the gradient direction $\nabla_{\boldsymbol{w}} L(D; \boldsymbol{w})$

The Workhorse of Machine Learning: *Gradient Descent*

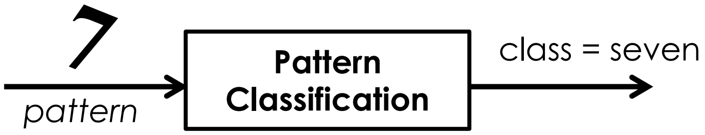
```
$\mathbf{w} \leftarrow \mathbf{w}_0$
$i \leftarrow 0$
while $i < maxiter$ do
    $\mathbf{w} \leftarrow \mathbf{w} - \eta \, \nabla_{\mathbf{w}} L(\mathbf{X}, \mathbf{y})$
    $i \leftarrow i + 1$
end while
return $\mathbf{w}$
```



Classification with Machine Learning

Pattern Recognition as a Classification Problem

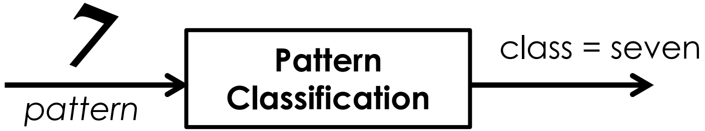
- **Pattern Classification:** assigning a “pattern” (input data) to a category/class



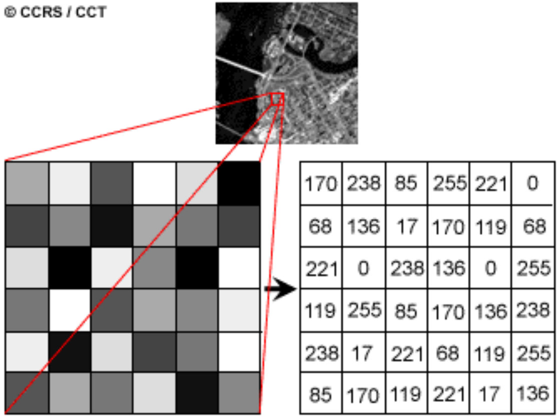
- In this picture, the pattern is the specific grouping of pixels that represent the number 7

Pattern Recognition as a Classification Problem

- Pattern classification is about assigning class labels to patterns



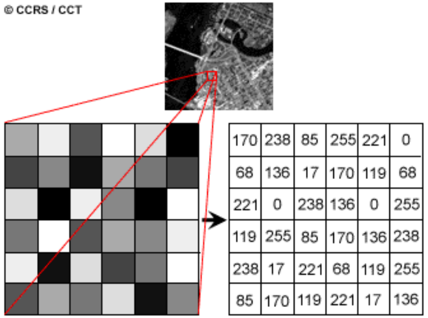
- Patterns are described by a set of measurements called *features* (or attributes)
  - For images, feature/input values could correspond to the brightness of each pixel



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Basic Concepts: Class and Features

- Each input sample is described by a feature vector with “d” elements: $\mathbf{x} = (x_1, x_2, \ldots, x_d)$.



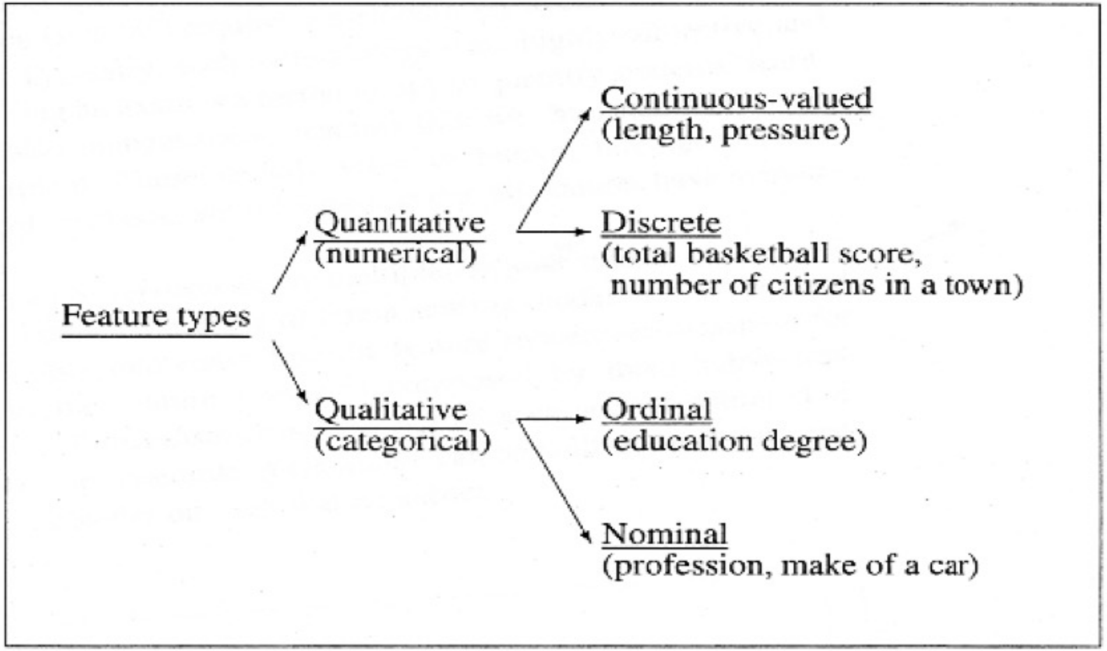
© CCRS / CCT

170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

$$\mathbf{x} = (x_1, x_2, \ldots, x_d) = (170, 238, 85 \ldots 136)$$

- **Class**: intuitively, a class contains similar patterns, whereas patterns from different classes are dissimilar (e.g., dogs and cars)
  - In this course, we assume that there are *c* possible classes, normally denoted with $y_1 \ldots y_c$
  - Each sample belongs to one of the “c” classes - We say that each input sample has a **class label**

Different Feature Types



Feature types

Quantitative (numerical)

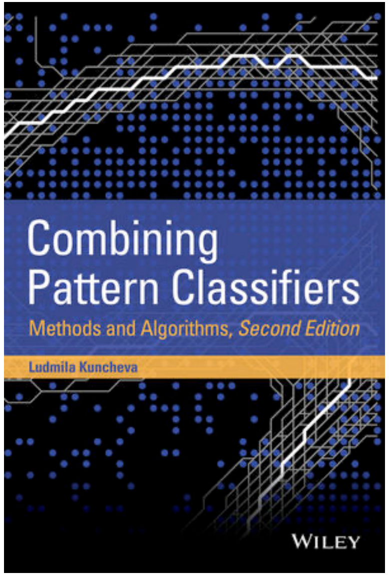
Continuous-valued (length, pressure)

Discrete (total basketball score, number of citizens in a town)

Qualitative (categorical)

Ordinal (education degree)

Nominal (profession, make of a car)


Combining Pattern Classifiers
Methods and Algorithms, Second Edition
Ludmila Kuncheva
WILEY

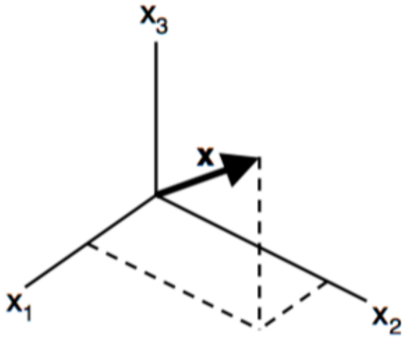
Ludmila Kuncheva, *Combining pattern classifiers*, Wiley, 2004

- **Statistical pattern classification** uses (mostly) *numerical* features

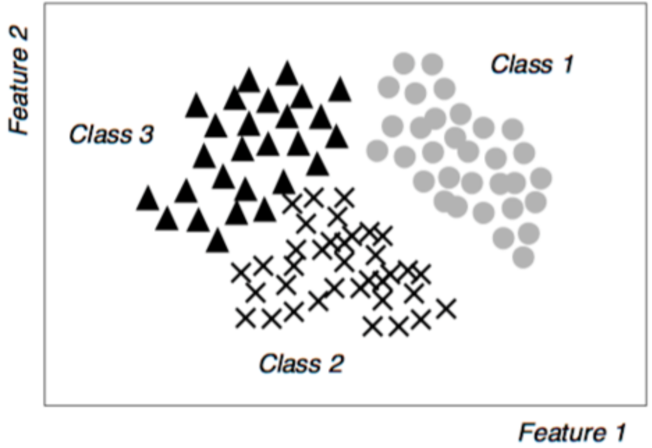
Basic Concepts: Feature Vector and Feature Space

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \\ x_d \end{bmatrix}$$

Feature vector



Feature space (3D)



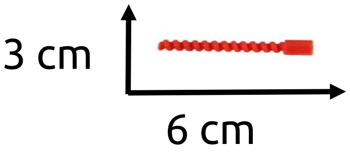
Scatter plot (2D)

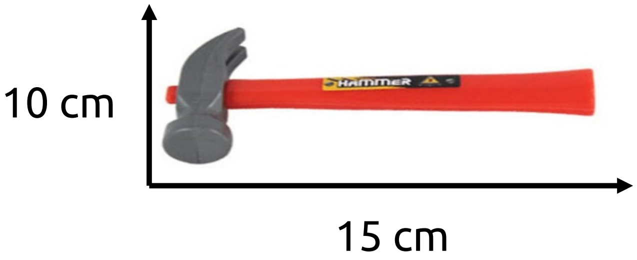
A Toy Example

Let’s assume we aim to build an ML model to discriminate between screws and hammers

Feature Extraction

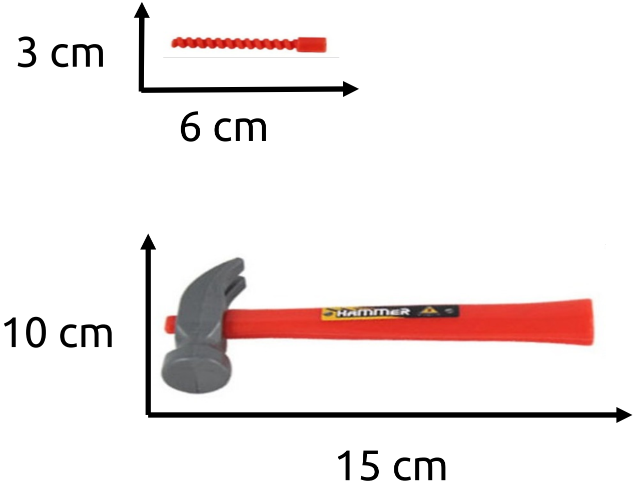
We consider height and width as our input features

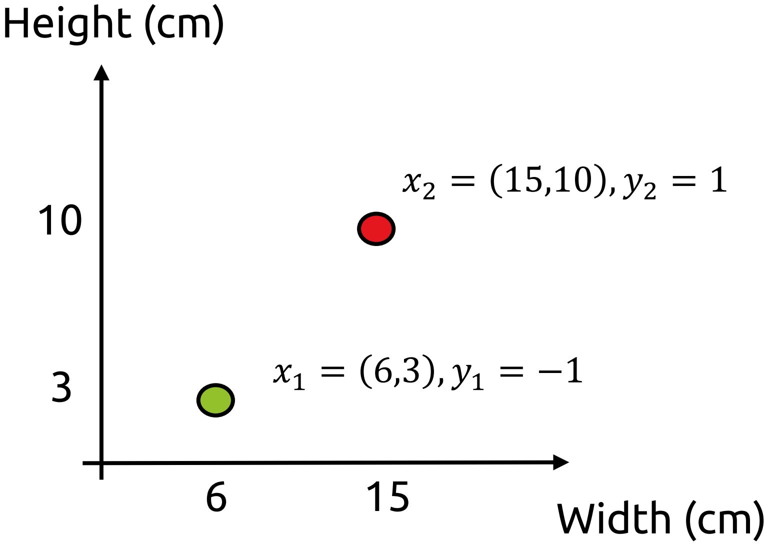




Feature Extraction

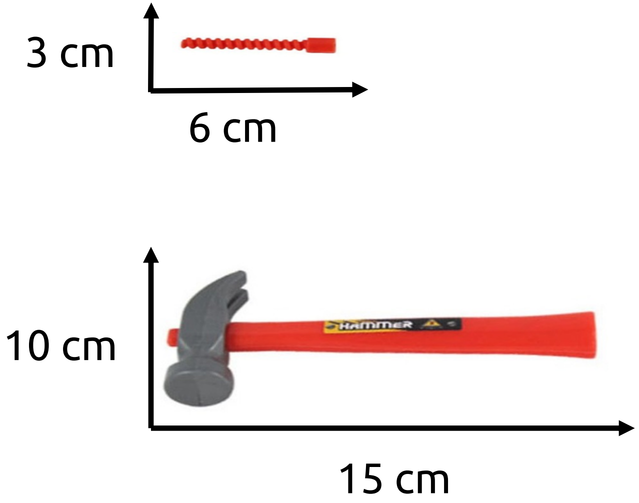
Now we can represent the two samples in the feature space

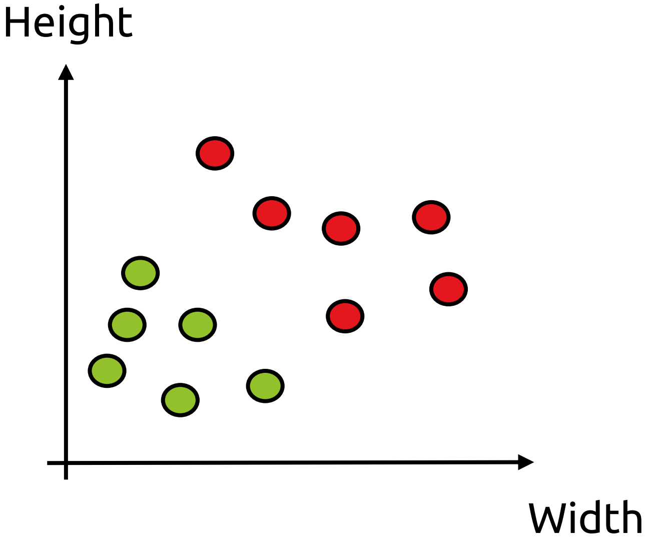




Feature Extraction

… and repeat for different screws and hammers…





Training Dataset

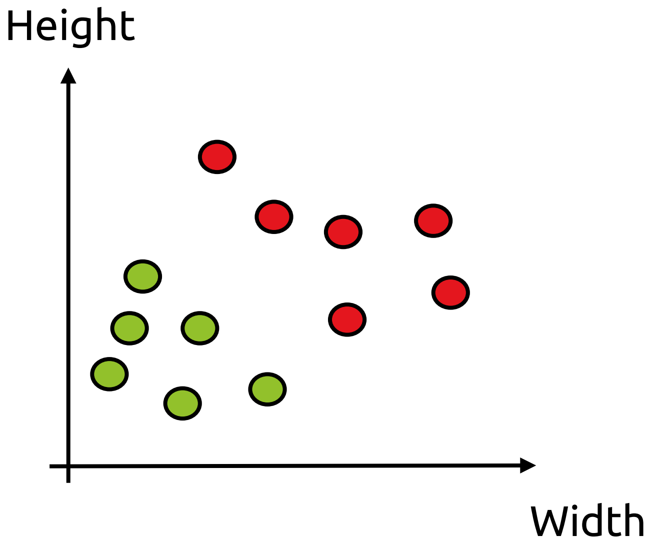
- The information to design a machine-learning model is usually in the form of a labeled data set (called training set): $\mathcal{D} = (\boldsymbol{x}_i, y_i)_{i=1}^{n}$

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nd} \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 \\ \cdots \\ y_n \end{bmatrix}$$

- In the previous example, $\mathcal{D}$ is the data set of screws and hammers that we collected…

Model Training

- Training a model amounts to finding a function that splits the training points according to their class label
- Can you find one?



Model Training

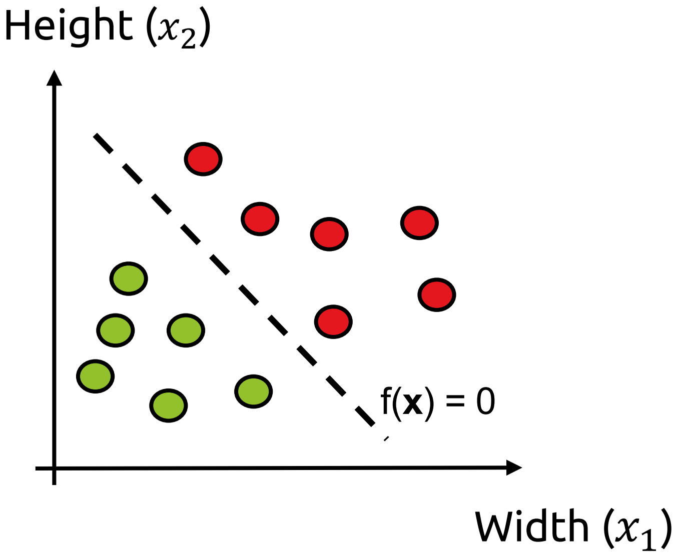
- Can you find one? Yes, of course…

- Let's pick $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = w_1x_1 + w_2x_2 + b$

- If $f(\boldsymbol{x}) \geq 0$ the model predicts «hammer» and «screw» otherwise

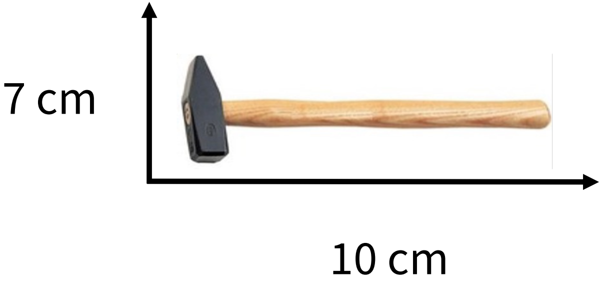
- Model training aims to estimate the model parameters $(\boldsymbol{w}, b)$ from the training data,
  - using either a probabilistic approach or solving an optimization problem

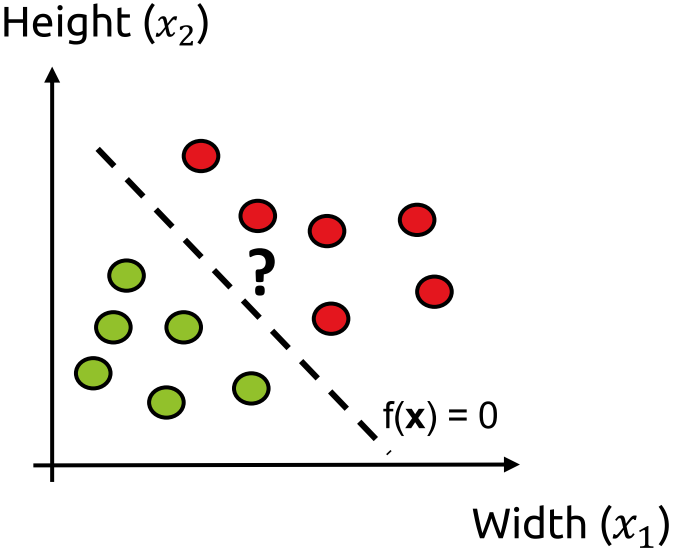
- This model makes zero errors on the training data. *Can we trust it?*



Model Evaluation

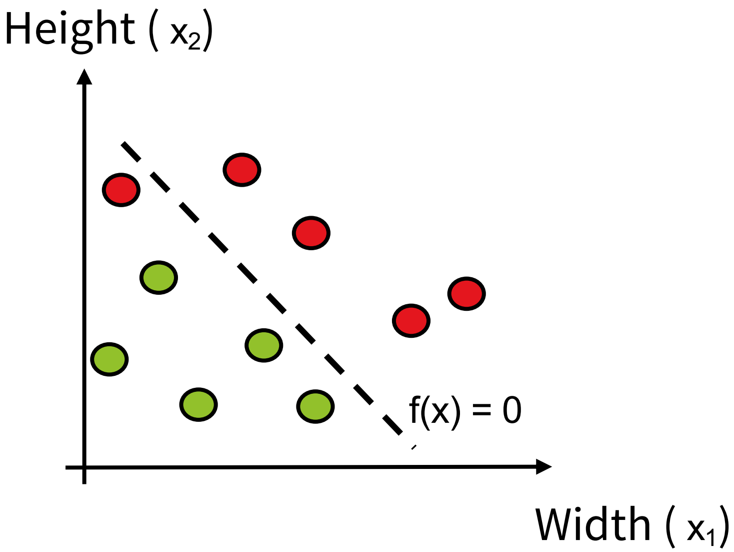
- Once the model is trained, it should be evaluated on never-before-seen input samples (a.k.a. *test* samples) to check if it can *generalize…*





Model Evaluation

- We evaluate our model on the test set, and report only 1 mistake out of 10
- The classification accuracy of this model is 90% (or its test error is 10%)



ML Model Design and Evaluation

- Let's recap the steps we just followed…
  - Data collection (and labeling)
  - Feature extraction
  - Training-test splits
  - Model Training
  - Model Evaluation

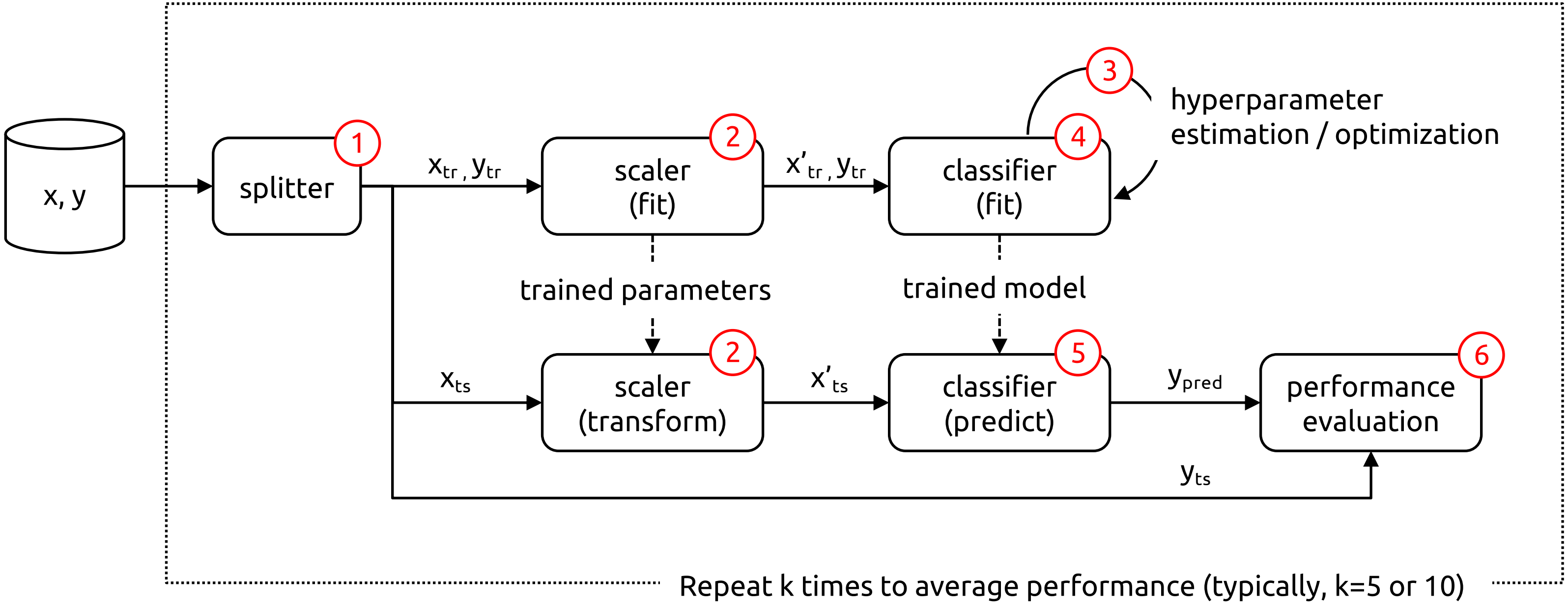
- Typically, feature values also need to be scaled (data normalization)

- **Important:** The underlying assumption is that the training and the test data are drawn from the same, unknown probability distribution $p(X, Y)$
  - This is known as the **stationarity** assumption – data is *independent* and *identically distributed* (iid)
  - Most of the ML algorithms/models are built under this assumption

Feature Scaling

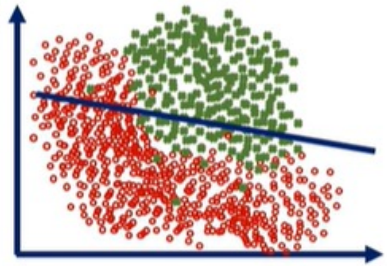
- Feature values are normally scaled within a bounded interval to facilitate model training
- **Min-max scaling:** $x' = \frac{x-m}{M-m}$, where x is the input feature, and
  - m and M are the min and max values of that feature over the whole training set
- **Z-score scaling:** $x' = \frac{x-\mu}{\sigma}$, where x is the input feature, and
  - μ and σ are the mean and standard deviation of that feature estimated from the training set

ML Model Design

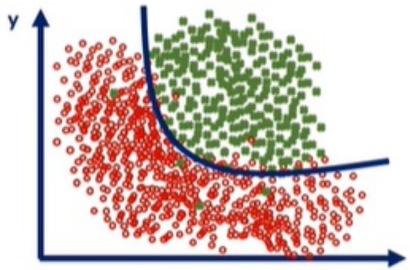


Model Training: Underfitting vs. Overfitting

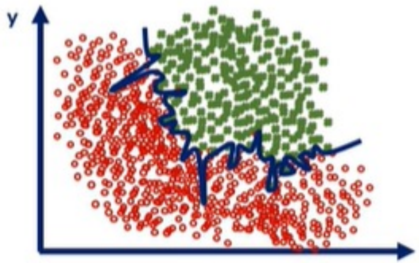
- Why do we need testing on a separate data split?



Underfitting
- high training error
- high test error



Good fit
- low training error
- low test error



Overfitting
- low training error
- high test error

Generalization Error - Overfitting

- The best values of the model's parameters are learned by minimizing the loss incurred on a ***training set*** consisting of some number of *examples* collected for training

- However, doing well on the training data does not guarantee that we will do well on (**unseen**) **test** data

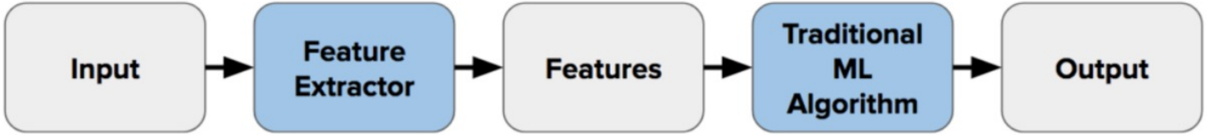
- So we split the available data into two partitions: the training data (for fitting model parameters) and the test data (which is held out for evaluation), and then measure:
  - **Training Error -** The error on that data on which the model was trained
  - **Test Error -** This is the error incurred on an **unseen** test set (**generalization error**). This can deviate significantly from the training error. When a model performs well on the training data but fails to generalize to unseen data, we say that it is ***overfitting***

https://d2l.ai, Chapter 1

What Is the Main Idea Behind *Deep Learning?*

Hand-crafted vs. Non-handcrafted (Learned) Features

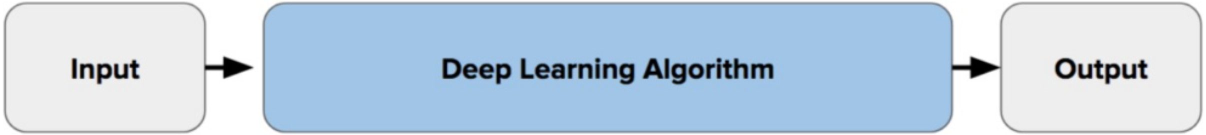
- **Handcrafted** features are manually engineered by the human designer



Input → Feature Extractor → Features → Traditional ML Algorithm → Output

Processing flow for extraction of **handcrafted** features

- Today, we can extract **non-handcrafted** features that are automatically learned from a machine learning algorithm

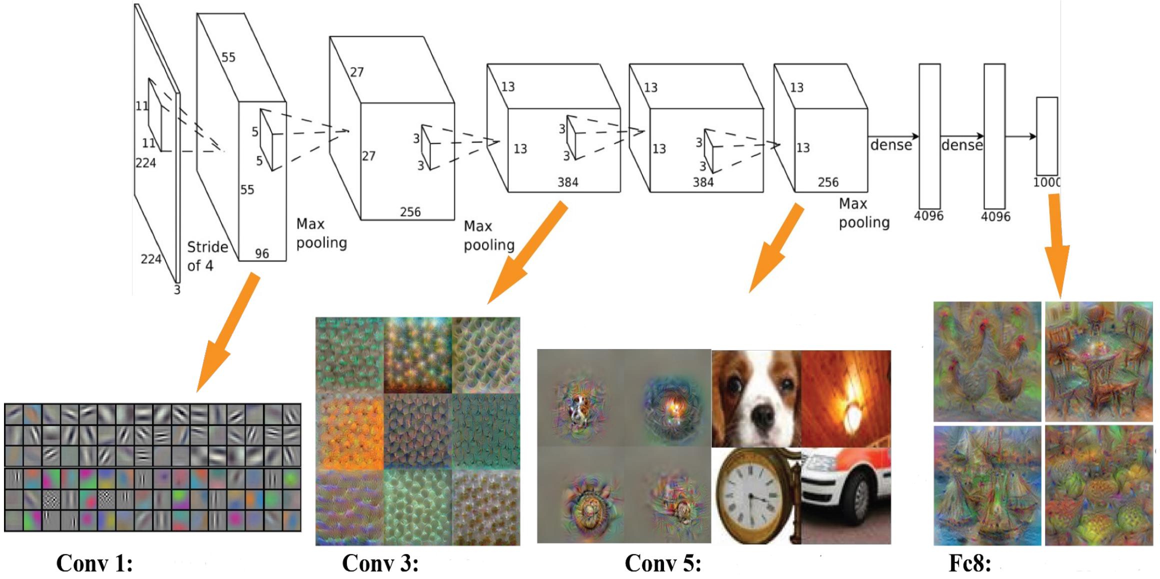


Input → Deep Learning Algorithm → Output

Processing flow for learning **non-handcrafted** features («learned» features)

Learning Non-handcrafted Features

- **Non-handcrafted** features can be automatically learned with **deep neural networks** (we will see more later)



224
224
3
11
11
Stride of 4
55
55
96
5
5
Max pooling
27
27
256
3
3
Max pooling
13
13
384
3
3
13
13
384
3
3
13
13
256
Max pooling
dense
dense
4096
4096
1000
Conv 1:
Conv 3:
Conv 5:
Fc8: