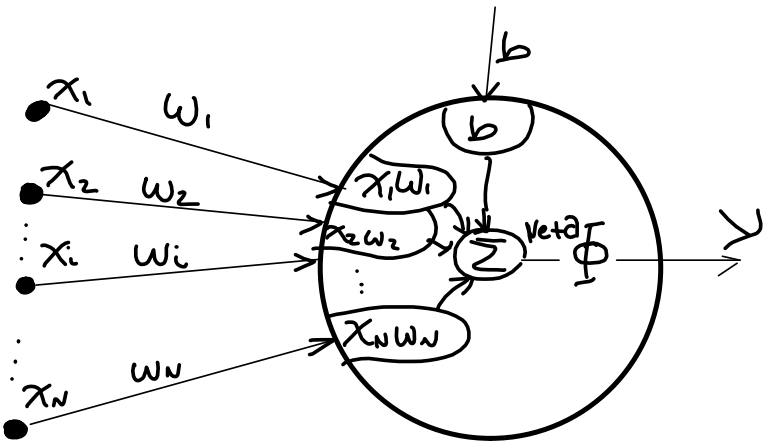
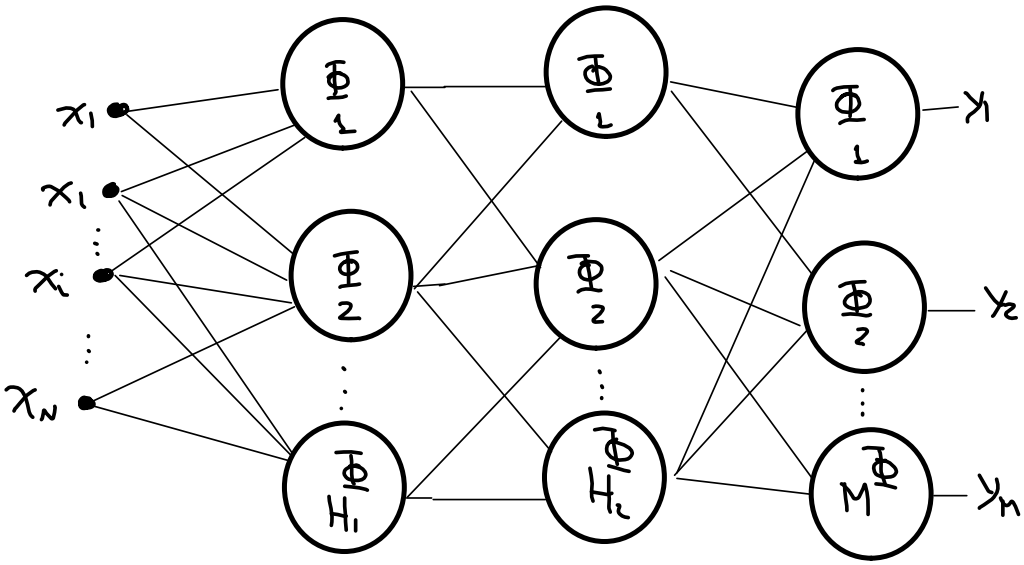


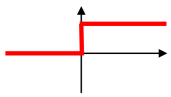
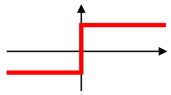
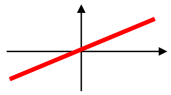
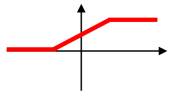
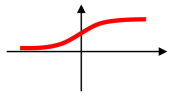
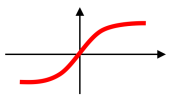
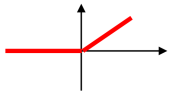
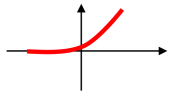
$$Neta = x_1w_1 + x_2w_2 + \dots + x_iw_i + \dots + x_Nw_N + b$$

$$= \sum_{i=1}^{N} x_iw_i + b$$

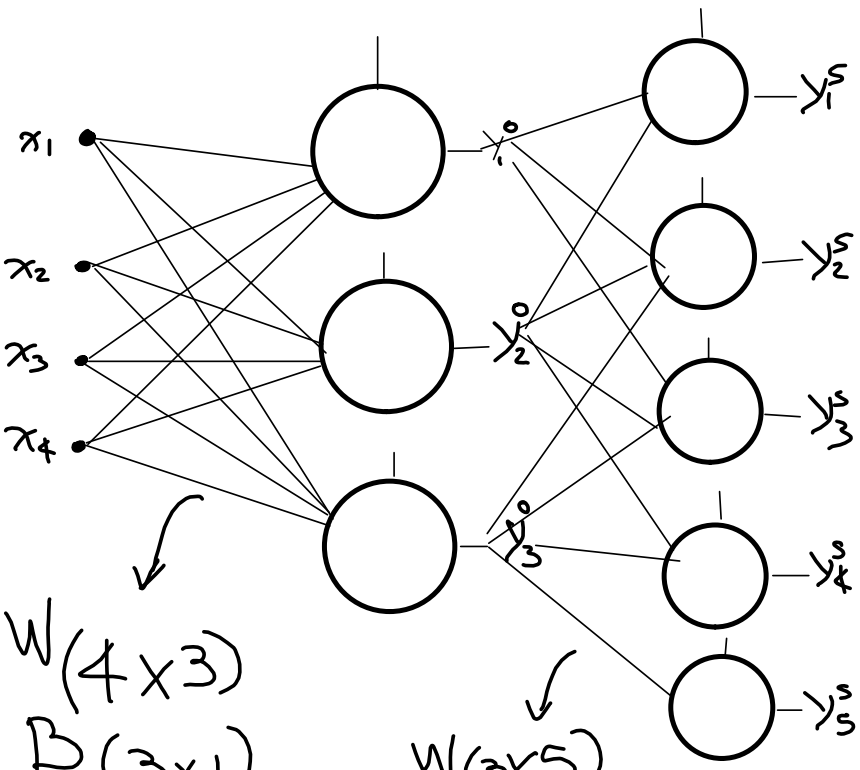
Φ = Función de activación

$$y = \Phi(Neta)$$


x_1
x_i
x_N
Φ_1^1
Φ_2^1
$\Phi_{H_1}^1$
Φ_1^2
Φ_2^2
$\Phi_{H_2}^2$
Φ_1^M
Φ_2^M
Φ_M^M
y_1
y_2
y_M

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 Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	

Copyright © Sebastian Raschka 2016 (http://sebastianraschka.com)



$W_{(4x3)}$

$B_{(3x1)}$

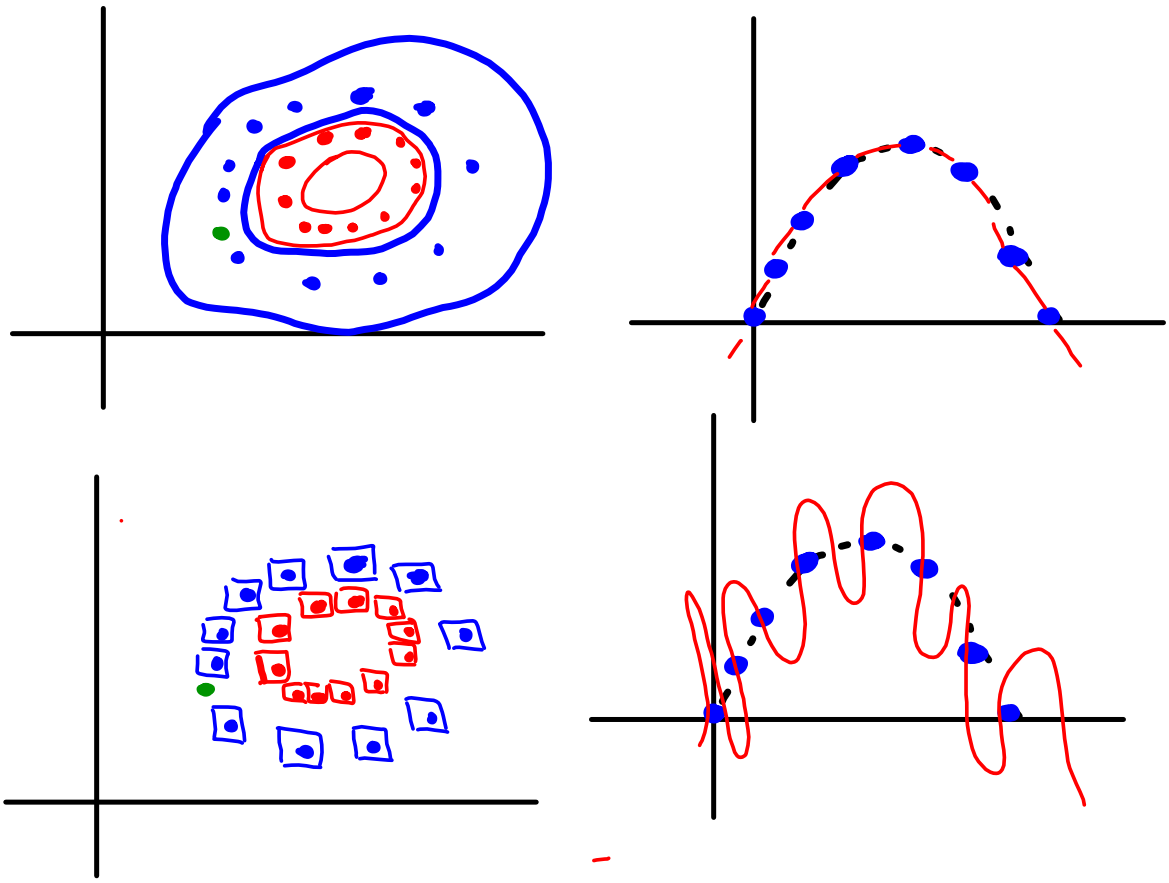
$X_{(4x1)}$

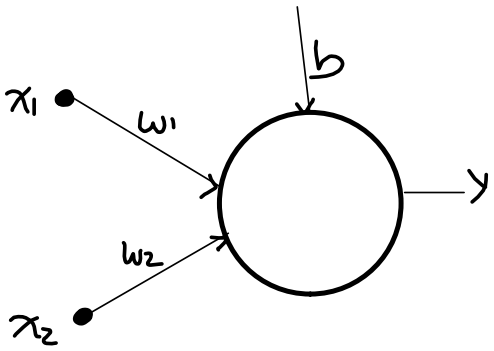
$W_{(3x5)}$

$B_{(5x1)}$

$X = Y_{anterior\ (3x1)}$

$$Net\,a_{(3x1)} = \underset{(3x4)}{W^T} \underset{(4x1)}{X} + B_{(3x1)}$$





$$Net_d = x_1 w_1 + x_2 w_2 + b = 0$$

$$\Rightarrow x_2 = -\frac{w_1}{w_2} x_1 - \frac{b}{w_2} \quad \Rightarrow y_k = m x_R + I_n$$

$$m = -\frac{w_1}{w_2}$$

$$I = -\frac{b}{w_2}$$

● = 0

● = 1