Advanced Machine Learning

Bilel GUETARNI, PhD

bilel.guetarni@junia.com

JUNIA ISEN

PowerPoint slides from **Halim Benhabiles, PhD** and **Farès BOUGOURZI, PhD**

Course session organization and evaluation

- 7 sessions of 4h [theory + lab]
- Written exam 2h
- Final score on 20

Material for this course

- Teams – Files – Course material
  - Slides

- Online documentation
  - Deep learning - Ian Goodfellow
  - Deep learning with python – François Chollet
  - http://neuralnetworksanddeeplearning.com/

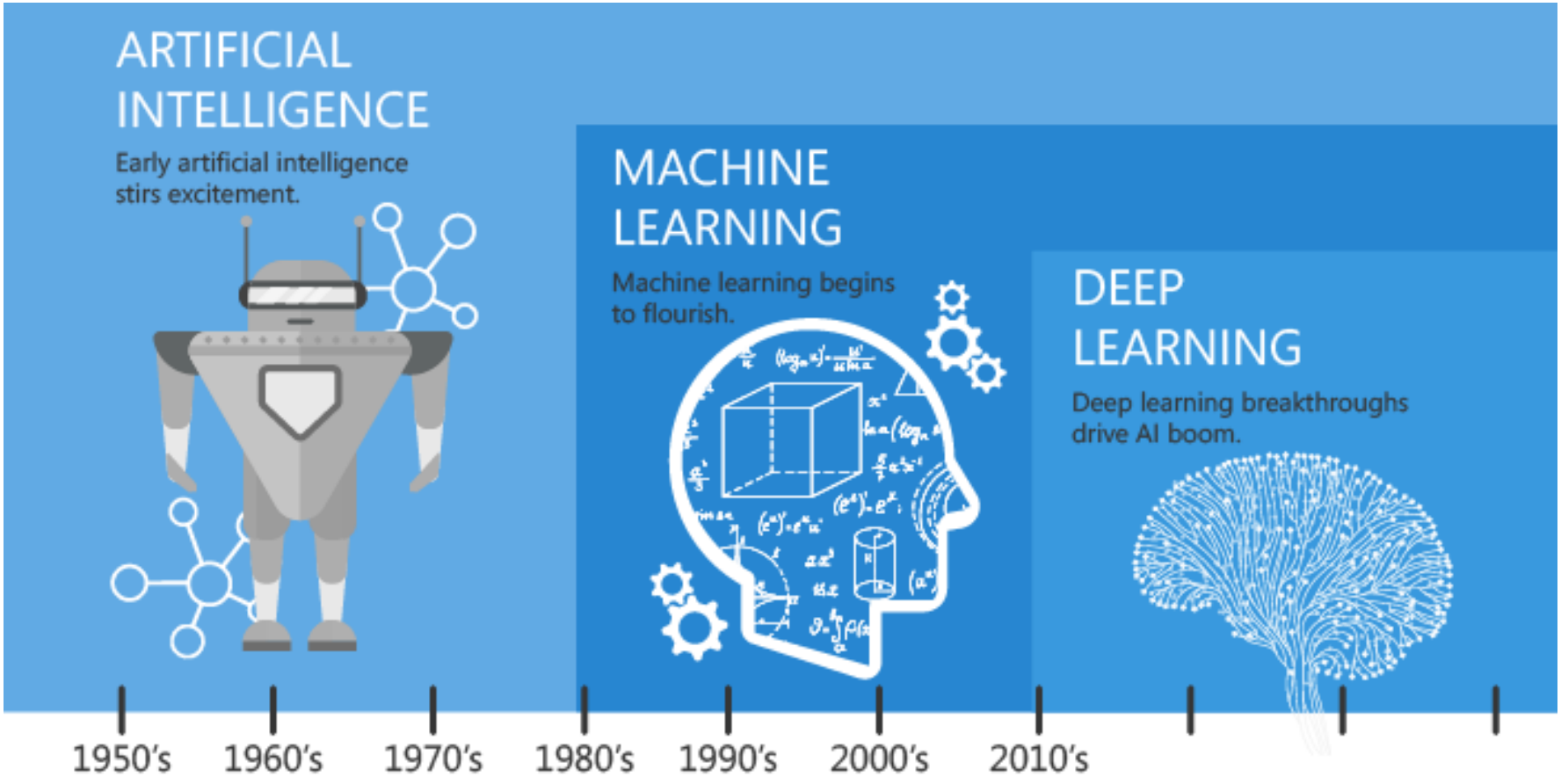
- Global goal

Design and develop a customized neural network with its hyperparameters, train it on several datasets related to classification task and evaluate its performances.

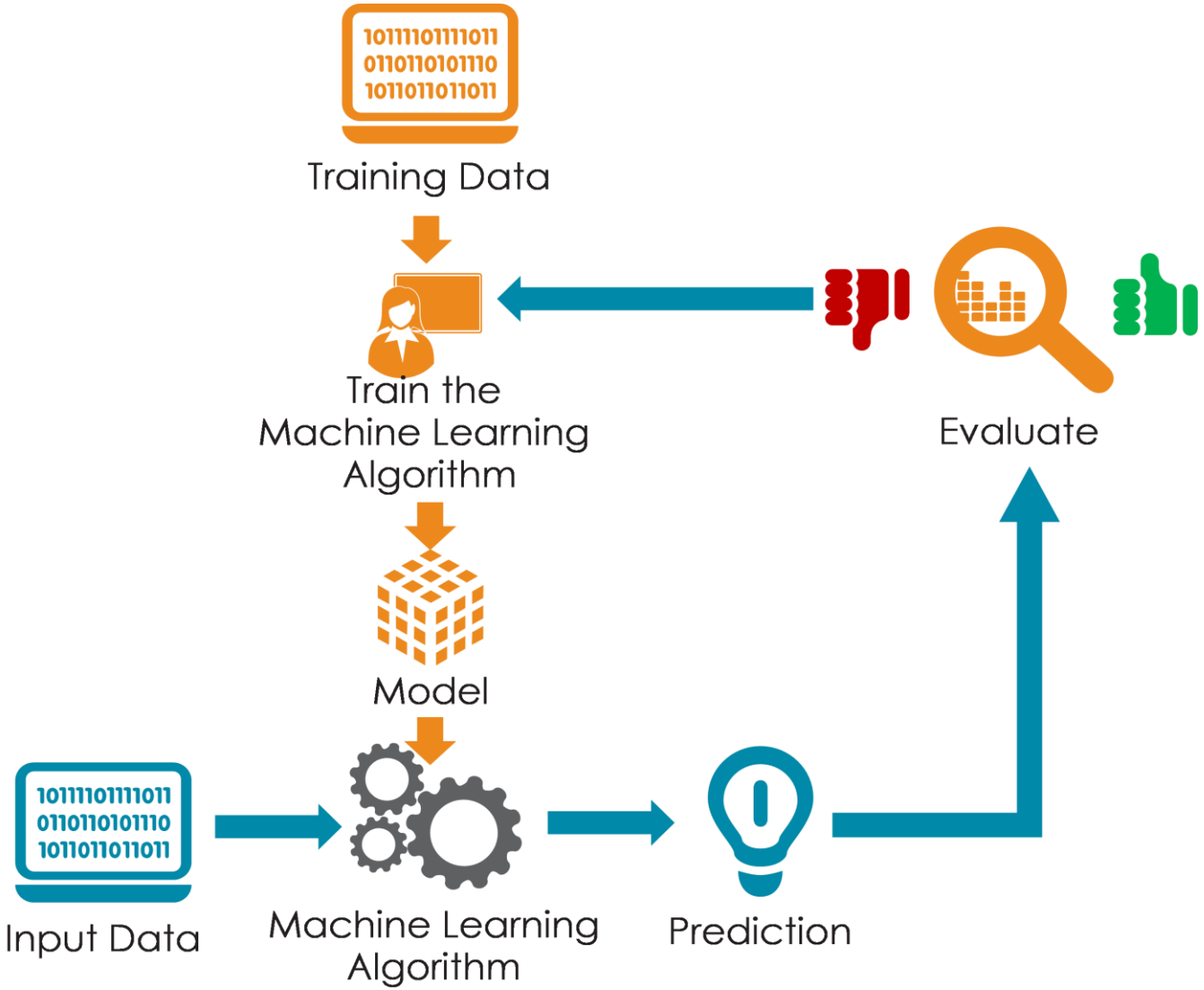
Chapters

- MLP and associated components
- Standard CNN
- Transfer learning strategies
- Common CNN architectures (e.g. ResNet, VGG, etc.)
- Transformers
- Generative adversarial nets: GAN

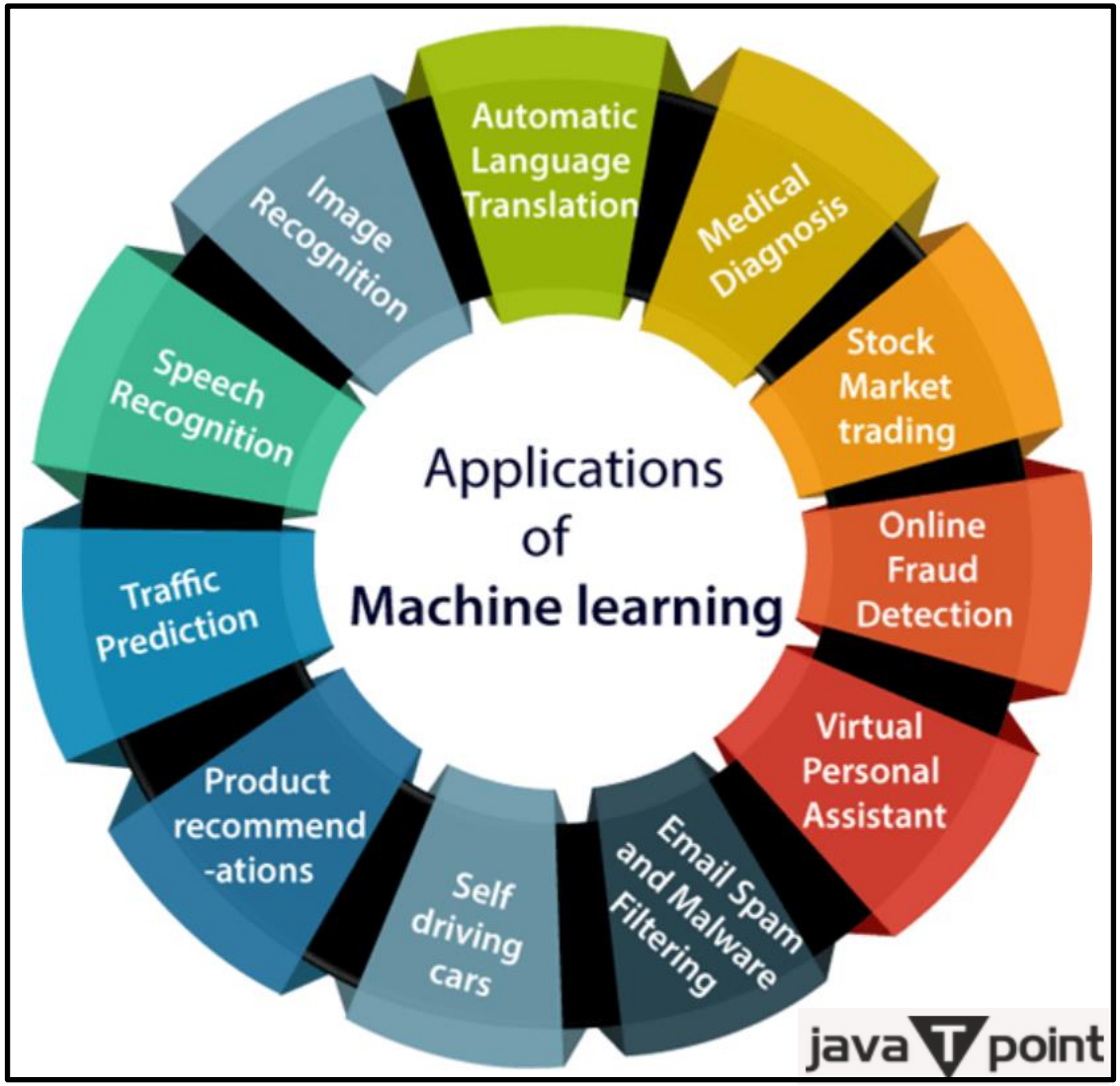
AI vs ML vs AML (Deep Learning)



An example of machine learning based system

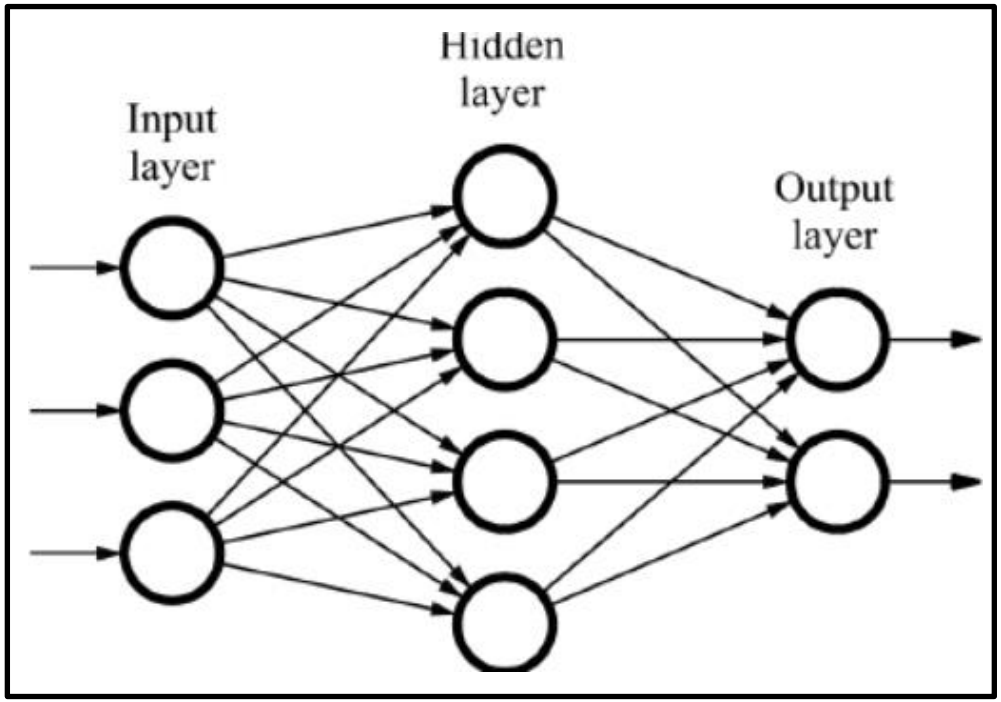


Machine learning applications



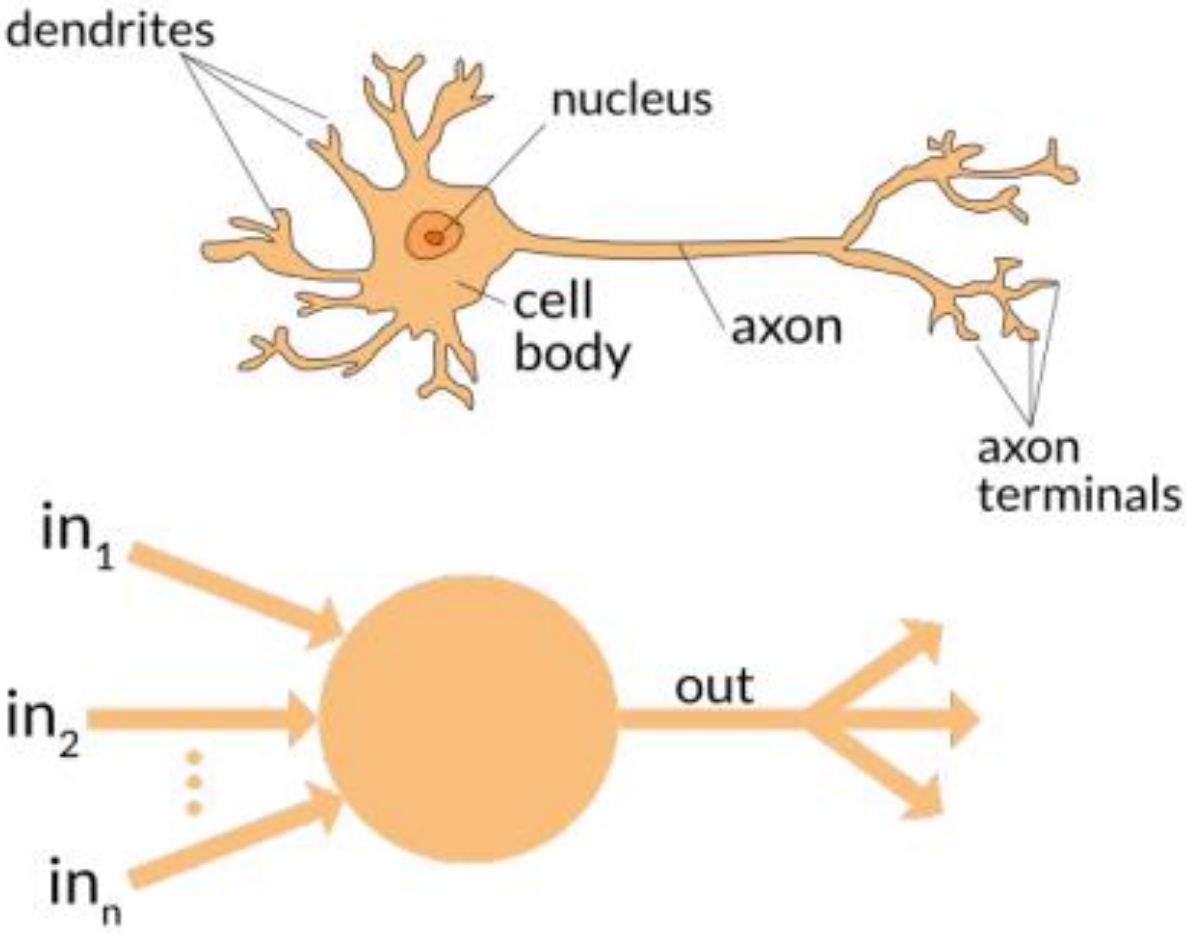
Machine learning technics (one advanced technic example)

Neural network in human brain
(Courtesy of Google Semantics)

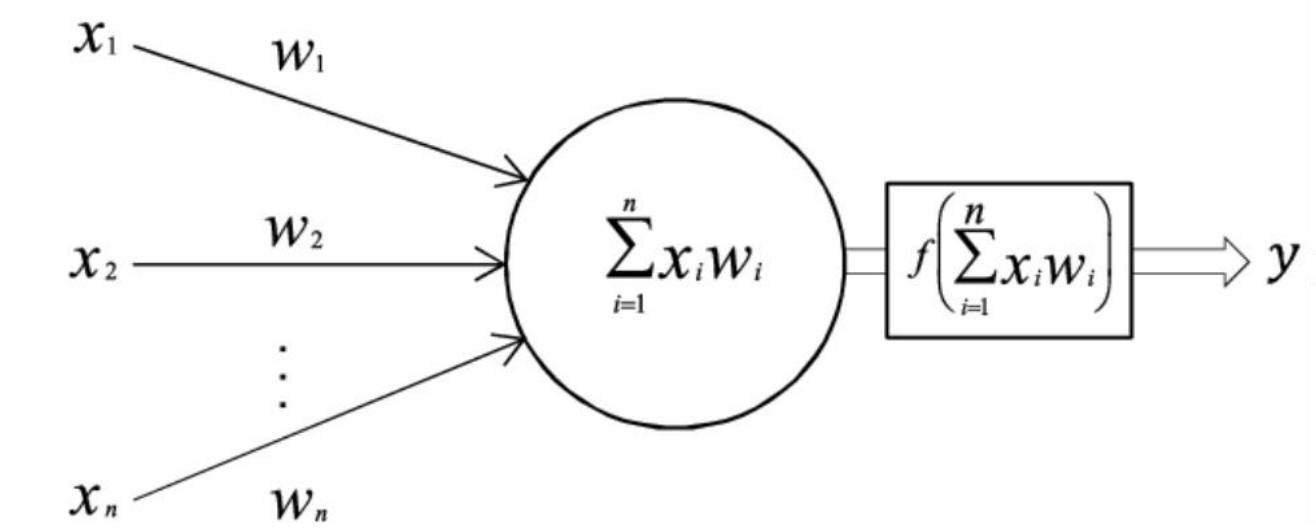


Artificial neural network

Biological neuron vs artificial neuron



Neuron modeling



inputs

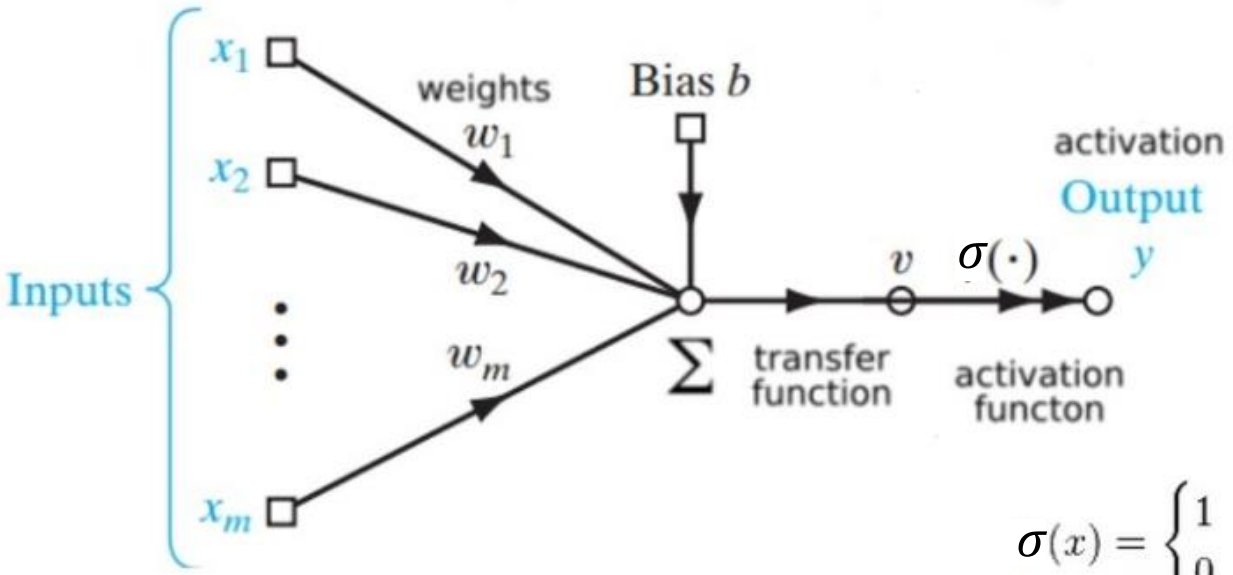
Weights or Synaptic coefficients

Transfer function Weighted sum

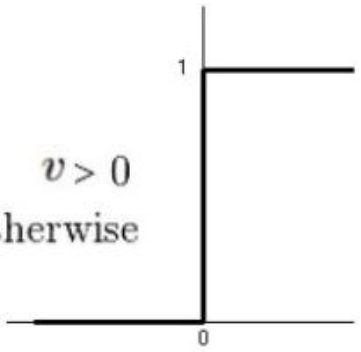
Activation Function

Output

Linear perceptron (heavyside function $\sigma(x)$)



$$\sigma(x) = \begin{cases} 1 & \text{if} \quad v > 0 \\ 0 & \text{otherwise} \end{cases}$$



$$X = \begin{pmatrix} X_0 \\ \vdots \\ X_m \end{pmatrix} \qquad W = \begin{pmatrix} W_0 \\ \vdots \\ W_m \end{pmatrix}$$

$$\mathbf{v} = \sum_{j=0}^{m} \boldsymbol{w_j x_j} + \boldsymbol{b}$$

Perceptron training algorithm (Widrow-Hoff law)

- We note S the training base
- S is composed of couples (X, c)
  - X input vector (x_0, x_1)
  - c expected output
- Set randomly initial weights (w_0, w_1, b)
  - Repeat until <u>all examples are well predicted*</u>
    - Take each example (X, c) from S
    - Calculate the output o of the perceptron based on the input X
    - Update the weights (w_0, w_1, b)
      - For i= 0 to 1
        - $w_i = w_i + \eta * (c - o) * x_i$
      - $b = b + \eta * (c - o)$
      - EndFor
  - EndRepeat

Learning rate

Implement the algorithm to learn 'OR'

a	b	OR
0	0	0
0	1	1
1	0	1
1	1	1

* Or for a given number of iterations

Lab session

- Implement your perceptron training algorithm (Widrow-Hoff law) to learn ‘OR’ function and test it
  - using different set of weights:
    - w = [-1, +1] b = 0.5 then 0
    - w = [-0.8,0.1], b = 0.5 then 0
  - using different learning rates [0.002, 0.02, 0.2, 2]
  - by changing the order of the examples

Implement the algorithm to learn ‘OR’

a	b	OR
0	0	0
0	1	1
1	0	1
1	1	1

- Do the same job to learn
  - AND function
  - XOR function

- Provide your own analysis based on the different results