Introduction

Practicalities, overview, text and image data.

Pekka Marttinen

Aalto University

Outline of this lecture

- Practicalities
- Overview of machine learning
- Basic concepts
- Text data
- Image data

Welcome

- Lecturers: Pekka Marttinen, Dariush Salami, Stephan Sigg
- Course home page in MyCourses:
  - https://mycourses.aalto.fi/course/view.php?id=37071
- Schedule:
  - See *MachLearn_student_timetable.pdf* in myCourses

Course structure

- Lectures $\times 10$
- Assignments, 5 sets
  - Programming exercises and mathematical derivations
- Project
  - Solving a machine learning problem
  - Writing a report
  - Can be done in pairs

Grading

- Assignments: 50%
- Project: 47%, including:
  - Project report, returned in two stages
  - Peer-grading
  - Details in MyCourses
- Giving course feedback: 3%
- Total: 100%
  - Preliminary grade boundaries: 1:50%, 2:60%, 3:70%, 4:80%, 5:90%.
- No exam.

Workload

- Lectures: 10×(2h+3h)=50h, including
  - Preparing for lectures
  - Revising after the lectures
  - Reading the book
- Assignments: 5×(2h+7h)=45h
- Project work: 30h
- Peer-grading: 8h
- Total: 133h, 133/26.7=5.0cr

Implementation (1/2)

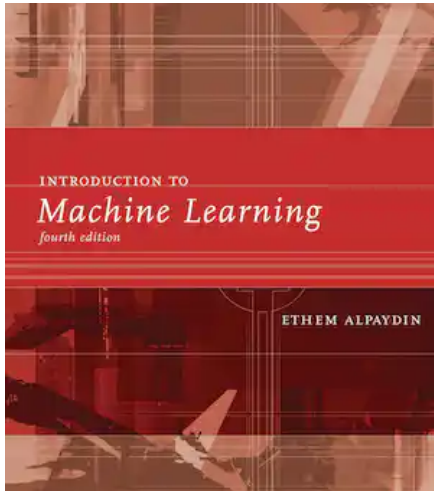
- Lectures:
  - In-person, check the location in MyCourses
  - Streaming and recordings are made available in MyCourses
- Assignment sessions:
  - 5 TA sessions weekly, 1-2 online and the rest in-person
  - Attend these for personal assistance
  - Pre-registration in MyCourses
  - Answers to previous week's assignments
  - Hints for solving the current week's assignments
- Assigments are implemented using JupyterHub and MyCourses
  - See more information in the respective MyCourses page

Implementation (2/2)

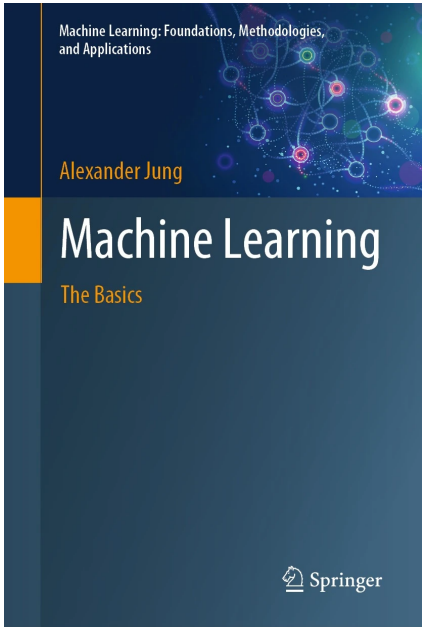
- Project
  - One weekly TA session dedicated for project help (see dates in MyCourses)
  - See details about the project and peer-grading in MyCourses
- Slack
  - You can post questions in the respective channels in Slack
  - One channel for each assignment, each lecture, and the project.
  - Students are welcome to reply to each other's questions: Be polite and constructive and don't reveal the whole answer
  - TAs/Lecturers will also answer if something remains unclear
  - Link to join the Slack is available in MyCourses for all registered students

Material

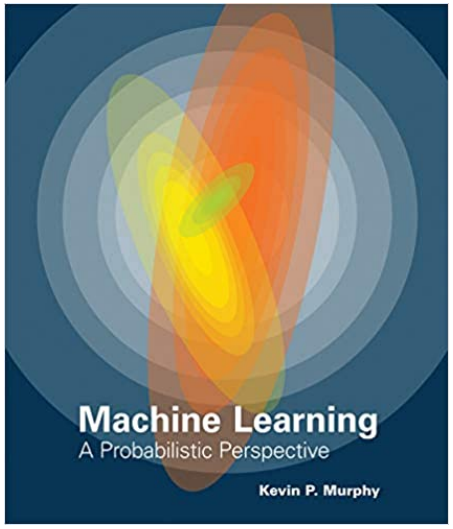
- The course does not follow any single book.
- Recommended reading material is communicated in the end of the slides of each lecture.
- Useful books that cover many lectures (but not all):



Alpaydin: Introduction to Machine Learning



Jung: Machine Learning: The Basics



Murphy: Machine Learning: A Probabilistic Perspective

Age of "Big data"

- 30M human genomes sequenced, each 6400M letters[1]
- Spotify has 70M tracks, 60k new tracks uploaded every day[2]
- Youtube has 800M videos[3]
- Estimated 63k Google search queries every second![4]

Image: Pete Linforth on Pixabay

[1] https://www.science.org/content/podcast/looking-back-20-years-human-genome-sequencing

[2] https://www.hypebot.com/hypebot/2021/02/60000-tracks-are-uploaded-to-spotify-every-day.html

[3] https://earthweb.com/how-many-videos-are-on-youtube/

[4] https://blog.hubspot.com/marketing/google-search-statistics

Machine learning

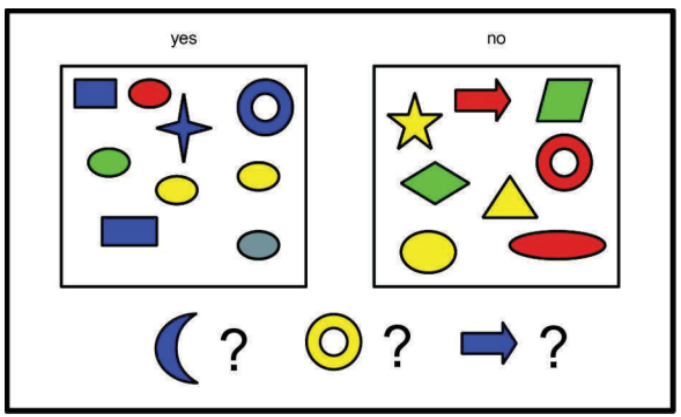
- Murphy: "A set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty"
- Alpaydin:
  - To solve a problem on computer, we need an algorithm
  - For some tasks we don't have an algorithm, e.g. detect spam email.
  - But we have lots of examples (data) of emails that are spam or not spam.
  - Use the data to automatically learn the algorithm for this task.
  - By detecting patterns and regularities in data that help us understand the process or make predictions.

Three types of machine learning

- **Supervised learning**
  - Given data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, learn to predict the **output** y_i using **input** $\mathbf{x}_i$.
  - elements of x_i are called **features**, **attributes**, or **covariates**.
  - y_i is called the **response** or **label**.
- **Unsupervised learning**
  - Given only inputs $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ (**unlabeled** data) the goal is to find some 'interesting' patterns in the data.
- **Reinforcement learning**
  - Broadly: training an agent to take actions that maximize some reward in the long term. For example: training a robot to escape a maze.

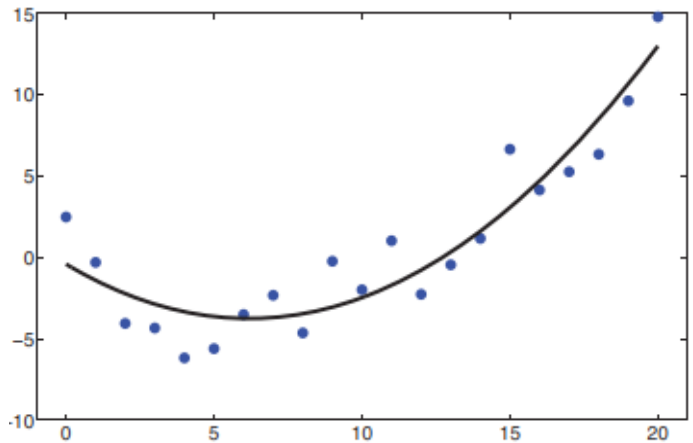
Supervised learning

Classification (y_i binary or categorical)



Murphy, Fig 1.1

Regression (y_i real-valued)



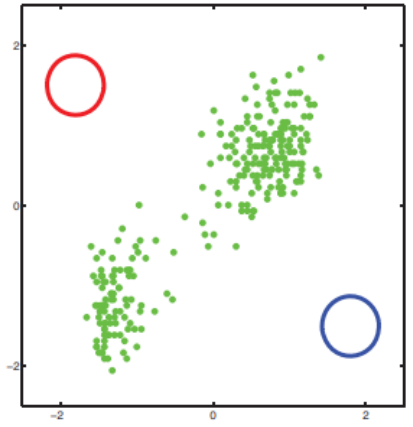
Murphy, Fig. 1.7

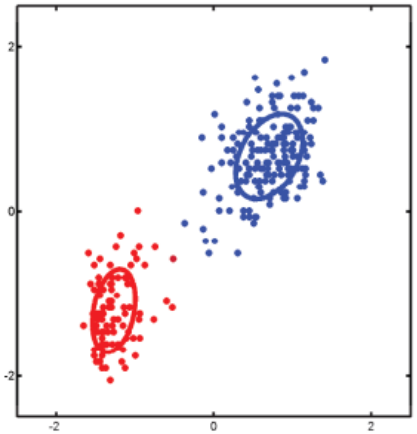
Unsupervised learning: Clustering

Old Faithful geyser

Wikipedia

Waiting time vs. eruption duration





Murphy, Fig. 11.11

Unsupervised learning: Dimensionality reduction

- Explain high-dimensional data in a lower-dimensional subspace which captures the "essential" parts of the original data

(a)

(b)

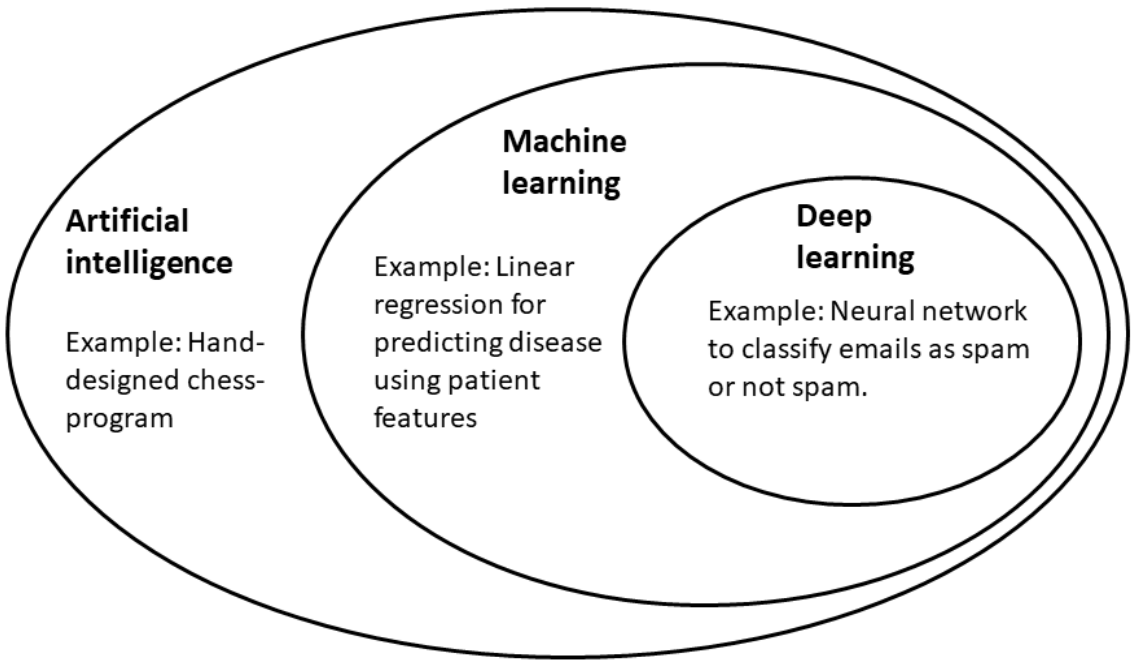
Murphy, Fig 1.10

Artificial Intelligence

- Artificial intelligence (AI) system interacts with its environment to solve some task or achieve a goal.
- Applications
  - web search engines
  - recommendation systems (Youtube, Amazon,...)
  - understanding human speech
  - self-driving cars
  - automated decision making
  - strategic games (chess, Go)
  - spam filtering
- **Challenge in AI**: solving a task in real-world may require thorough knowledge that is subjective and intuitive and difficult to articulate in any formal way.

Artificial Intelligence vs. machine learning

- Machine learning
  - Gives an AI system ability to automatically acquire their own knowledge, by extracting meaningful patterns from raw data.
  - Allows tackling problems involving knowledge of the real world.
- *Machine learning is the only viable approach to building AI systems that can operate in complicated real-world environments* -Goodfellow et al., Deep Learning



Outline of the course

- Introduction
- Regression (SL)
- Classification (SL)
- Feature learning, visualization (UL)
- Nonparametric methods (SL)
- Deep learning (SL)
- Clustering (UL)
- Probability theory
- Reinforcement learning 1
- Reinforcement learning 2

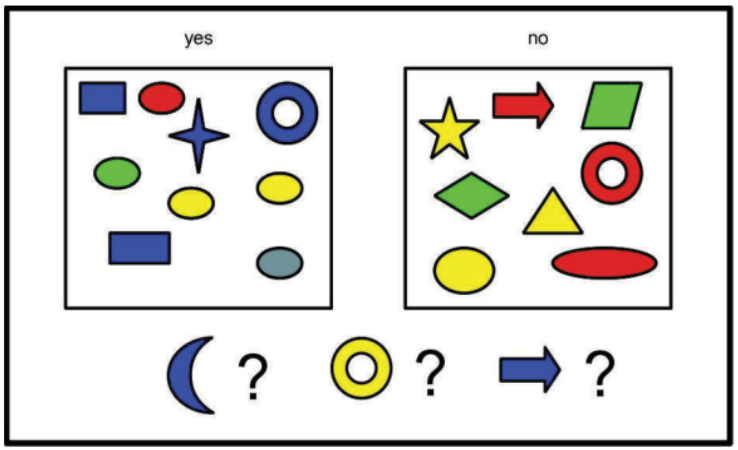
SL: Supervised learning
UL: Unsupervised learning

Data matrix (1/3)

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1d} \\ x_{21} & x_{22} & & & \\ x_{31} & & \ddots & & \\ \vdots & & & & \\ x_{n1} & & & & x_{nd} \end{bmatrix}$$

- Usually (but not always!) rows are individual data points; columns are characteristics of the data points (e.g. measurement, feature, attribute, variable, covariate)

Data matrix (2/3)



(a)

D features (attributes)

N cases

Color	Shape	Size (cm)	Label
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	0

(b)

Murphy Fig. 1.1

Data matrix (3/3)

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1d} \\ x_{21} & x_{22} & & & \\ x_{31} & & \ddots & & \\ \vdots & & & & \\ x_{n1} & & & & x_{nd} \end{bmatrix}$$

- Sometimes data are immediately available as a matrix.
- For example:
  - Rows: students (n in total)
  - Columns: grades in different exams (d exams in total)

Feature vector

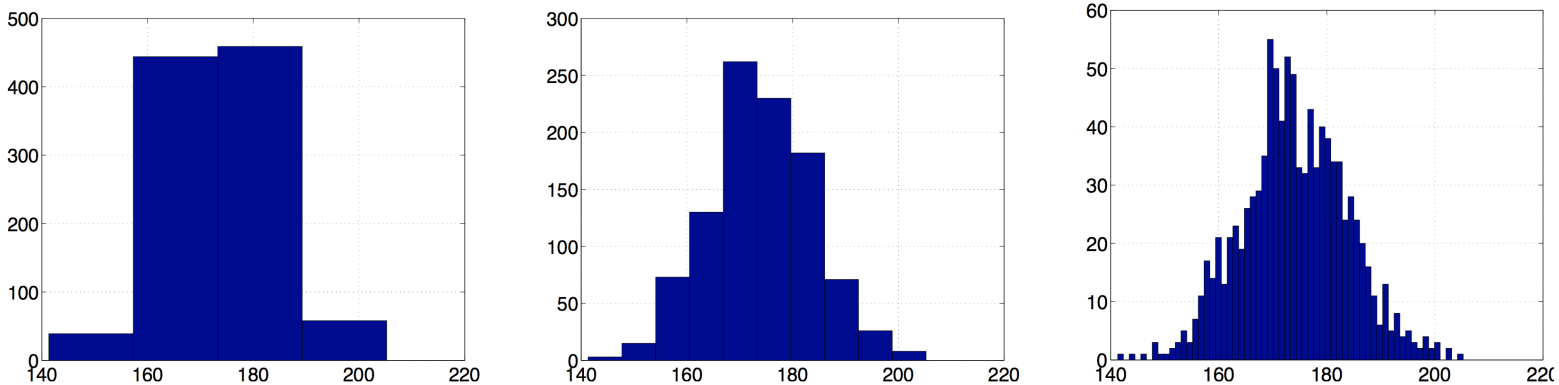
- One row of the data matrix, written as a column vector, represents one data point $\mathbf{x}_i$.

$$\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^\mathsf{T}$$

- $\mathbf{x}_i$ is called e.g. a **feature vector** or a **representation**.
- Everything can not be represented as vectors but a lot of things can.
- To emphasize that $\mathbf{x}_i$ is a vector, it is often written in bold.
- We assume the following operations to be familiar: vector addition $\mathbf{x} + \mathbf{y}$, vector inner product $\mathbf{x}^\mathsf{T}\mathbf{y}$, vector length (norm) $\|\mathbf{x}\|$, matrix multiplication $A\mathbf{x}$, matrix addition $A + B$, matrix product AB, matrix eigenvalues and eigenvectors $A\mathbf{x} = \lambda\mathbf{x}$.

Histogram (1/2)

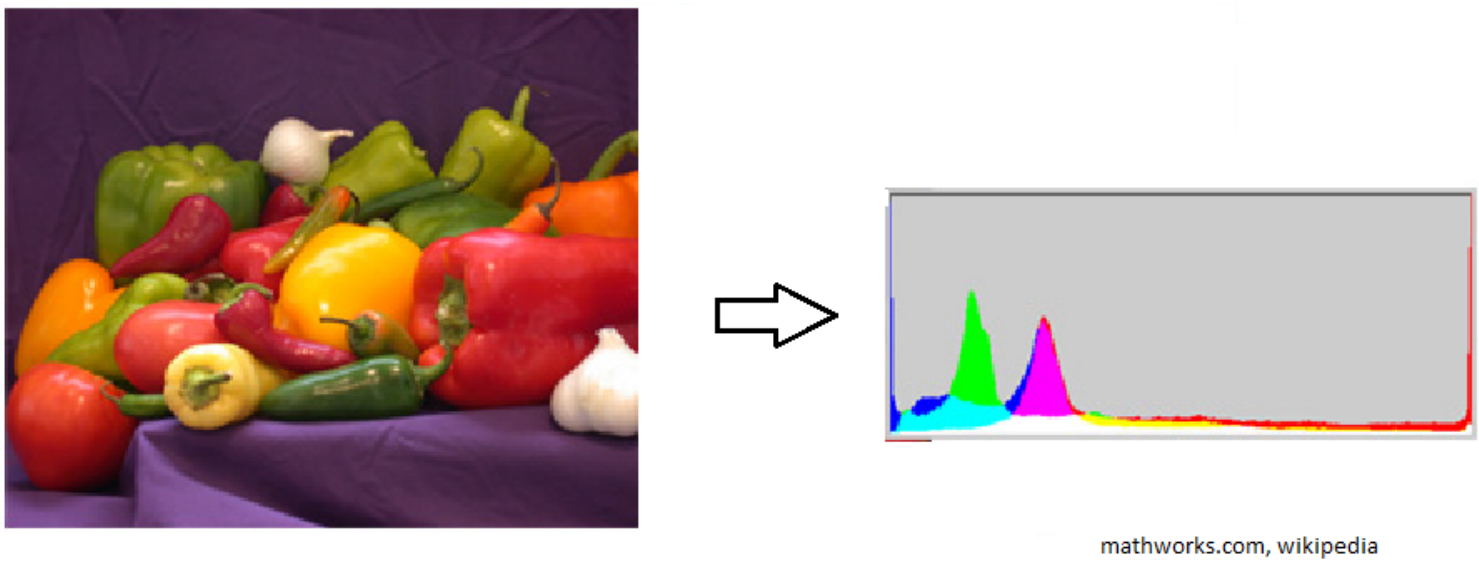
- Histogram is a common way to create features and explore data, by dividing values of a variable in different bins
- Height of a box in a histogran is equal to the number of values in the corresponding bin.



Random numbers $N = 1000$ drawn from the Normal distribution.
The same data represented with 4, 10, and 60 bins.

Histogram (2/2)

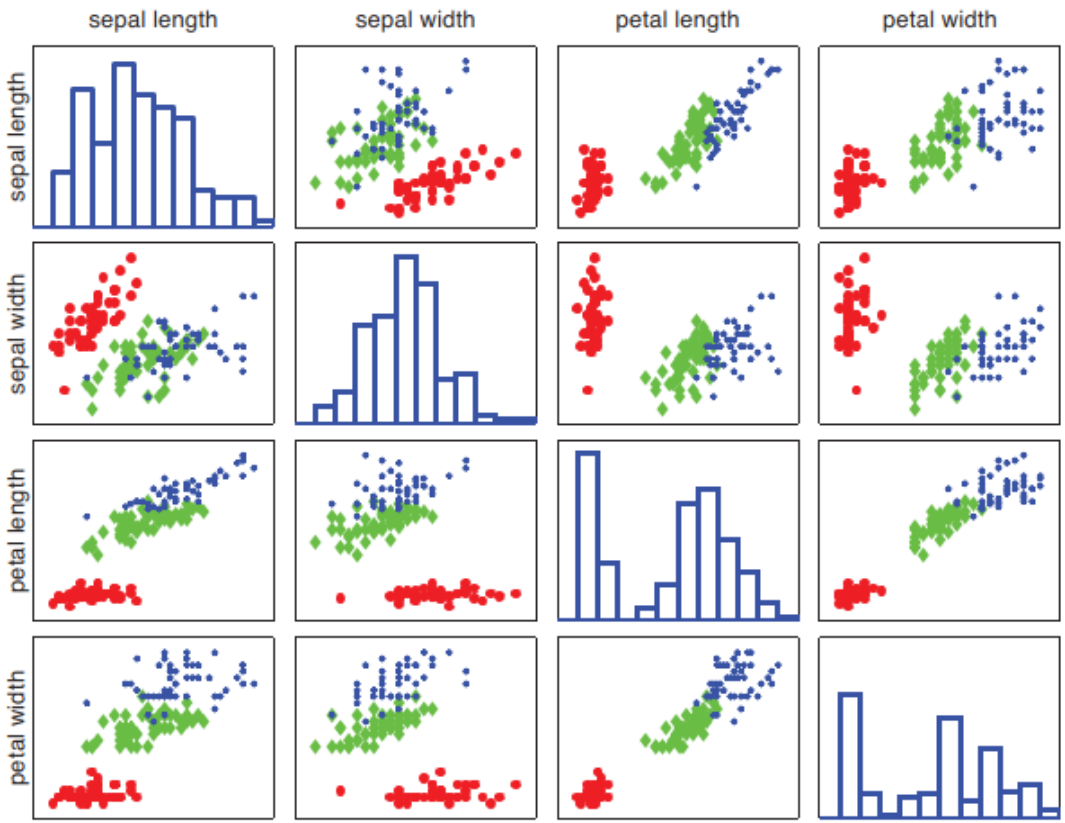
- A color histogram of an image shows the number of pixels in each color
- Histograms can be used also for text and audio



Left: Original image. Right: x-axis shows the RGB values (red-green-blue), y-axis is the number of pixels.

Scatter plot

- **A scatter plot** shows the **joint distribution** of two variables, by plotting the value of one variable against another
- Example: Three types of iris flowers



Murphy, Fig 1.4

Murphy, Fig. 1.3

Text as a vector

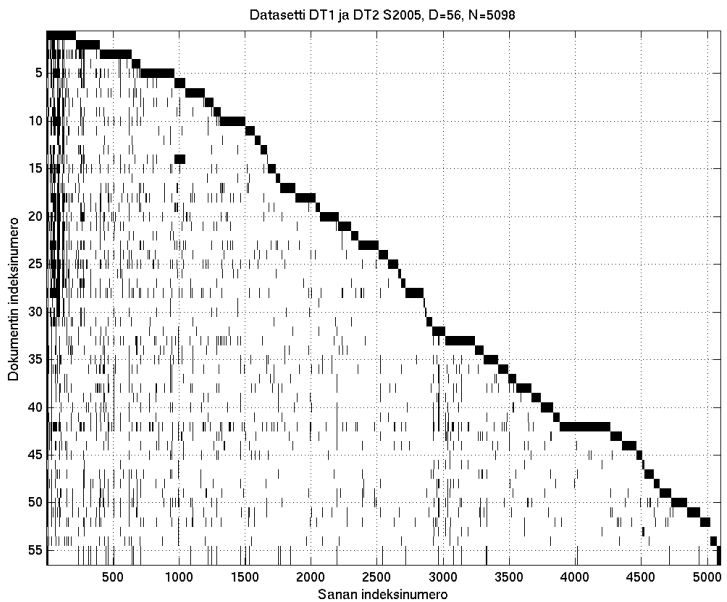
- A **term-document matrix**:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \cdots & x_{1d} \\ x_{21} & x_{22} & & & \\ x_{31} & & \ddots & & \\ \vdots & & & & \\ x_{n1} & & & & x_{nd} \end{bmatrix}$$

- Rows correspond to text documents, columns corresponds to words, and x_{ij} is the number of occurrences of word j in document i.
- **Bag of words** feature vector: $x_{ij} = 1$ iff document i contains word j

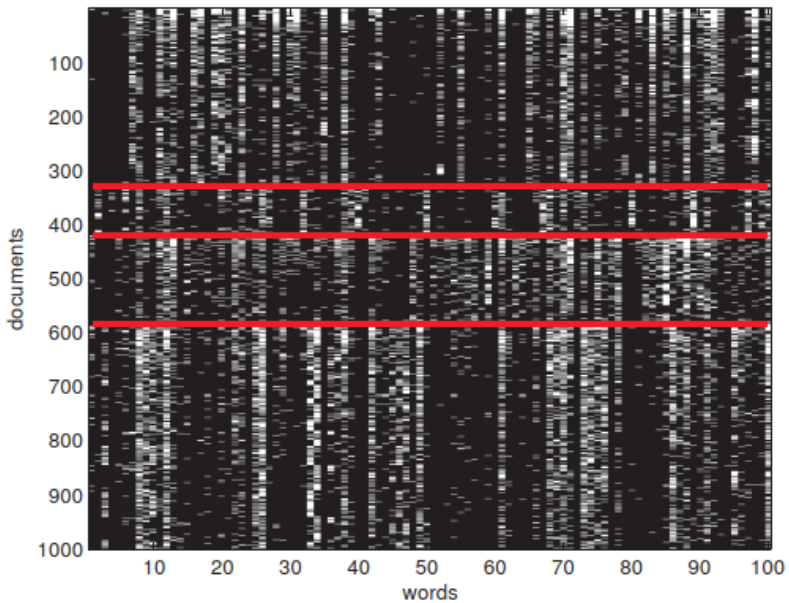
Example: student assignments

- Example: A term-document matrix. Word frequency ≥ 0 is represented with a black dot.
- **Left-ordered form**: columns are ordered according to appearance of words in documents: the first document has more than 200 words, the second document has also roughly 200 words of which 150 did not appear in the first, etc.
- Note: the data include a document which is almost identical to another document. Which one?



Example: news articles

- Bag of words feature vectors for articles in different news groups
  - We see words that are indicative of the class.



Murphy, Fig 1.2

tf-idf text representation

- The most common words (and, so, ...) can be removed manually.
- The words (terms) are often weighted according to their importance using the *inverse document frequency*

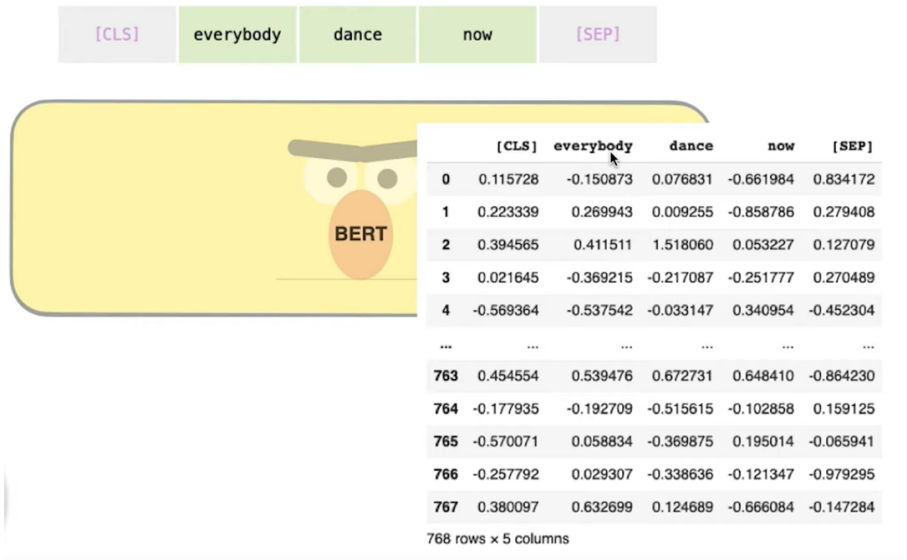
$$w(word) = \log(N / df(word)),$$

where $w(word)$ is the weight, $df(word)$ is the document frequency (how many documents have the word) and N is the number of documents.

- Each word frequency is multiplied by its corresponding weight w .
- For example, the word 'and' which likely appears in every document gets value 0.
- *tf-idf* representation (term frequency - inverse document frequency).

Neural network -based text representations

- Nowadays, several large neural network models, pre-trained with massive data sets, have been published for Natural Language Processing (NLP).
- For example, **BERT** learns a vector representation for a word, sentence, or the whole document.



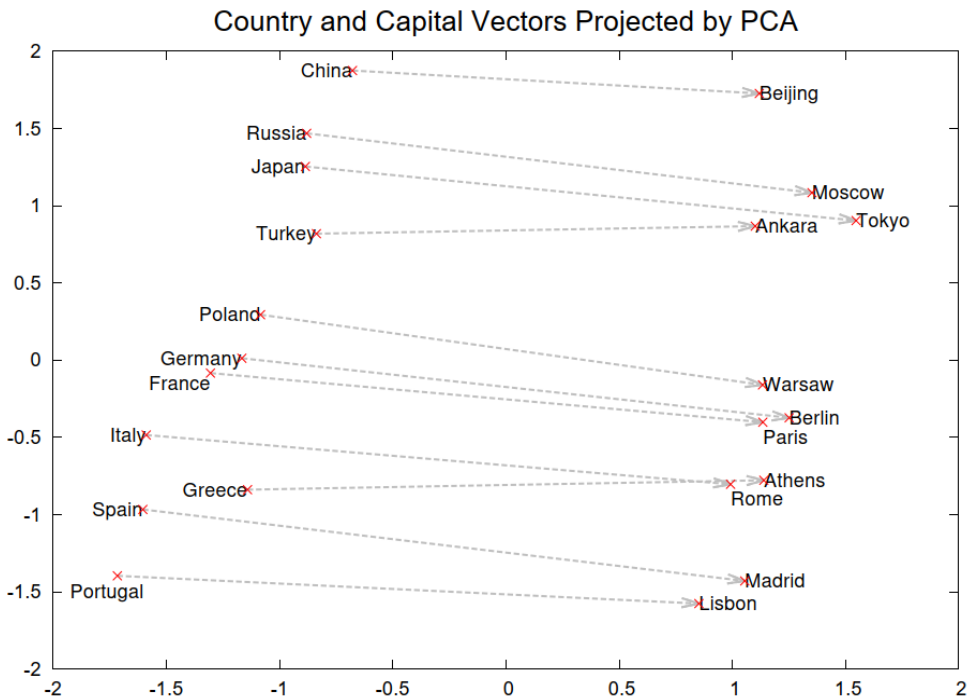
	[CLS]	everybody	dance	now	[SEP]
0	0.115728	-0.150873	0.076831	-0.661984	0.834172
1	0.223339	0.269943	0.009255	-0.858786	0.279408
2	0.394565	0.411511	1.518060	0.053227	0.127079
3	0.021645	-0.369215	-0.217087	-0.251777	0.270489
4	-0.569364	-0.537542	-0.033147	0.340954	-0.452304
...	...	...	...	...	...
763	0.454554	0.539476	0.672731	0.648410	-0.864230
764	-0.177935	-0.192709	-0.515615	-0.102858	0.159125
765	-0.570071	0.058834	-0.369875	0.195014	-0.065941
766	-0.257792	0.029307	-0.338636	-0.121347	-0.979295
767	0.380097	0.632699	0.124689	-0.666084	-0.147284

768 rows × 5 columns

https://www.youtube.com/watch?v=ioGry-89gqE&t=3s and
https://jalammar.github.io/illustrated-transformer/

Example: Countries and capitals

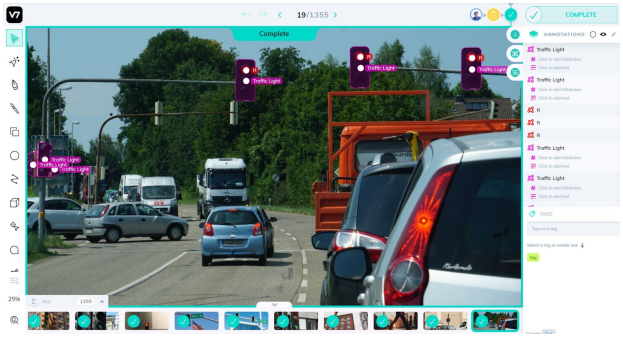
- 2D PCA projection of 1000D word vectors learned using **skip-gram** neural network model
- For example: Madrid - Spain + France = Paris

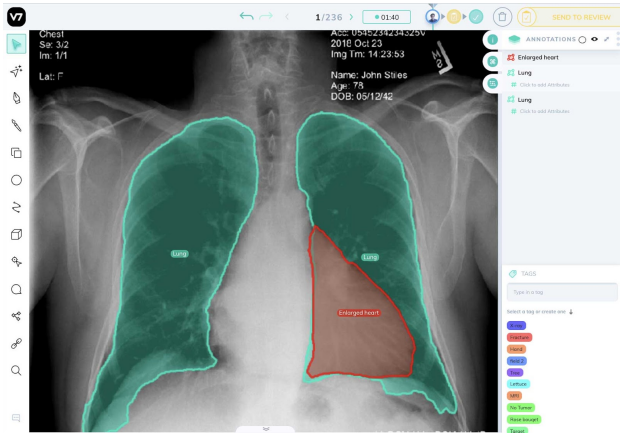


Mikolov et al. (2013). NeurIPS

Computer vision applications

- **Transportation**: self-driving cars, traffic monitoring,...
- **Healthcare**: X-Ray analysis,Digital pathology,...
- **Manufacturing**: defect detection, barcode reading,...
- **Construction**
- **Agriculture**
- **Retail**
- And many more[a]

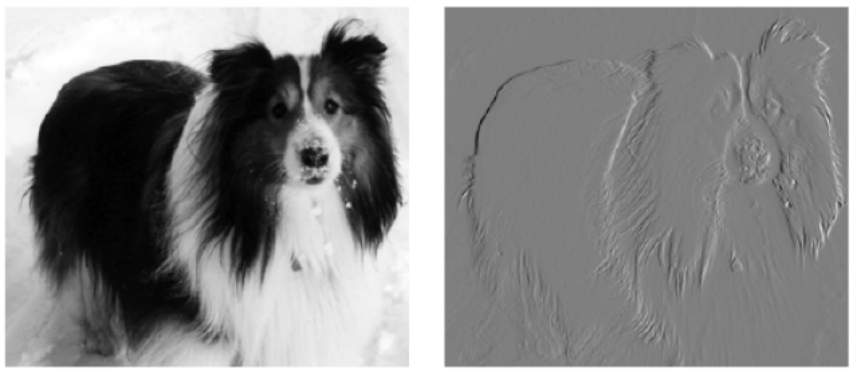




[a] https://www.v7labs.com/blog/computer-vision-applications

A simple example: edge detection

- **Left**: original image
- **Right**: the result after subtracting the intensity of the neighboring pixel on the left from each pixel.
  - The method identifies vertical edges in the image.
  - The result is called a *feature map*.



Goodfellow, Fig 9.6

Local features for images

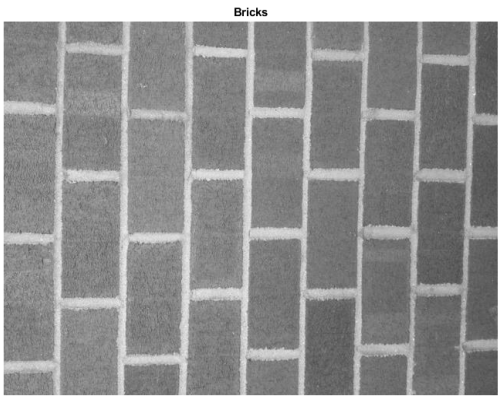
- The edge detection was an example of a *local feature detection* method.
  - Combines information from multiple pixels in a local neighborhood to define features.
- Many different local feature methods.
  - Can detect edges, corners, regions with uniform intensity, etc.
  - For more information, see tutorials[5][6].

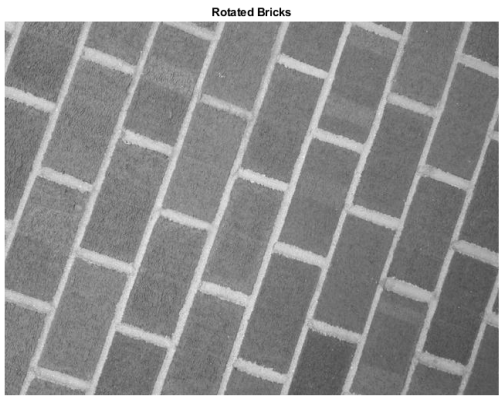
[5] OpenCV-Python Tutorials: Feature Detection and Description

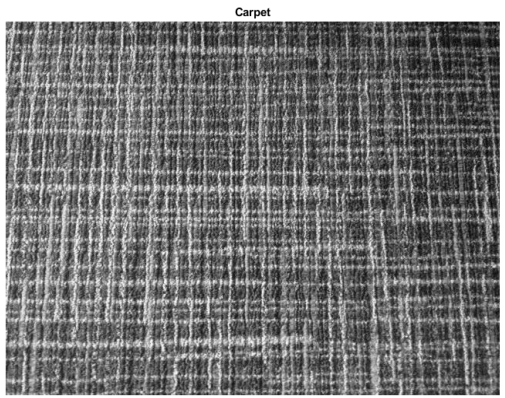
[6] Matlab tutorial: Local Feature Detection and Extraction

Example: Local features for differentiating images by texture (1/3)

- We would want to define a feature vector images that can distinguish images with different kinds of textures, e.g., brick vs. carpet.[7]



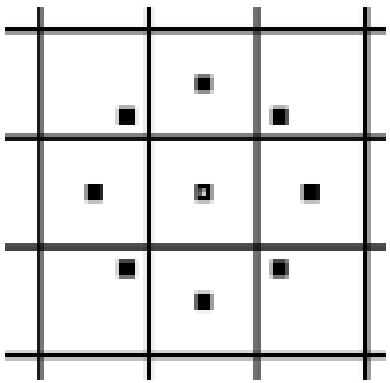


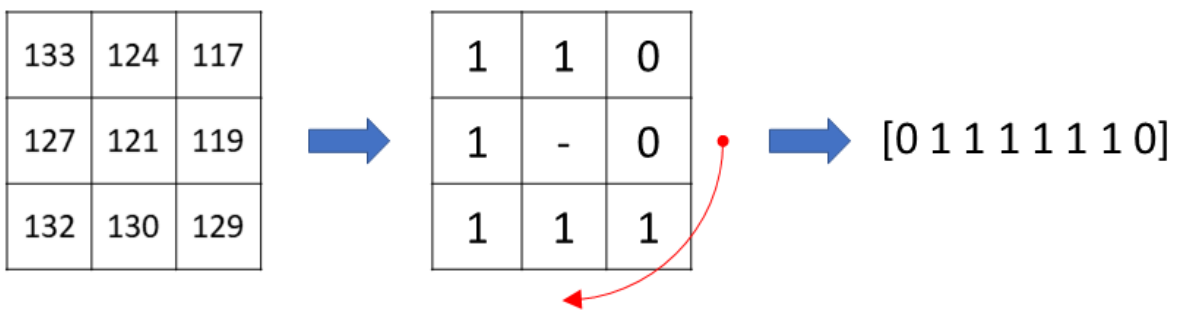


[7] Example from mathworks.com for extractLBPFeatures.

Example: Local features for differentiating images by texture (2/3)[8]

- **Local binary patterns**
  - Look at the 8 neighbors of each pixel
  - If the neighbor is greater than the center pixel, mark it as 1, otherwise mark it as 0.
  - In total 2^8 possible neighbor patterns, e.g., [1 0 0 0 0 1 1 0].
  - Remove symmetric patterns.

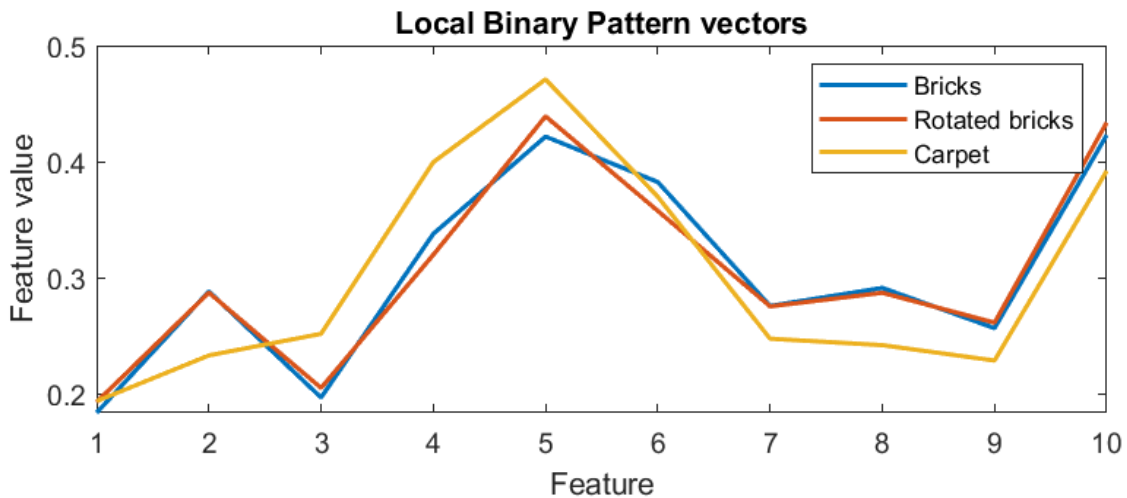




[8] Ojala et al. (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns

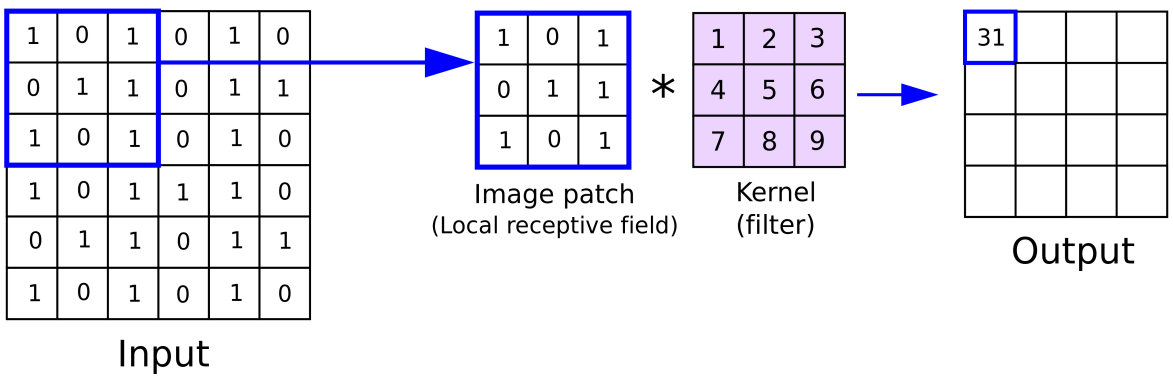
Example: Local features for differentiating images by texture (3/3)

- Define a 10-dimensional feature $\mathbf{x}_i$ vector for each image i:
  - proportions of the 9 most important patterns in the whole image and the proportion of all other patterns.
  - normalized to a unit length: $\|\mathbf{x}_i\| = 1$.



Convolution (1/2)

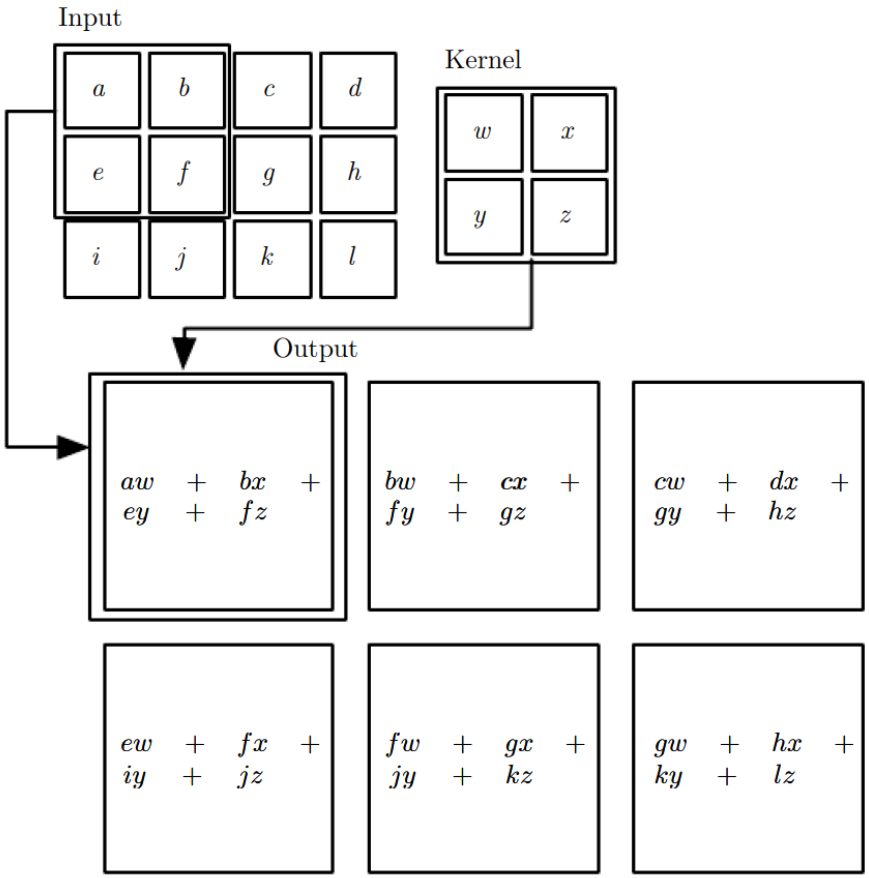
- **Convolution** is an generic operation which forms new features as the weighted sum of neighboring pixels
  - Weights are defined by a **kernel** (or **filter**)
  - The output (or **feature map**) is obtained by multiplying each image patch elementwise with the kernel.
  - A typical kernel size with image data is 3*3.



https://anhreynolds.com/blogs/cnn.html

Convolution (2/2)

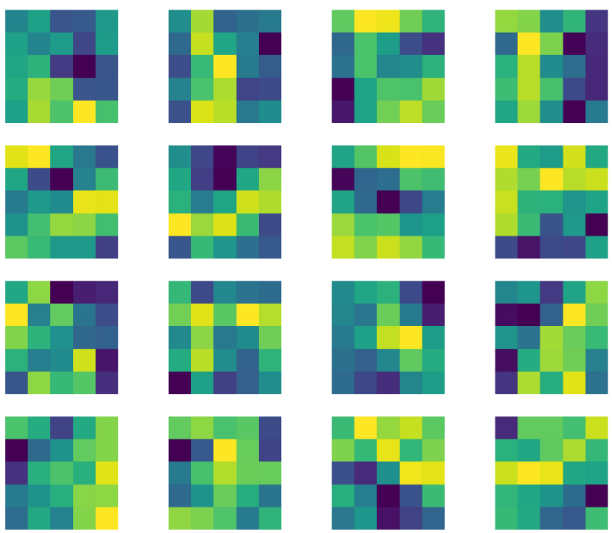
In practice the elements of the kernel (**weights**, **parameters**) are not set by hand but learned from data to result in useful features.

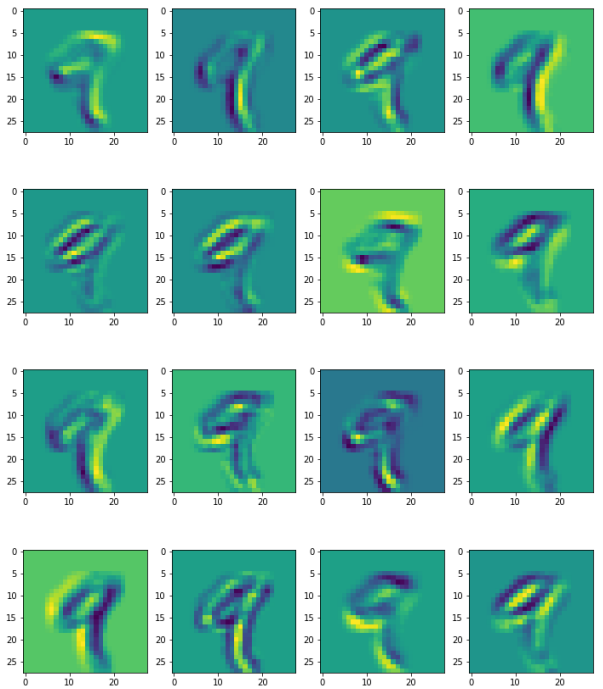


Goodfellow et al. (2015) Deep Learning.

Convolution, example

- A model trained to classify images of digits 0,1,...,9.
- Learned filters and a feature map (after one convolutional layer) for an input image of digit 9.[9]

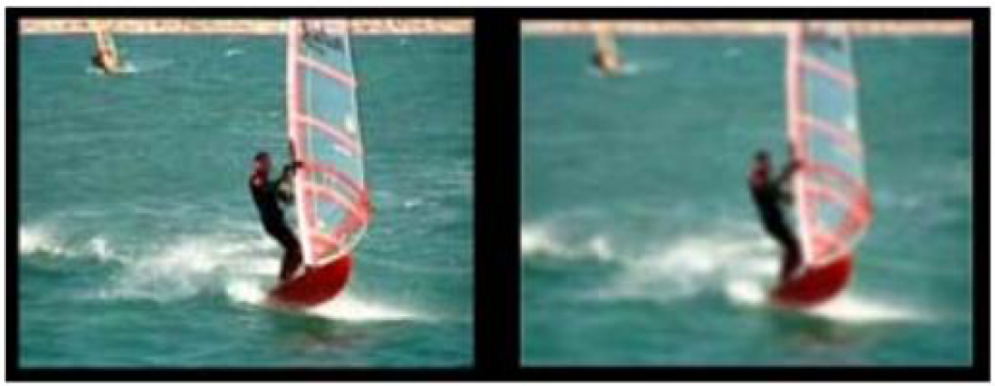




[9] https://medium.com/dataseries/visualizing-the-feature-maps-and-filters-by-convolutional-neural-networks-e1462340518e

Convolution for smoothing

- Convolution can be used to average across neighboring pixels, resulting in a **smoothed** image.[10]



Gaussian Blur

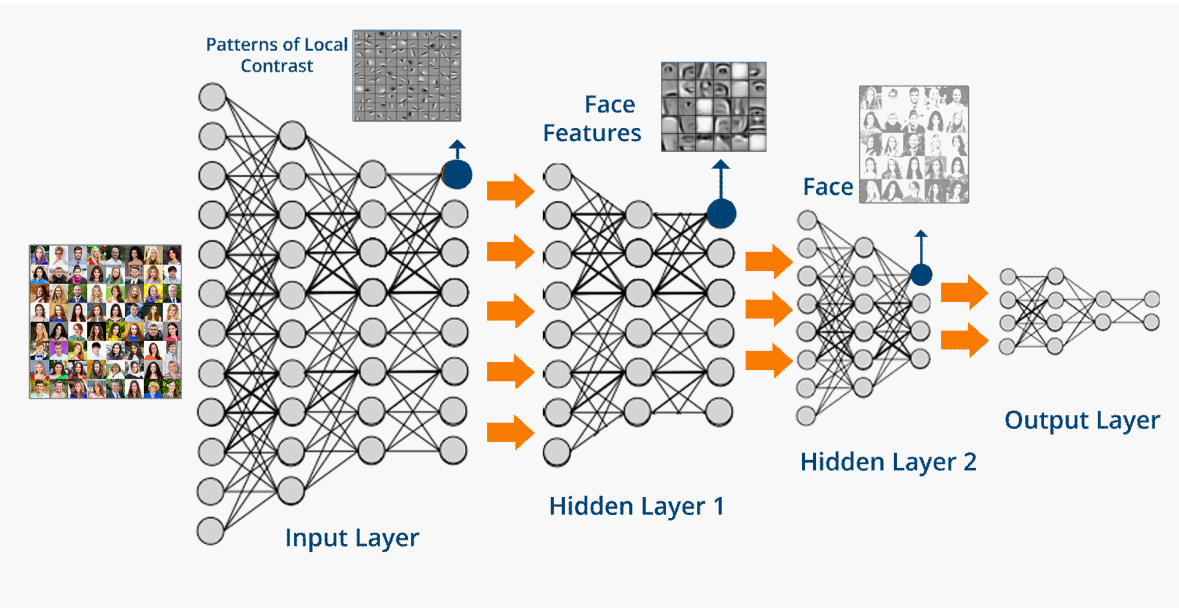
0	1	0
1	4	1
0	1	0

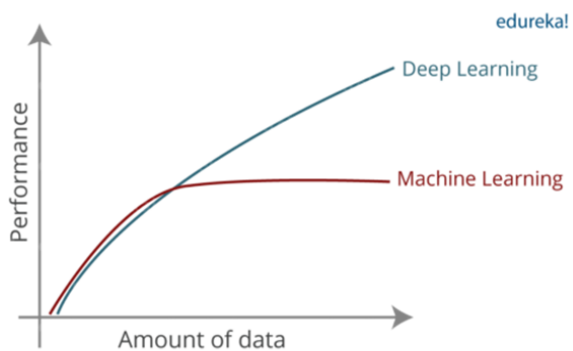
Weighted 3x3 smoothing kernel with Gaussian blur

[10] https://stackoverflow.com/questions/34851259/how-to-make-a-smooth-kernel-in-convolution-neural-networks-with-mxnet-framework

Deep learning

- Hierarchical models with multiple layers.
- Trained with massive data.
- Automated extraction of abstract interpretable features.[11]





[11] Figures: https://www.edureka.co/blog/what-is-deep-learning

Summary

- Machine learning means *learning from data* how to solve a given task
- Three types of machine learning: supervised, unsupervised, reinforcement learning
- Basic concepts: data matrix, feature vector
- Visualizations: histogram, scatter plot
- Features for text and images
- Convolution operation

Recommended reading

- Murphy: *Machine Learning: A Probabilistic Perspective*
  - Sections 1.1-1.3, p. 1-15
  - A nice introduction to machine learning.
- Goodfellow et al.: *Deep learning*
  - Section 9.1, p. 322-324
  - Explains the convolution operation