UNIVERSITÀ PARTHENOPE
DiST
DIPARTIMENTO DI SCIENZE E TECNOLOGIE

Machine Learning (part II)

Hebbian Learning

Angelo Ciaramella

Introduction

- Artifical Neural Networks (ANNs) are at the very core of Deep Learning
  - powerful
  - scalable

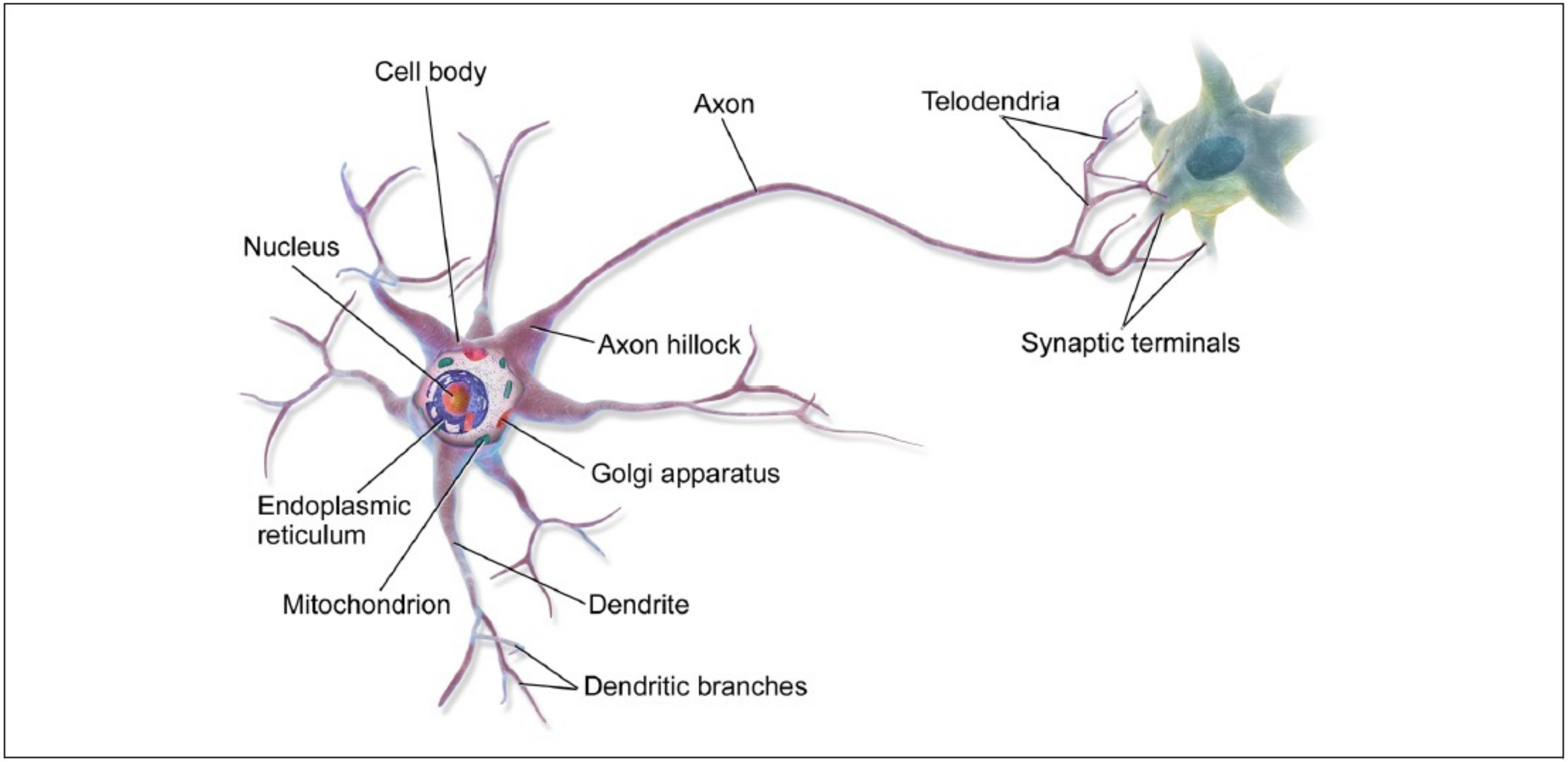
- Complex ML tasks
  - classifying billions of images (e.g., Google Images)
  - powering speech recognition services (e.g., Apple's Siri),
  - recommending the best videos to watch to hundreds of millions of users every day (e.g., YouTube)
  - learning to beat the world champion at the game of Go by playing millions of games against itself (DeepMind's Alpha-Zero)

Biological neuron

- Biological neurons
  - behave in a rather simple way
  - are organized in a vast network of billions of neurons
  - each neuron typically connected to thousands of other neurons

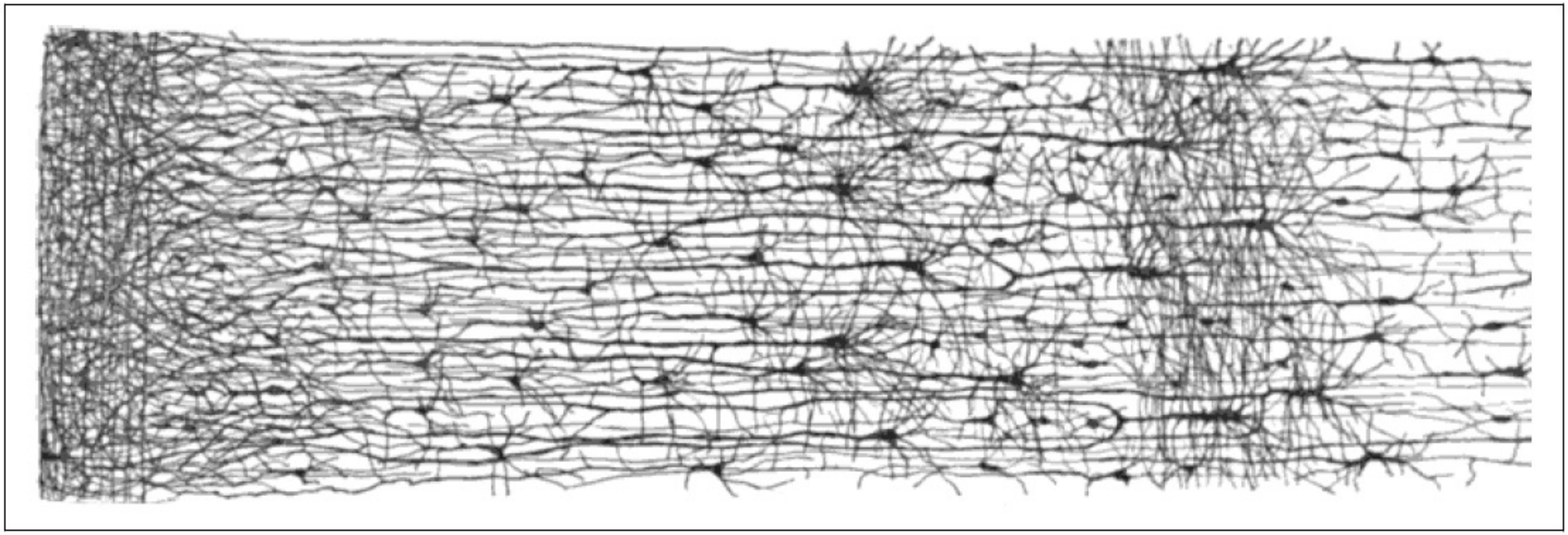
- Neural networks
  - Highly complex computations can be performed by a vast network of fairly simple neurons

Biological neuron



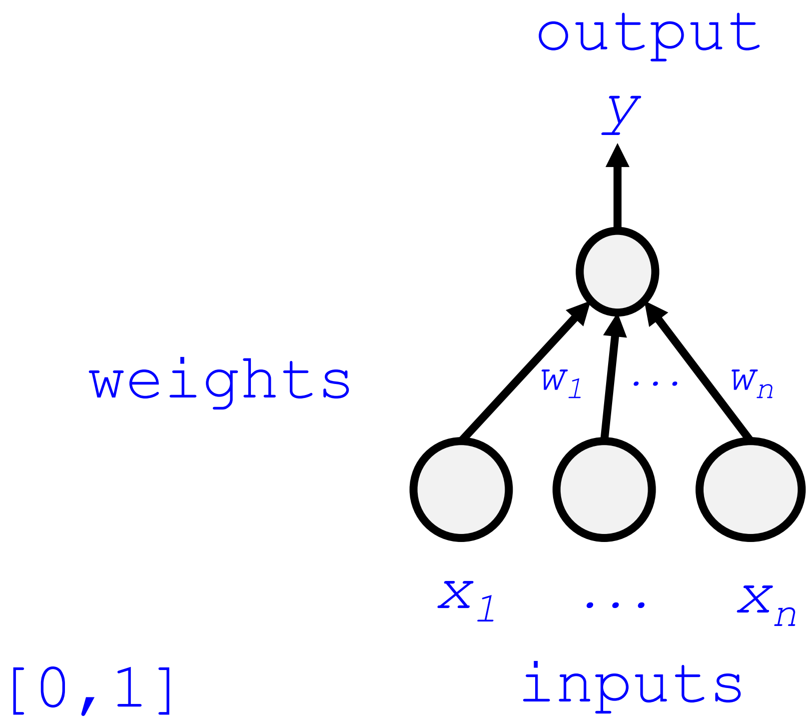
Biological Neuron

Biological Neural Network



Multiple layers in a biological neural network (human cortex)

Artificial neuron



$$\mathbf{w} = \begin{bmatrix} w_1 \\ \dots \\ w_n \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix}$$

Artificial neuron

- sum

$$z = \sum_{i=1}^{n} w_i \, x_i = \mathbf{w}^{\mathrm{T}} \mathbf{x}$$

- output

$$y = f(\mathbf{w}, \mathbf{x}) = \theta(z)$$

- activation functions

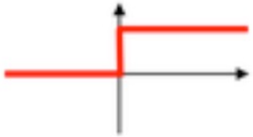
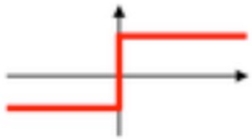
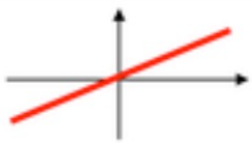
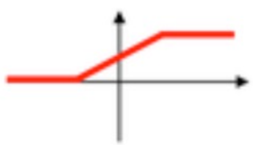
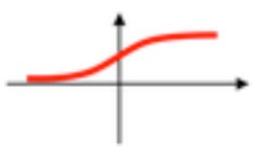
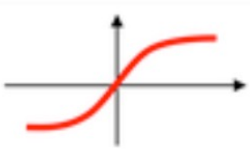
$$\theta = \begin{cases} 1 & if \ z \geq 0 \\ 0 & if \ z < 0 \end{cases}$$

Heaviside

$$\theta = \begin{cases} -1 \ if \ z < 0 \\ 0 \ if \ z = 0 \\ +1 \ if \ z > 0 \end{cases}$$

Signum

Activation functions

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

Weights adaption

- McCulloch-Pitts Neuron
  - fixed weights
- Learning
  - weights adaption
  - learning approach

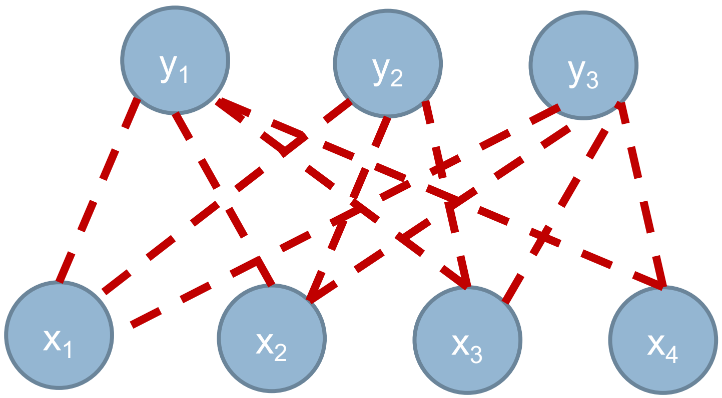
Learning

- First learning hypotheses
  - Donald O. Hebb
  - 1949 – Book titled: *The organization of behavior neurophysiological evidence*
- Principle

If two connected neurons are simultaneously active, the synaptic efficacy of the connection is reinforced

Hebb's rule

Synapses



$\Delta w_{ij} = \eta y_i x_j$

Learning rule

μ learning rate

Hebb's algorithm

- Initialize the synaptic weights

$$w_{ij}=0$$

- Calculate synaptic changes

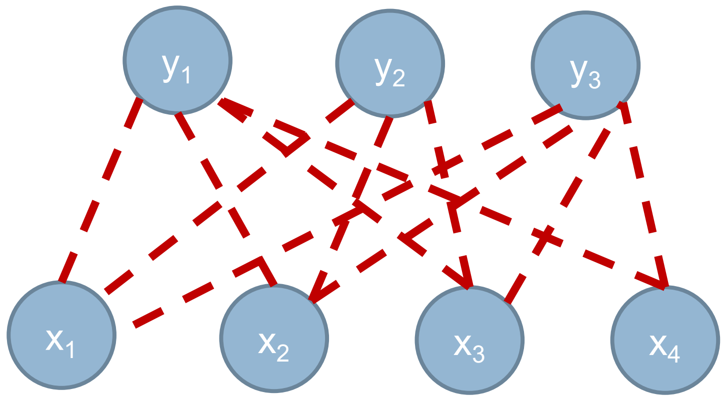
$$\Delta w_{ij}=\eta y_i x_j$$

- Update the synaptic weights

$$w_{ij}(t) = w_{ij}(\mathrm{t}-1) + \Delta w_{ij}$$

Hebb's rule

Synapses



x	y
1 0 0 1	1 0 0
0 1 0 0	0 1 0
0 0 10	0 0 1

Learning example

Considerations

- Limitations
  - The Hebb rule allows to learn only orthogonal patterns
  - Mixed responses are called interferences

- Some improvements
  - Postsynaptic rule
  - Presynaptic rule

Postsynaptic rule

- Postsynaptic rule
  - Stent-Singer
    - neurophysiologicals that highlighted the mechanism in biological circuits
  - rule
    - increased when the postsynaptic and presynaptic units are active
    - decreased when the postsynaptic unit is active but the presynaptic unit is inactive
  - reduction of the interference phenomenon
  - too many inhibitory synapses
    - it is not found in biological systems but in all the artifcial neural networks

Presynaptic rule

- Presynaptic rule
  - increased when the postsynaptic and presynaptic units are active
  - decreased when the presynaptic unit is active but the postynaptic unit is inactive
  - It works well when many different and partially overlapping patterns need to be associated with the same pattern

Postsynaptic rule

$$\Delta w_{ij} = \eta\left(y_i x_j + (x_j - 1) y_i\right)$$

postsynaptic rule

$$\Delta w_{ij} = \eta\left(y_i x_j + (y_i - 1) x_j\right)$$

presynaptic rule

Hebbian learning and NNs

- NNs based on the Hebb's rule
  - Hopfield network
    - recurrent artificial NN described by Little in 1974
    - popularized by John Hopfield in 1982
    - content-addressable («associative») memory systems with binary threshold nodes
    - They are guaranteed to converge to a local minimum
      - converge to a false pattern (wrong local minimum) rather than the stored pattern (expected local minimum
    - provide a model for understanding human memory

Hebbian learning and NNs

- NNs based on the Hebb's rule
  - Oja's rule
    - Finnish computer scientist Erkki Oja
    - Is a model of how neurons in the brain or in artificial neural networks change connection strength
    - solves stability problems of Hebbian learning
    - generates an algorithm for
      - Principal Component Analysis (PCA)
      - non-linear PCA
      - Independent Component Analaysis (ICA)