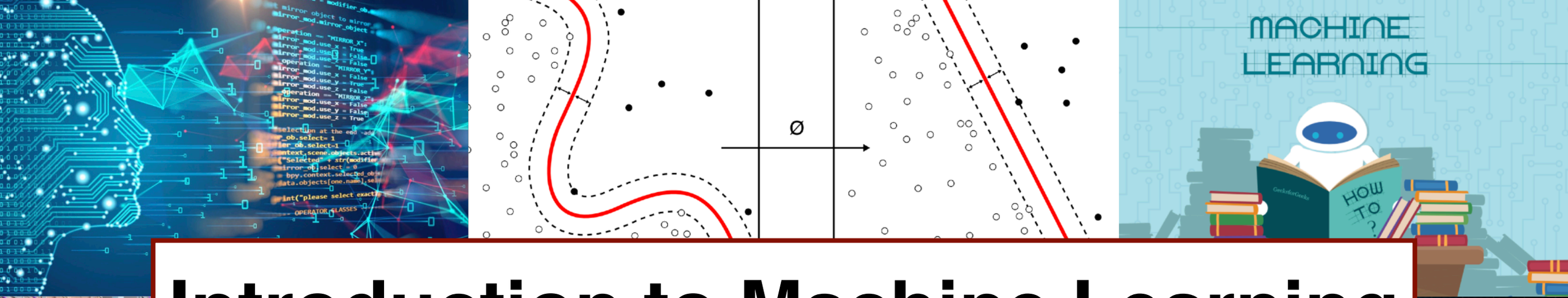


Introduction to Machine Learning

SCP8084699 - LT Informatica

Artificial Neural Networks I

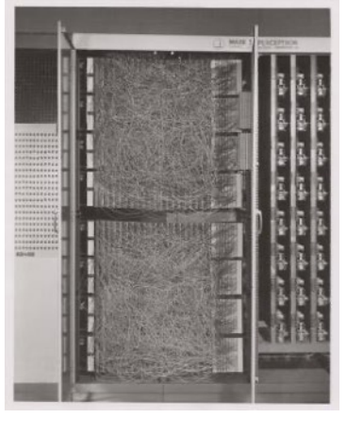
Prof. Lamberto Ballan

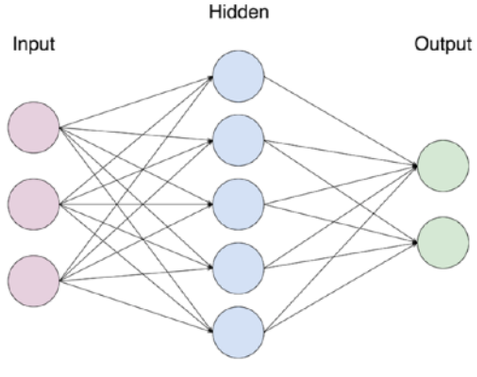
University of Padova, A.Y. 2022/23 April 6, 2023

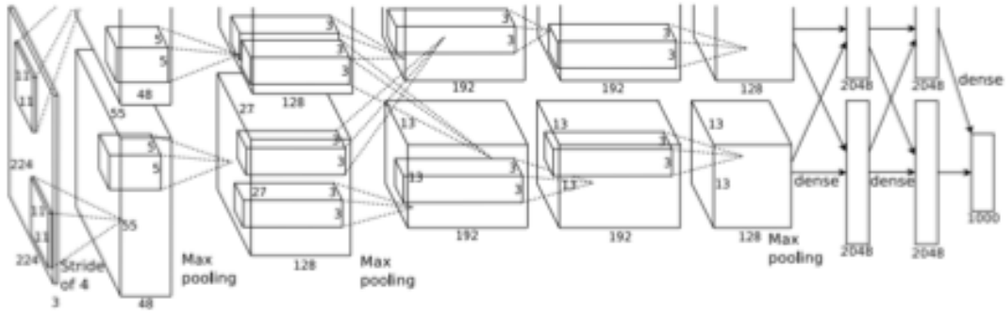
DIPARTIMENTO MATEMATICA
Department of Mathematics “Tullio Levi-Civita”

Neural Networks: origins

- Algorithms that are modelled (loosely) after the brain
  - Neural networks have been introduced in late 1950s
  - They were widely used in 1980s and early 1990s; their popularity largely diminished in late 1990s
  - Recent “resurgence”: late 2000s/early 2010s (deep learning revolution)
  - Artificial neural networks are not nearly as complex or intricate as the actual brain structure

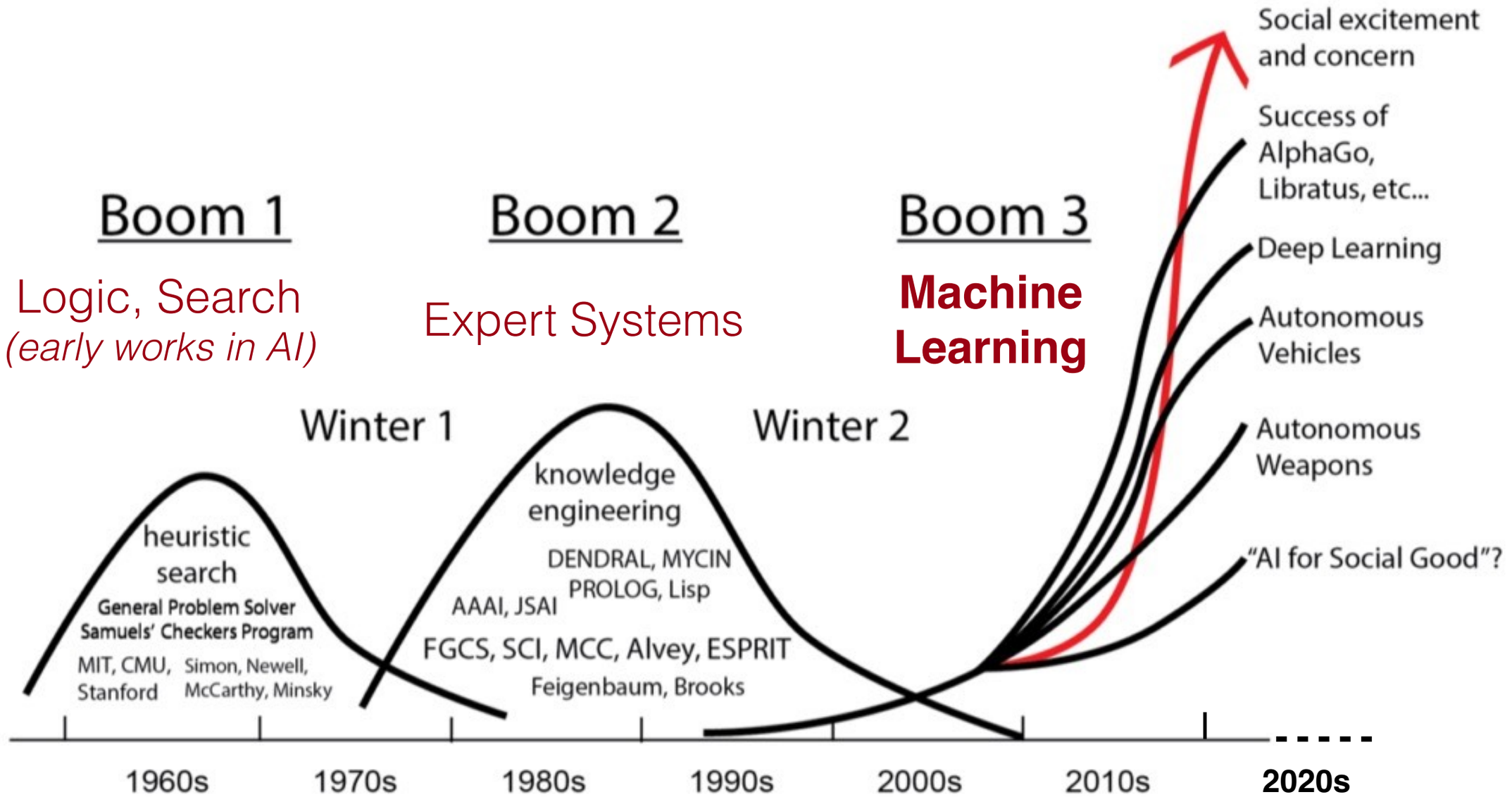






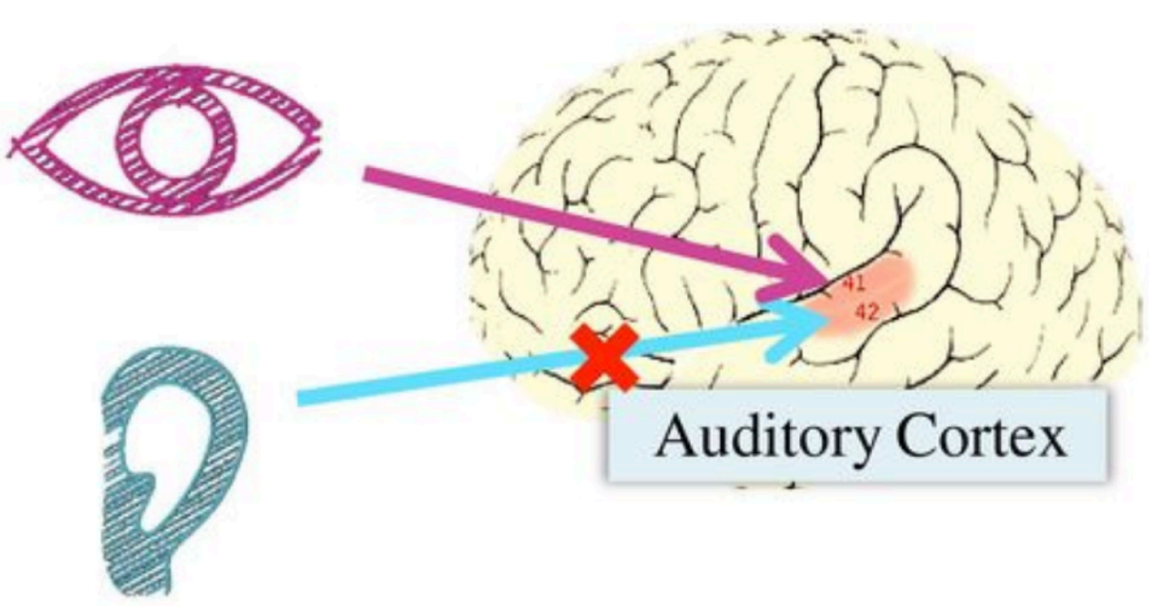
Neural Networks: origins

- Neural Networks and AI winters



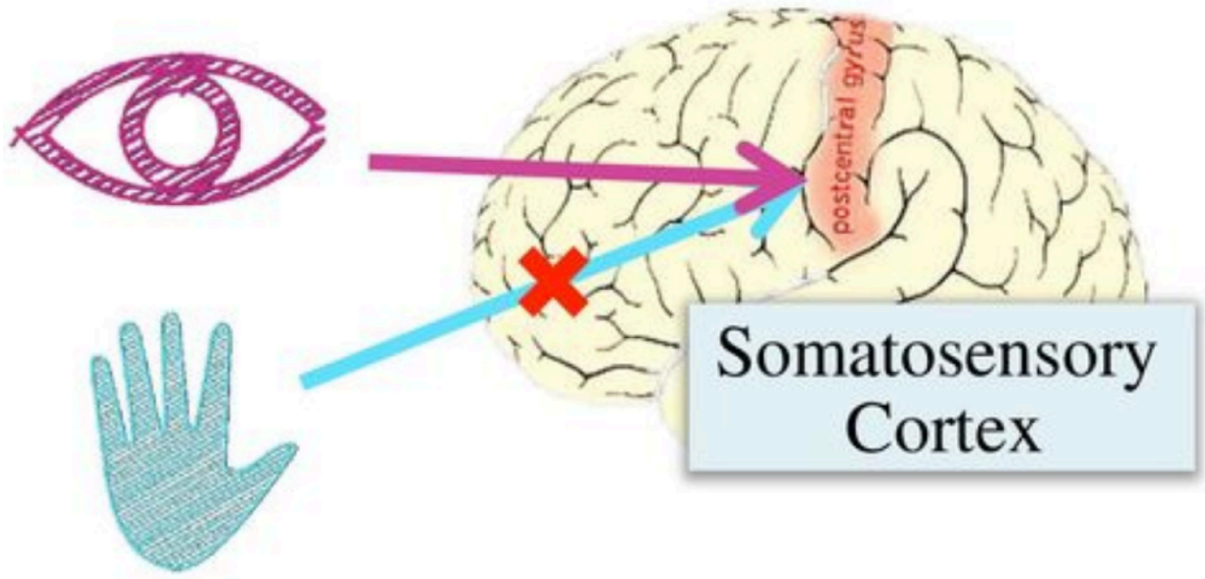
Neural Networks: origins

- The “one learning algorithm” hypothesis



Auditory cortex learns to see

[Roe et al., 1992]

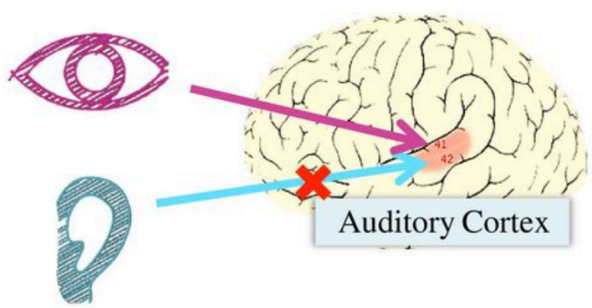


Somatosensory cortex learns to see

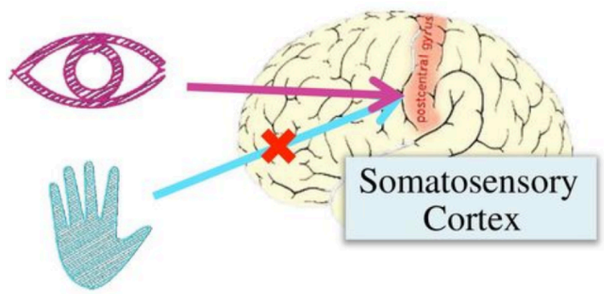
[Metin and Frost, 1989]

Neural Networks: origins

- The “one learning algorithm” hypothesis
  - Because of these experiments there is this sense that if the same part of the brain can process sight or sound or touch, then (maybe) there is *one learning algorithm*
  - Therefore, instead of designing hundreds of different algorithms, we need to approximate this one algorithm



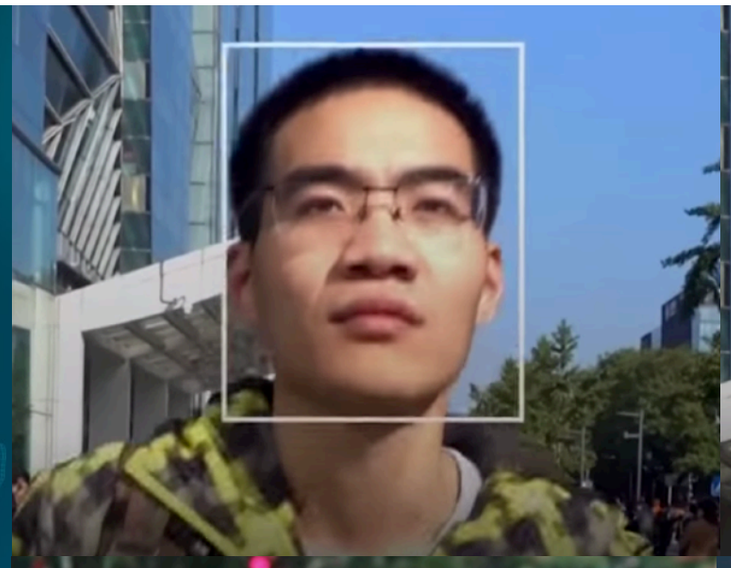
[Roe et al., 1992]

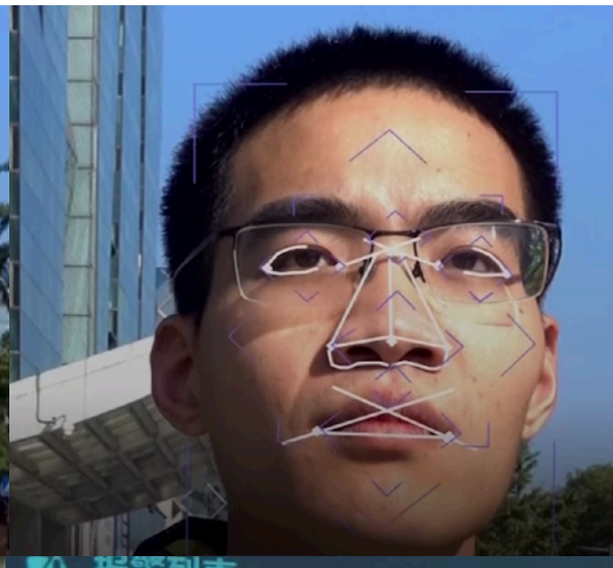


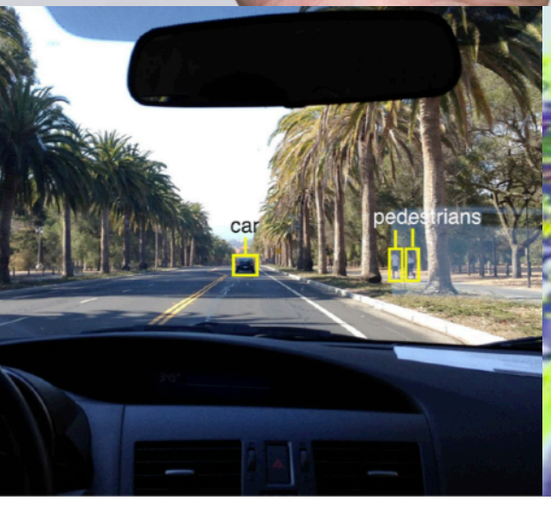
[Metin and Frost, 1989]

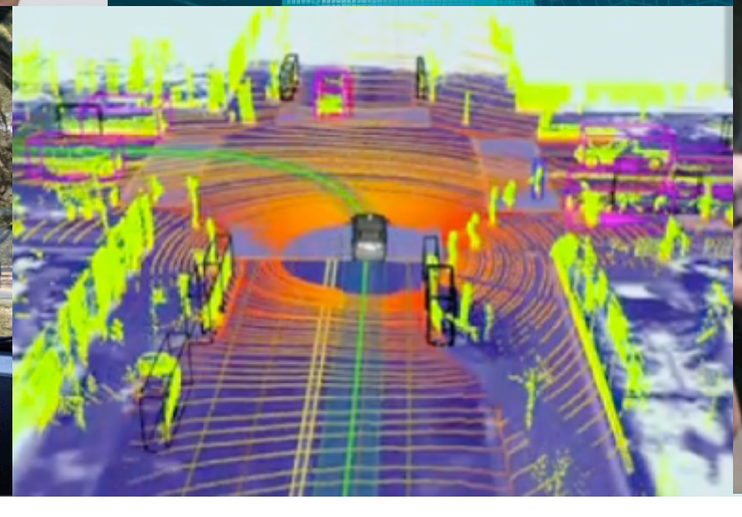
Neural Networks: applications

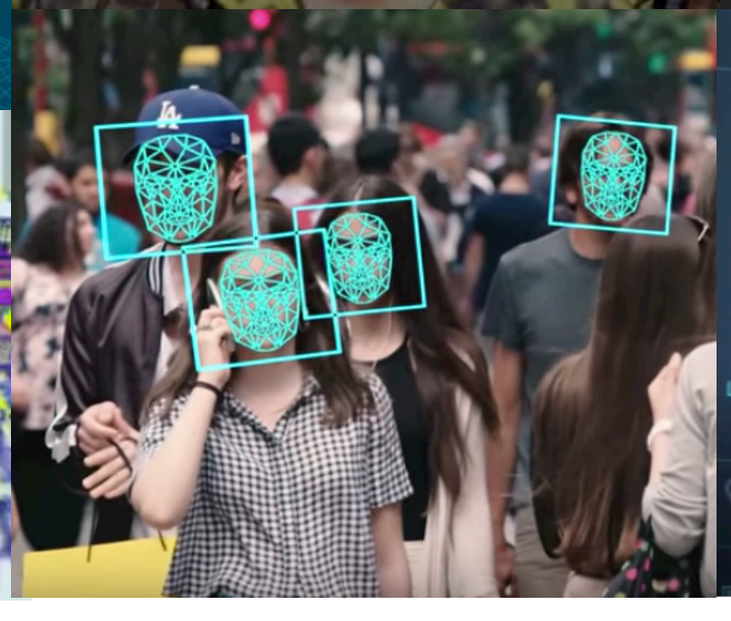
- Neural networks are everywhere…





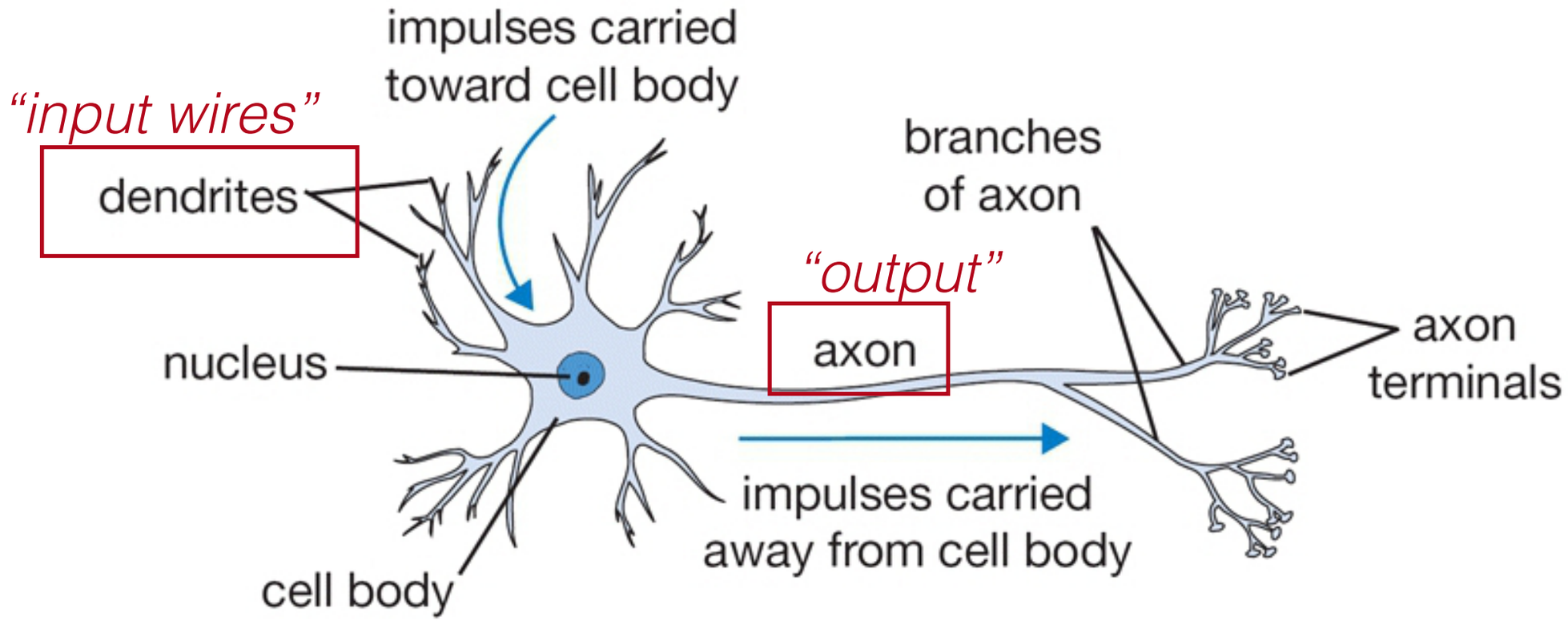






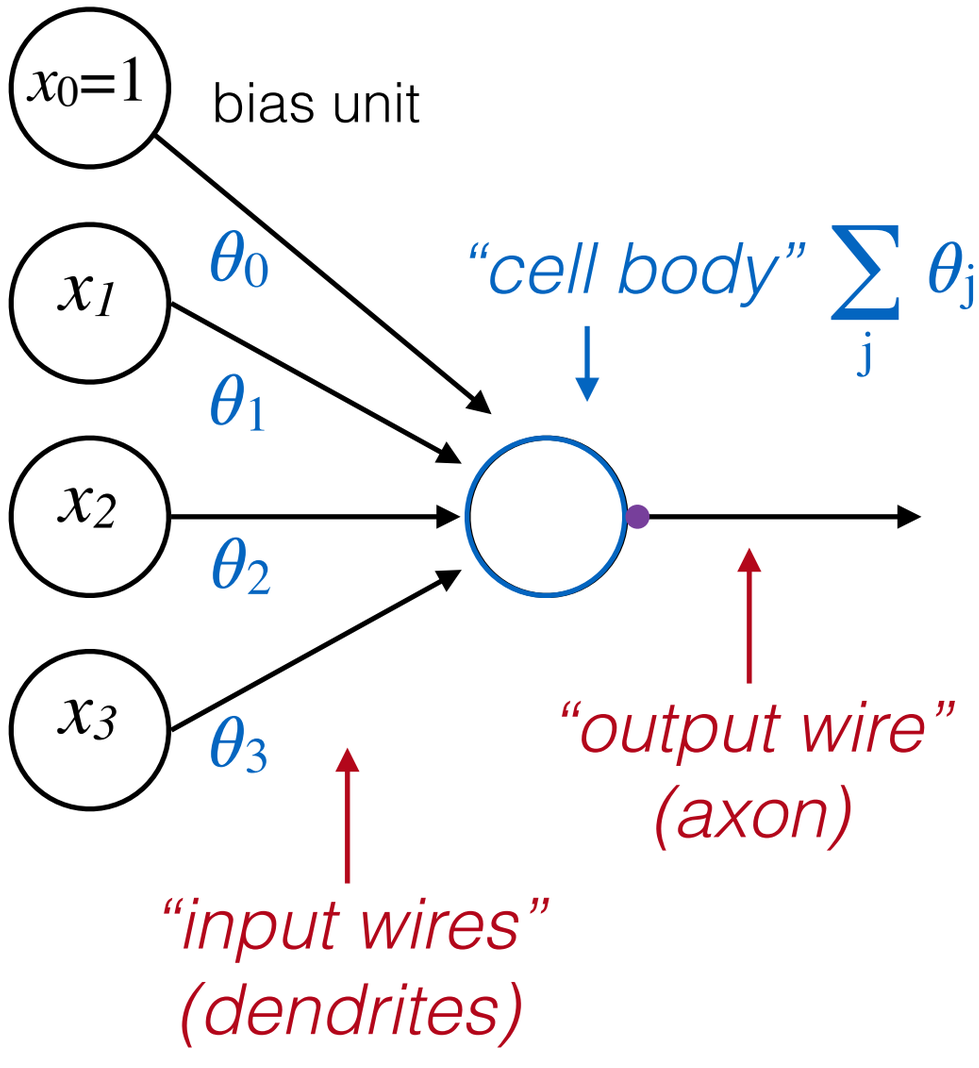
Neural Networks

- Let's start by looking at what a single neuron in the brain looks like



Artificial Neural Networks

- We define a simple model for an artificial neuron (computational unit)



$$\mathbf{x} = \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix}$$

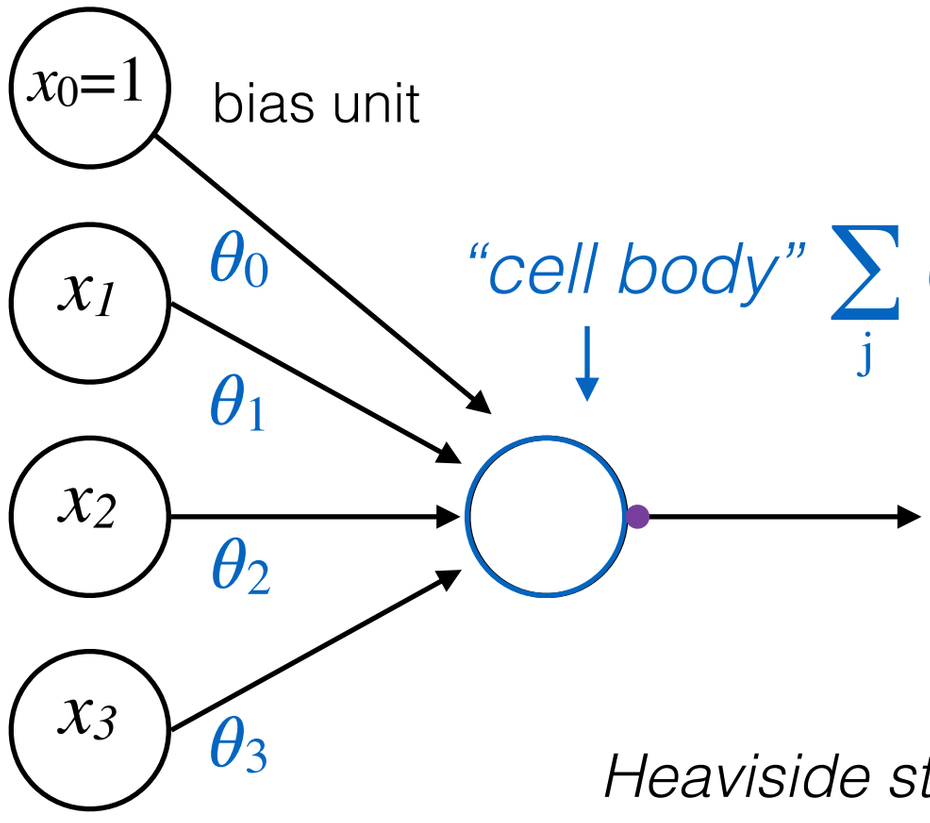
Parameters ("weights")

$$h_\theta(\mathbf{x}) = f\left(\boldsymbol{\theta}^T \mathbf{x}\right)$$

activation function f

Artificial Neural Networks

- We define a simple model for an artificial neuron (computational unit)



$$\mathbf{x} = \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix}$$

Parameters ("weights")

$$h_\theta(\mathbf{x}) = f\left(\boldsymbol{\theta}^T \mathbf{x}\right)$$

Heaviside step function

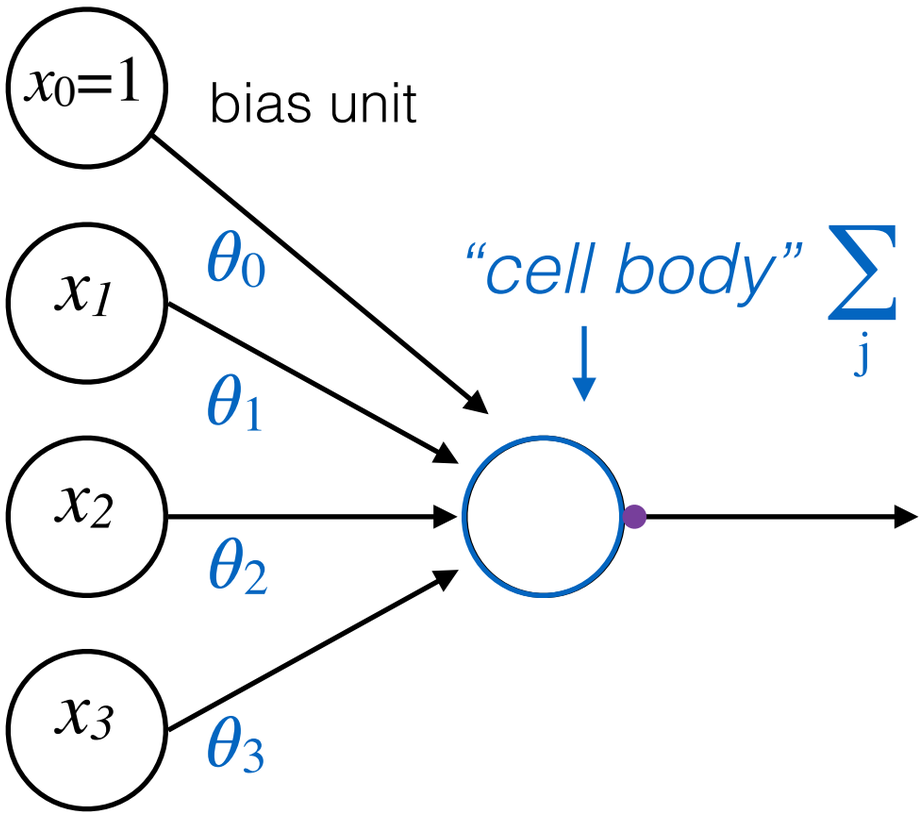
$$h_\theta(\mathbf{x}) = \begin{cases} 1 & \text{if } \boldsymbol{\theta}^T \mathbf{x} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Perceptron

F. Rosenblatt, 1958

Artificial Neural Networks

- We define a simple model for an artificial neuron (computational unit)



$$\mathbf{x} = \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix}$$

Parameters ("weights")

$$h_\theta(\mathbf{x}) = f\left(\boldsymbol{\theta}^T \mathbf{x}\right)$$

Logistic Unit

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

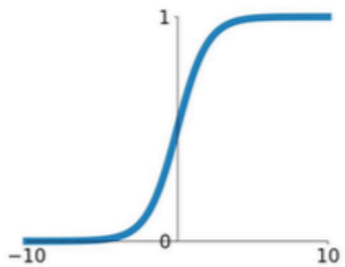
"activation function"
"non-linearity"

Neural Networks: activation functions

- Nowadays there are several activation functions you may encounter:

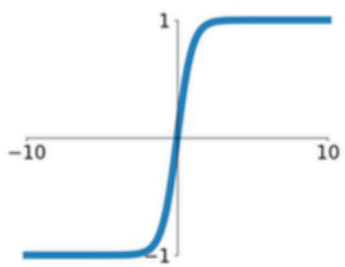
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



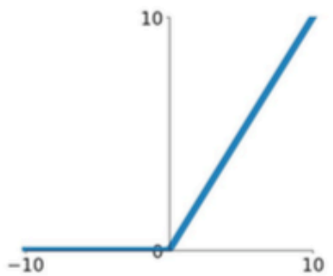
tanh

$$\tanh(x)$$



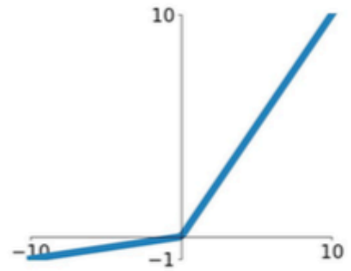
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

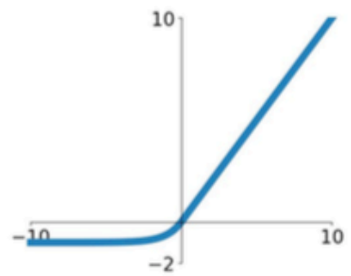


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

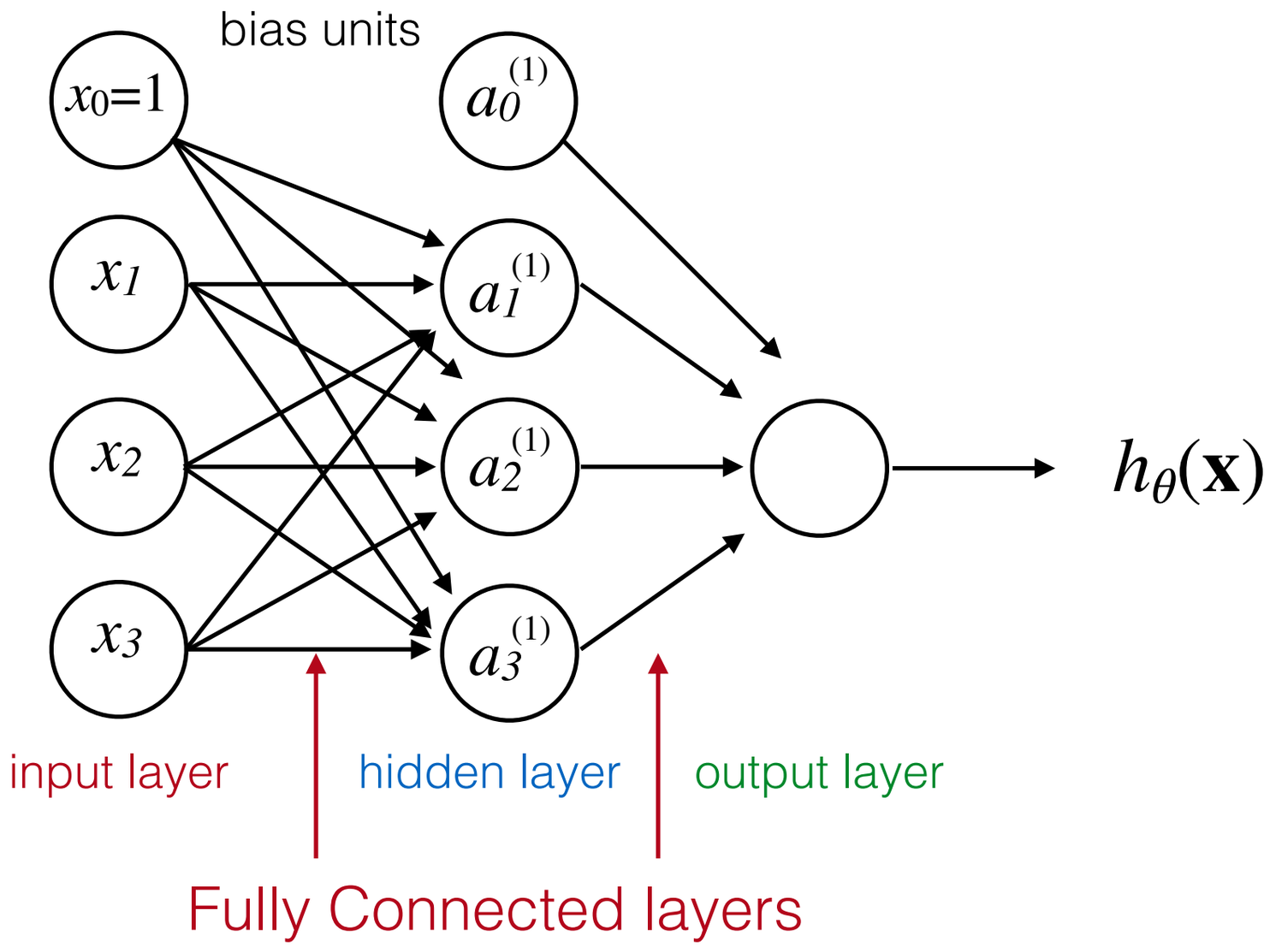
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Neural Networks: definition

- A (artificial) neural network is just a group of this different neurons strong together



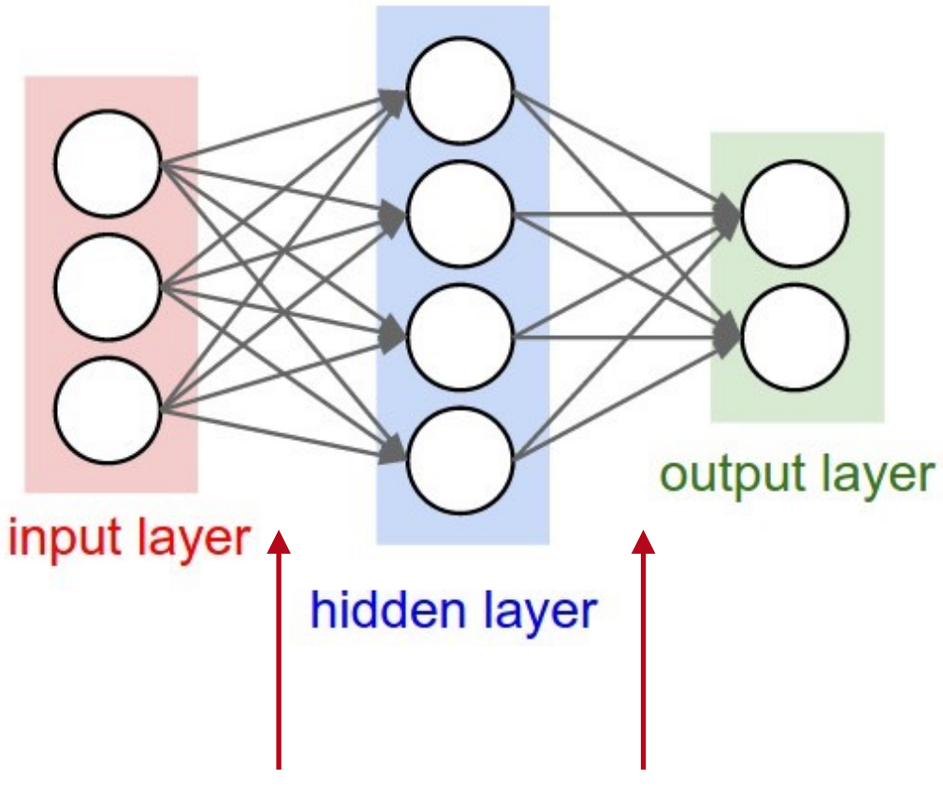
Adapted from slide by A. Ng

Neural Networks: architectures

- Here the term *architecture* refers to how the different neurons are connected to each other

2-layer neural network

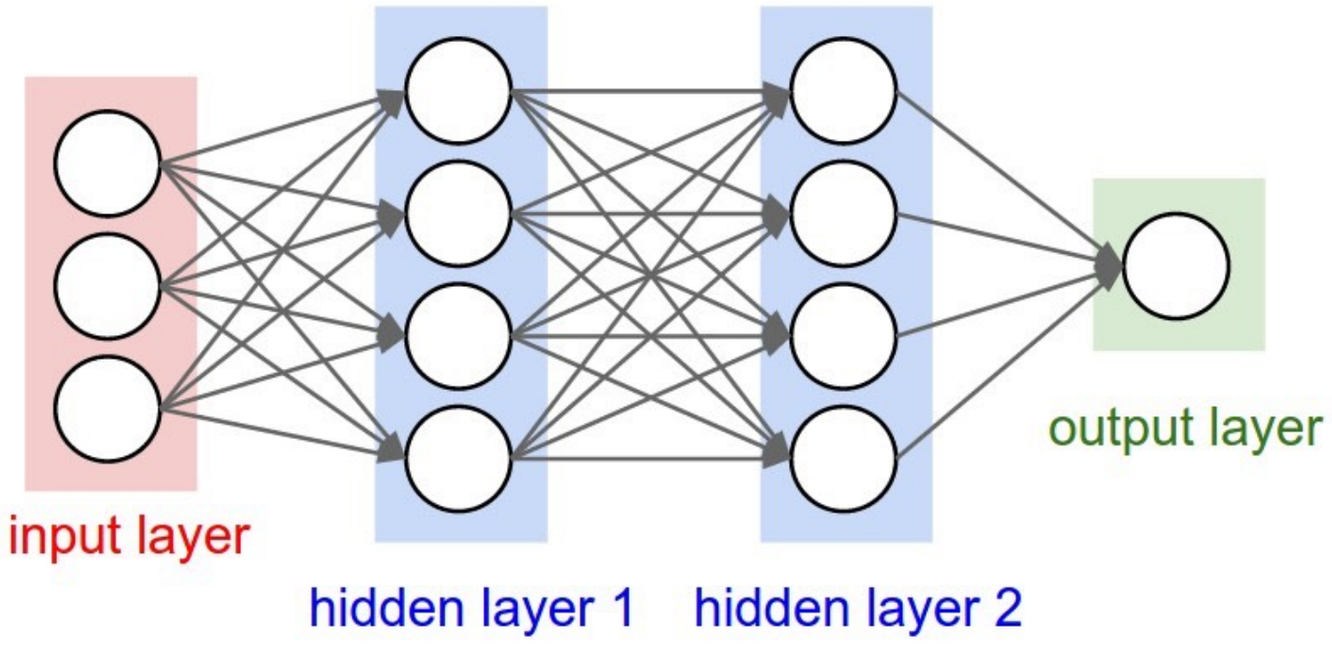
or 1-hidden-layer neural net



Fully Connected layers

3-layer neural network

or 2-hidden-layer neural net

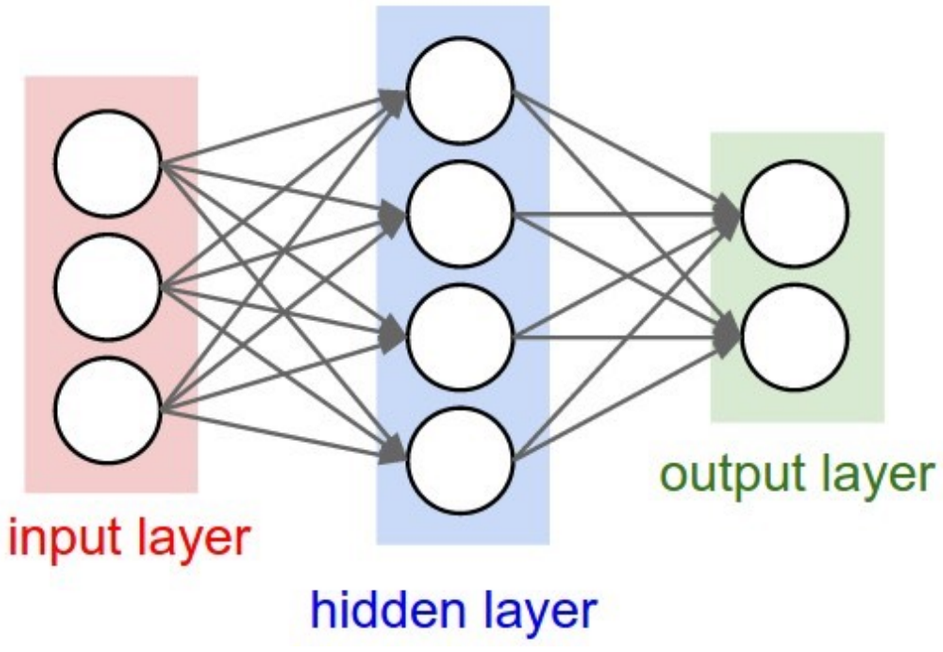


<u>Note:</u> when we say N-layer neural network, we do not count the input layer

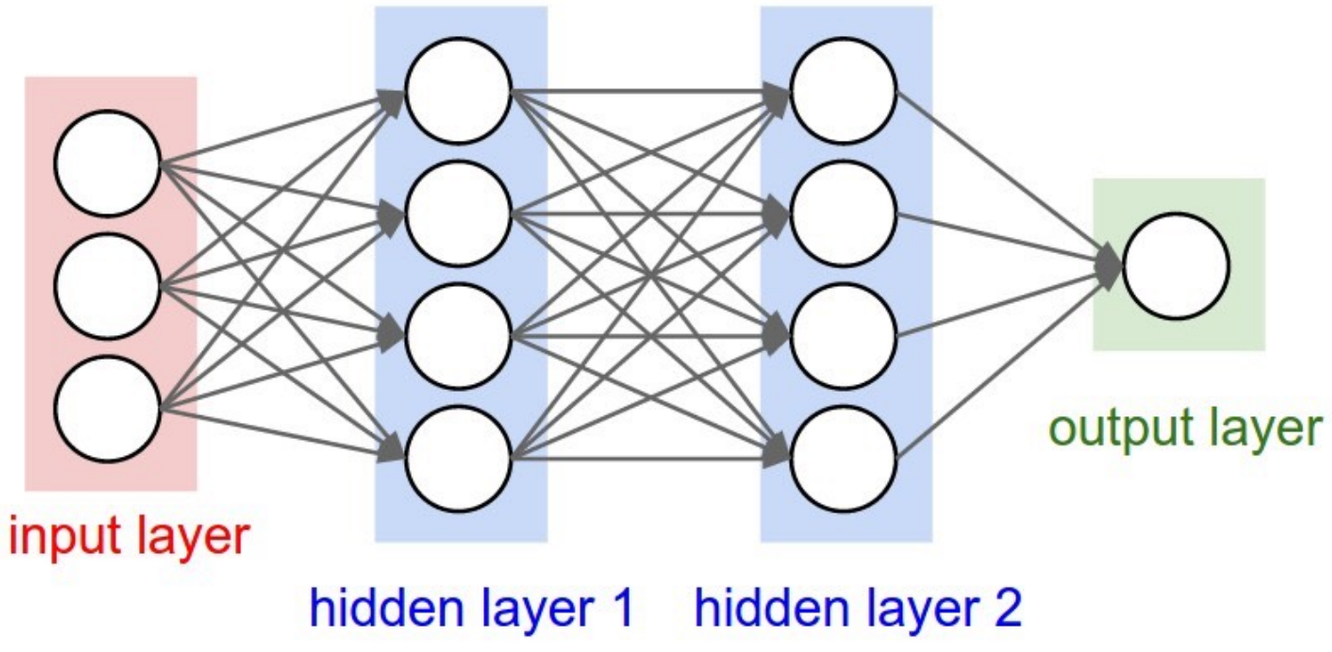
Neural Networks: architectures

- **Sizing neural networks:** the two metrics that are commonly used to measure the size of a NeuralNet are the #neurons or (more commonly) the #parameters

(a) **2-layer neural network**



(b) **3-layer neural network**



(b) has 4+4+1 neurons; how many (learnable) parameters?

(b) #parameters: 3x4 + 4x4 + 4x1 = 32 + 9 *(biases)* = 41

Contact

- **Office:** Torre Archimede, room 6CD3
- **Office hours** (ricevimento)**:** Friday 9:00-11:00

✉ lamberto.ballan@unipd.it

🏠 http://www.lambertoballan.net

🏠 http://vimp.math.unipd.it

@ twitter.com/lambertoballan