Adversarial Machine Learning Course

Georgia Fargetta, Ph.D.

Erasmus +
Master Degree in Computer Science
University of Rouen, France

Info&contacts

Georgia Fargetta, *Ph.D.*

Email: *georgia.fargetta@unict.it*

Personal webpage: http://www.dmi.unict.it/fargetta/index.html

Università di Catania

SICILIAE STVDIVM GENERALE 1434

IMAGE PROCESSING LABORATORY

IPlab@UNICT

http://Iplab.dmi.unict.it – Email: iplab@dmi.unict.it

Core Competences: **Computer Vision and Multimedia**

Overall Team (~15 – Lead by Prof. Sebastiano Battiato)

- **Results:** ~30 patents, >300 papers
- **Current topics:**
  - Data Analisys
  - Multimedia Forensics and Security
  - Context Aware Enhancement
  - Social Media Mining
- **R&D projects:**
  - Funded projects: ENIAC (1), PO/FESR (4), KDT (1) MISE HZ2020(1), others (4), PON NRR (PE AI, Cyber, - CN HPC)
- **International Events:** IFOSS (since 2022), ICVSS (since 2007), ICIAP 2017, ACIVS 2015, VISAPP

Team IPLAB@UNICT

3 FULL PROFESSORS
2 ASSOCIATE PROFESSOR
4 ASSISTANT PROFESSOR
3 POST DOC
15 PHD STUDENTS

Digital and Multimedia Forensics
Image and Video Understanding
Data Analysis and Applications
Computer Vision and Applications (e.g. First Person View, Medical Imaging, etc.)
Social Media Mining
Video Analytics (e.g. Video Surveillance, etc.)
Archeomatica (Imaging for Cultural Heritage)

COMPANY OVERVIEW

Digital forensics Consulting
Multimedia forensics and **Security**
Advanced **Data Recovery**
Research and Development

PARTNERS

Tech**rgy**

ACHIEVEMENTS

2 PhD Fellowships: PON FSE-FESR 2014-2020 (Dottorati industriali)

More than 20 MD Thesis: on digital forensics and multimedia security topics

More than 50 Forensics Cases/y: proudly and ethically investigated by our experts

iCTLab S.r.l.

WWW.ICTLAB.SRL

HISTORY

Founded in 2016
Spinoff of University of Catania

R&D PROJECTS

Deepfake Analysis: for both authentication and security applications

Multimedia Analytics: semantic analysis of images, audio and video for investigations and extraction of insights.

Vision-Based Monitoring: we employ SOTA computer vision solutions to empower industrial monitoring in many contexts

Georgia Fargetta - Unive

IFOSS
International Forensics Summer School

ETHICAL AND LEGAL CHALLENGES IN AI-DRIVEN FORENSIC SCIENCE

JULY 14-20, 2024

Watch a preview!

School Directors

PROF. SEBASTIANO BATTIATO, PH.D.
University of Catania

PROF. DONATELLA CURTOTTI, PH.D.
University of Foggia

PROF. GIOVANNI ZICCARDI, PH.D.
University of Milan

Speakers

ALESSANDRO TRIVILINI
Scuola universitaria professionale della svizzera italiana (SUPSI)

MARTIN DRAHANSKÝ
Faculty of Information Technology, Brno University of Technology

PROF. DR. DIDIER MEUWLY
University of Twente

others coming soon..

School location

The school will take place at Sampieri, Sicily
https://www.hotelbaiasamuele.it/en/

Social Network

IFOSS

@ifoss_official

@ifoss_official

IFOSS

www.ifoss.it

info@ifoss.it

Outline

- *Introduction to Artificial Neural Networks*
- *Generative Adversarial Networks (GAN)*
- *Deepfakes and countermeasures*
- *Adversarial Machine Learning*
- *Adversarial Machine Learning and Game Theory*

Introduction to Machine Learning and Deep Neural Networks

Georgia Fargetta, Ph.D.

Erasmus +
Master Degree in Computer Science
University of Rouen, France

Machine Learning & Big Data

One of the most important components of a data analysis process is constituted by the quantitative and qualitative characteristics of the data.

The proliferation of devices acquiring and communicating information leads to a growth in data that must be transmitted, stored, and interpreted.

Often, the outcome of data analysis determines subsequent behaviors and actions.

Considering the quantity and variety of information, the analysis of these data requires specific processes and techniques.

Machine Learning & Big Data

The term "Big Data" refers to a set of data that grows along three dimensions (3V):

Volume: the quantity of data generated over time from heterogeneous sources;

Variety: the generated data can take on various forms (e.g., text, numbers, maps, audio, video, email, etc.);

Velocity: the speed at which data is generated is continuously increasing. Consequently, so is the speed required to analyze them.

D. Laney (2001)

Machine Learning & Big Data

The definition has later been extended with two additional Vs:

Veracity: it refers to the quality/credibility of the data (e.g., Social Networks);

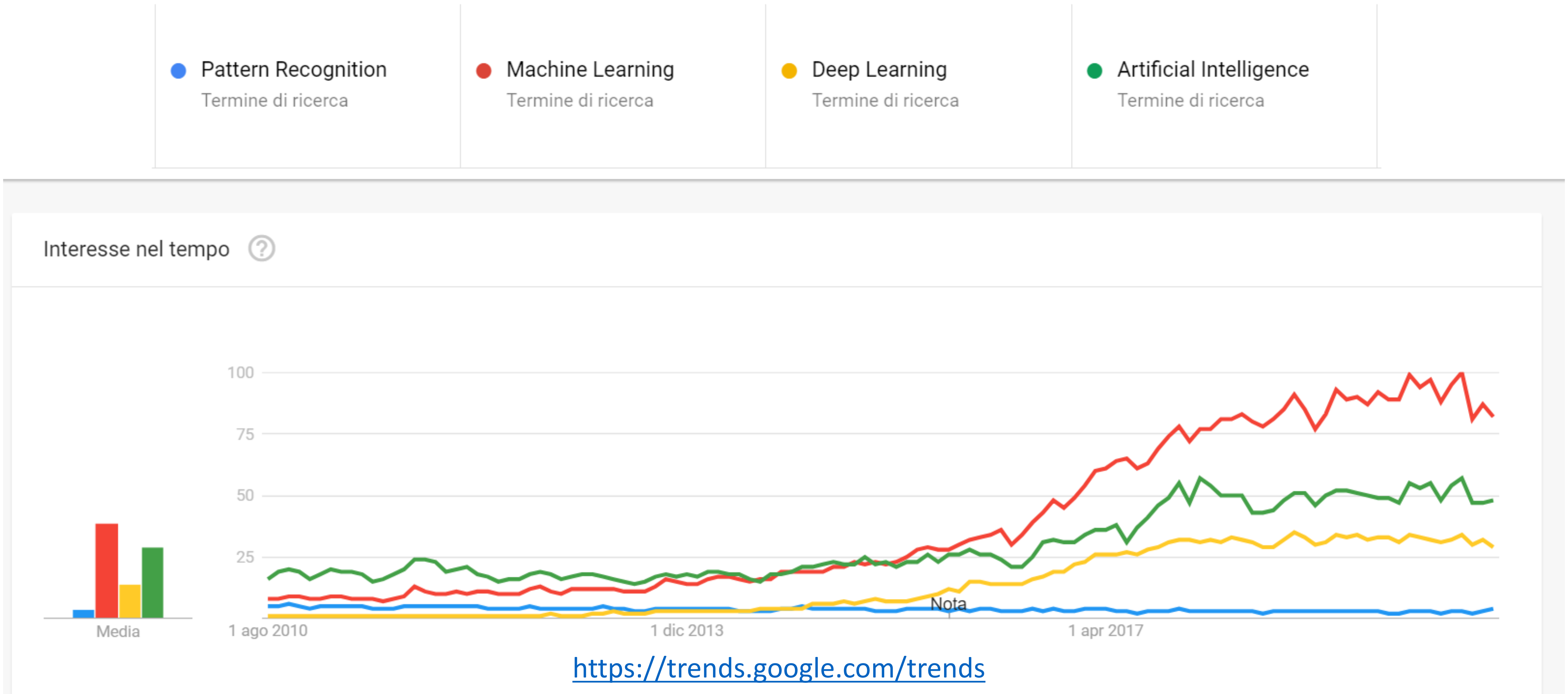
Value: it is crucial to understand if business value can be derived from the data.

Machine Learning & Big Data

Applicative examples of ML made possible by Big Data:

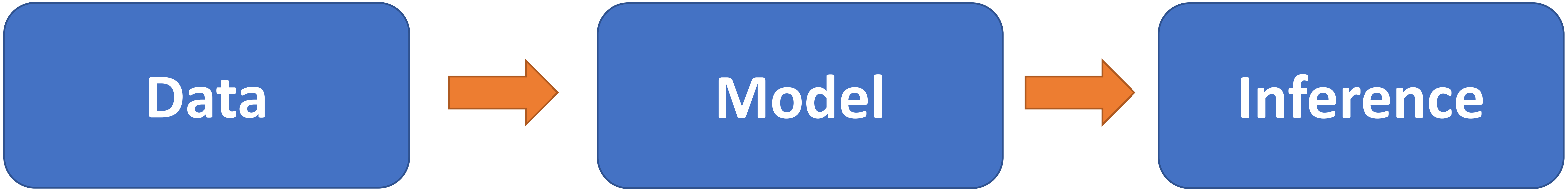
- Medicine: monitoring the spread of diseases;
- Security: analysis of electronic payments;
- Environment: analysis of meteorological and/or pollution data;
- Marketing: user profiling and targeted campaigns (e.g., recommendation systems);
- Transportation: traffic analysis;
- Sports: performance analysis and statistics of athletes/teams.

PR vs. ML vs. DL vs. AI

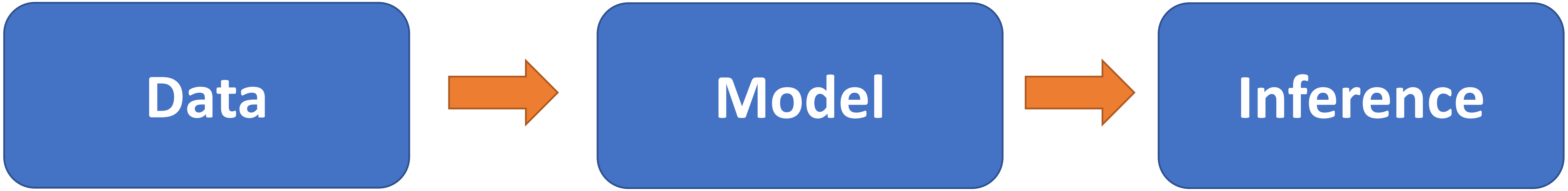


https://trends.google.com/trends

Inference formulation



Inference formulation



Feature definition
Data collection
Data exploration & cleaning
Normalization
....

Model definition
Training phase
Test phase
....

Classification
Regression
Clustering
Anomaly Detection
....

Machine Learning

In general, an ML problem involves a set of data to analyze and attempts to make predictions on new, previously unseen data.

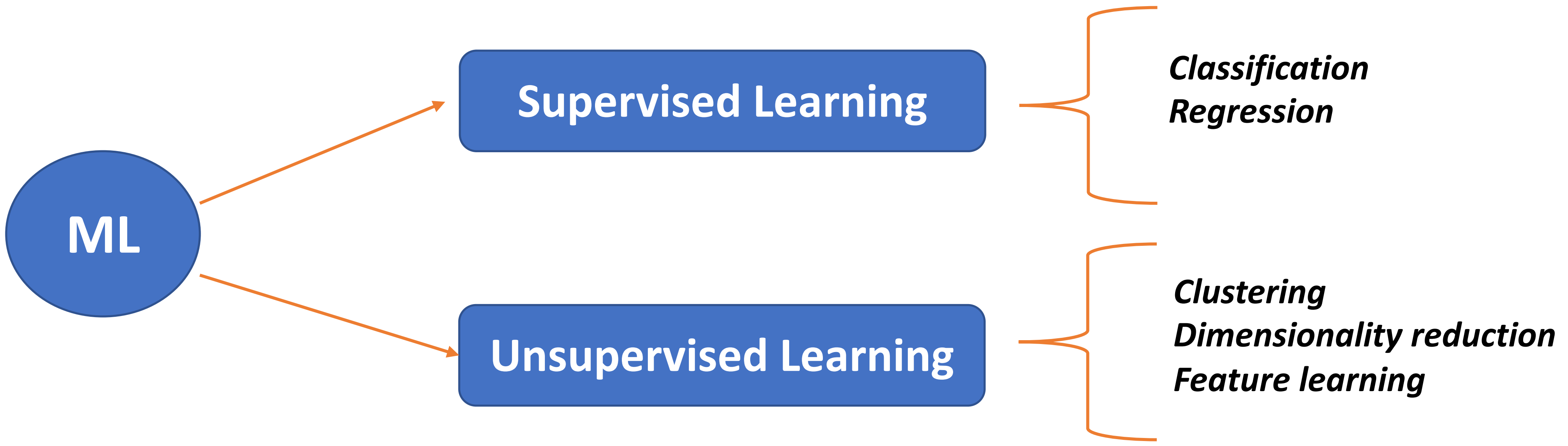
If the data is represented by more than one variable, it is referred to as multivariate data.
In both cases, we refer to the attributes used to describe the data as features.

We can categorize ML problems into different categories based on the data available and the desired type of output.

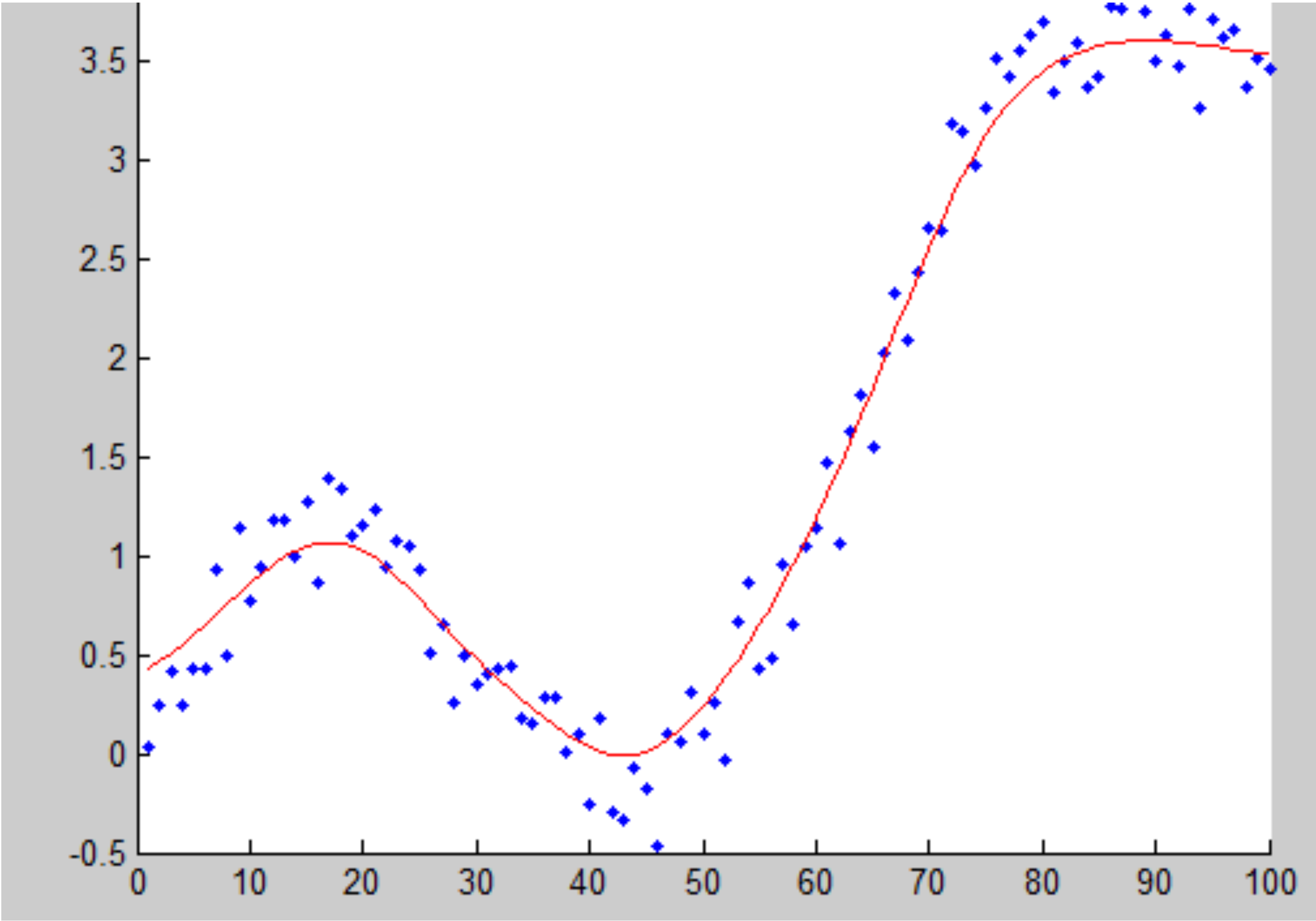
Machine Learning

In general, an ML problem involves a set of data to analyze and attempts to make predictions on new, previously unseen data. If the data is represented by more than one variable, it is referred to as multivariate data. In both cases, we refer to the attributes used to describe the data as features.

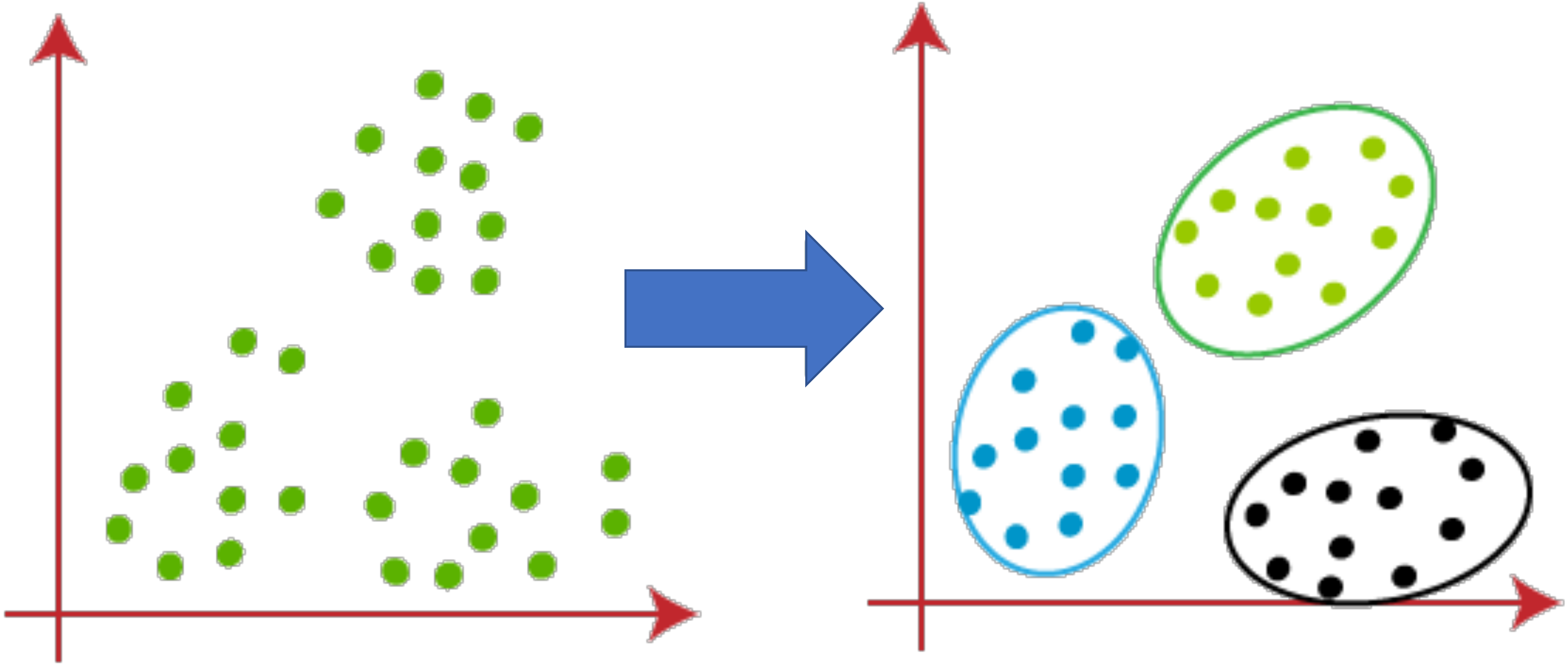
We can categorize ML problems into different categories based on the data available and the desired type of output.



Regression



Clustering



Classification

Statistical population: set of elements that are the subject of study upon which the analysis is conducted.

Feature: directly observable aspect related to a phenomenon for which a quantitative or categorical measure can be recorded (e.g., height, weight, temperature, color, humidity, etc.).

Class: abstract and general concept that summarizes the observations assigned to it (e.g., man, woman, dog, car, etc.).

From the observation of various **features**, the classifier reaches the decision to label the observed data into a more abstract and general category, called a **class**. Classifying can, therefore, be thought of as recognizing, in an observation, characteristics typical of members of a class.

Classification

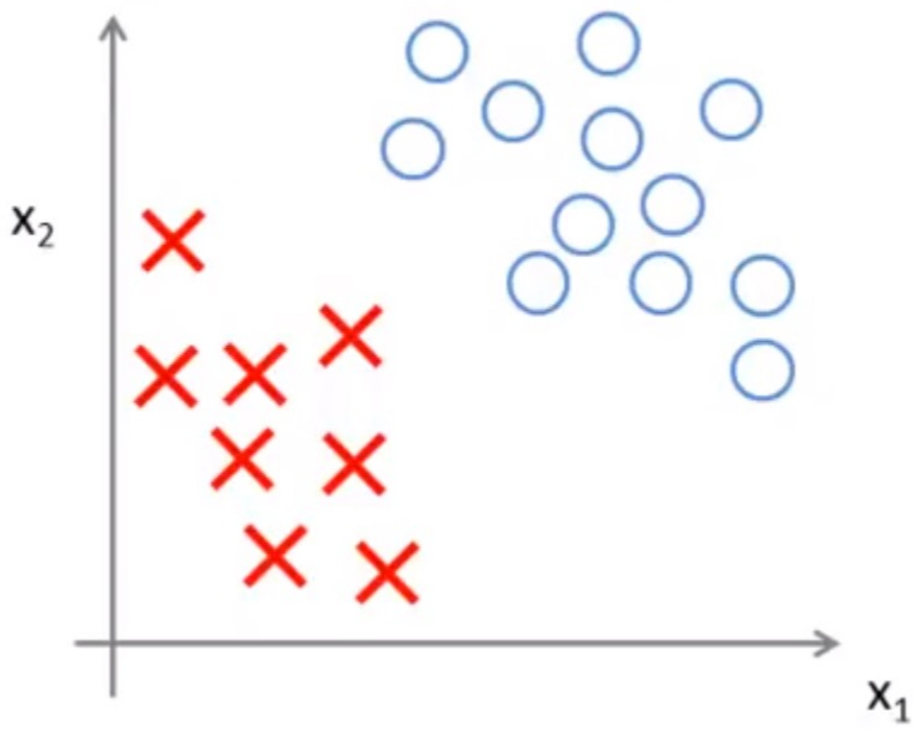
The rules of a classification model/schema depend on a learning phase called **training**: from observing a well-labeled set of data, an attempt is made to deduce rules applicable to unlabeled data or data that will be encountered in the future.

Key steps:

1. Split the data into training and test sets.
2. Normalize the data (utilizing information from the training set).
3. Define a model that performs the classification of data.
4. Test the effectiveness of the model using the test data.

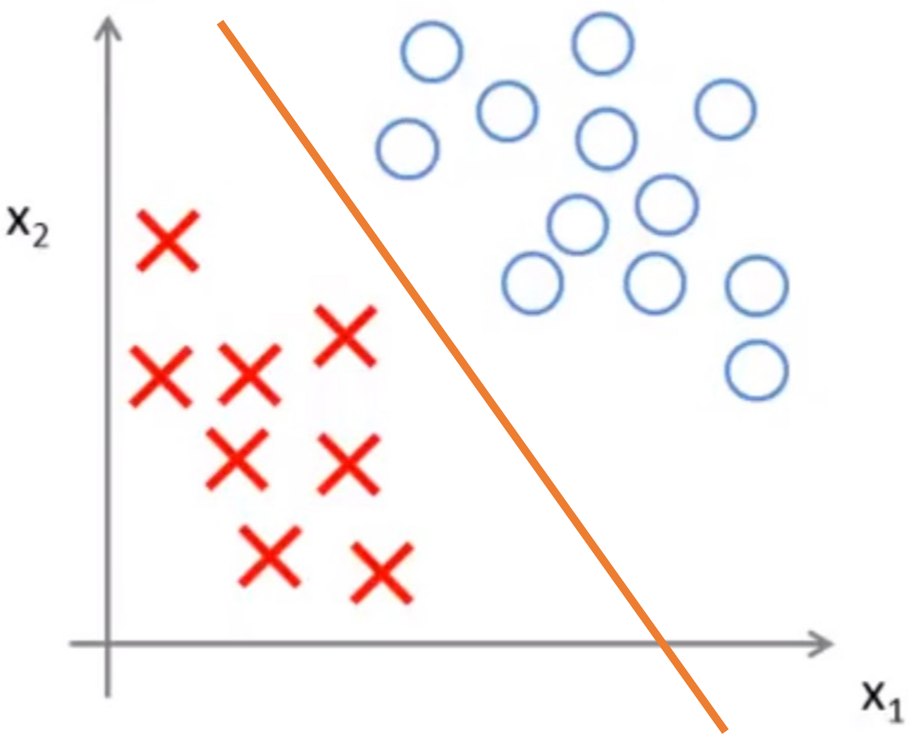
Classification

The goal is to find a function that correctly separates the data based on their class membership (i.e., the class).



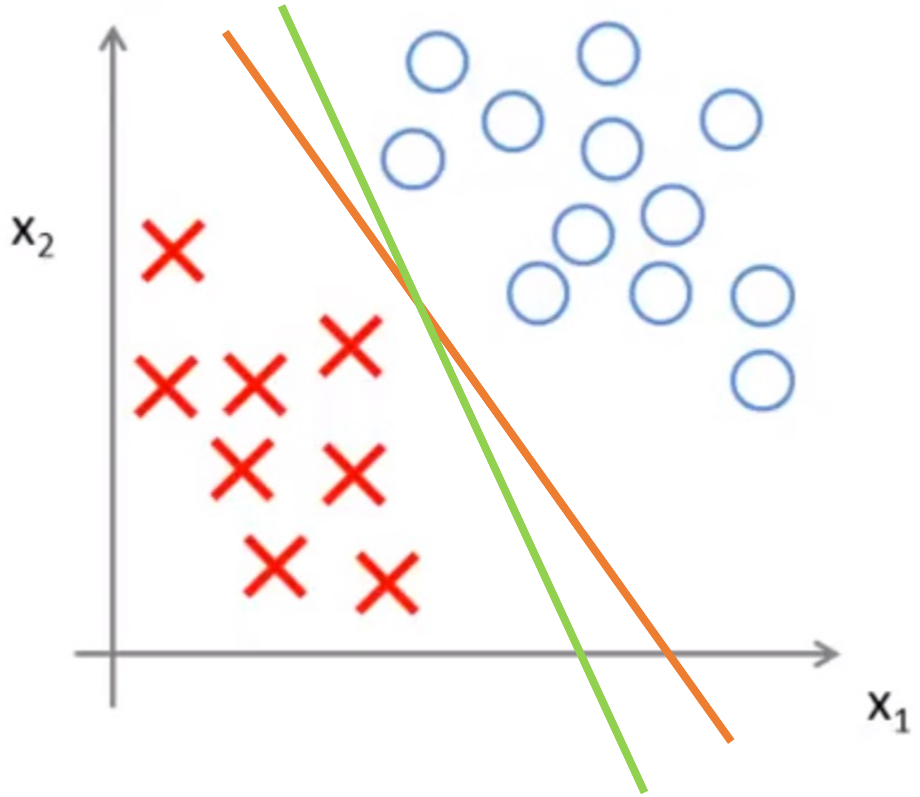
Classification

The goal is to find a function that correctly separates the data based on their class membership (i.e., the class).



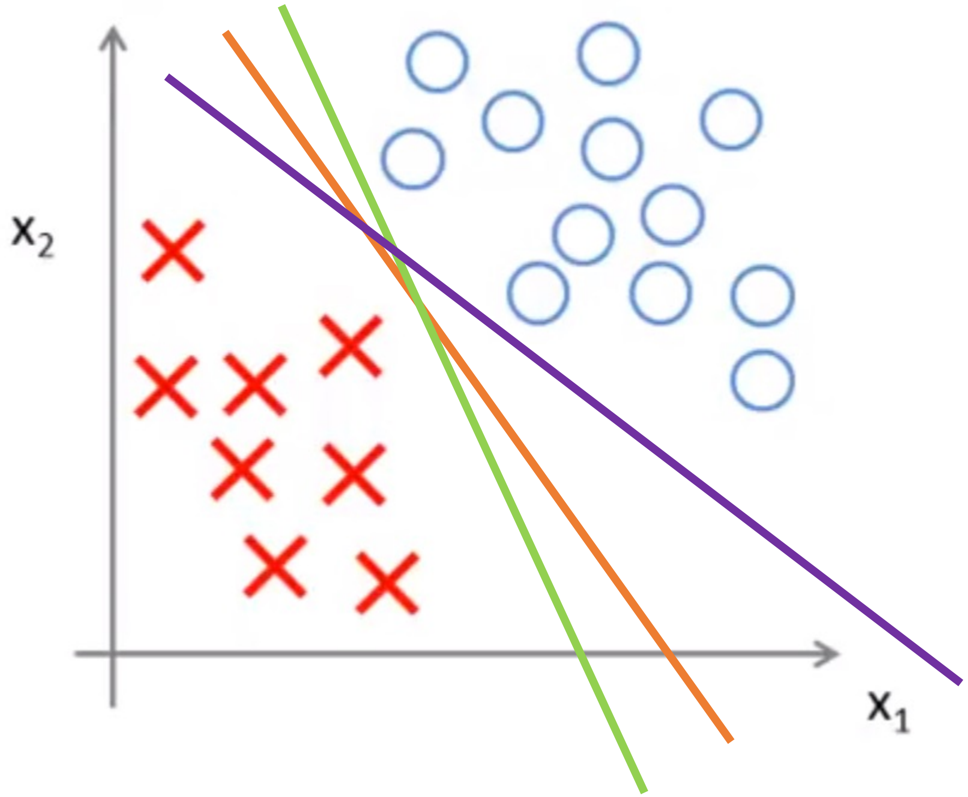
Classification

The goal is to find a function that correctly separates the data based on their class membership (i.e., the class).



Classification

The goal is to find a function that correctly separates the data based on their class membership (i.e., the class).



Data Splitting

The data is divided into training and test sets. The first group of data is used to train a model. The performance of the trained model is then evaluated on the test set.

The training set is always larger than the test set (70-80% of the data).

Data Splitting

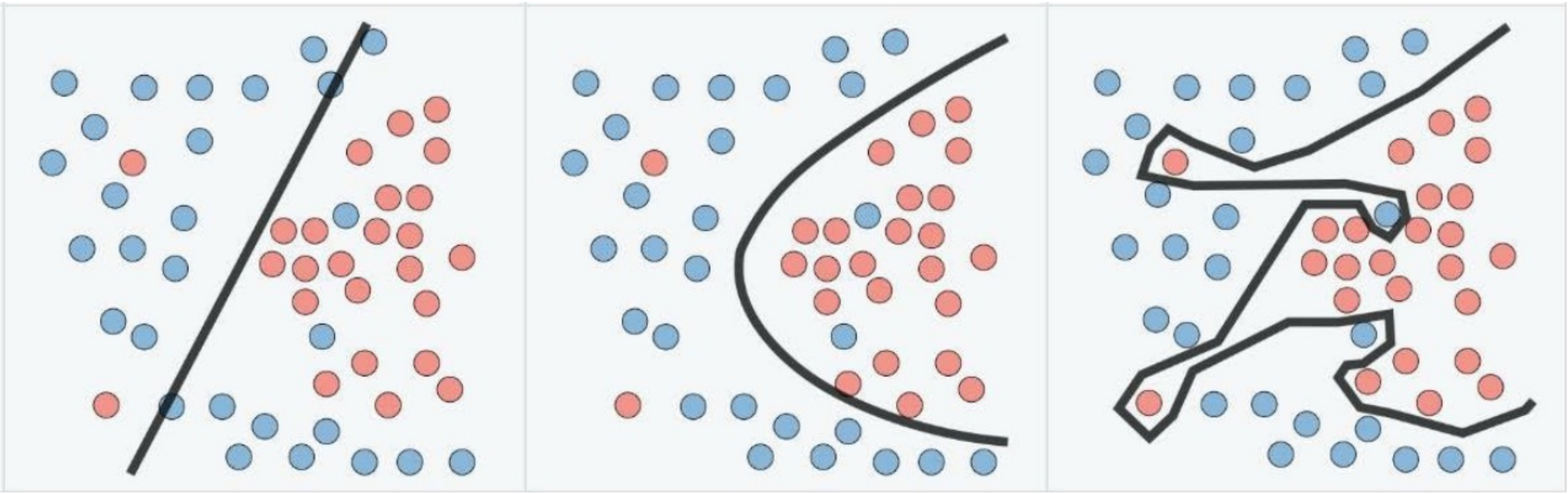
The data is randomly divided into training and test sets.

It is essential to ensure that both sets are large enough to represent all variations in the data (e.g., all classes, outliers). Otherwise, there is a risk of encountering **overfitting**.

Overfitting occurs when a model performs very well on the training set, but its performance on the test set is significantly lower. This is caused by the model being overly tailored to the examples seen during training.

Overfitting

Underfitting ***Overfitting***



Data Splitting

K-fold Cross Validation

It is a very simple method to avoid overfitting.

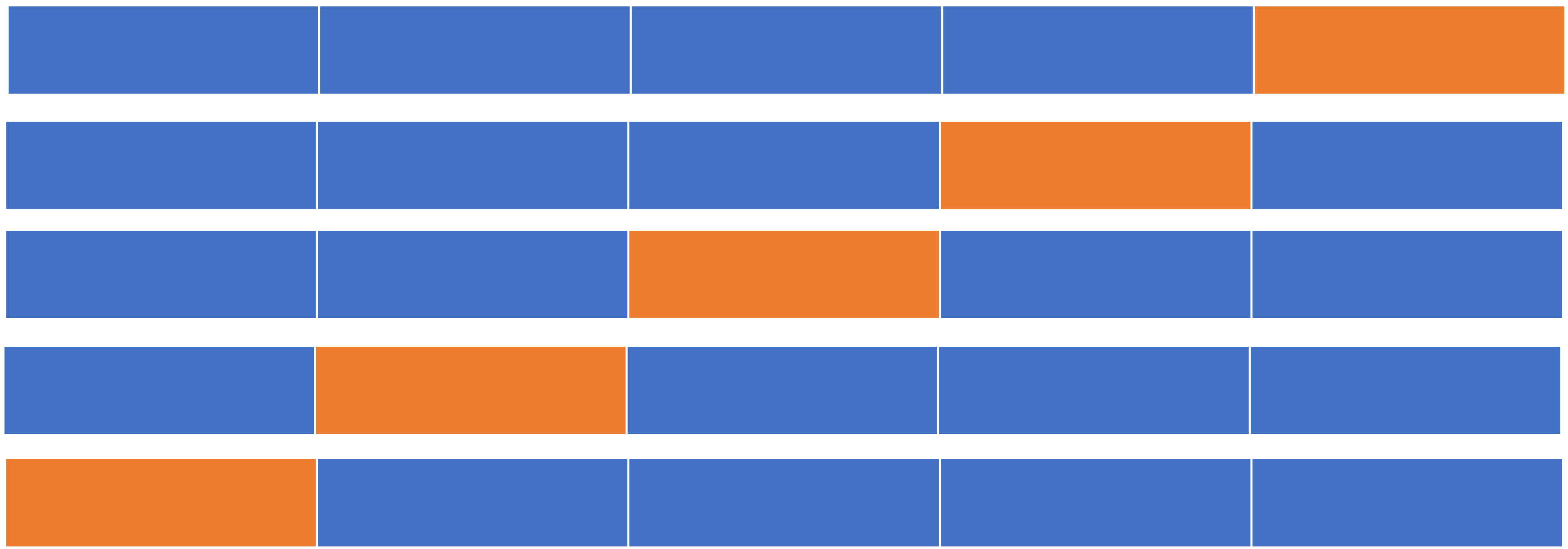
1. Randomly divide the data into k equal-sized folds.
2. Train the model on k-1 subsets.
3. Test the model on the one subset not included in the training.
4. Repeat steps (2) and (3) by changing the subset used as the test.
1. 5. Express the final result as the average of the k obtained results.

Data Splitting

K-fold Cross Validation

Data Splitting

K-fold Cross Validation



K-Nearest Neighbors Classifier

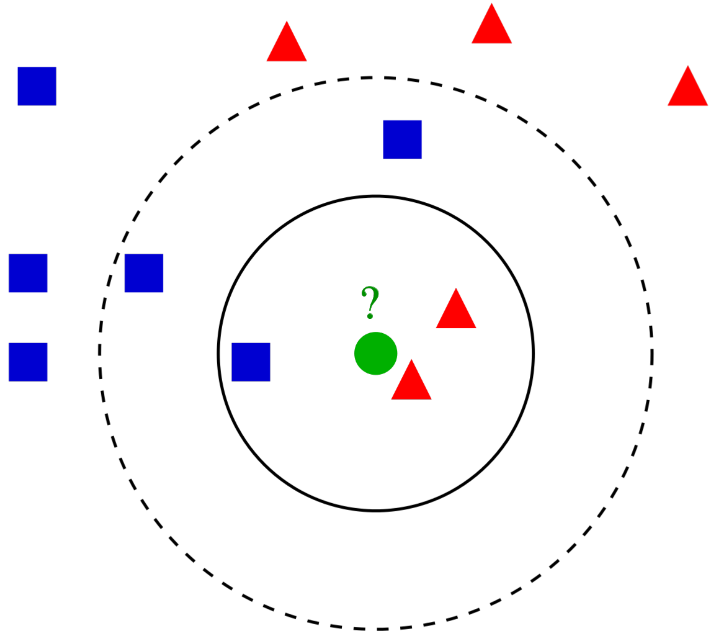
KNN (K-Nearest Neighbors) is one of the simplest classification methods and does not require a training phase

The algorithm classifies new data (test set) based on their distance from known data (training set).

Given a new data point x to be classified:

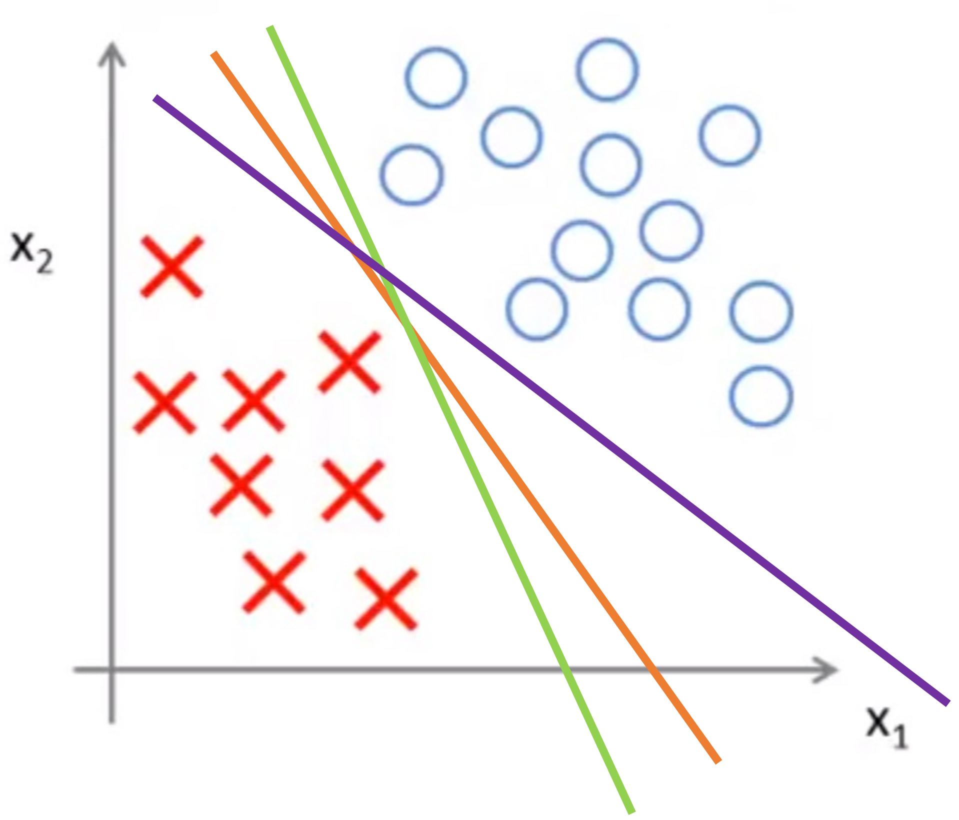
1. Find the k elements in the training set closest to x.
2. Assign x to the class that holds the majority.

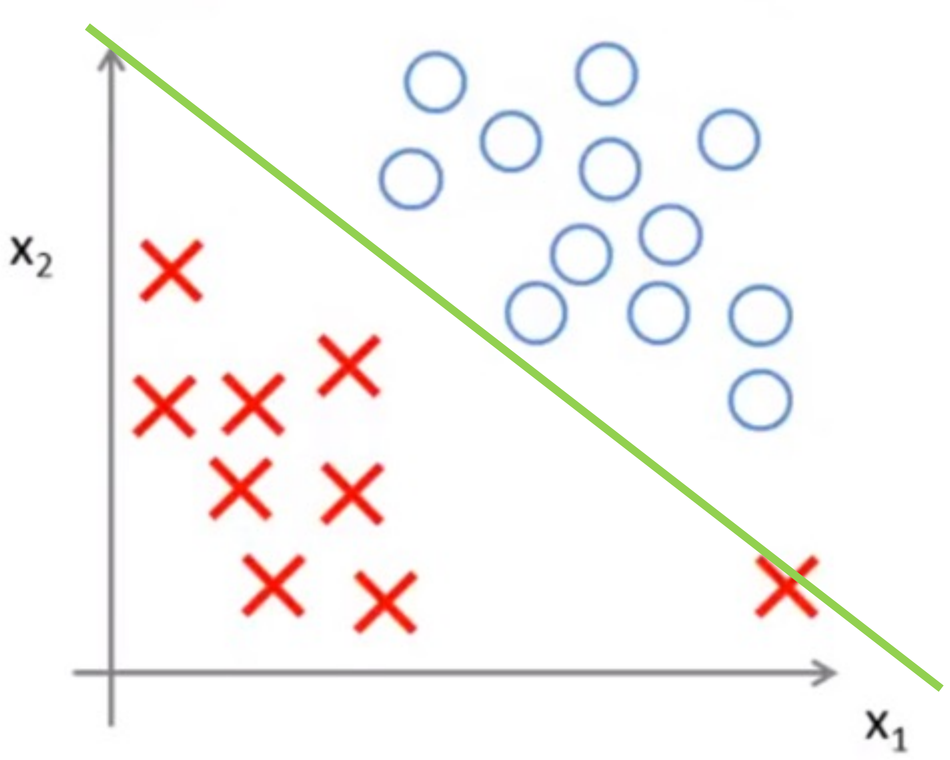
The value of k should always be odd.



Approssimate Solutions

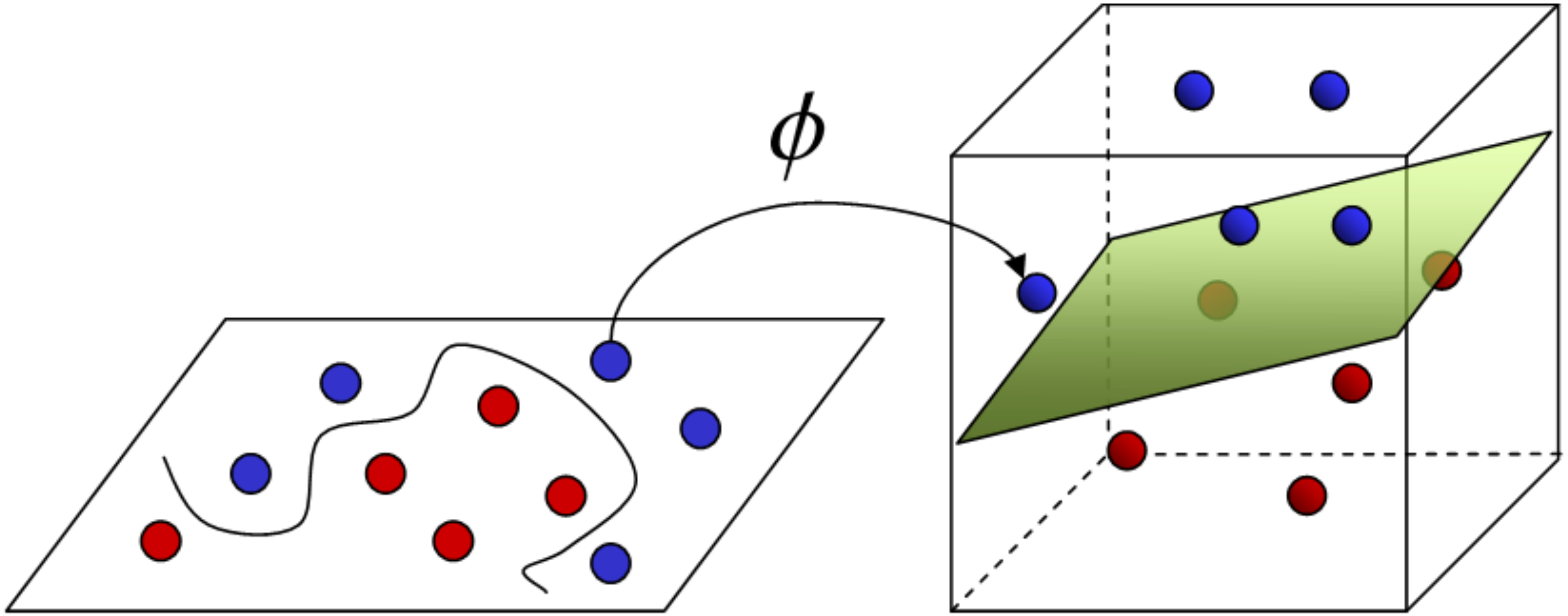
When working with real-world data, we often settle for an approximate solution, meaning that we accept making occasional errors, aiming to minimize the average error.





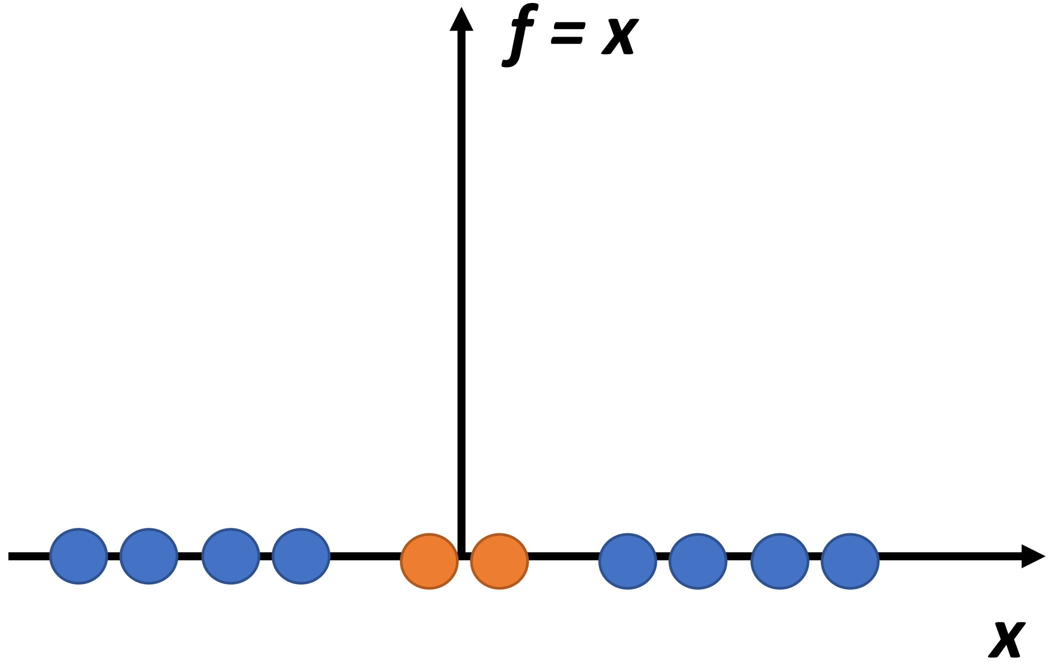
Input Space vs Feature Space

Often the original input data are transformed into a higher dimensional space using a nonlinear mapping to make data separable in the new feature space.

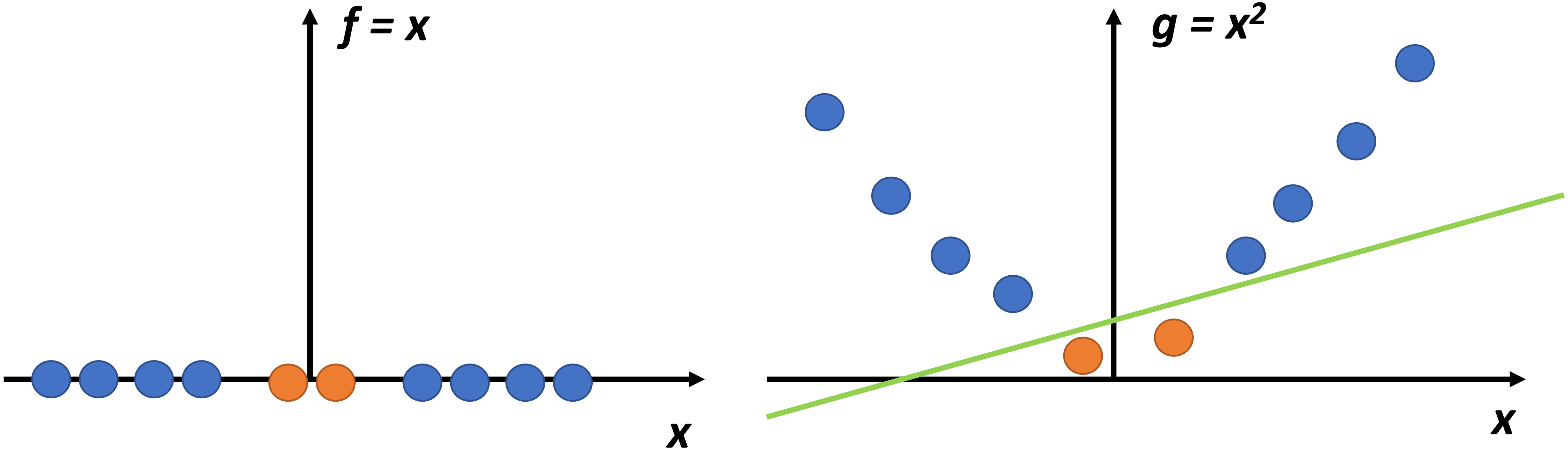


In general, the ***real*** decision boundary becomes more complex as the feature dimension space becomes larger.

Approssimate Solutions



Approssimate Solutions



Gradient Descent

La discesa del gradiente è un metodo di ottimizzazione che serve a trovare il minimo di una funzione.

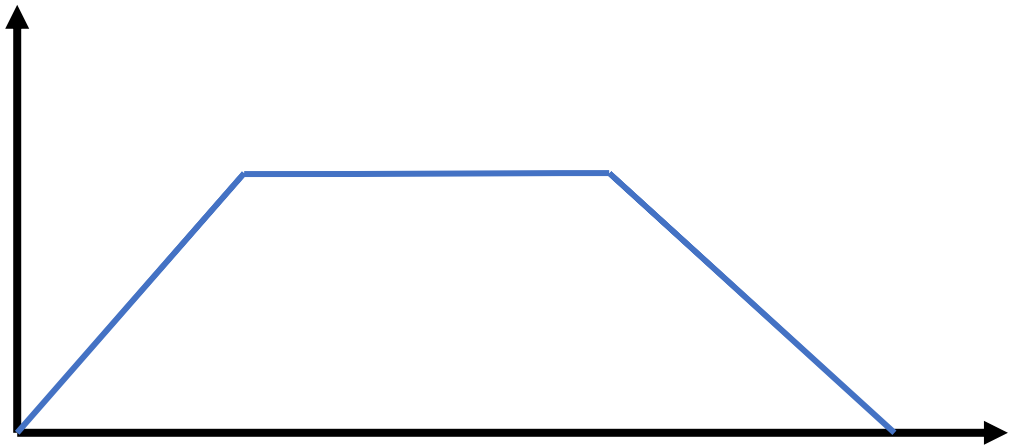
Viene utilizzato in molti algoritmi di Machine Learning, ed è la base di funzionamento delle reti neurali artificiali.

Il suo funzionamento è basato sul calcolo delle derivate.

Gradient Descent

Its operation is based on the calculation of derivatives.

Function f(x)

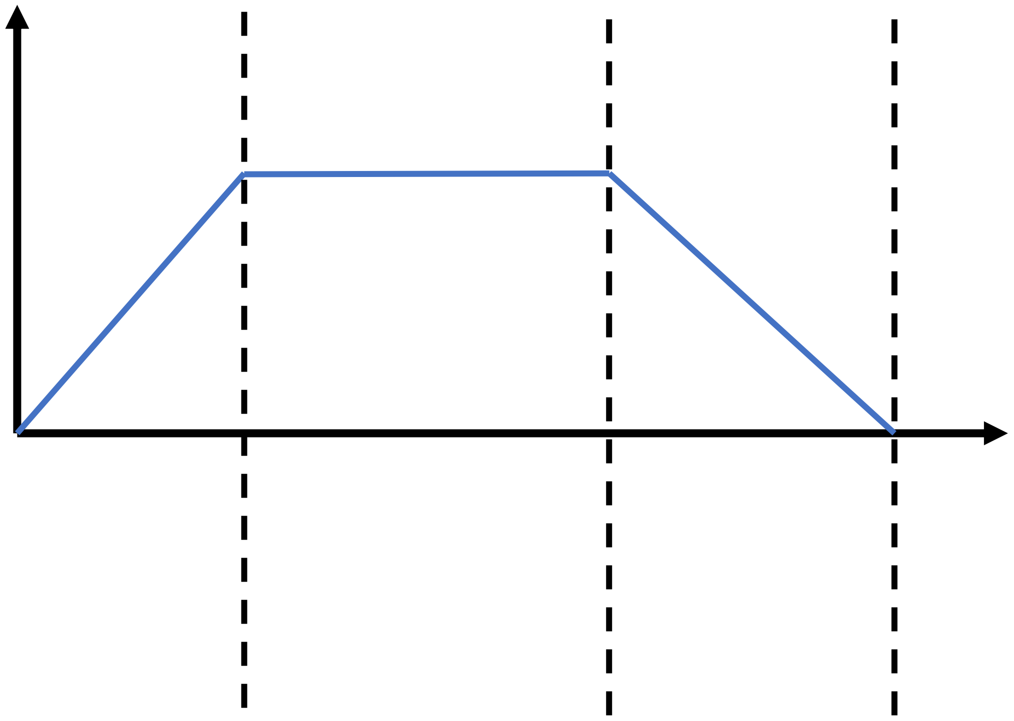


First Derivative f'(x)

Gradient Descent

Its operation is based on the calculation of derivatives.

Function f(x)

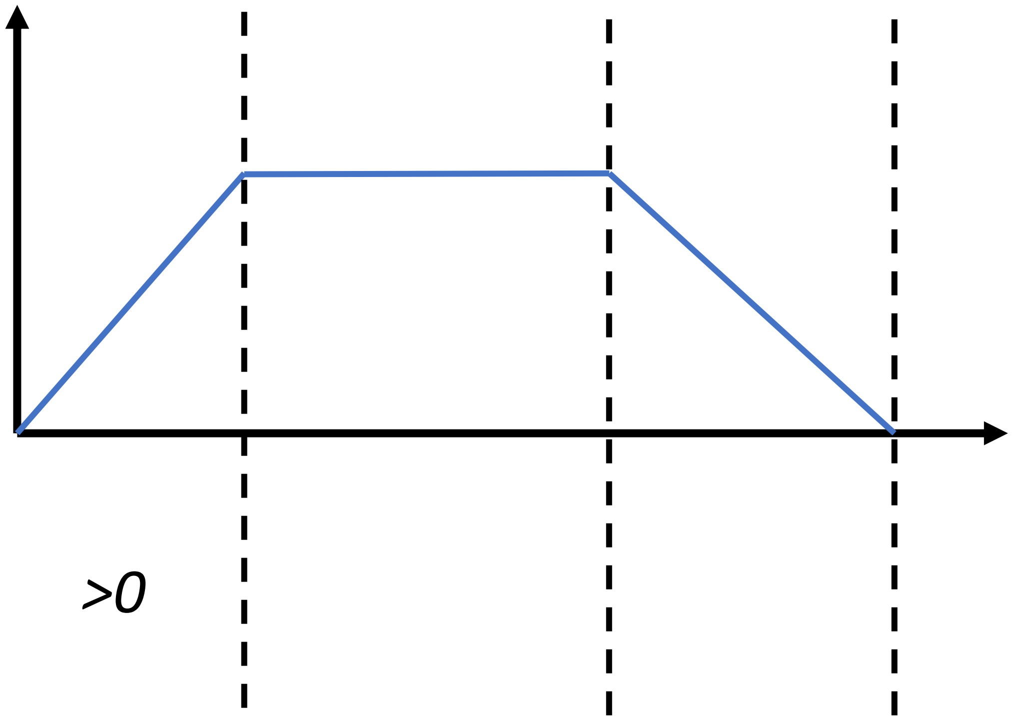


First Derivative f'(x)

Gradient Descent

Its operation is based on the calculation of derivatives.

Function f(x)

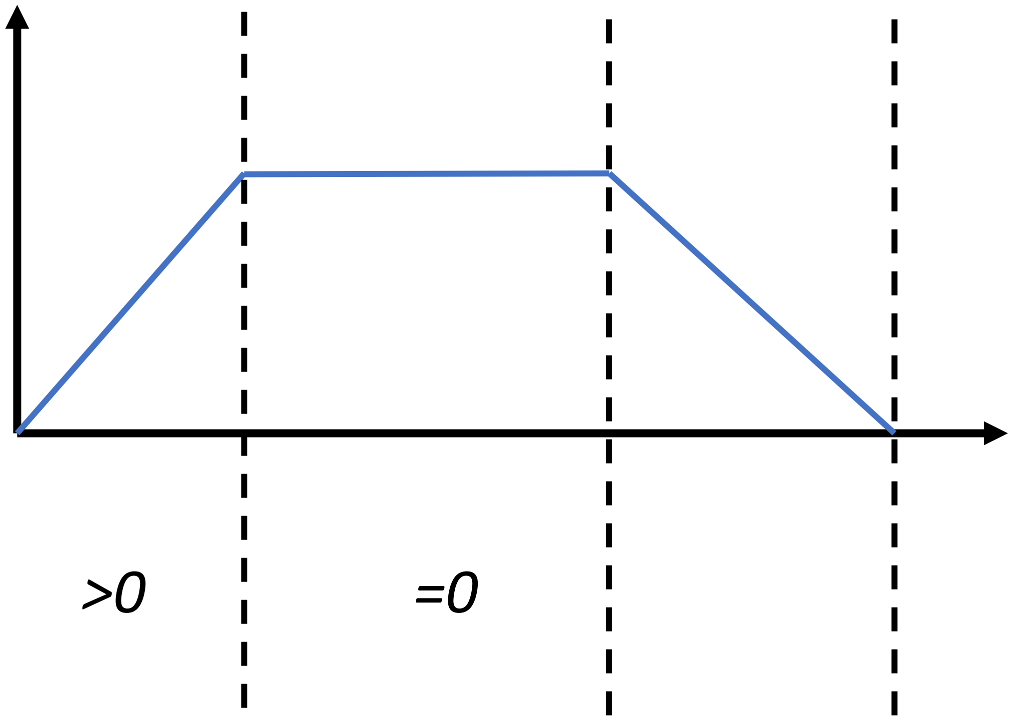


First Derivative f'(x)

Gradient Descent

Its operation is based on the calculation of derivatives.

Function f(x)

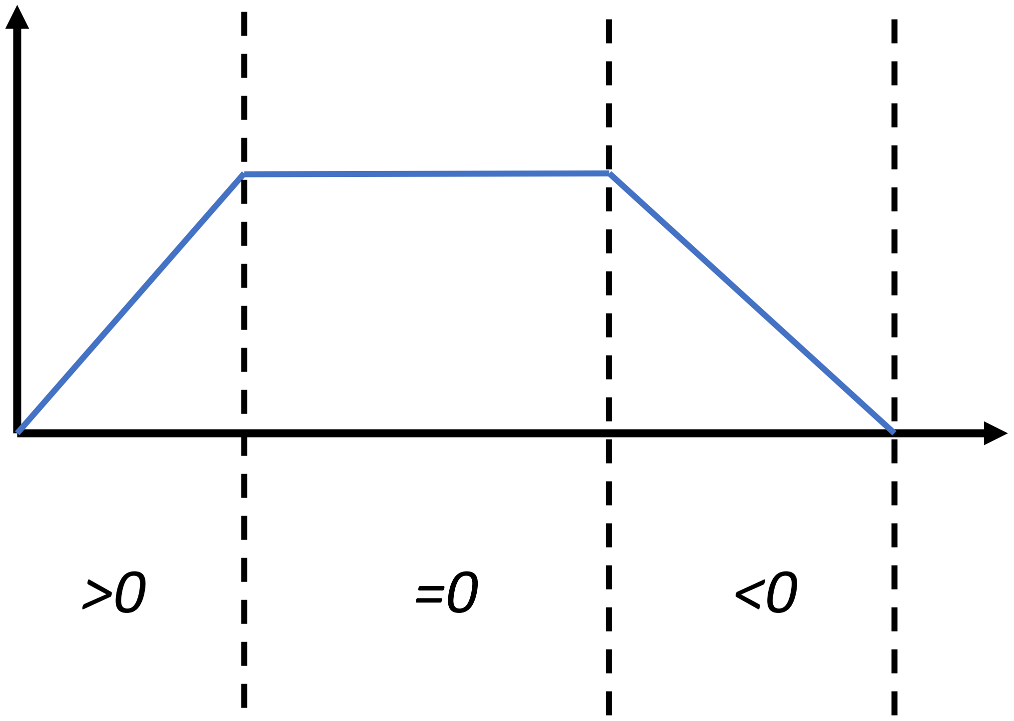


First Derivative f'(x) *>0* *=0*

Gradient Descent

Its operation is based on the calculation of derivatives.

Function f(x)

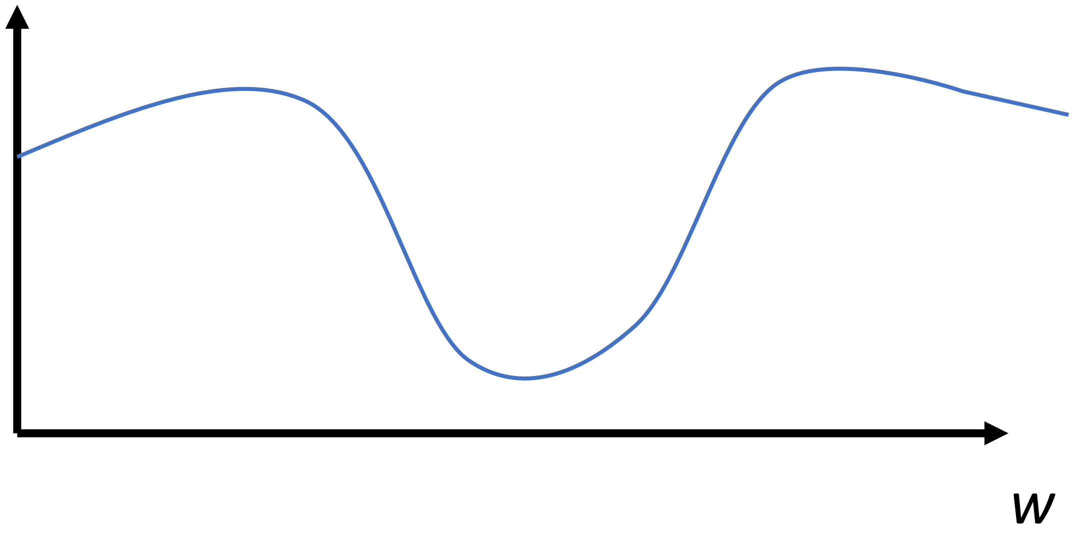


First Derivative f'(x)

Gradient Descent

We want to find the minimum of the function

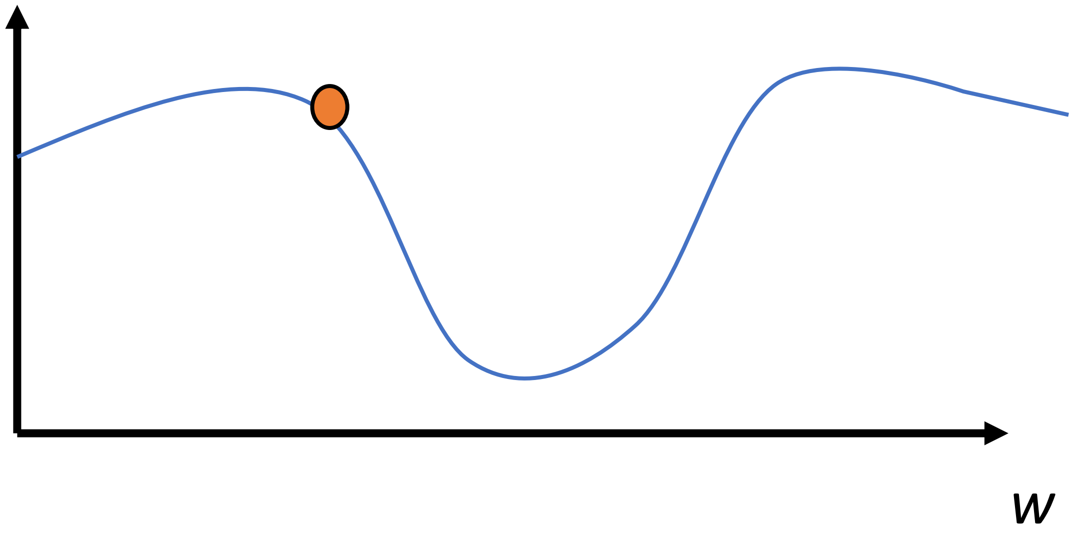
Function f(x)



Gradient Descent

We want to find the minimum of the function

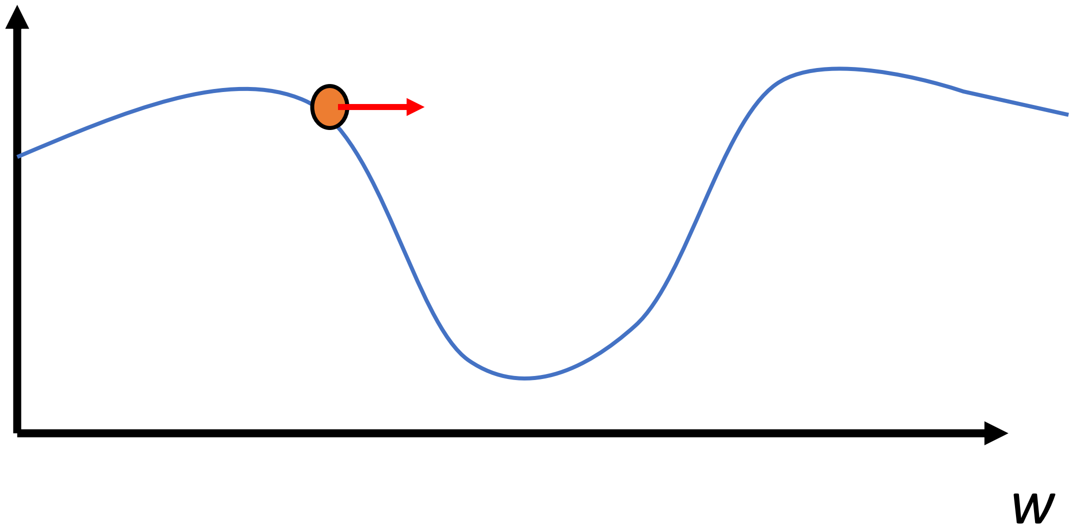
Function f(x)



Gradient Descent

We can leverage the value of the derivative at that point: I calculate the derivative and move in the opposite direction of the gradient until converging to a minimum.

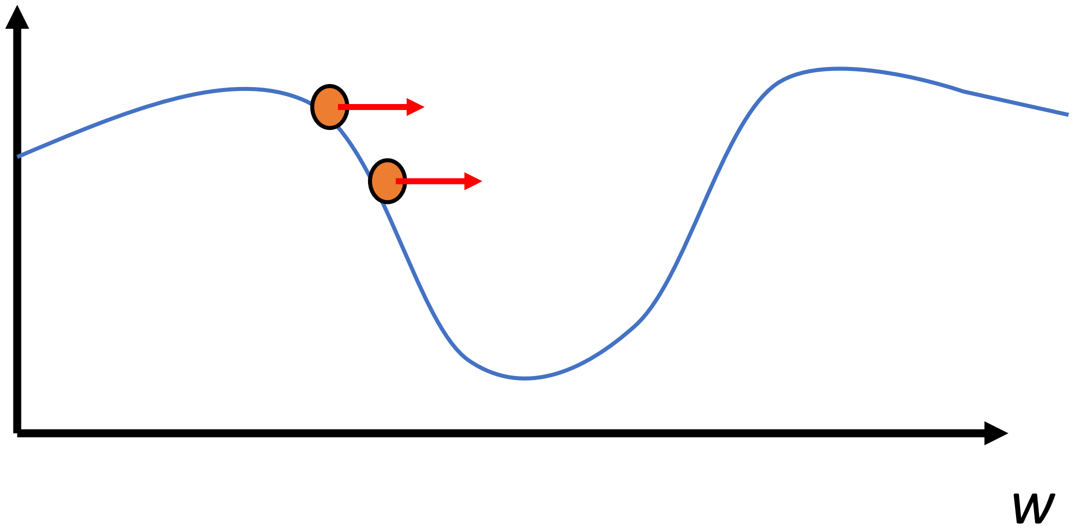
Function f(x)



Gradient Descent

We can leverage the value of the derivative at that point: I calculate the derivative and move in the opposite direction of the gradient until converging to a minimum.

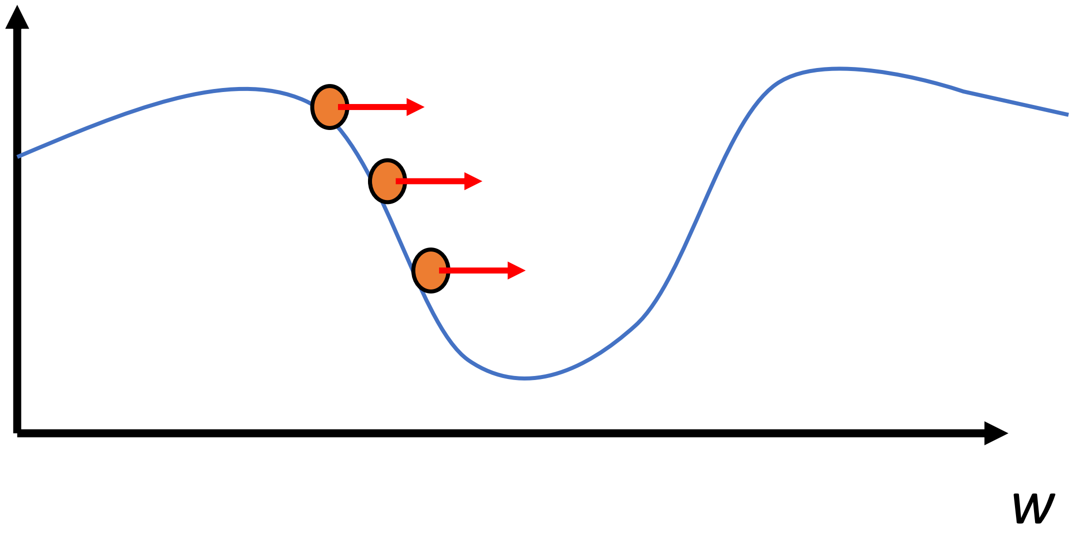
Function f(x)



Gradient Descent

We can leverage the value of the derivative at that point: I calculate the derivative and move in the opposite direction of the gradient until converging to a minimum.

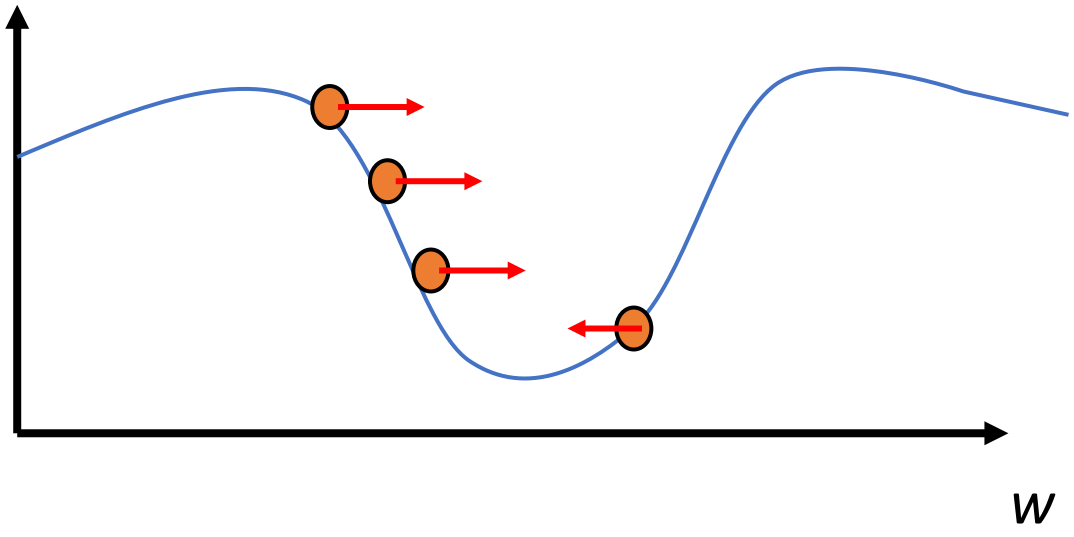
Function f(x)



Gradient Descent

We can leverage the value of the derivative at that point: I calculate the derivative and move in the opposite direction of the gradient until converging to a minimum.

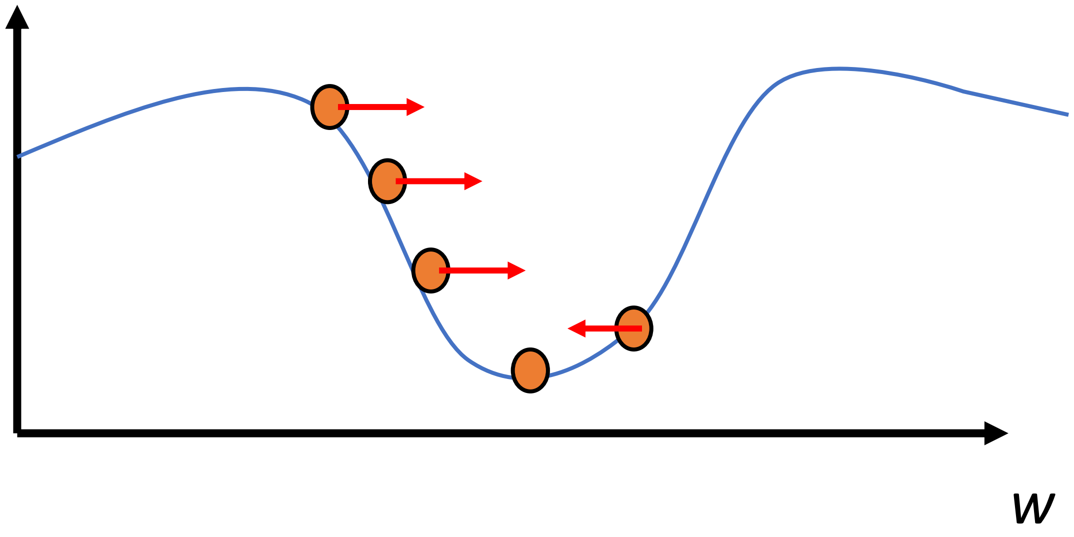
Function f(x)



Gradient Descent

We can leverage the value of the derivative at that point: I calculate the derivative and move in the opposite direction of the gradient until converging to a minimum.

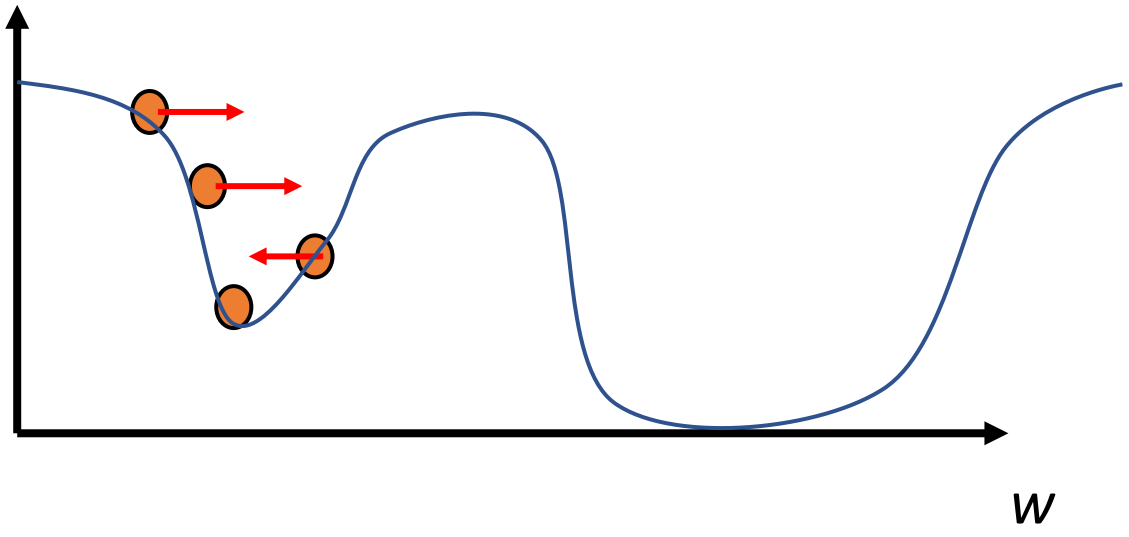
Function f(x)



Gradient Descent

The gradient descent find only the local minimum not the global one.

Function f(x)



Gradient Descent

Gradient Descent:

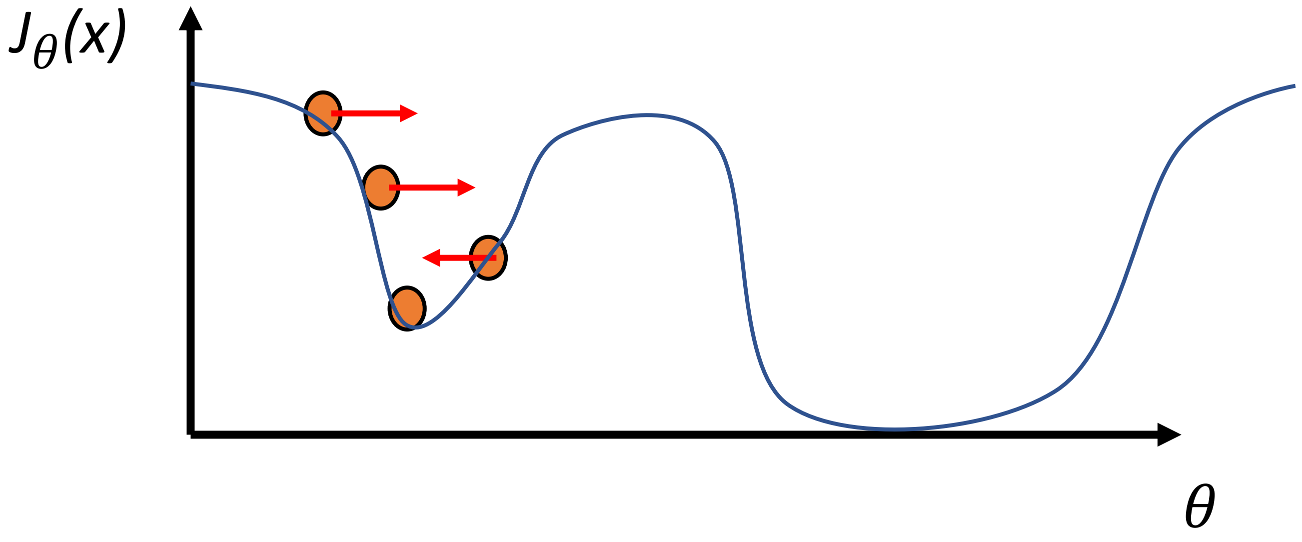
1. Initialize parameters randomly.
2. For each input/output pair (x, y):
   - -Calculate the model's predicted output, y’.
   - - Calculate the error, e = y - y'.
   - - Use the error to update weights through the derivative.

$$w_{new} = w_{old} - \alpha \frac{dJ(x)}{dw}$$

3. Repeat until the error is minimized.

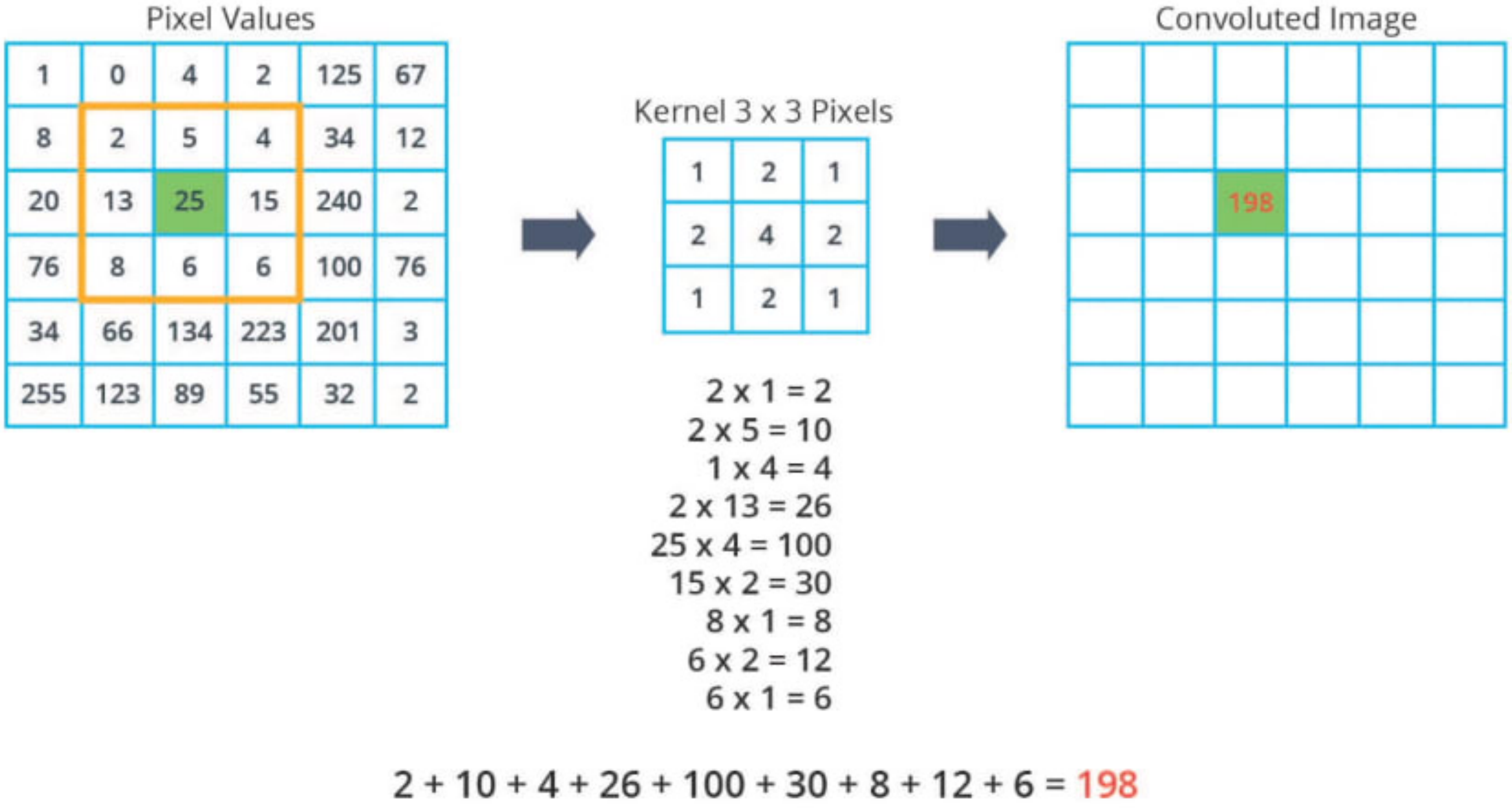
Gradient Descent

$$\boxed{\theta_{\text{new}} = \theta_{\text{old}} - \alpha \frac{d\, J_\theta(x)}{d\theta}}$$



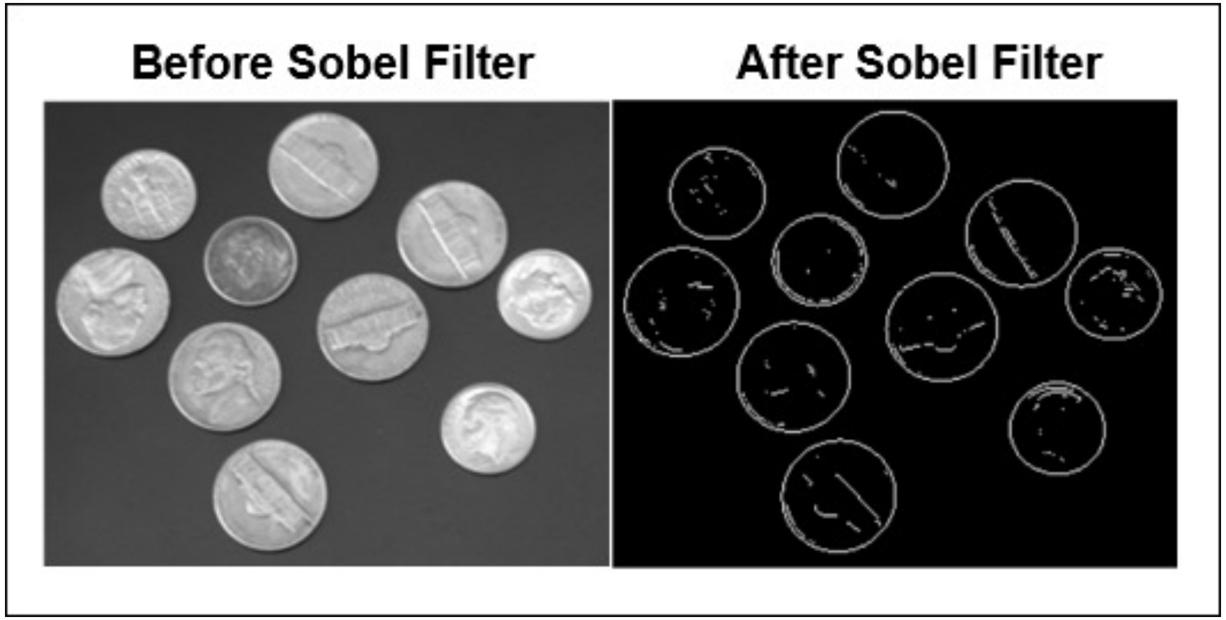
Convoluzione 2D

È una operazione fondamentale nelle reti neurali artificiali, e anch’essa causa una certa approssimazione dei dati di input che vengono trasformati.



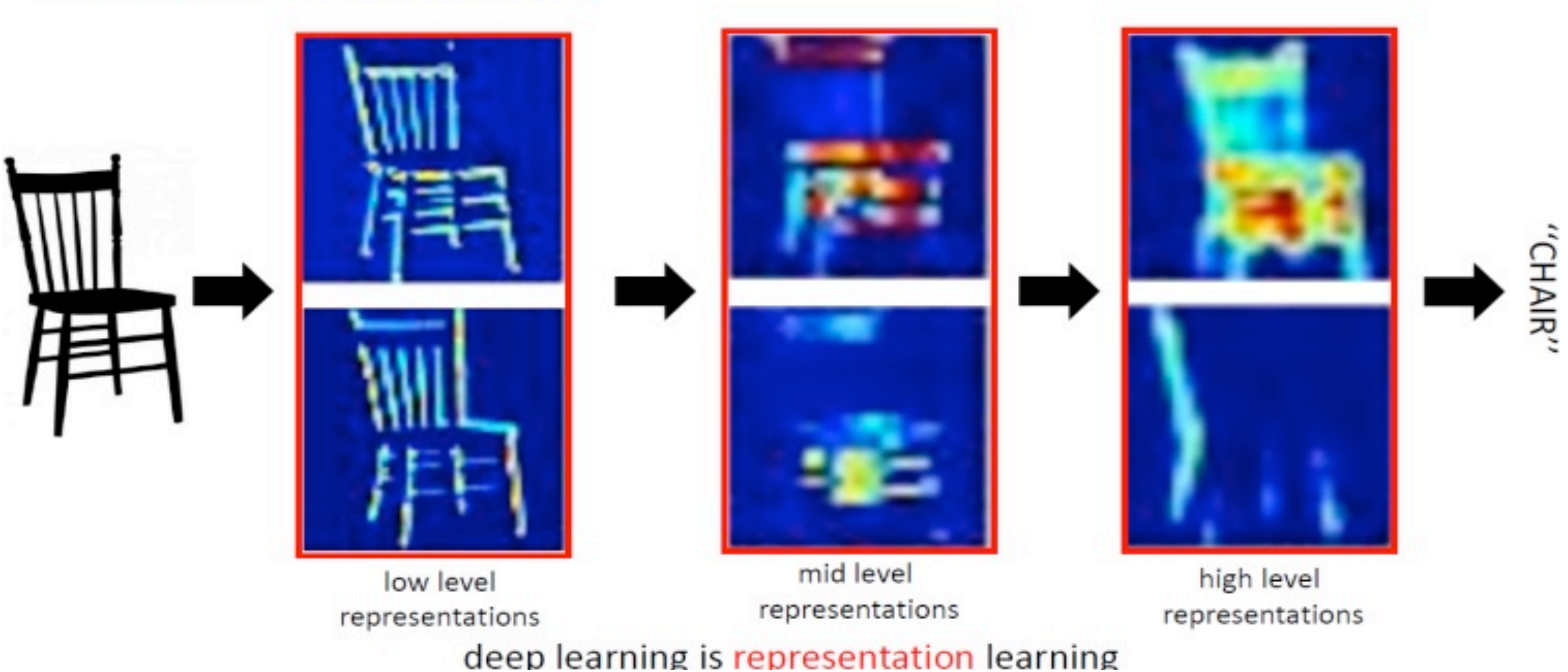
Convoluzione 2D

The purpose of convolution is to highlight the features that are most relevant to us.



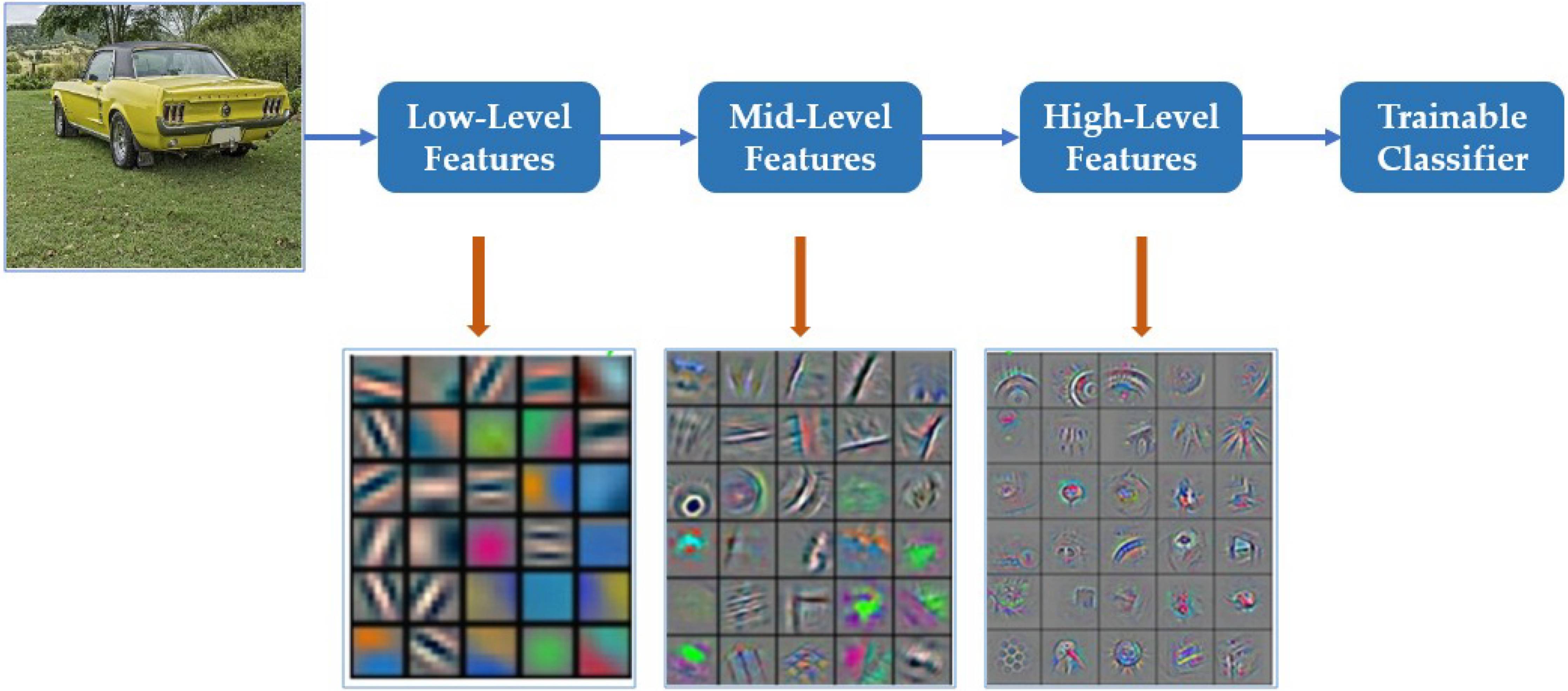
2D Convolution

CNNs learn the convolution parameters (i.e., kernel values) which extract meaning features from the (training) data.

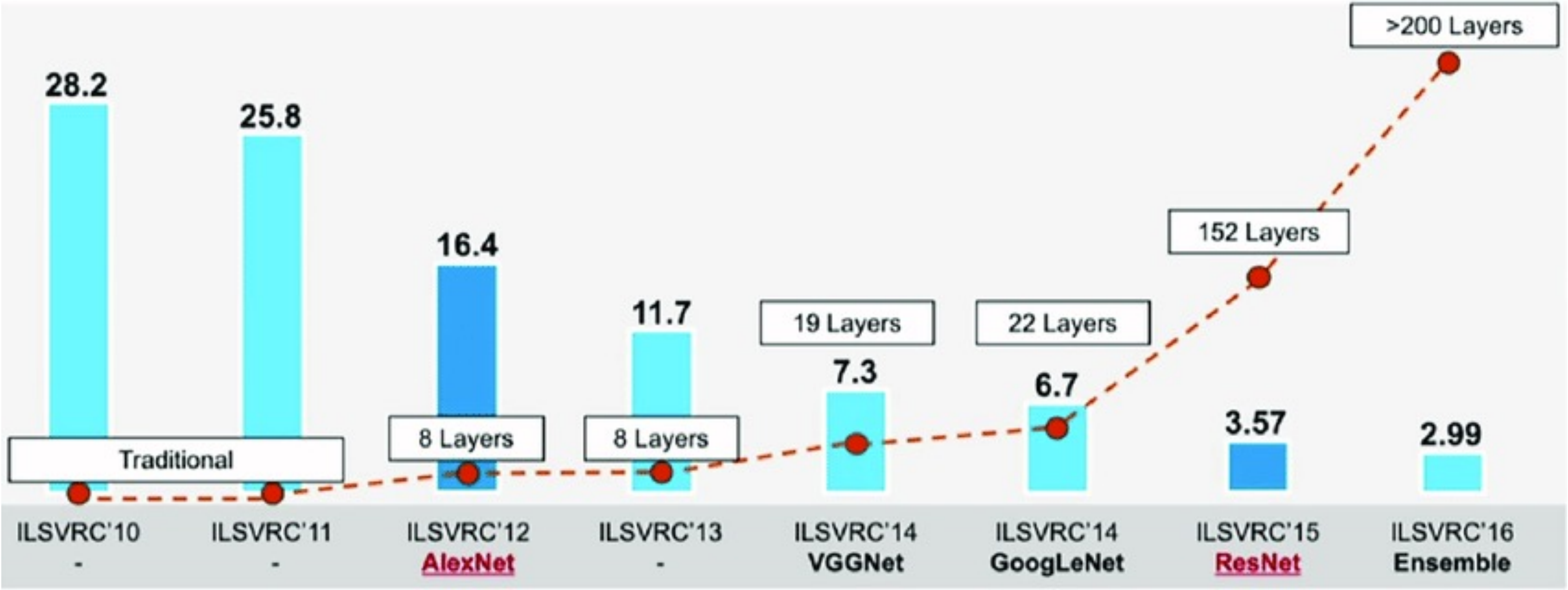


Convoluzione 2D

The features learned in the first layers are very basic.
As we go deeper in the arch. The features represent more semantic information.

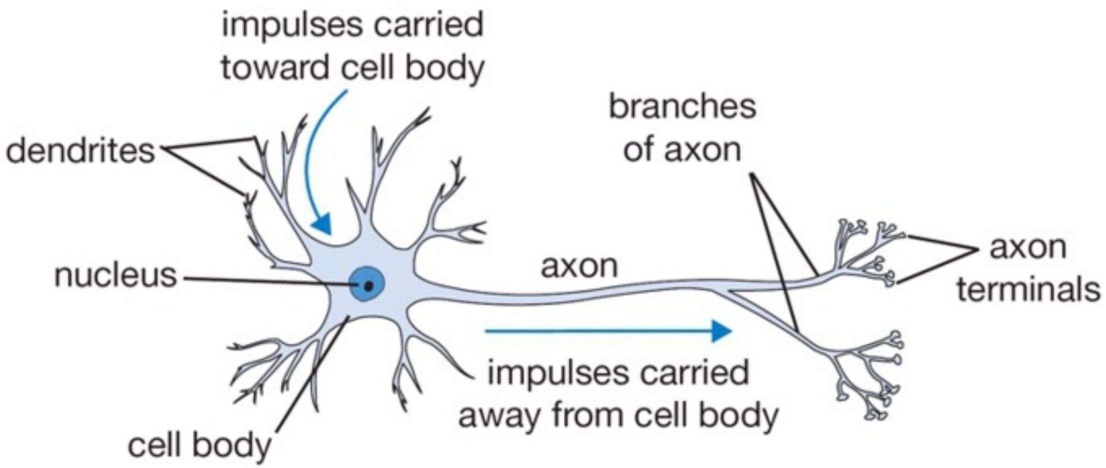


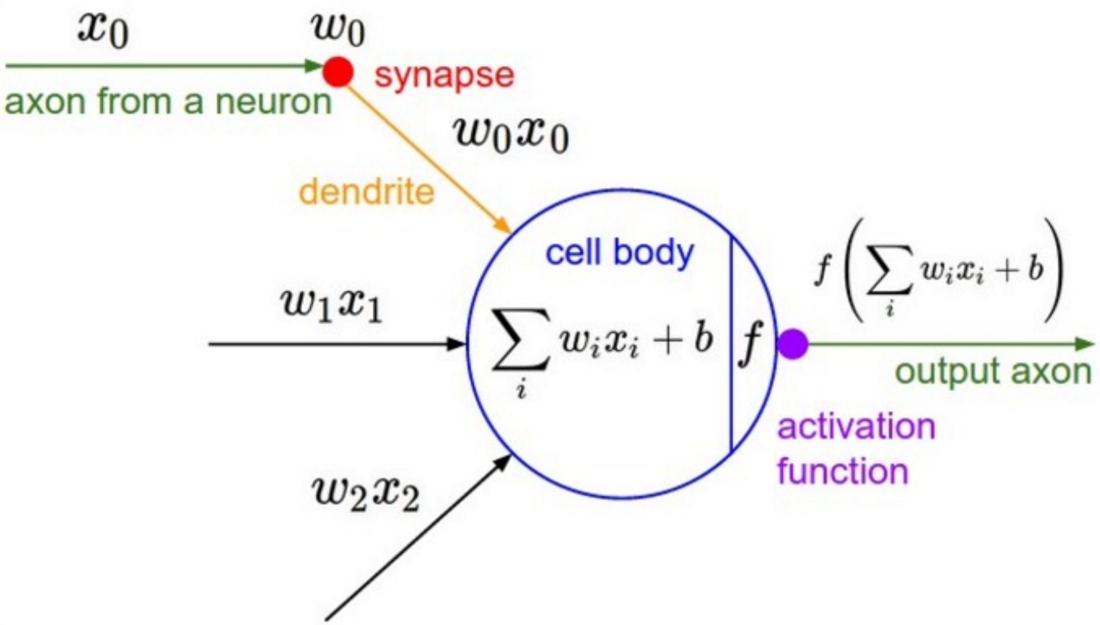
ImageNet Classification Challenge (ILSVRC)



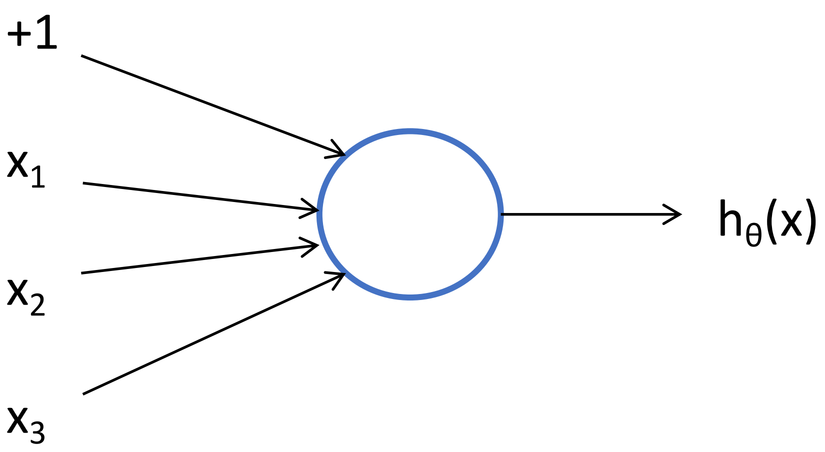
Break time!

Perceptron


impulses carried
toward cell body
dendrites
branches
of axon
nucleus
axon
axon
terminals
impulses carried
away from cell body
cell body


x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$f\left(\sum_i w_i x_i + b\right)$
$w_1 x_1$
$\sum_i w_i x_i + b$
f
output axon
activation
function
$w_2 x_2$

Perceptron



Input: $x = (x_1, x_2, x_3)$ output: $y \in \{0,1\}$ bias unit: $x_0 = b = +1$

Hypothesis: $h_\theta(x) = f(\theta^T x) = f(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3)$

Activation Function: $f(z)$ (per esempio $f(z) = \frac{1}{1+e^{-z}}$)

Activation Functions

Sigmoid function:

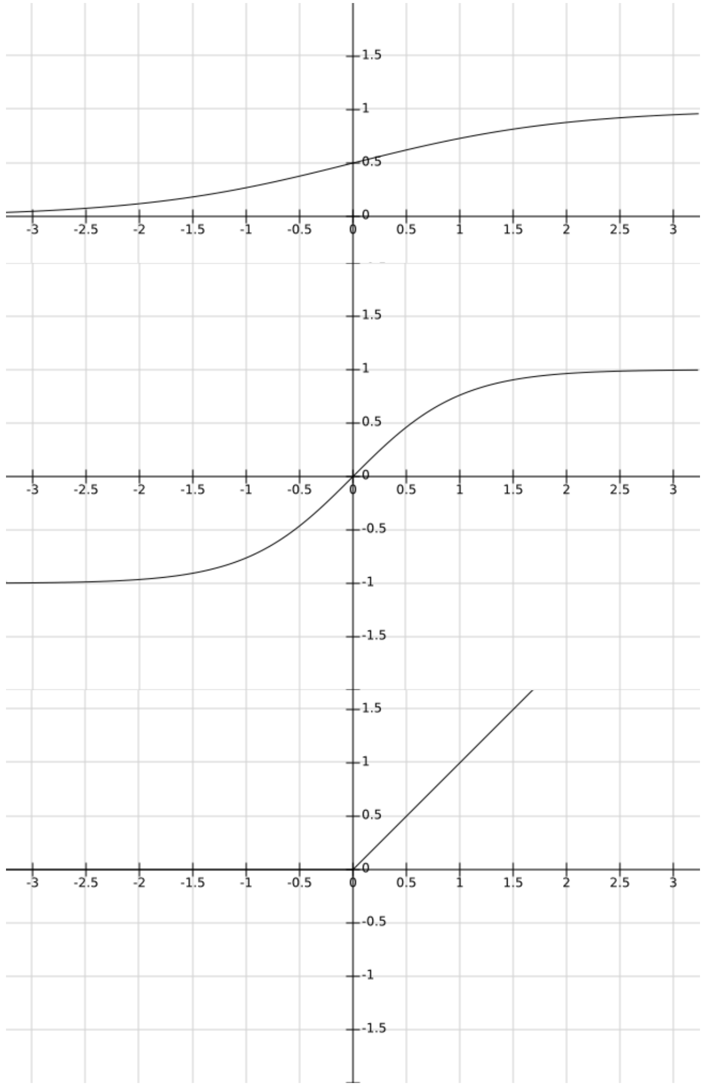
$$f(z) = \frac{1}{1 + e^{-z}}$$

Tanh(z):

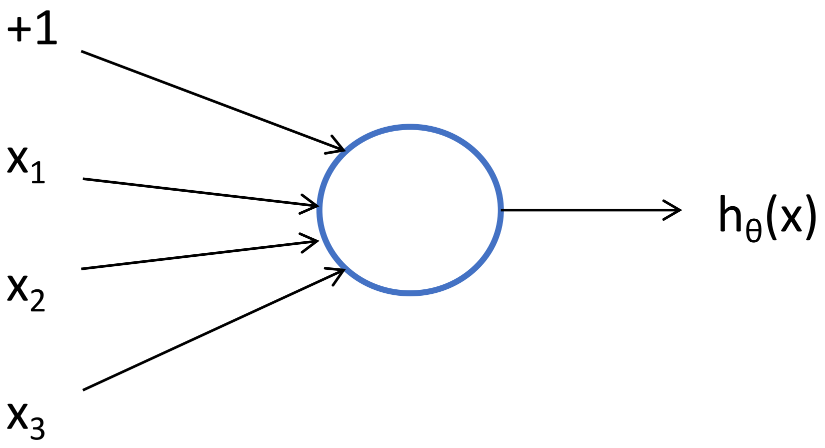
$$f(z) = \tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

ReLU (Rectified Linear Unit):

$$f(z) = \max\{0, z\}$$



Training Perceptron



Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots (x^{(i)}, y^{(i)}), \dots (x^{(m)}, y^{(m)})\}$

Loss Function: $\mathrm{J}(\theta) = -\frac{1}{m} \sum_i \left[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + (1 - y^{(i)}) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$

Application of the gradient descent

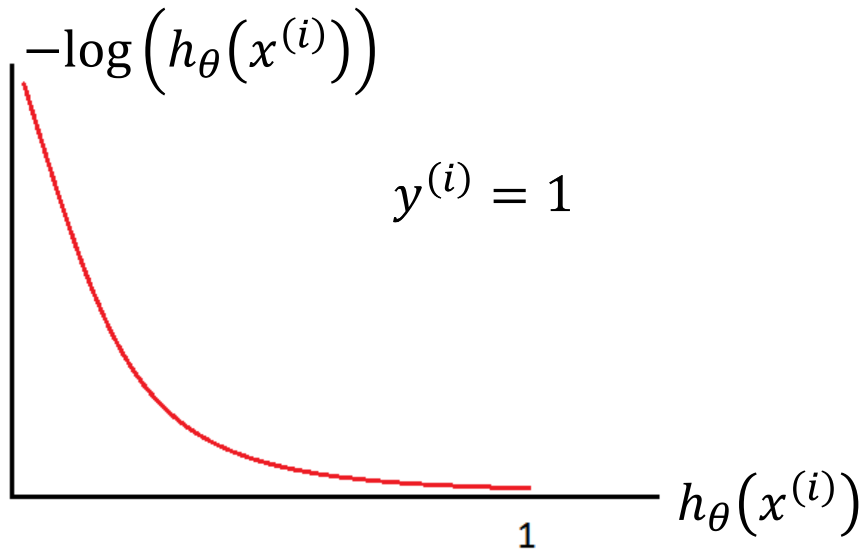
$$\theta_j = \theta_j - \alpha \frac{d\mathrm{J}(\theta)}{d\theta_j}$$

Training Perceptron

$$y^{(i)} \in \{0,1\}$$

$$P\left(y^{(i)} = 1 \middle| x^{(i)}\right) = h_\theta\left(x^{(i)}\right)$$

$$P\left(y^{(i)} = 0 \middle| x^{(i)}\right) = 1 - h_\theta\left(x^{(i)}\right)$$



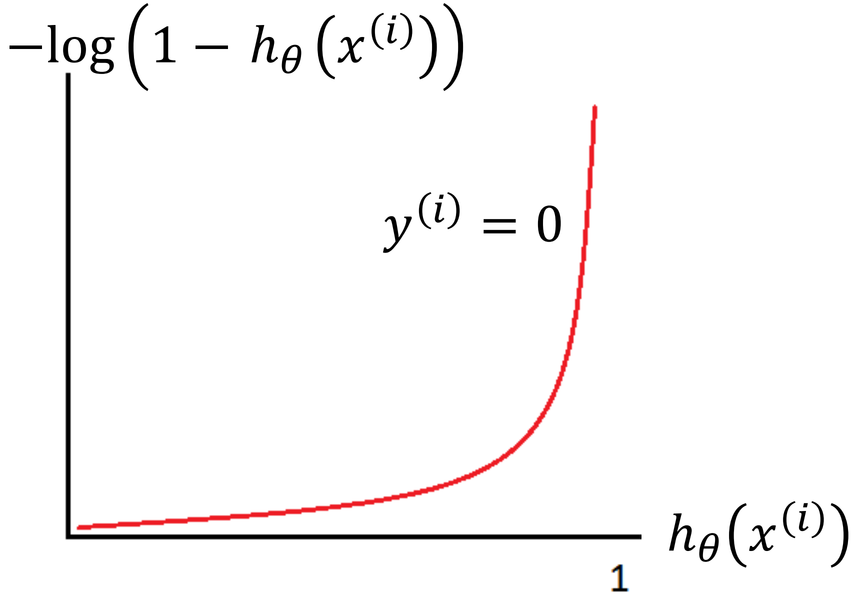
$$J(\theta) = -\frac{1}{m}\sum_i \left[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

Training Perceptron

$$y^{(i)} \in \{0,1\}$$

$$P\left(y^{(i)} = 1 \middle| x^{(i)}\right) = h_\theta\left(x^{(i)}\right)$$

$$P\left(y^{(i)} = 0 \middle| x^{(i)}\right) = 1 - h_\theta\left(x^{(i)}\right)$$



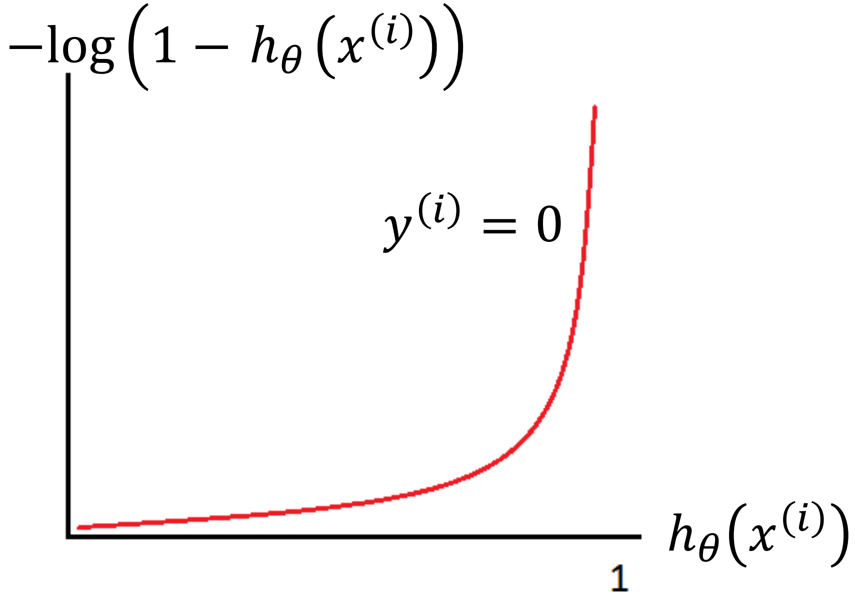
$$J(\theta) = -\frac{1}{m}\sum_i \left[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

Training Perceptron

$$y^{(i)} \in \{0,1\}$$

$$P\left(y^{(i)} = 1 \middle| x^{(i)}\right) = h_\theta\left(x^{(i)}\right)$$

$$P\left(y^{(i)} = 0 \middle| x^{(i)}\right) = 1 - h_\theta\left(x^{(i)}\right)$$

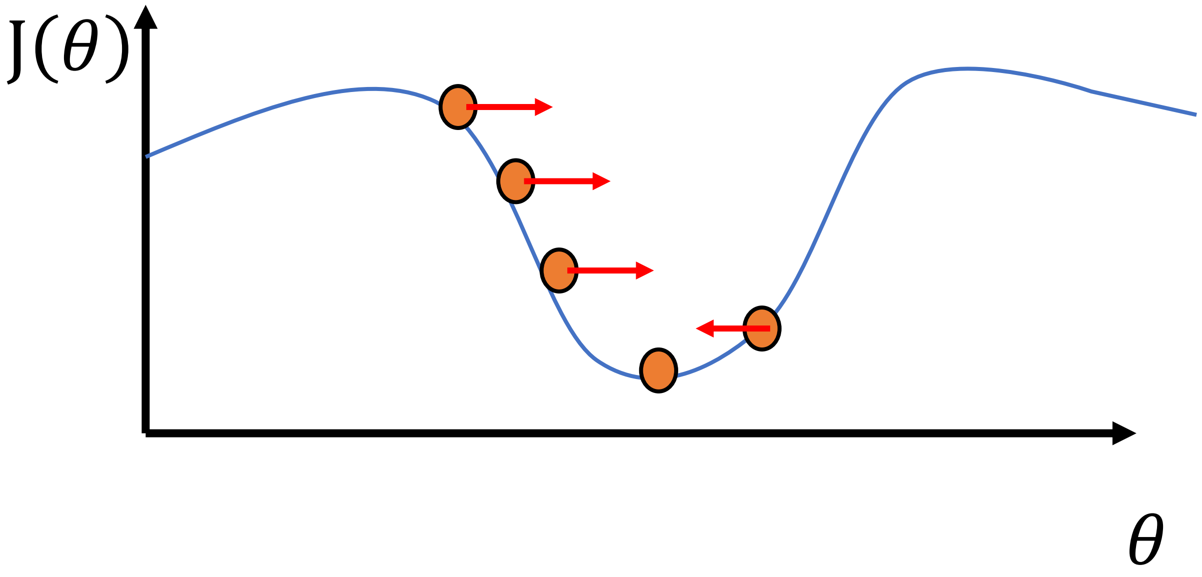


$$J(\theta) = -\frac{1}{m}\sum_i \left[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right]$$

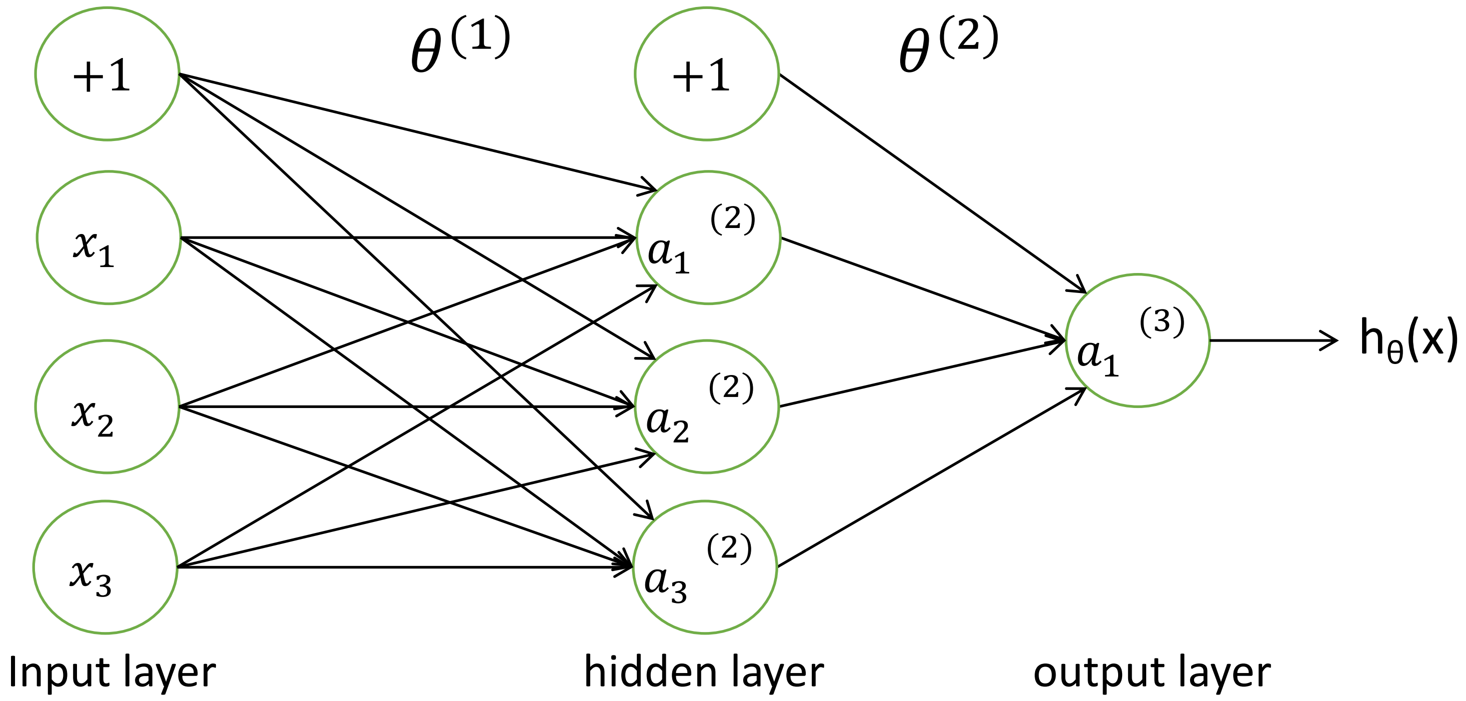
Nota anche come Binary Cross Entropy (BCE) loss

Training Perceptron

The gradient descent allows us to find the parameters θ that minimize the loss function $\mathrm{J}(\theta)$.



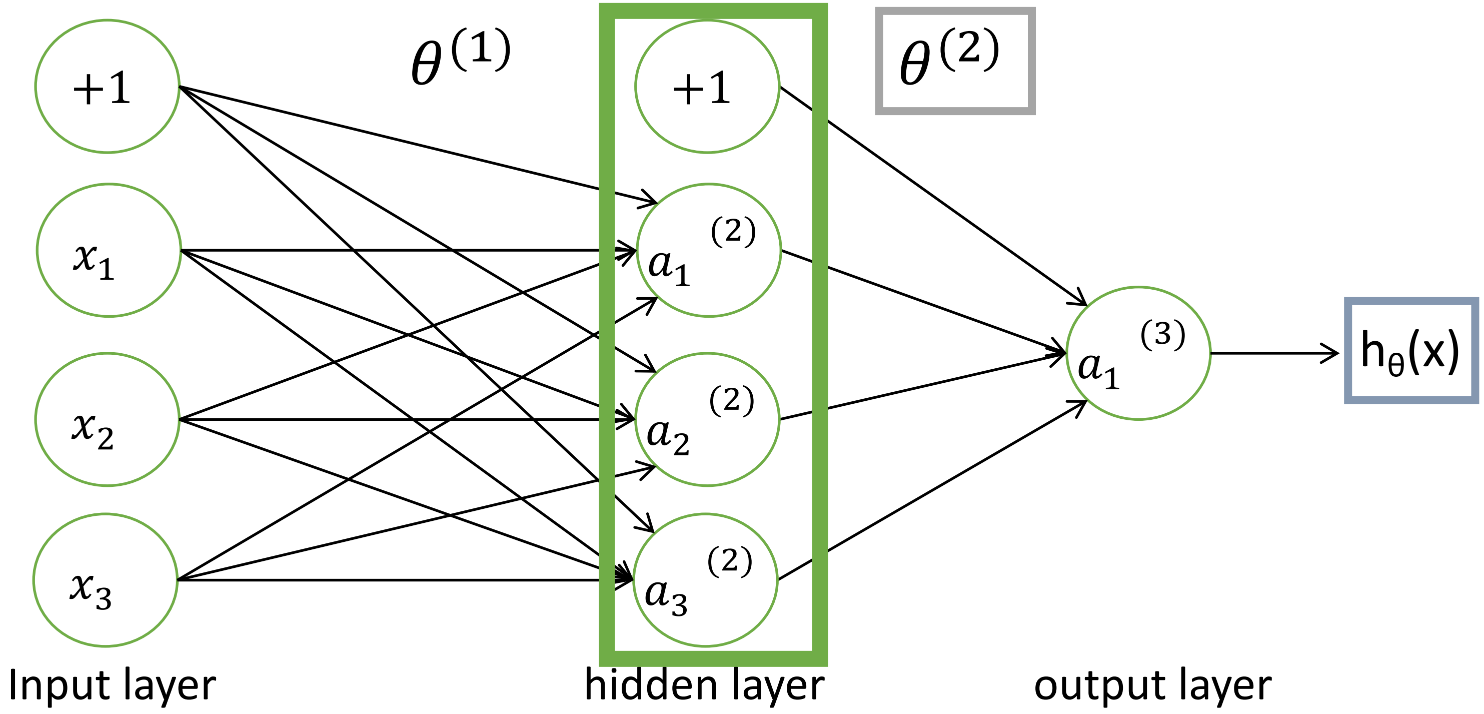
Artificial Neural Network (ANN)



$a_i^{(l)}$: activation unity i layer l

$\theta^{(l)}$: map matrix parameters $l \rightarrow l+1$

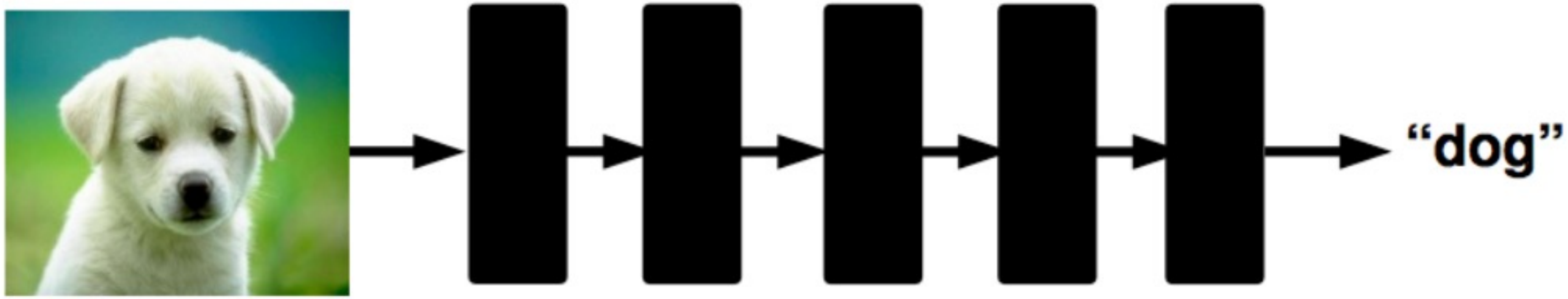
Artificial Neural Network (ANN)



$a_i^{(l)}$: activation unity i layer l
$\theta^{(l)}$: map matrix parameters $l \rightarrow l+1$

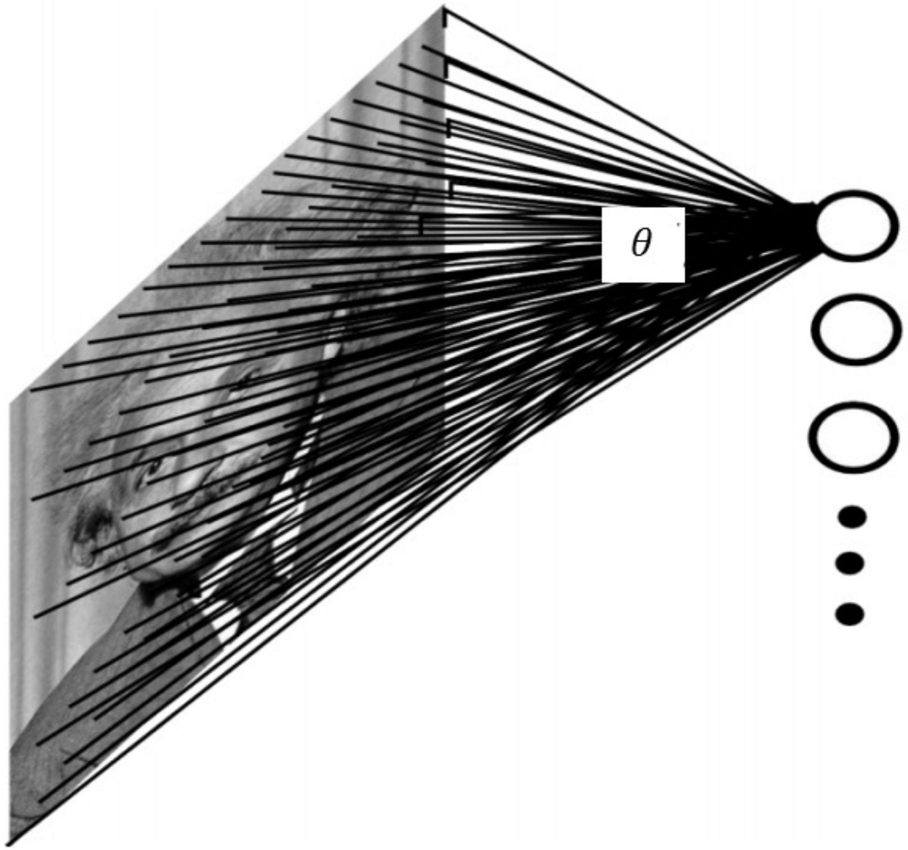
$$\underline{h_\theta(x)} = a_1^{(3)} = \underline{f(\theta_{10}^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} a_2^{(2)} + \theta_{13}^{(2)} a_3^{(2)})}$$

Convolutional Neural Networks



- CNNs (Convolutional Neural Networks) are very similar to the neural networks seen so far: they are composed of layers of connected neurons, and the connections are characterized by trainable weights.
- In CNNs, it is explicitly assumed that the inputs are images. This allows us to define architectures that take advantage of the spatial structure of the images.

Convolutional Neural Networks



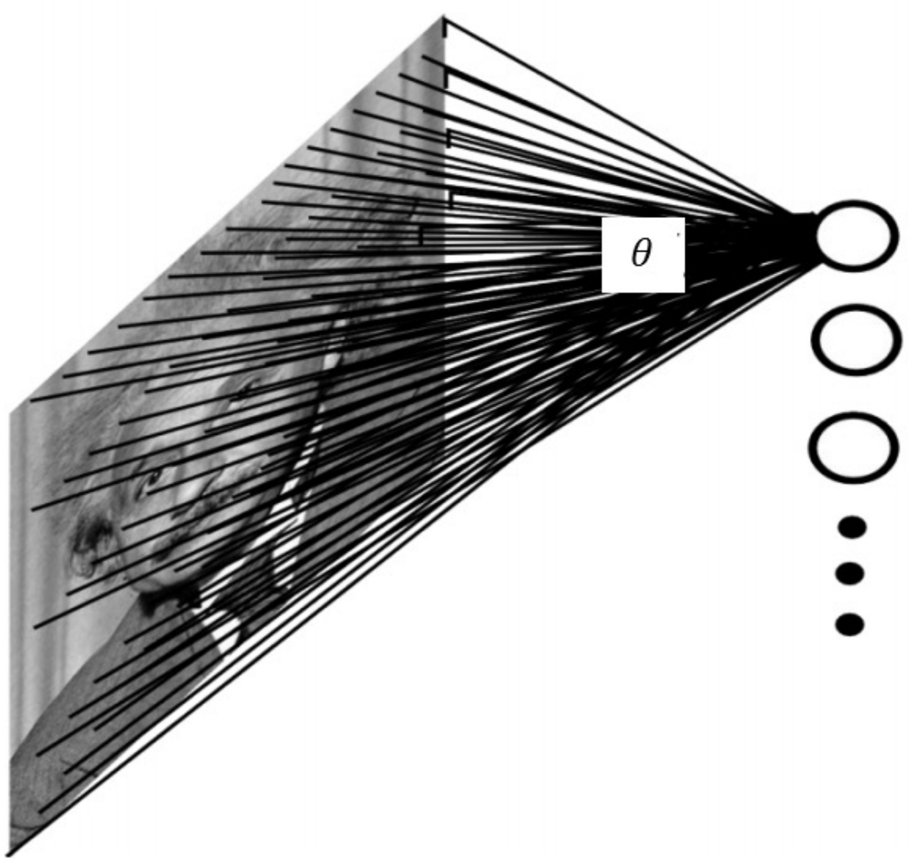
In traditional neural network each node of a layer is fully connected with each node of each adiacent layer.

Example:

Image 1000x1000

1M parameters ***per neuron***

Convolutional Neural Networks



In traditional neural network each node of a layer is fully connected with each node of each adiacent layer.

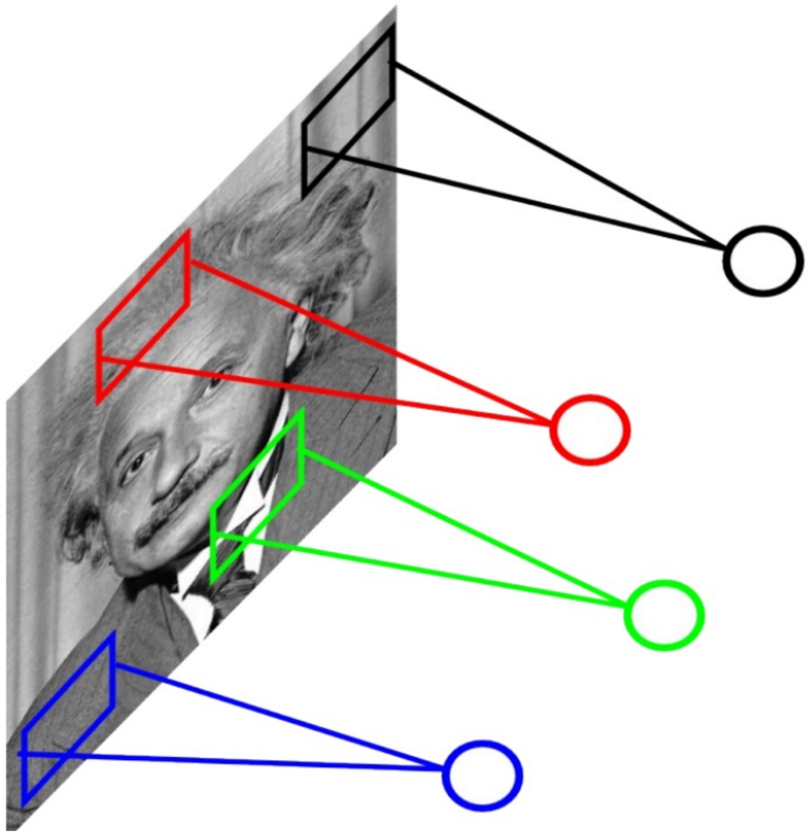
Example:

Image 1000x1000

1M parameters ***per neuron***

Solutions of CNN:
- Local receptive field
- Shared weights
- Pooling Layers

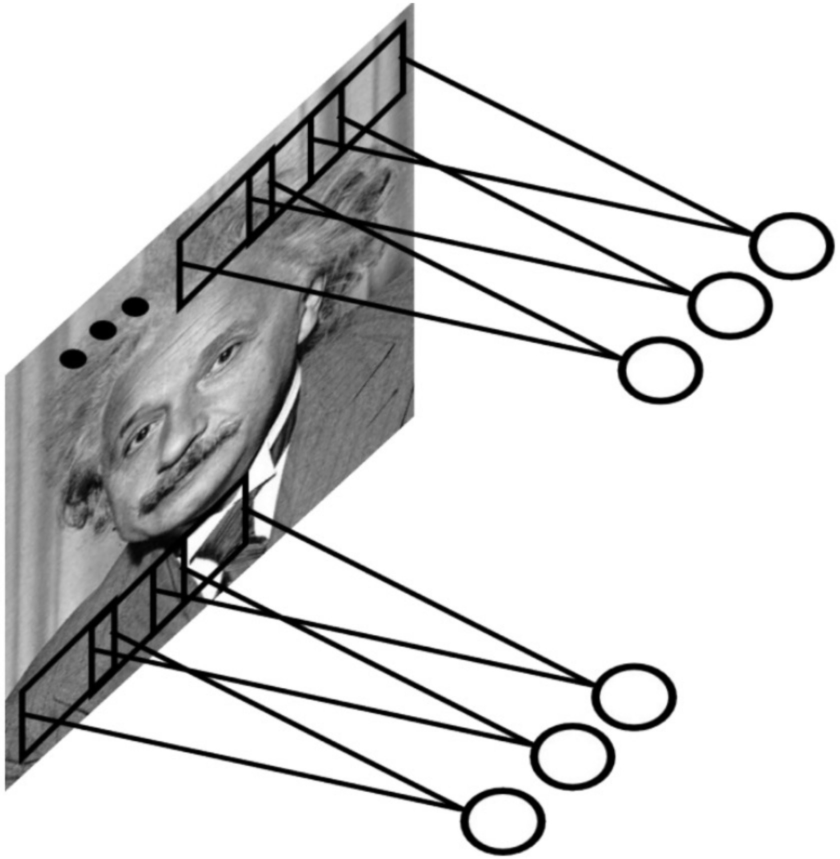
Convolutional Neural Networks



Local Receptive Field: Each neuron in a hidden layer is (completely) connected to a small region of the input (called a local receptive field), and each connection learns a weight.

Example: With a 5x5 receptive field, each neuron has 25 connections.

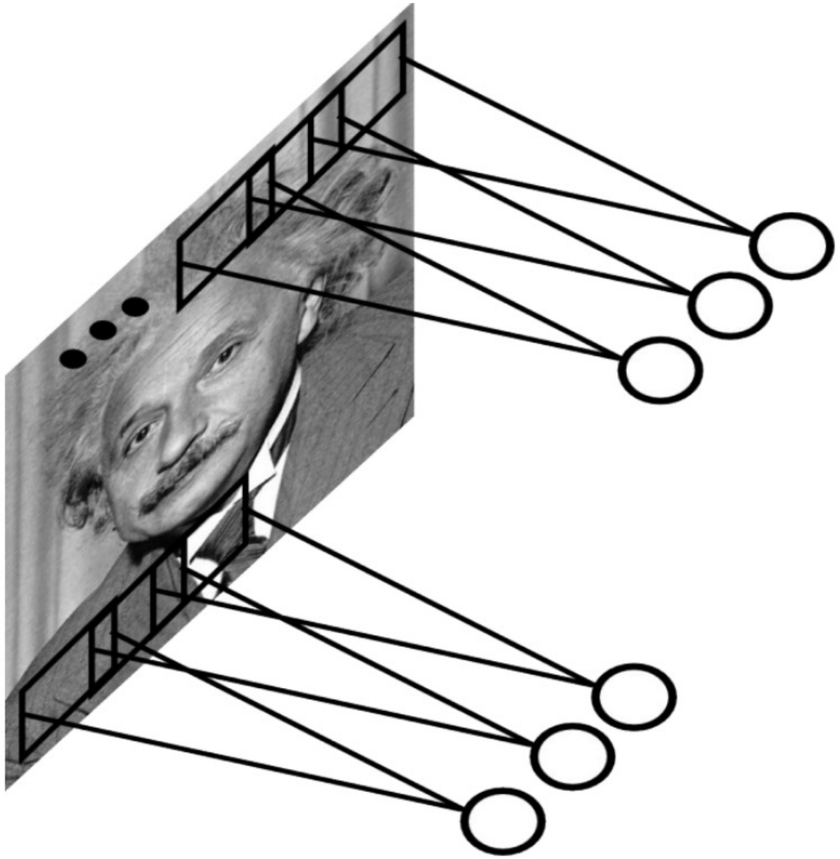
Convolutional Neural Networks



Shared Weights: Since interesting features (edges, blobs, etc.) can be found anywhere in the image, neurons in the same layer share weights.

This means that all neurons in the same layer will recognize the same feature, located at different points in the input.

Convolutional Neural Networks



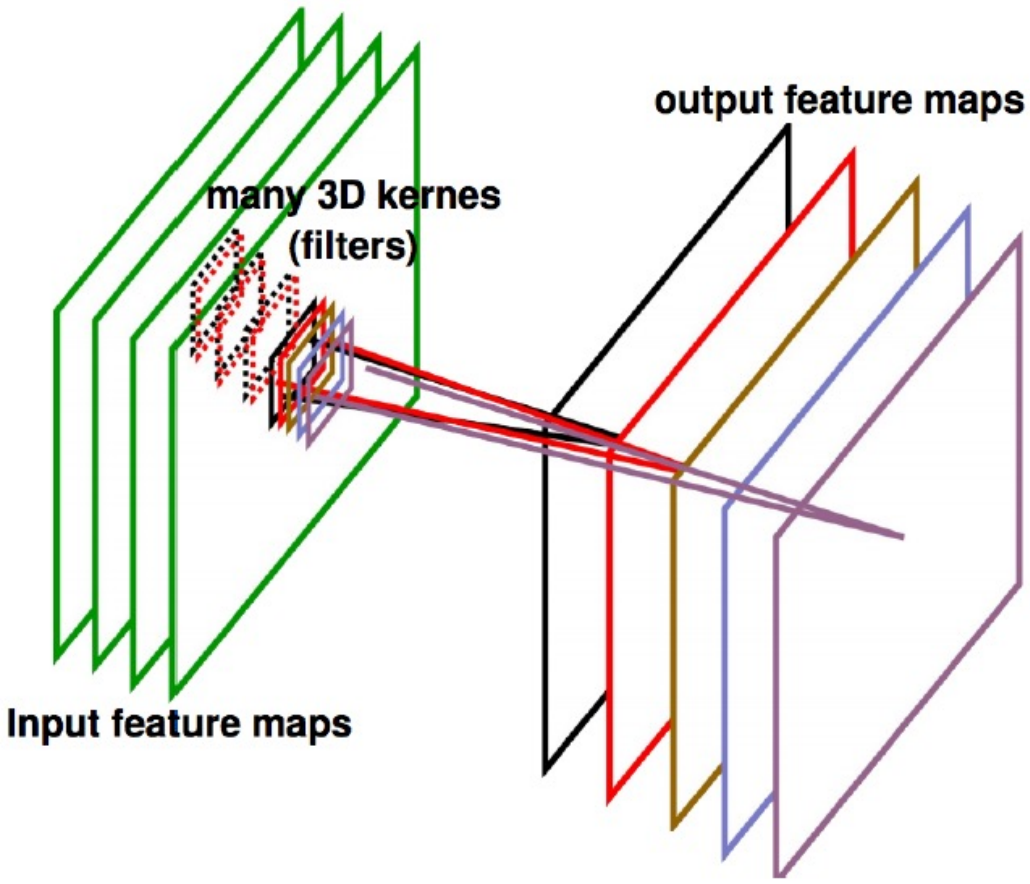
Shared Weights: Since interesting features (edges, blobs, etc.) can be found anywhere in the image, neurons in the same layer share weights.

This means that all neurons in the same layer will recognize the same feature, located at different points in the input.

Same map applies in different positions
➔ **convolution**

We call the convolution output as ***feature map***.

Convolutional Neural Networks

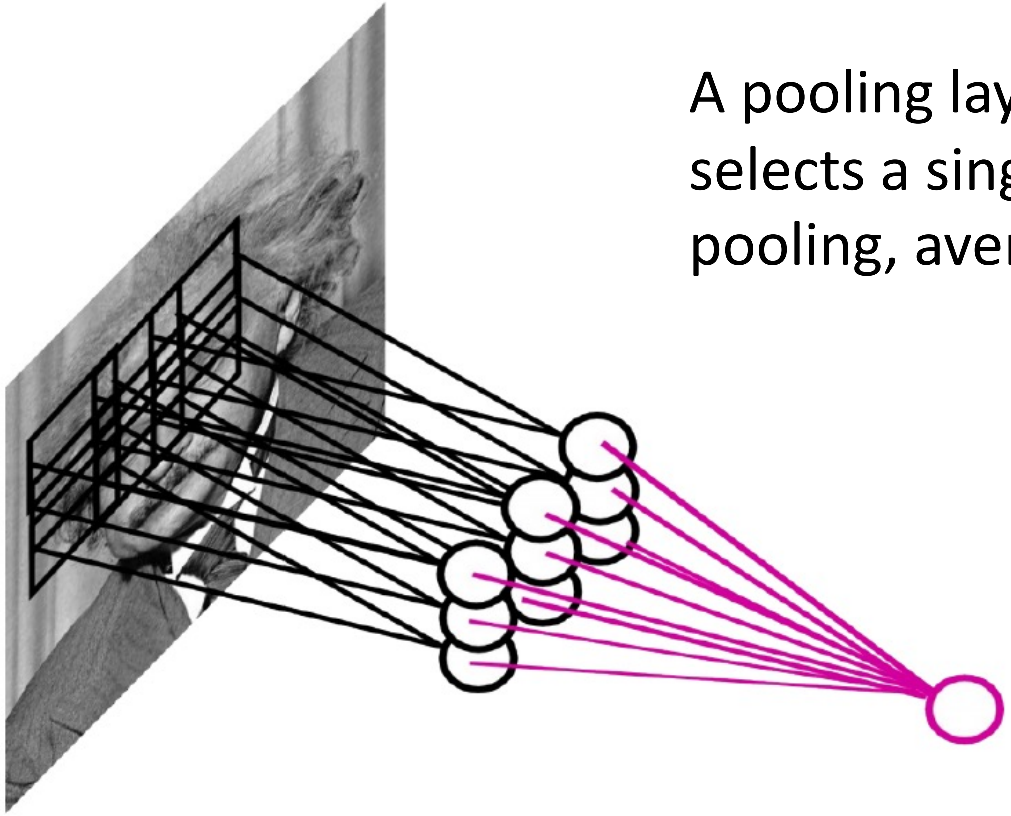


Each filter captures **a** feature present in the previous layer.

Therefore, to extract different features, we need to train multiple convolutional filters.

Each filter returns a feature map that highlights different characteristics.

Convolutional Neural Networks

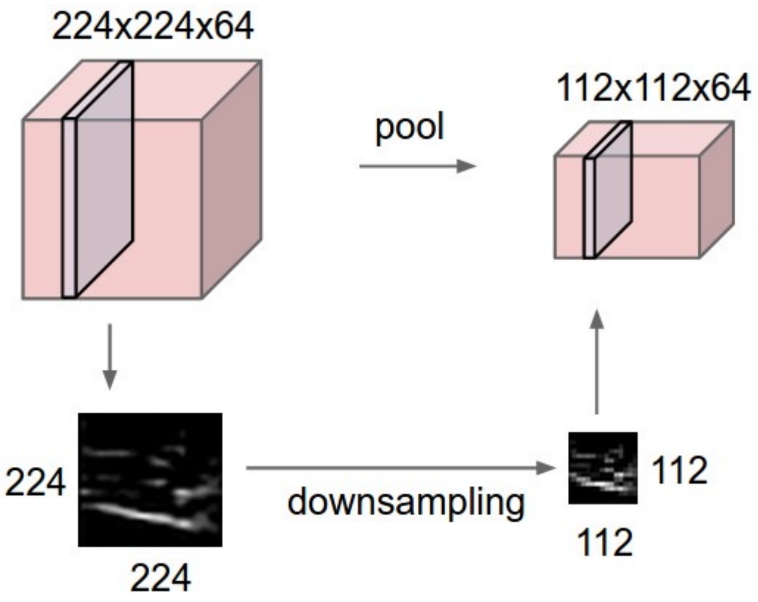


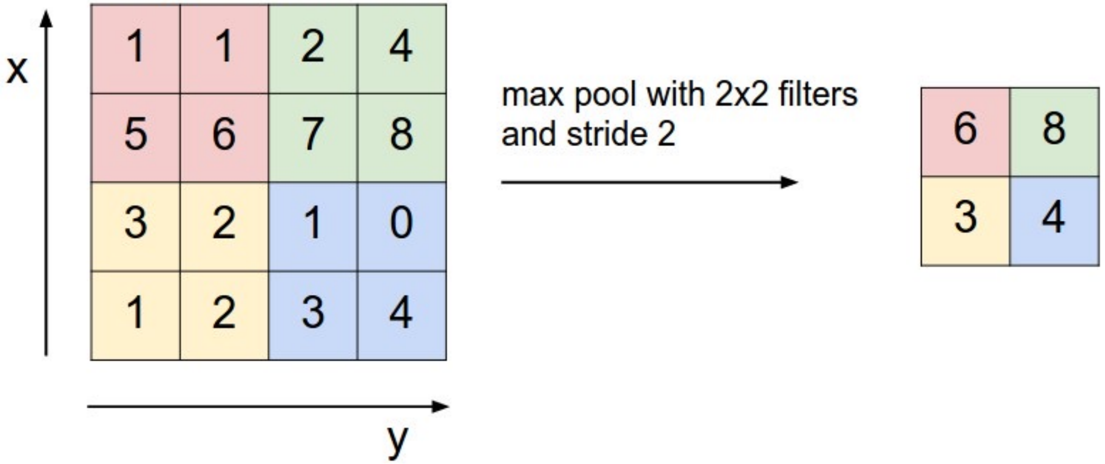
CNNs also use pooling layers positioned immediately after the convolutional layers.

A pooling layer divides the input into regions and selects a single representative value (max-pooling, average-pooling).

- Reduces computations in the subsequent layers
- Increases the robustness of features with respect to spatial position.

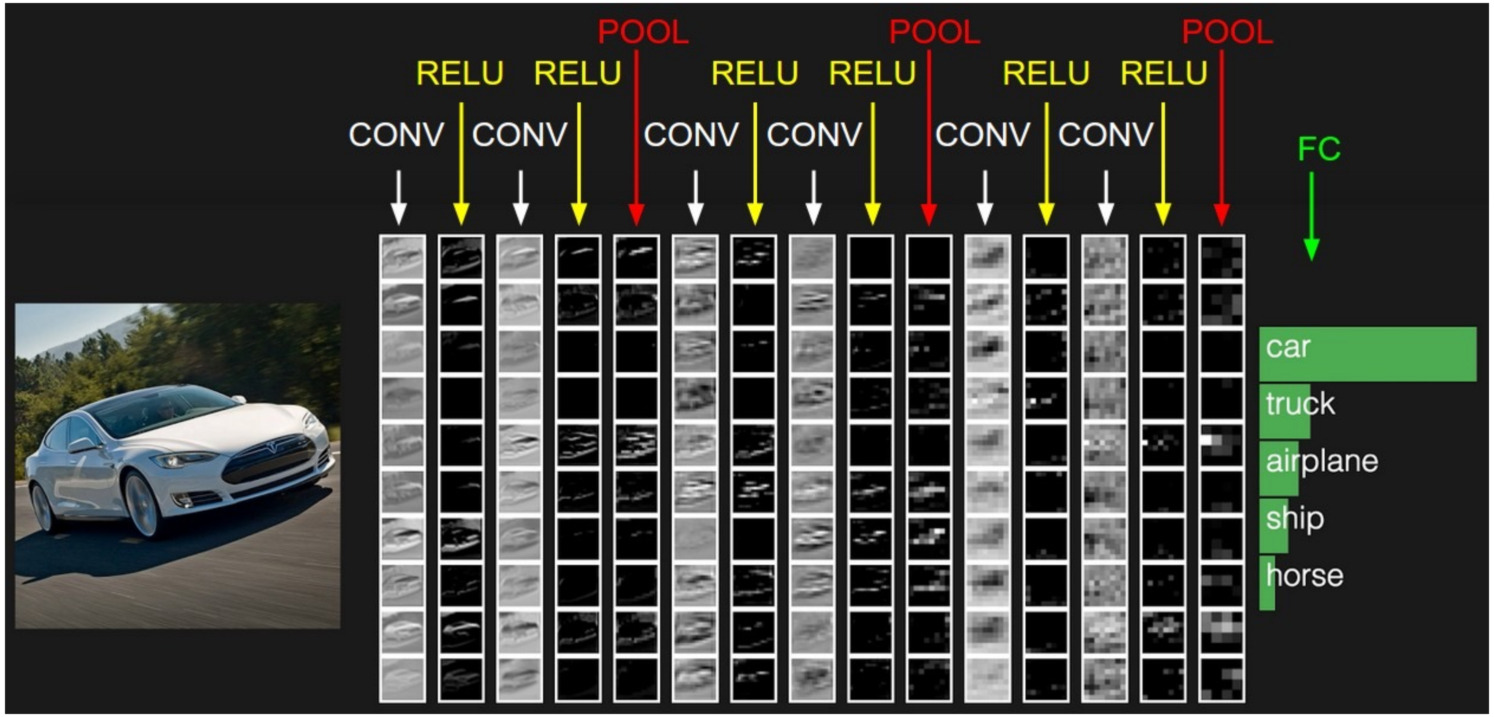
Convolutional Neural Networks





Pooling subsamples spatially each input feature map.

Convolutional Neural Networks



The last layer consists of a **fully connected** (FC) layer, and its output has a dimension equal to the number of classes.
Therefore, the last layer provides a score for each of the existing classes.

Convolutional Neural Networks

Donahue, Jeff, et al. ***"Long-term recurrent convolutional networks for visual recognition and description."*** *arXiv preprint arXiv:1411.4389* (2014).

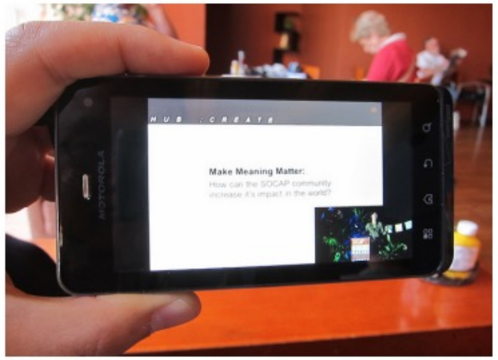
A female tennis player in action on the court.

A group of young men playing a game of soccer

A man riding a wave on top of a surfboard.

A baseball game in progress with the batter up to plate.

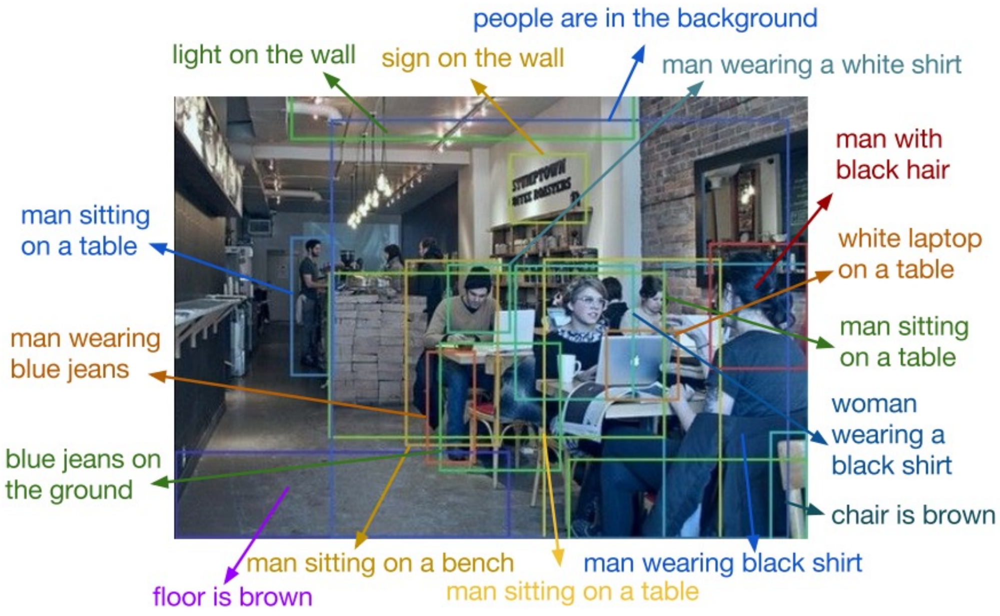
A brown bear standing on top of a lush green field.

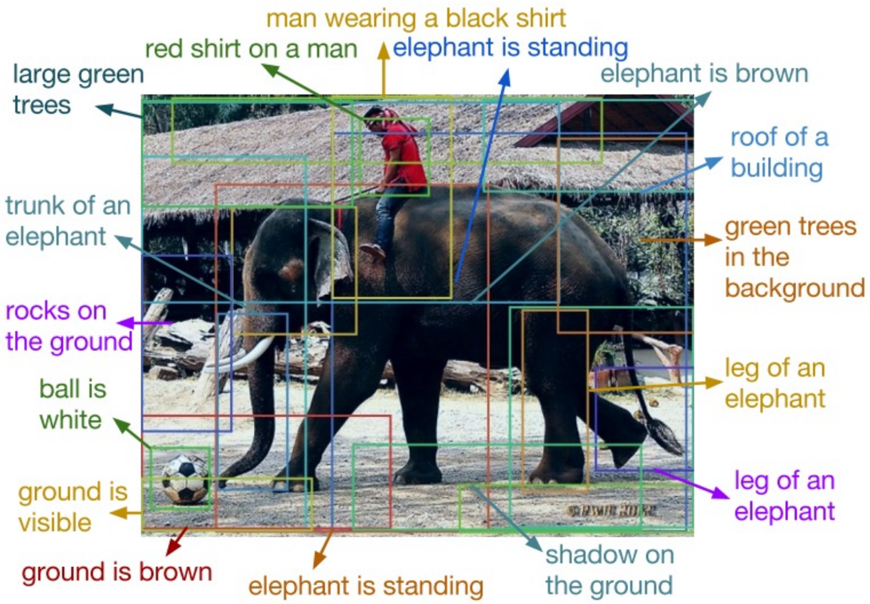


A person holding a cell phone in their hand.

Convolutional Neural Networks

Johnson, Justin, Andrej Karpathy, and Li Fei-Fei. ***"DenseCap: Fully Convolutional Localization Networks for Dense Captioning."*** *arXiv preprint arXiv:1511.07571* (2015).



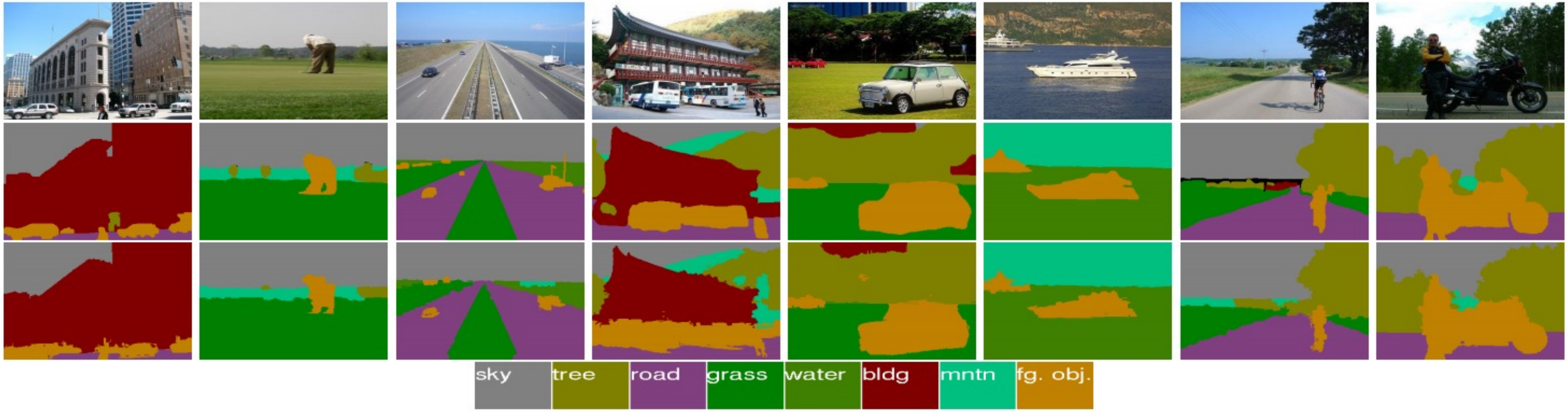


Convolutional Neural Networks

Karpathy, Andrej, Armand Joulin, and Fei Fei F. Li. ***"Deep fragment embeddings for bidirectional image sentence mapping."*** *Advances in neural information processing systems*. 2014.

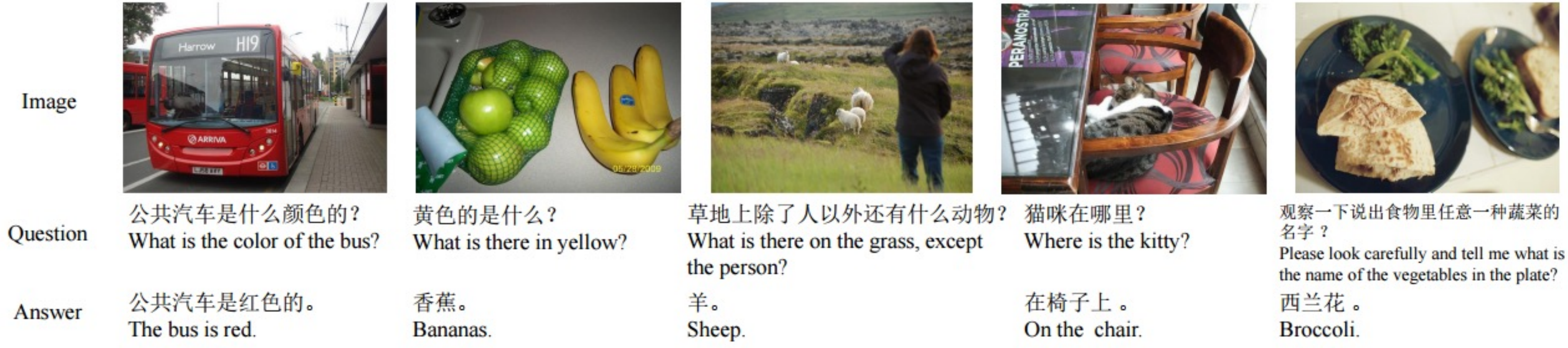
Convolutional Neural Networks

Liu, Fayao, Guosheng Lin, and Chunhua Shen. ***"CRF learning with CNN features for image segmentation."*** *Pattern Recognition* (2015).



Convolutional Neural Networks

Gao, Haoyuan, et al. ***"Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering."*** *arXiv preprint arXiv:1505.05612*(2015).



Convolutional Neural Networks

Zeiler, Matthew D., and Rob Fergus. ***"Visualizing and understanding convolutional networks."*** *Computer Vision–ECCV 2014*. Springer International Publishing, 2014. 818-833.

clarifai

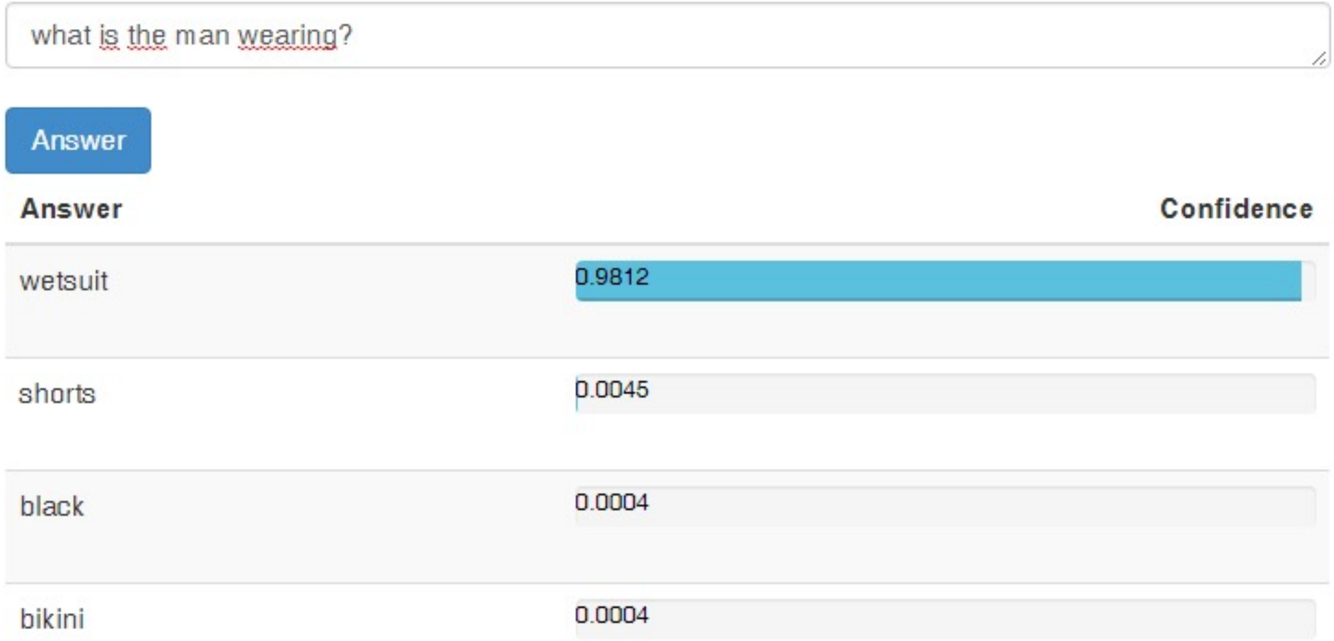
Predicted Tags

breakfast · no person · food · delicious · dawn · plate · homemade · nutrition · bread · lunch

Similar Images

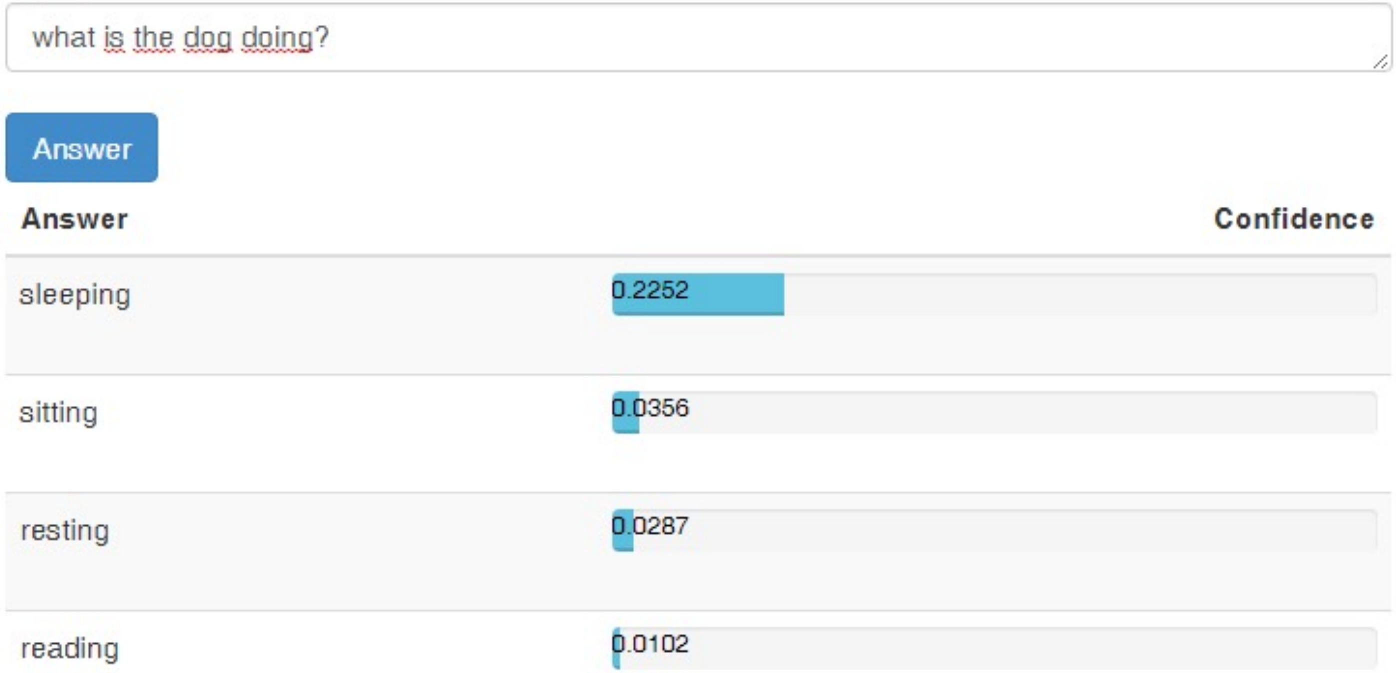
Convolutional Neural Networks

Stanislaw Antol , Aishwarya Agrawal , Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh ***"Visual Question Answering."*** (2015).



Convolutional Neural Networks

Stanislaw Antol , Aishwarya Agrawal , Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh ***"Visual Question Answering."*** (2015).



References

- *"Generative Adversarial Networks." Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. ArXiv 2014.*
- *Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).*
- *Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv:1511.06434 (2015).*
- *Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.*
- *Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." Proceedings of the IEEE international conference on computer vision. 2017.*
- *Choi, Yunjey, et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." Proceedings of the IEEE*