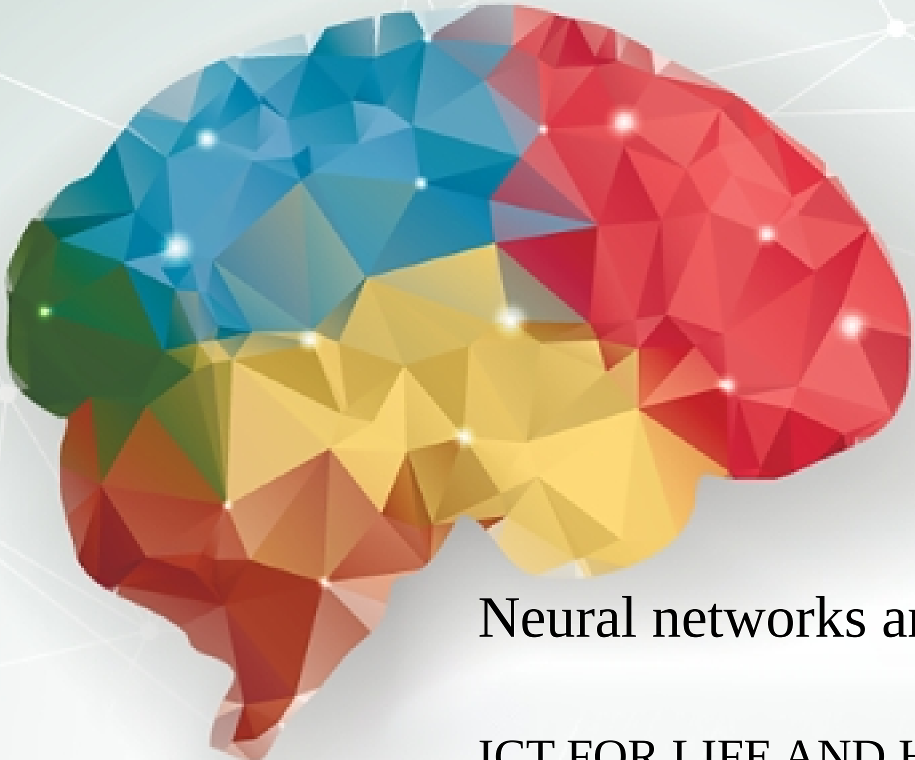


Neural networks and deep learning

ICT FOR LIFE AND HEALTH - Department of Information Engineering

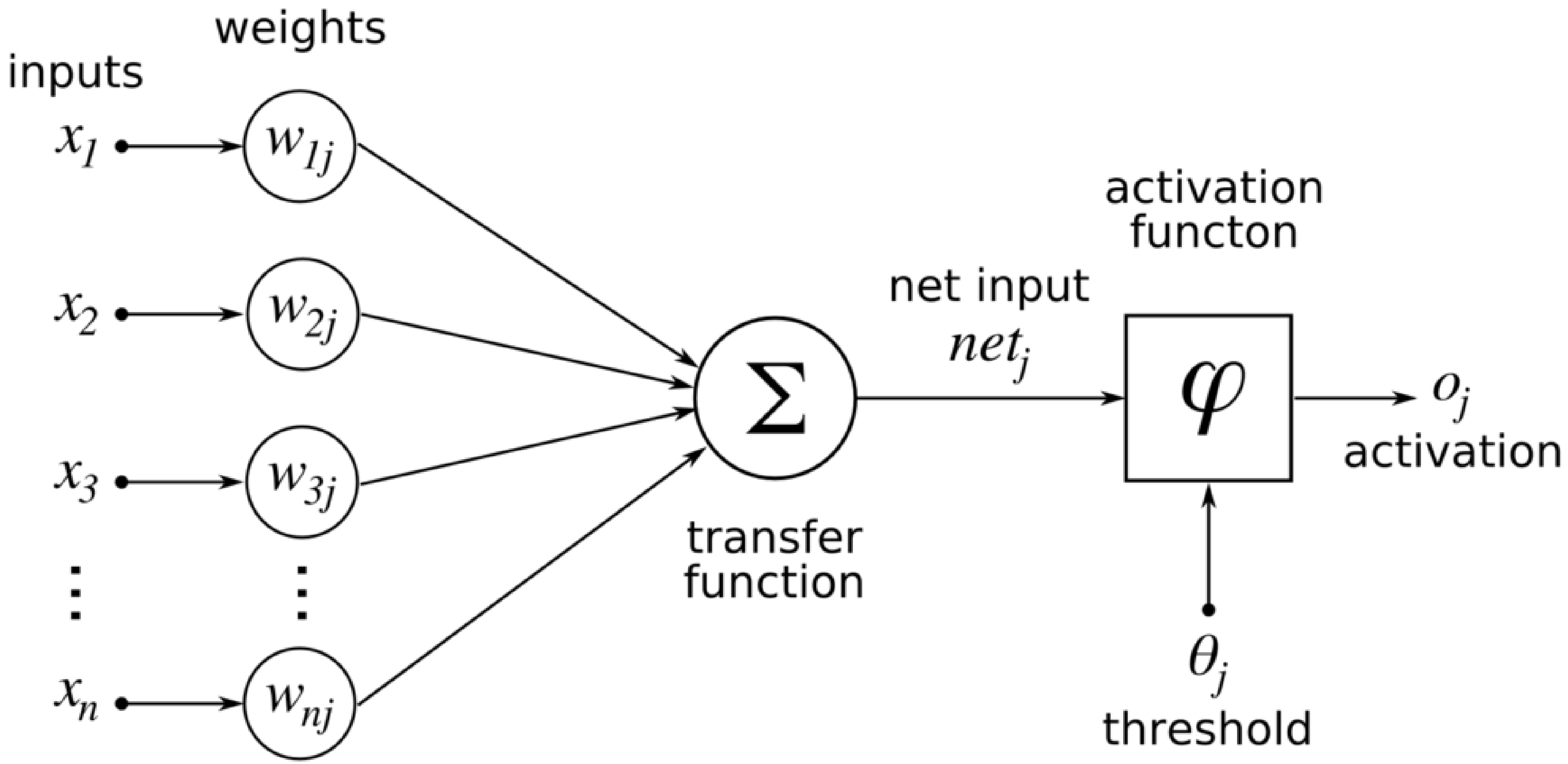
PHYSICS OF DATA - Department of Physics and Astronomy

COGNITIVE NEUROSCIENCE AND CLINICAL NEUROPSYCHOLOGY - Department of Psychology

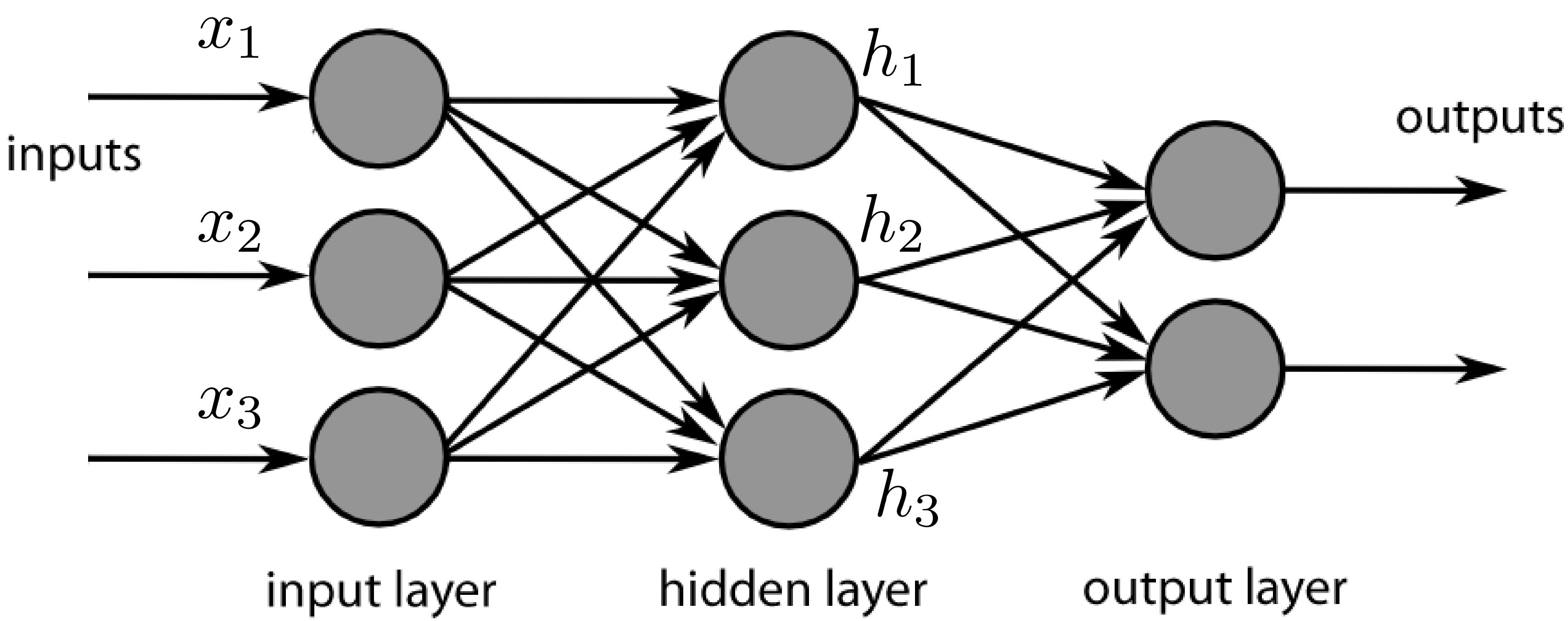
A.A. 2019/20 (6 CFU)

Dr. Alberto Testolin,
Dr. Federico Chiariotti (chiariot@dei.unipd.it)

Artificial neuron



Fully connected feed-forward network



$$x_1 w_{11} + x_2 w_{21} + x_3 w_{31} + b_1 = h_1$$
$$x_1 w_{12} + x_2 w_{22} + x_3 w_{32} + b_2 = h_2$$
$$x_1 w_{13} + x_2 w_{23} + x_3 w_{33} + b_3 = h_3$$

Fully connected feed-forward network

$$\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \\ w_{13} & w_{23} & w_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} =$$

$$= \begin{bmatrix} w_{11} & w_{21} & w_{31} & b_1 \\ w_{12} & w_{22} & w_{32} & b_2 \\ w_{13} & w_{23} & w_{33} & b_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix}$$

$H \qquad W_h \qquad X$

$$\boxed{H = W_h X}$$

Fully connected feed-forward network

- Network with 2 hidden layer (regression)

$$Y = W_o\ \varphi(W_{h2}\ \varphi(W_{h1}X))$$

$$\boxed{\begin{aligned} H_1 &= W_{h1}X \\ Z_1 &= \varphi(H_1) \\ H_2 &= W_{h2}Z_1 \\ Z_2 &= \varphi(H_2) \\ Y &= W_oZ_2 \end{aligned}}$$

Activation function (e.g. sigmoid)

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

$$\varphi'(x) = \varphi(x)(1 - \varphi(x))$$

Gradient Descent

- Loss function to minimize (MSE, like OLS)

Network output True value

$$l = \frac{1}{2}(Y - \hat{Y})^2 \qquad l'(Y) = (Y - \hat{Y})$$

Learning rate

$$\boxed{W_o \leftarrow W_o - \lambda \frac{\partial l}{\partial W_o}} \quad \boxed{W_{h1} \leftarrow W_{h1} - \lambda \frac{\partial l}{\partial W_{h1}}} \quad \boxed{W_{h2} \leftarrow W_{h2} - \lambda \frac{\partial l}{\partial W_{h2}}}$$

Chain rule refresh

$$f(x) = g(h(z(x)))$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial h} \cdot \frac{\partial h}{\partial z} \cdot \frac{\partial z}{\partial x}$$

Gradient Descent

$$Y = W_o \, \varphi(W_{h2} \, \varphi(W_{h1} X)) \qquad l = \frac{1}{2}(Y - \hat{Y})^2$$

$$\begin{aligned} H_1 &= W_{h1} X \\ Z_1 &= \varphi(H_1) \\ H_2 &= W_{h2} Z_1 \\ Z_2 &= \varphi(H_2) \\ Y &= W_o Z_2 \end{aligned}$$

$$\frac{\partial l}{\partial W_o} = \frac{\partial l}{\partial Y} \frac{\partial Y}{\partial W_o}$$

$$\frac{\partial l}{\partial W_{h2}} = \frac{\partial l}{\partial Y} \frac{\partial Y}{\partial Z_2} \frac{\partial Z_2}{\partial H_2} \frac{\partial H_2}{\partial W_{h2}}$$

$$\frac{\partial l}{\partial W_{h2}} = \frac{\partial l}{\partial Y} \frac{\partial Y}{\partial Z_2} \frac{\partial Z_2}{\partial H_2} \frac{\partial H_2}{\partial Z_1} \frac{\partial Z_1}{\partial H_1} \frac{\partial H_1}{\partial X}$$

Gradient Descent

$$D_1 = \frac{\partial l}{\partial Y} = Y - \hat{Y}$$

$$D_2 = \frac{\partial Y}{\partial W_o} = Z_2$$

$$D_3 = \frac{\partial Y}{\partial Z_2} = W_o$$

$$D_4 = \frac{\partial Z_2}{\partial H_2} = \varphi'(H_2)$$

$$D_5 = \frac{\partial H_2}{\partial W_{h2}} = Z_1$$

$$D_6 = \frac{\partial H_2}{\partial Z_1} = W_{h2}$$

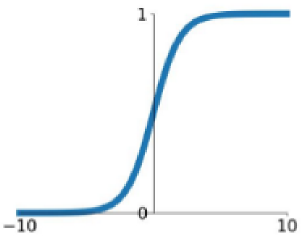
$$D_7 = \frac{\partial Z_1}{\partial H_1} = \varphi'(H_1)$$

$$D_8 = \frac{\partial H_1}{\partial X} = X$$

Activation Functions

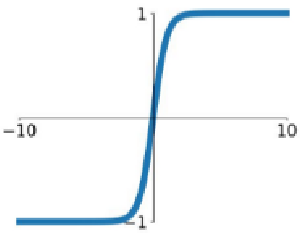
Sigmoid

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



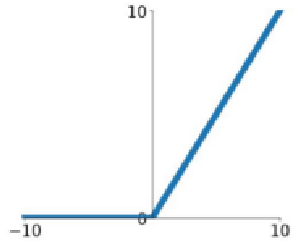
tanh

$\tanh(x)$



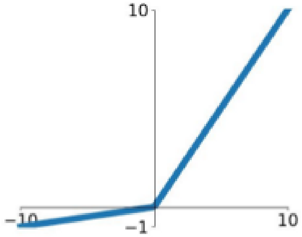
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

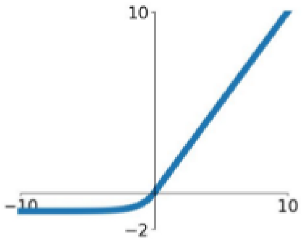


Maxout

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

ELU

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



$$\frac{d}{dx} tanh(x) = 1 - tanh(x)^2$$

$$\frac{d}{dx} ReLU(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$$

Homework: function estimation contest

1. **Download the dataset from the Elearning page**
2. **Train your neural network (no cheating → don't use pre-built frameworks)**
3. **Write the report**
4. **We will run a test on a hidden set of values to create the rankings**

Don't worry, the rankings are just for fun!

Report and homework

1. 3-page report
   -Brief introduction
   -Parameter search (parameters you tried, activation function…)
   -Final model (network structure, activation function, loss, etc.)
   -Plots and figures are appreciated
2. Code used to train the neural network
3. Testing script: you should provide a script with the trained model, so that running python trained_model.py outputs the MSE of the points in the file test_set.py)