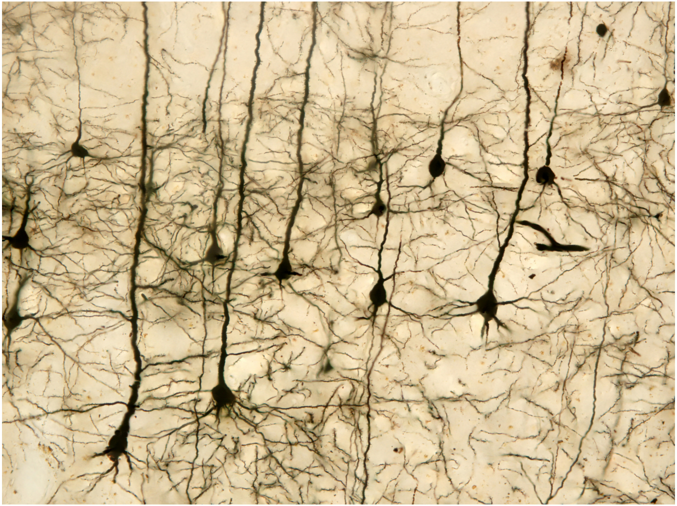
# Artificial Intelligence and Automation

## Unit 2: Artificial Neural Networks ANN

Ph.D. Gerardo Marx Chávez-Campos

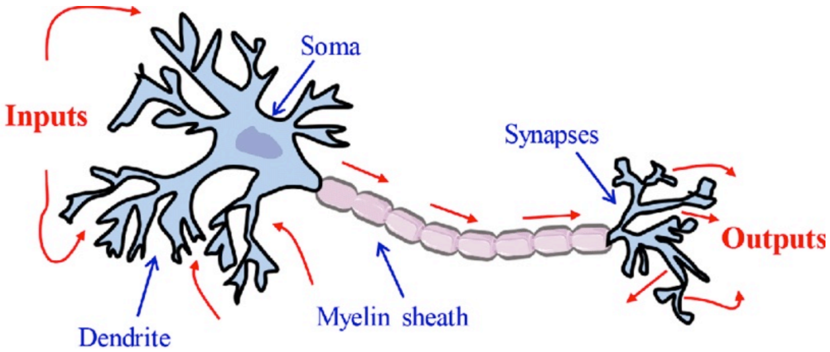
Instituto Tecnológico de Morelia: Ing. Mecatrónica

# Artificial Neuron

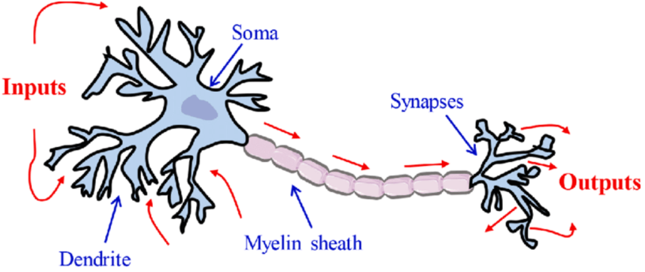
 1

[1] Derechos de autor: Copyright (c) 2020 Jose Luis Calvo/Shutterstock.

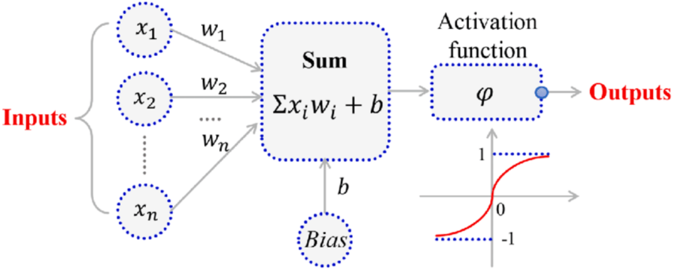
# Artificial Neuron



# Artificial Neuron

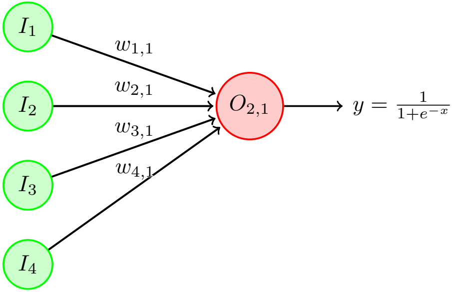



(a) Biological neuron




(b) Artificial neuron

# Artificial Neuron

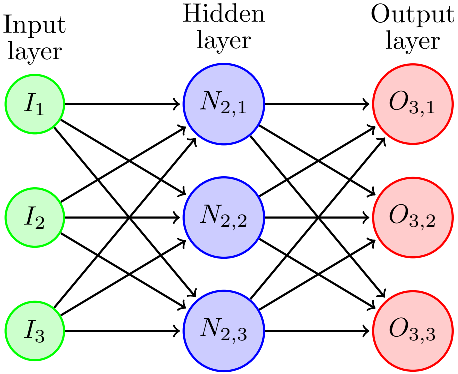



$$X = \sum_{i=1}^{n} I_i w_{ij} + b_i \quad (1)$$

$$O_i = \sigma(X) = \frac{1}{1 + e^{-x}} \quad (2)$$

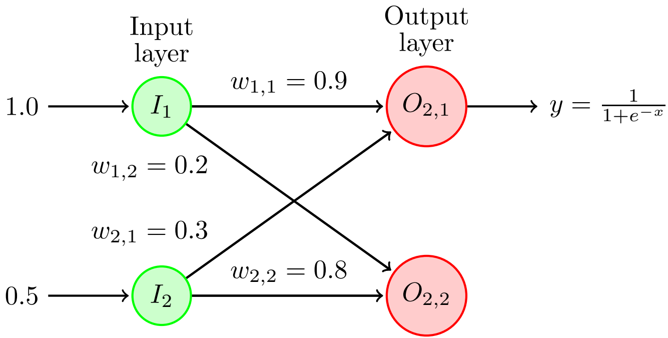
# Artificial Neural Networks

- Inputs and outputs
- Neuron and its activation function
- Weights and bias



# Signals in the Neural Network

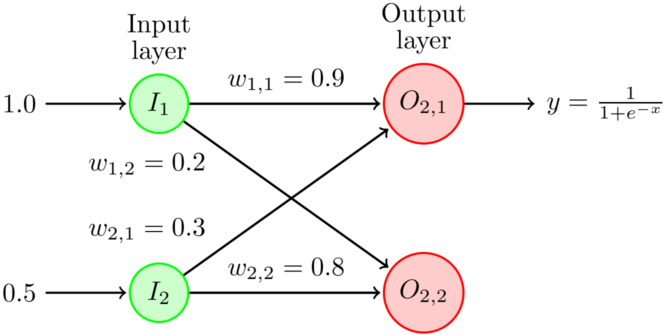
Considers the next example and compute first the output $O_{2,1}$




$$x = I_1 \cdot w_{1,1} + I_2 \cdot w_{2,1}$$

# Signals in the Neural Network

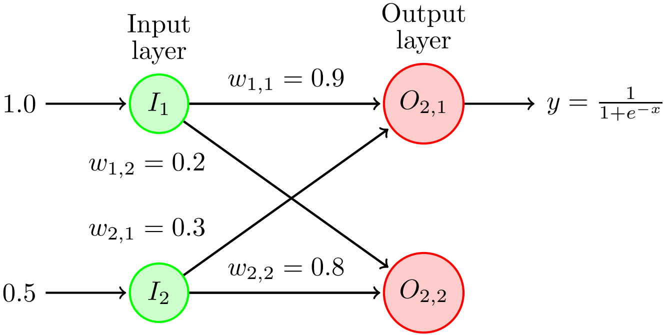
Considers the next example and compute first the output $O_{2,1}$




$$x = I_1 \cdot w_{1,1} + I_2 \cdot w_{2,1}$$
$$x = 1.0 \cdot 0.9 + 0.5 \cdot 0.3$$
$$x = 1.05$$

# Signals in the Neural Network

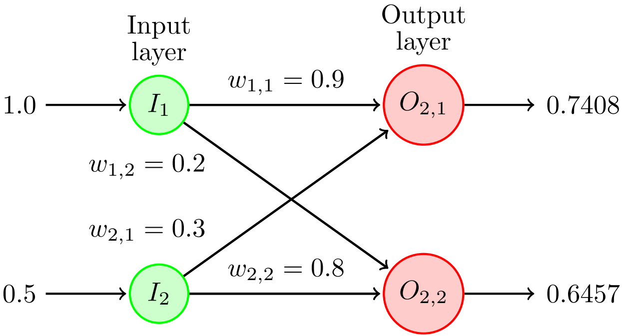
Considers the next example and compute first the output $O_{2,1}$




$$x = I_1 \cdot w_{1,1} + I_2 \cdot w_{2,1}$$
$$x = 1.0 \cdot 0.9 + 0.5 \cdot 0.3$$
$$x = 1.05$$
$$y = \frac{1}{1 + 0.3499} = 0.7407$$

# Signals in the Neural Network

## Results



# Matrix Multiplication

Then, $W$ is the matrix of weights, $I$ is the matrix of inputs, and $X$ is the resulting matrix of combined moderated signals into layer 2.

$$W \cdot I = X \tag{3}$$

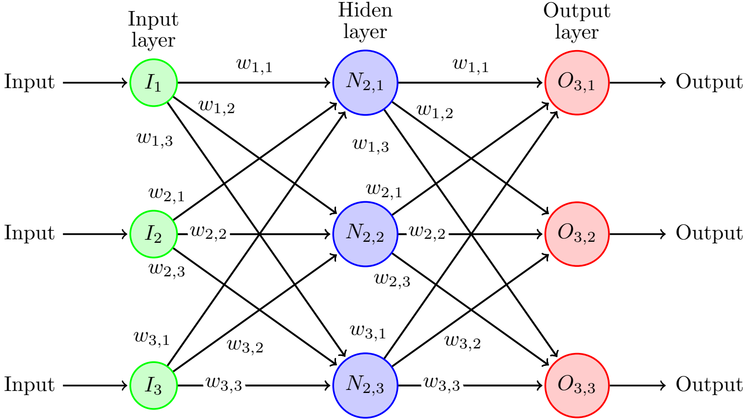
$$\begin{bmatrix} w_{1,1} & w_{2,1} \\ w_{1,2} & w_{2,2} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} (I_1 w_{1,1}) + (I_2 w_{2,1}) \\ (I_1 w_{1,2}) + (I_2 w_{2,2}) \end{bmatrix} \tag{4}$$

Finally, the output of the layer is:

$$O = \text{sigmoid}(X) \tag{5}$$

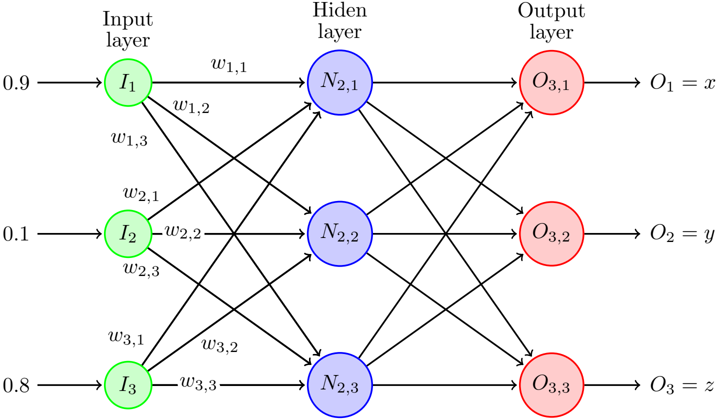
# A Three Layer Matrix Multiplication

## Terminology



# Three layer example

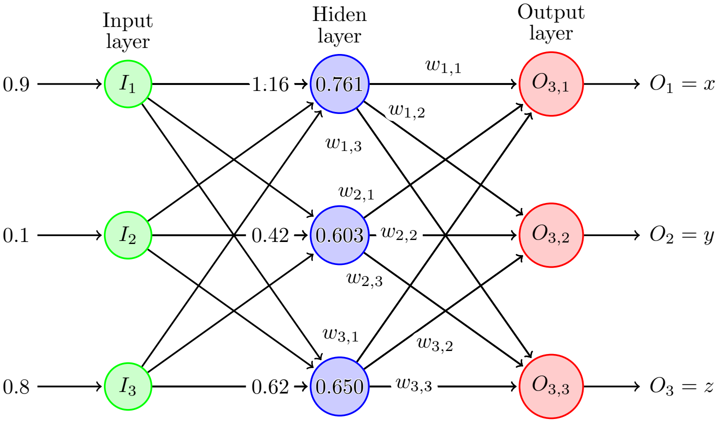
## Input-Hidden Layer




$$w_{11} = 0.9, \quad w_{12} = 0.2, \quad w_{13} = 0.1,$$
$$w_{21} = 0.3, \quad w_{22} = 0.8, \quad w_{23} = 0.5,$$
$$w_{31} = 0.4, \quad w_{32} = 0.2, \quad w_{33} = 0.6,$$

# Three layer example

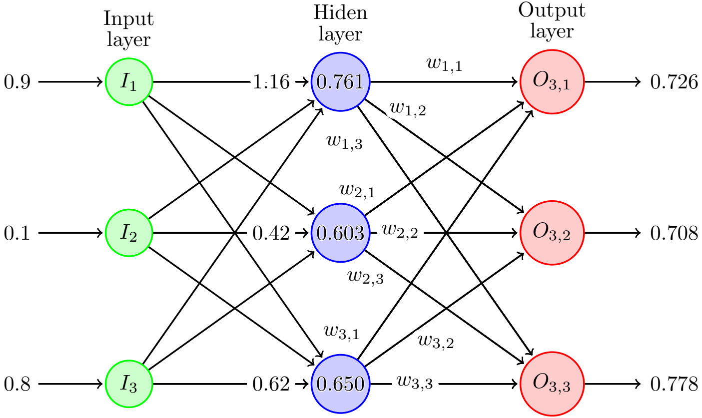
## Hidden-Output Layer



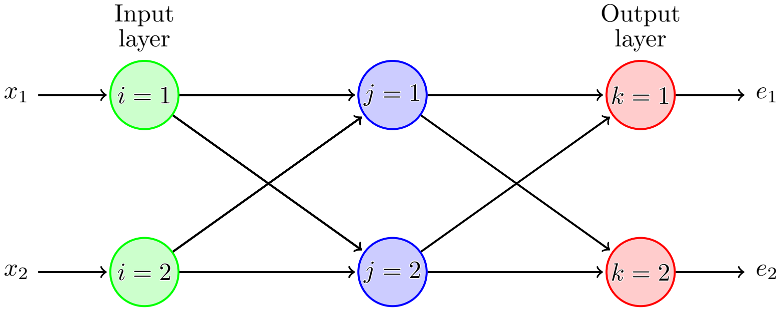

$$w_{11} = 0.3, \quad w_{12} = 0.6, \quad w_{13} = 0.8,$$
$$w_{21} = 0.7, \quad w_{22} = 0.5, \quad w_{23} = 0.2,$$
$$w_{31} = 0.5, \quad w_{32} = 0.2, \quad w_{33} = 0.9,$$

# Three layer example

## Resulting Output



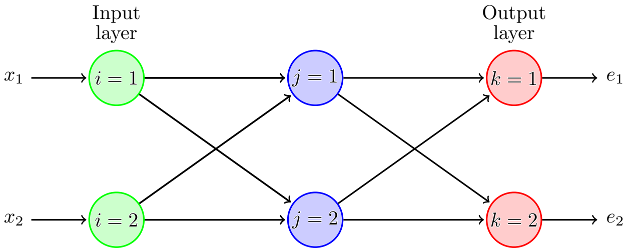
# The Error




$$e_k = t_k - o_k \tag{6}$$

# The Error

## Backpropagation




$$e_o = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{7}$$

$$e_h = \begin{bmatrix} \frac{w_{11}}{w_{11}+w_{21}} & \frac{w_{12}}{w_{12}+w_{22}} \\ \frac{w_{21}}{w_{21}+w_{11}} & \frac{w_{22}}{w_{22}+w_{12}} \end{bmatrix} \tag{8}$$

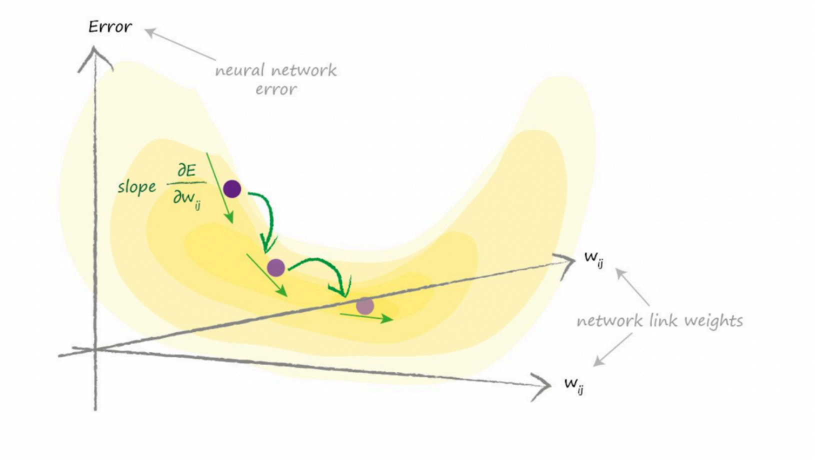
$$e_h = \begin{bmatrix} \frac{w_{11}}{w_{11}+w_{21}} & \frac{w_{12}}{w_{12}+w_{22}} \\ \frac{w_{21}}{w_{21}+w_{11}} & \frac{w_{22}}{w_{22}+w_{12}} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{9}$$

$$e_h = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \tag{10}$$

$$e_h = W_{ho}^T \cdot e_o \tag{11}$$

# The Error

## Gradient concept




$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} \sum_n (t_n - o_n)^2 \tag{12}$$

# The Error

## Gradient formula

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} \sum_n (t_n - o_n)^2$$

- $o_n$ only depends on the links (weigths) connected to it
- $o_k$ depends on $w_{jk}$

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} (t_k - o_k)^2 \tag{13}$$

# The Error

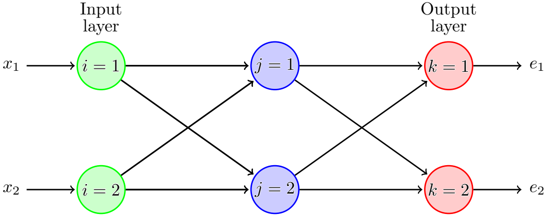
## Gradient formula

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}}(t_k - o_k)^2$$

- $t_k$ is constant
- $o_k$ depends on $w_{jk}$

# The Error

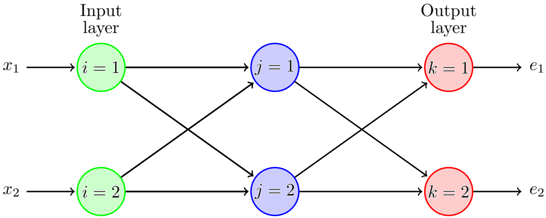
## Gradient formula: Chain Rule




$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial o_k} \cdot \frac{\partial o_k}{\partial w_{jk}}$$

# The Error

## Gradient formula: Chain Rule




$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial o_k} \cdot \frac{\partial o_k}{\partial w_{jk}}$$

$$\frac{\partial E}{\partial w_{jk}} = -2(t_k - o_k) \cdot \frac{\partial}{\partial w_{jk}} \sigma \left( \sum_j w_{jk} o_j \right)$$

**$o_j$ is the output of the previous hidden layer node; the input of the current layer!!**

# The Error

## Gradient formula: Chain Rule

$$\frac{\partial}{\partial x}\sigma\left(x\right) = \sigma(x)\left(1 - \sigma(x)\right)\frac{\partial x}{\partial x}$$

# The Error

## Gradient formula: Chain Rule

$$\frac{\partial}{\partial x}\sigma\left(x\right) = \sigma(x)\left(1 - \sigma(x)\right)\frac{\partial x}{\partial x}$$

$$\frac{\partial E}{\partial w_{jk}} = -2(t_k - o_k) \cdot \sigma\left(\sum_j w_{jk} o_j\right)\left(1 - \sigma\left(\sum_j w_{jk} o_j\right)\right) o_j$$

$$\frac{\partial E}{\partial w_{jk}} = -e_k \cdot \sigma\left(\sum_j w_{jk} o_j\right)\left(1 - \sigma\left(\sum_j w_{jk} o_j\right)\right) o_j$$

# The Error

## Updating weights

$$\frac{\partial E}{\partial w_{jk}} = -e_k \cdot \sigma\left(\sum_j w_{jk} o_j\right)\left(1 - \sigma\left(\sum_j w_{jk} o_j\right)\right) o_j \quad (14)$$

- $e_k$ is the error at the output
- The $\sigma(\sum_j)$ refers to the previous layers output; the hidden layer $j$
- $o_j$ is the output of the first layers of nodes (inputs)

$$\frac{\partial E}{\partial w_{ij}} = -e_j \cdot \sigma\left(\sum_i w_{ij} o_i\right)\left(1 - \sigma\left(\sum_i w_{ij} o_i\right)\right) o_i \quad (15)$$

# The Error

## Updating weights

$$w_{jk} = w_{jk} - \alpha \frac{\partial E}{\partial w_{jk}} \tag{16}$$

$$\begin{bmatrix} \Delta w_{11} & \Delta w_{12} & \cdots & \Delta w_{1k} \\ \Delta w_{21} & \Delta w_{22} & \cdots & \Delta w_{2k} \\ \dots & \dots & \dots & \dots \\ \Delta w_{j1} & \Delta w_{j2} & \cdots & \Delta w_{jk} \end{bmatrix} = \begin{bmatrix} E_1\sigma_1(1-\sigma_1) \\ E_2\sigma_2(1-\sigma_2) \\ \dots \\ E_k\sigma_k(1-\sigma_k) \end{bmatrix} \begin{bmatrix} o_1 & o_2 & \cdots & o_j \end{bmatrix} \tag{17}$$

- $k$ values from next layer
- $j$ values from previous layer

$$w_{new} = w_{old} - \alpha \frac{\partial E}{\partial w_{jk}} \tag{18}$$

# References

Rashid, Tariq. Make your own neural network. CreateSpace Independent Publishing Platform, 2016.