Artificial Intelligence: biological brain and intelligent machines

What is Artificial Intelligence and how intelligent machines work in several disciplines.

Prof. RAFFAELLA FOLGIERI
Università degli Studi di Milano, Italy
Raffaella.Folgieri@unimi.it

MACHINE LEARNING

General definition: **a science of getting computers to learn without being explicitly programmed**

Formal definitions:

- Arthur Samuel (1959): ML is the field of study that gives computers the ability to learn without being explicitly programmed.
- Tom Mitchell (1998) defined a well-posed Learning Problem as follows:

a computer program is said to learn from experience **E**

with respect to some task **T**

and some performance measure **P**,

if its performance on **T**, as measured by **P**, improves with experience **E**.

MACHINE LEARNING

- In traditional computer science every program/software is created by humans
- In complex cases, this work requires big efforts to produce small results
- In Machine Learning (that stands for «automatic learning») computers are able to learn theirselves

QUIZ

Suppose to have an agent on your PC watching which documents you do or do not mark as personal, and based on that learns how to better mark documents as personal. What is the task T in this scenario?

◯ Classifying documents as personal or not personal

◯ Watching you label documents as personal or not personal

◯ The number (or fraction) of documents correctly classified as personal/not personal

◯ None of the above – not a machine learning problem

SOLUTION

Suppose to have an agent on your PC watching which documents you do or do not mark as personal, and based on that learns how to better mark documents as personal. What is the task T in this scenario?

● Classifying documents as personal or not personal T

○ Whatching you label documents as personal or not personal E

○ The number (or fraction) of documents correctly classified as personal/not personal P

○ None of the above – not a machine learning problem

Why Machine Learning?

- Limits in traditional programming:
  - **Too many difficulties** (ex. face recognition: we can do it, but we are not able to explain how)
  - **Unsufficient (a priori) knowledge** (ex. predict the cohesion force among molecules and proteins)
  - **Automatically customized procedures** (ex. assign a score to emails or web pages following users' preferences)
- **Data availability** (discover new knowledge from large datasets) and current Industry support
- **Replace monotonous tasks** which require some intelligence (ex. Recognizing handwritten characters)
- **Progress in available algorithms** and theory from research

Examples of applications

Banks, Telco, Retail	
Perspective customs	**Effective marketing campaign**
Customer satisfaction	**Credit risk decreasing**
Selected customers	**Fraud detection**
Selected payers	**Chum rate decreasing**

Medicine/Security (biomedicine, biometrics)

Screening
Diagnosis/Prognosis
Drug discovery
Security
Face recognition
Signature/fingerprint/iris ver.
DNA identification

ICT: interfaces, Internet	
Handwriting, speech	**Fraud detection**
Brain Computer Interfaces	**Effective interface design**
Ranking, recommendation	**User experience**
Spam filtering	**User profiling**
Text categorization and translation	

Big Data: Analytics & ML

Analitics: **Deductive reasoning**

- analyzes Big Data (often related) which relies on statistics and aggregations created according to the rules imposed by those who prepare the system

ML: **Inductive reasoning**

- allows the aggregation of seemingly unrelated data from which induction takes proactive suggestions for decision making (learning from experience) that can be used in practice and to maximize the expected results.

- ML priviledges the **predictive accuracy**, Analytics is about **statistical inference**.

ML: learning approaches

Supervised learning
- Example from which the machine can learn (input and known output)

Unsupervised learning:
- Input and unknown output

Reinforced learning:
- A “premium” is given when the machine performs well

ML vs deep learning

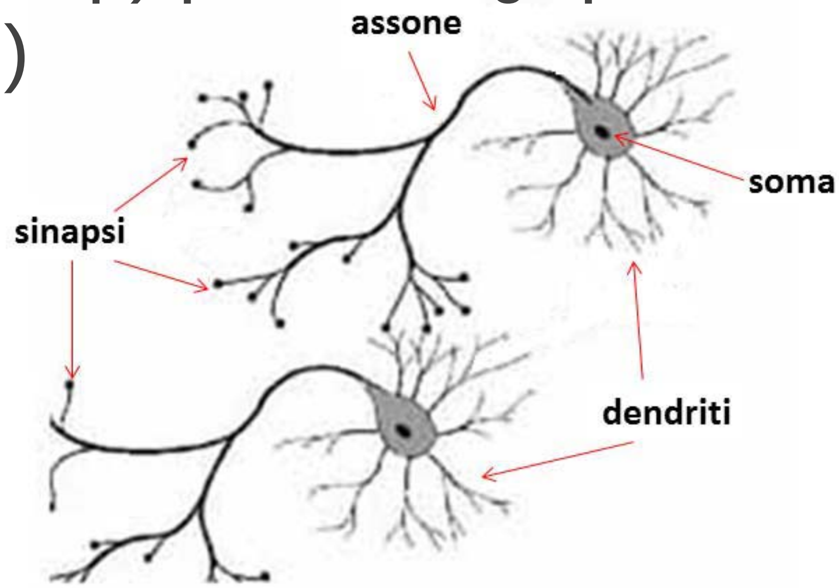
- Deep learning is a subset of machine learning, but their capabilities are different.
- If an ML algorithm returns an inaccurate prediction, then an engineer needs to step in and make adjustments. With a deep learning model, the algorithms can determine on their own if a prediction is accurate or not.
- How does deep learning work?
- A deep learning model is designed to continually analyze data with a logic structure similar to how a human would draw conclusions. To achieve this, deep learning uses a layered structure of algorithms called an **artificial neural network (ANN)**.

ML: Artificial Neural Networks

- Most used ML models
- Inspired by biology
- McCulloch and Pitts neuron and the Rosemblatt perceptron
- The XOR problem (Minsky & Papert)
- ‘80: new interest in ANN
- Multilayer perceptron

Neurons and cerebral areas

- two types of neuron action: excitatory and inhibitory
  - A neuron "fires" if the strength of an input (electrochemical) signal exceeds a certain threshold
- Cerebral Cortex divided into 52 discrete local areas (cyto-architectural map) processing specific data (Korbinian Brodmann)
- Such areas generate electrical and magnetic impulses inducing actions



Artificial Neural Network (1)

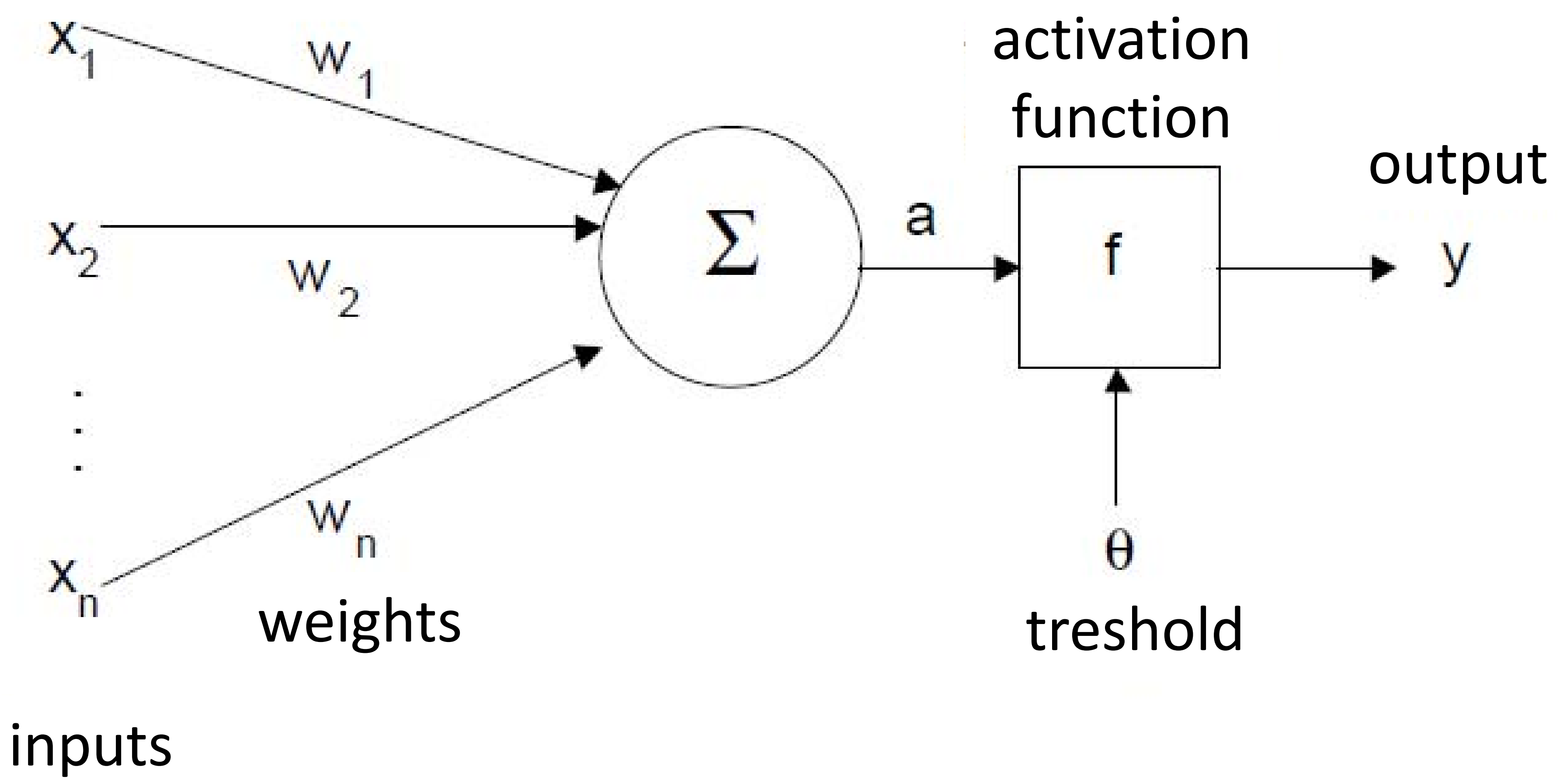
- In ANNs neurons are units elaborating, in a simple way, signals received in input from other neurons.
- Synapsis are connections amont units and the electic signal is a muber (usually between 0 and 1).
- Synapsis's conductivity is represented by the waight of the connection (ususaly indicated by **w**), obtained multiplying the signal for the weight itself, before reaching the following unit (w*x).

Artificial Neural Network (2)

In a formal way:

- n input channels $x_1, \dots, x_n$
- One weight w_i is associated to every channel (it stands for the synapsi)
- If $w_i > 0$, the channel is called *excitatory*, if $w_i < 0$, it is *inibitory*.
- The output is obtained applying an *activation function* to the weighted sum of all the inputs.

Artificial Neural Network (3)



Artificial Neural Network (4)

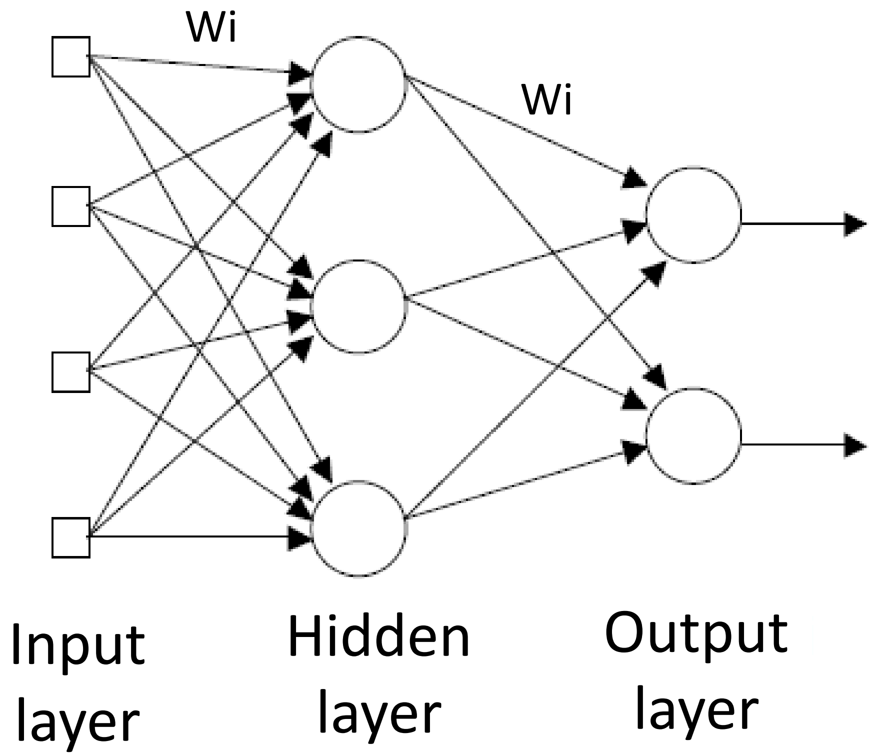
Indicating with $a = \sum_{i=1}^{n} w_i x_i$ the weighted sum of all the inputs, he have: $y = f(a) = f(\sum_{i=1}^{n} w_i x_i)$

The weighted sum is generally indicated with the word ***net***.

The mathematical functions are called ***transfer functions (***binary, linear, semi-linear, sigmoid, sign)

ANN: design (1)

General aspect of a multi-layer ANN



Weights are assigned randomly

No standard rules to define numbers of nodes and layers.

- The choice depends on data
- Trial and errors

ANN: design (2)

We will see Supervised ANN

Supervised learning

- Example from which the machine can learn (input and known output)
- vector $\underbrace{\langle x_1,\dots x_n,}_{\text{input}}\underbrace{y\rangle}_{\text{output}}$
- <number of legs, tail (Y/N), age, dog (Y/N)>

  input | output

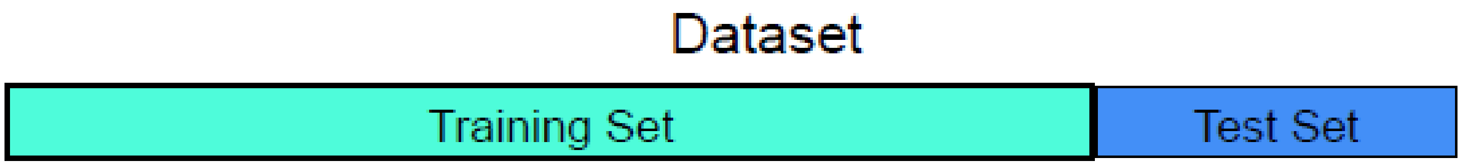
Data input: how?

Local mode: one vector/ node (too many nodes)

Distributed:one features (vector component) for each input node

Learning process

- Simplifying: weights are incremented or decremented to obtain the expected output.
- The dataset (examples) is divided into 2 subset: Training and Test (samples are randomly selected).
- Training set (used to train the ANN)
    - At least number of connection *2
- Testing set (used to verify the ANN's performance) to estimate the error
- Error: simplifying, number of obtained outputs different from the real ones



ANN: design (3) - preprocessing

- **Transform input data in data suitable for ANN (numeric)**
  - **Applying statistical or mathematical functions**
  - **Codifying textual data**
- **Selection of relevant data and removing outliers**
- Minimizing input data
  - **Feature extraction**
  - Principal components analysis
  - Waveform/Image analysis
- Codifying pre-processed data in inputs for the network

ANN: design (3) - parameters

- **Layers:** 1 for most of the problems (2 with discontinuous functions)
- **Number of neurons:**
  - Too low → data underfit (the ANN can't learn details)
  - Too many → data overfit (the ANN learns not significant details)
  - Better: start with few neurons and increment!
- **In a good training set:**
  - Data represents all the population
  - Data include the members of each class
  - Data contain variation or noise
  - Dimensions: not more than weights [(num input + 1) x num hidd. Neurons] + [(num hidd. Neurons +1) x num output]
  - Too many examples (ANN remembers the examples instead of the general idea)

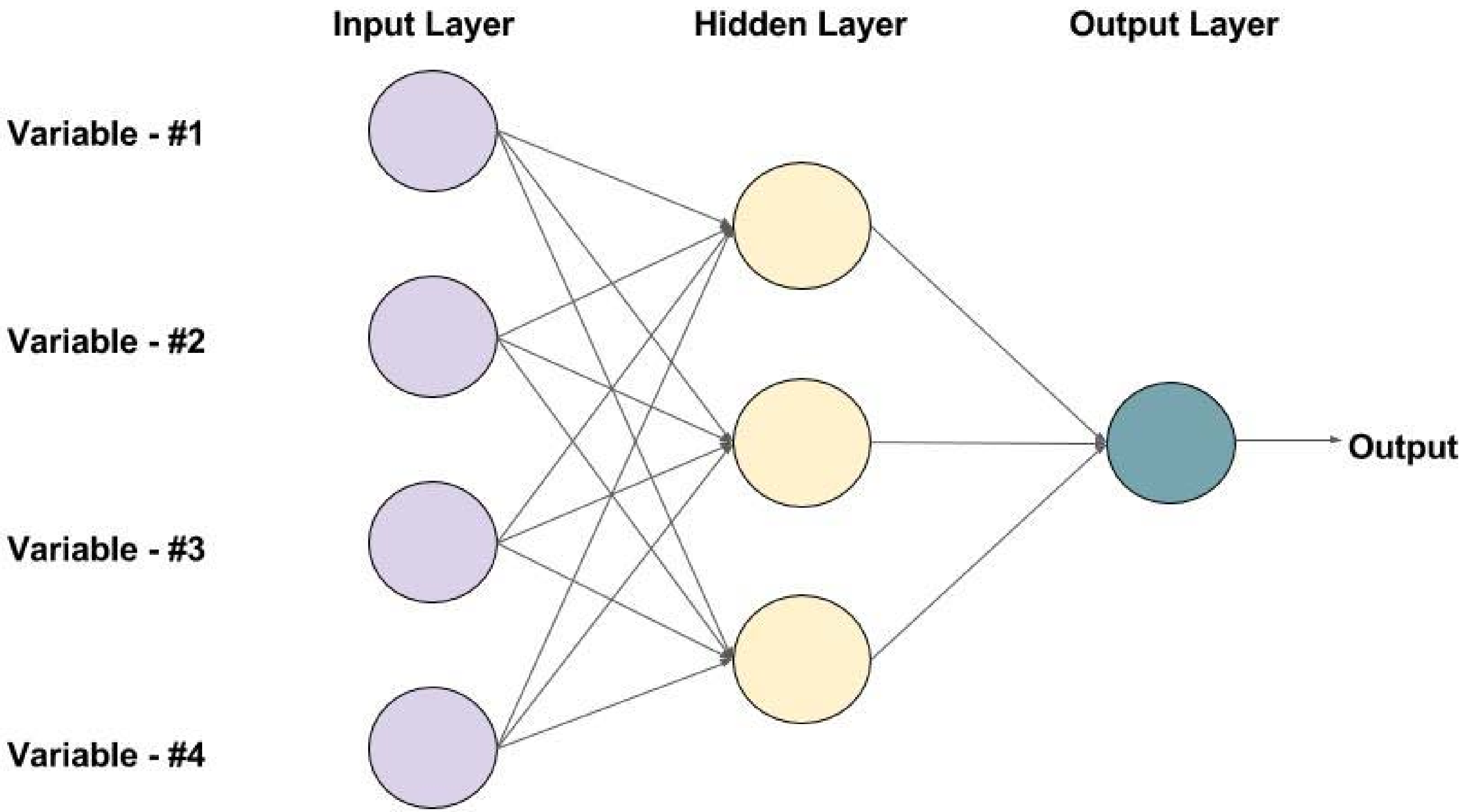
  Empirical indications

ANN: feedforward

In the *forward pass*:

- the signal flow moves from the input layer through the hidden layers to the output layer
- the decision of the output layer is measured against the ground truth labels

ANN: feedforward

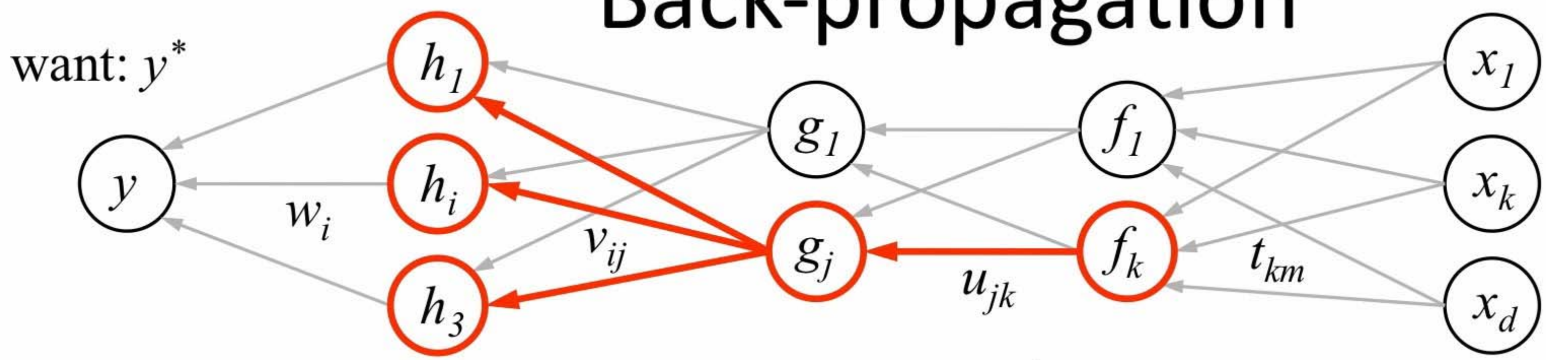


An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

ANN: backpropagation

- **The error is propagated backward**

Back-propagation



1. receive new observation $\boldsymbol{x} = [x_1 \ldots x_d]$ and target y^*
2. **feed forward:** for each unit g_j in each layer $1 \ldots L$
   compute g_j based on units f_k from previous layer: $g_j = \sigma\left(u_{j0} + \sum_k u_{jk} f_k\right)$
3. get prediction y and error $(y - y^*)$
4. **back-propagate error:** for each unit g_j in each layer $L \ldots 1$

(a) compute error on g_j

$$\underbrace{\frac{\partial E}{\partial g_j}}_{\text{should } g_j \text{ be higher or lower?}} = \sum_i \underbrace{\sigma'(h_i) v_{ij}}_{\text{how } h_i \text{ will change as } g_j \text{ changes}} \underbrace{\frac{\partial E}{\partial h_i}}_{\text{was } h_i \text{ too high or too low?}}$$

(b) for each u_{jk} that affects g_j

(i) compute error on u_{jk}

$$\frac{\partial E}{\partial u_{jk}} = \underbrace{\frac{\partial E}{\partial g_j}}_{\text{do we want } g_j \text{ to be higher/lower}} \underbrace{\sigma'(g_j) f_k}_{\text{how } g_j \text{ will change if } u_{jk} \text{ is higher/lower}}$$

(ii) update the weight

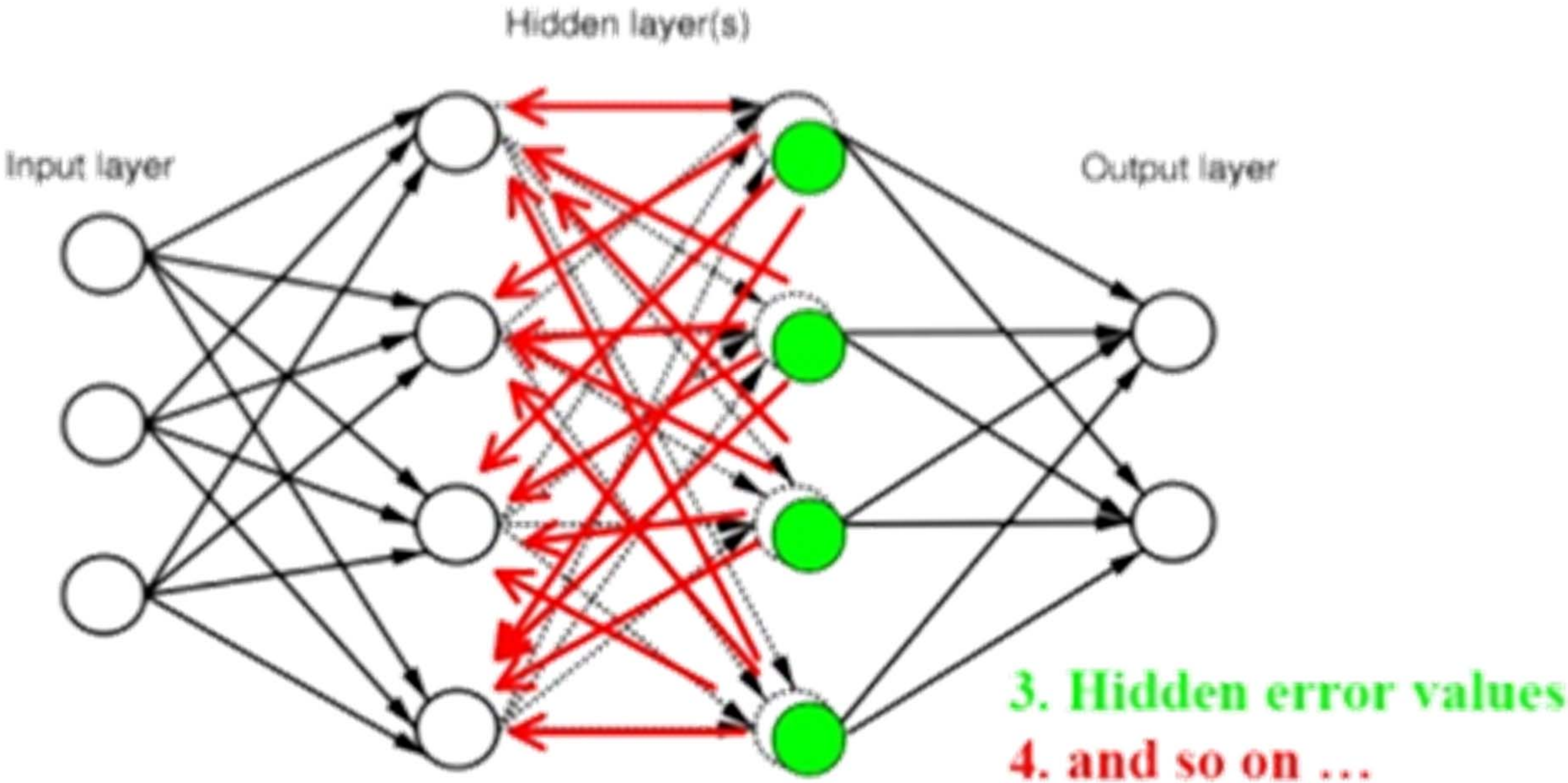
$$u_{jk} \leftarrow u_{jk} - \eta \frac{\partial E}{\partial u_{jk}}$$

ANN: backpropagation

- Using backpropagation and the chain rule of calculus, partial derivatives of the error function the various weights and biases are back-propagated through the ANN.
- That act of differentiation gives us a gradient, or a landscape of error, along which the parameters may be adjusted as they move the ANN one step closer to the error minimum.
- This can be done with any gradient-based optimization algorithm such as stochastic gradient descent.
- The network keeps playing this until the error can go no lower. This state is known as *convergence*.

ANN: backpropagation

simple

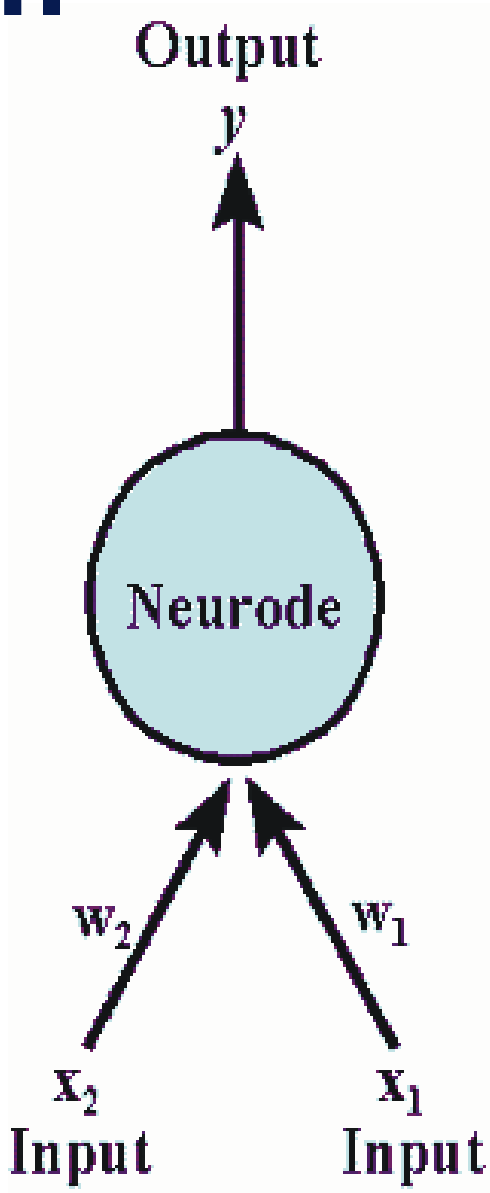


ML: Artificial Neural Networks

- ✓ML and deep learning
- ✓Most used ML models
- ✓Inspired by biology
- McCulloch and Pitts neuron and the Rosemblatt perceptron
- The XOR problem (Minsky & Papert)
- ‘80: new interest in ANN
- ✓Multilayer perceptron

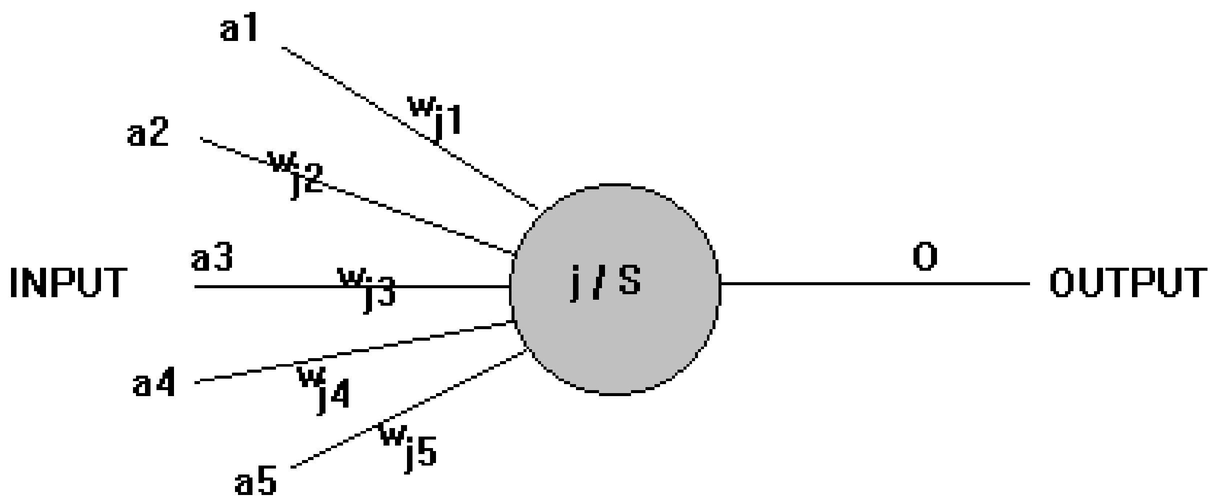
McCulloch/Pitts neuron

- 1943. It did not incorporate weighting the different inputs
- 1949, Hebb introduced weights at each of the inputs



Rosenblatt perceptron

- 1962: Rosenblatt, using the McCulloch-Pitts neuron and the findings of Hebb, went on to develop the first perceptron



Stop dreaming!

- 1969. Minsky and Papert: attacked the limitations of the perceptron.
- They showed that the perceptron could only solve linearly separable functions.
- Of particular interest was the fact that the perceptron still could not solve the XOR and NXOR functions.

Stop researching!

- As a result, very little research was done in the area until about the 1980’s.

- What happened in ‘80?

Multilayer perceptron and NN

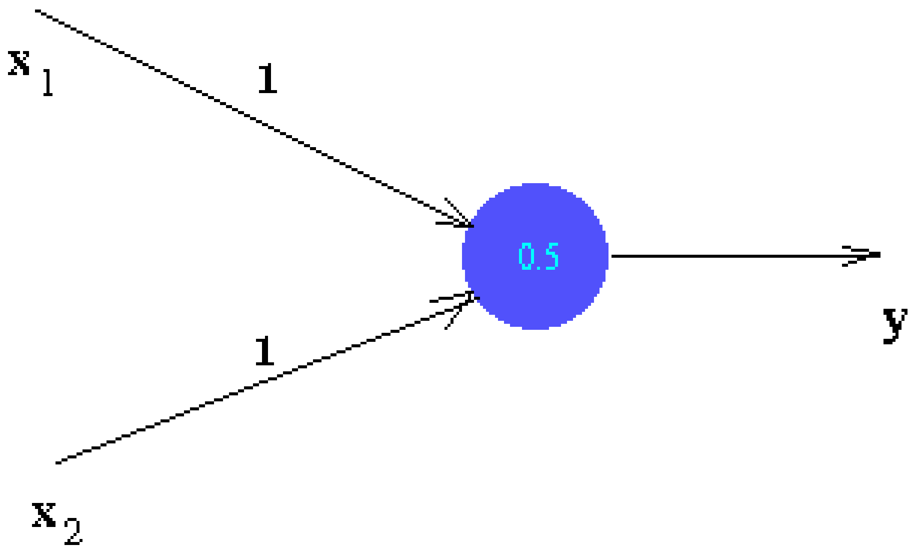
- Multilayer perceptron (Werbos 1974, Rumelhart, McClelland, Hinton 1986), also named feed forward networks.
- Creation of neural networks.
  - networks connecting the inputs of artificial neurons with the outputs of other artificial neurons. Able to solve more difficult problems, but they have grown considerably more complex.
- Powerful computers, parallel programming

The XOR problem

INPUT	DESIRED OUTPUT
0 0	0
0 1	1
1 0	1
1 1	0

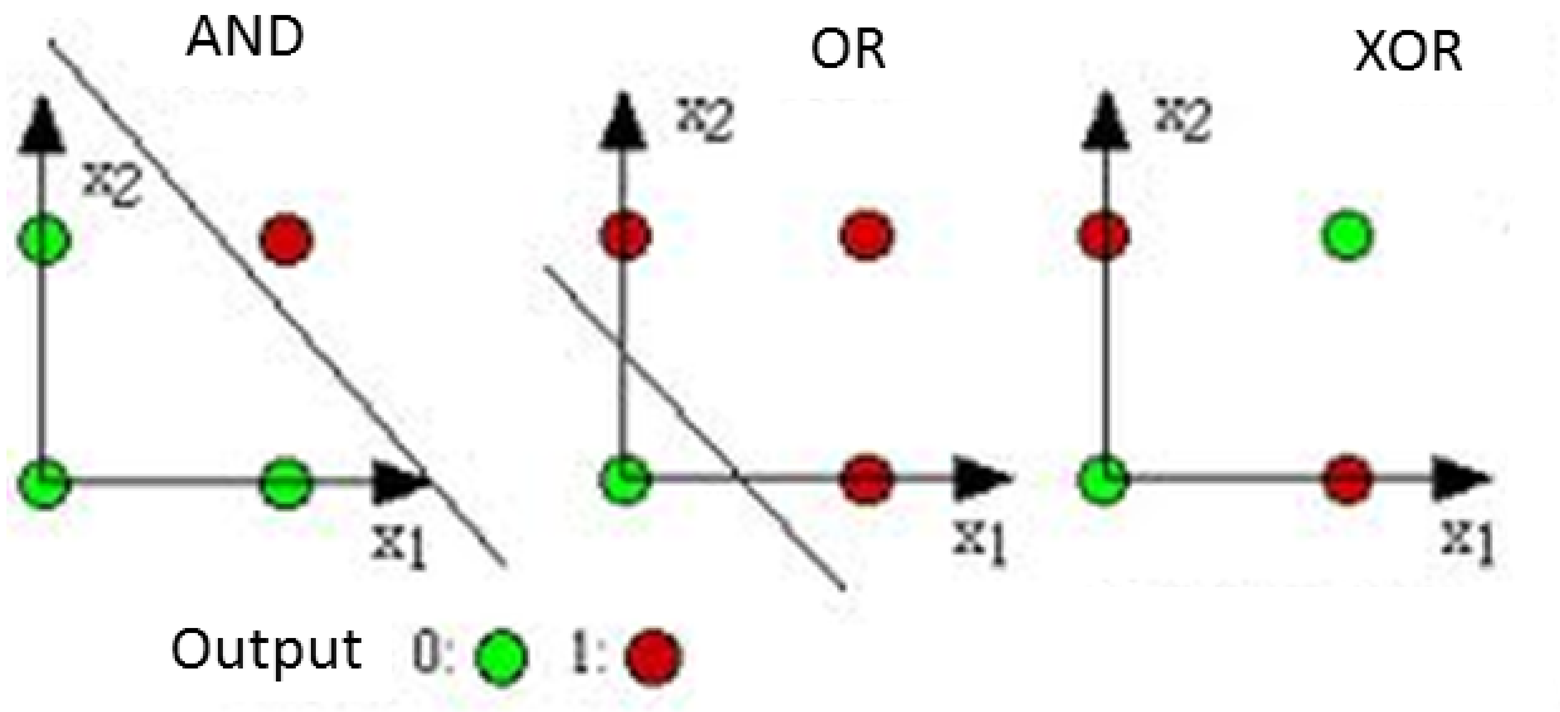
- If the inputs are both 0, then net input is 0 which is less than the threshold (0.5). So the output is 0 - desired output.
- If one of the inputs is 0 and the other is 1, then the net input is 1. This is above threshold, and so the output 1 is obtained.
- But the given perceptron

  fails for the last case.



The XOR problem

- Graphically:



- This is true for all the non linear separable problems
- Note: there is a mathematical trick (see SVM)

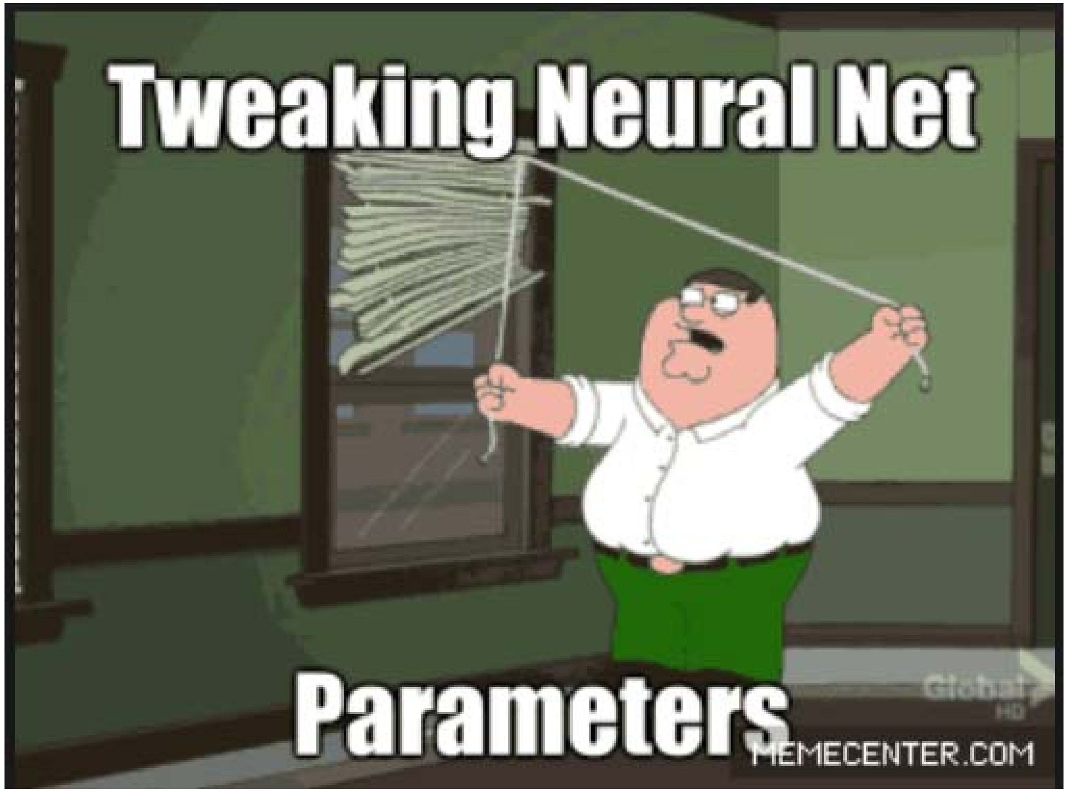
To recap the differences

- Machine learning uses algorithms to parse data, learn from that data, and make informed decisions based on what it has learned
- Deep learning structures algorithms in layers to create an “artificial neural network” that can learn and make intelligent decisions on its own
- Deep learning is a subfield of machine learning. While both fall under the broad category of artificial intelligence, deep learning is what powers the most human-like artificial intelligence

Dynamic outline

- Introduction
- Taxonomy
- AI fundamentals: Philosophy, History
- GOFAI vs modern AI
- Practical examples:
  - Turing Test and Chatbots
  - BCIs
  - ANNs (machine learning) NEXT LESSON
  - Ontologies
- Philosophy of AI & Ethic issues

See you next time…



Raffaella.Folgieri@unimi.it