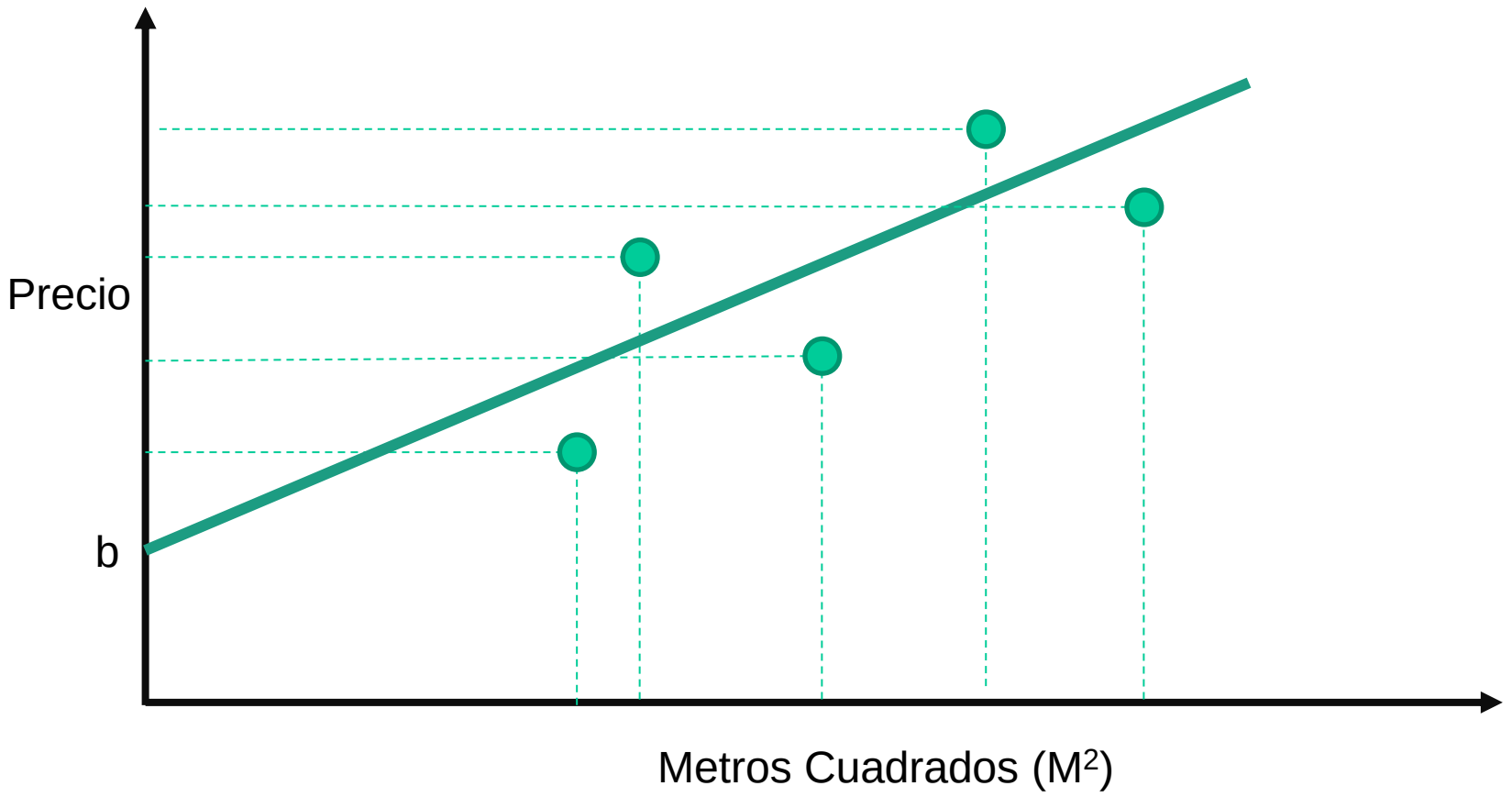
DATA SCIENCE

Redes Neuronales Profundas

UCEMA

Predecir el Precio de una Vivienda



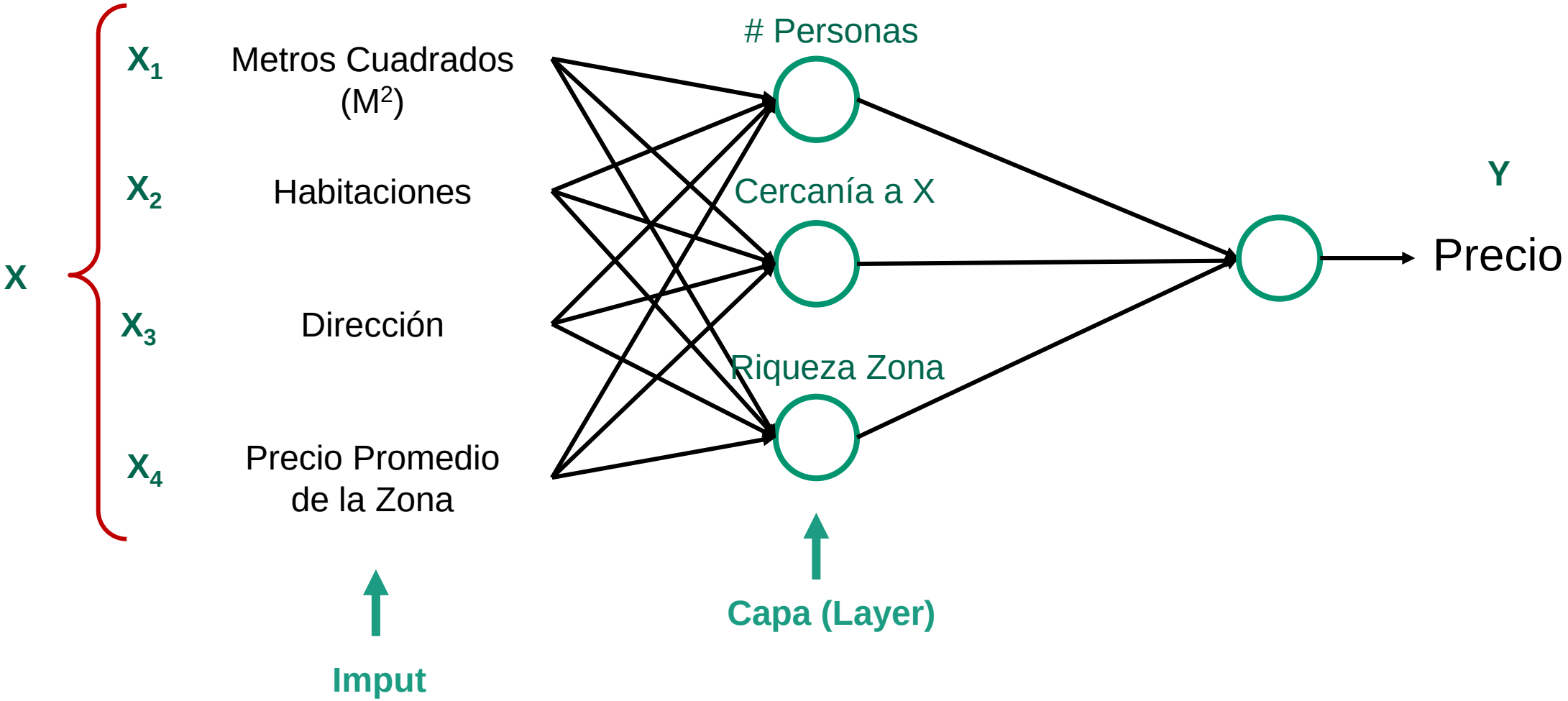
X Y

$M^2 * a + b = Precio$

M^2 → ○ → Precio

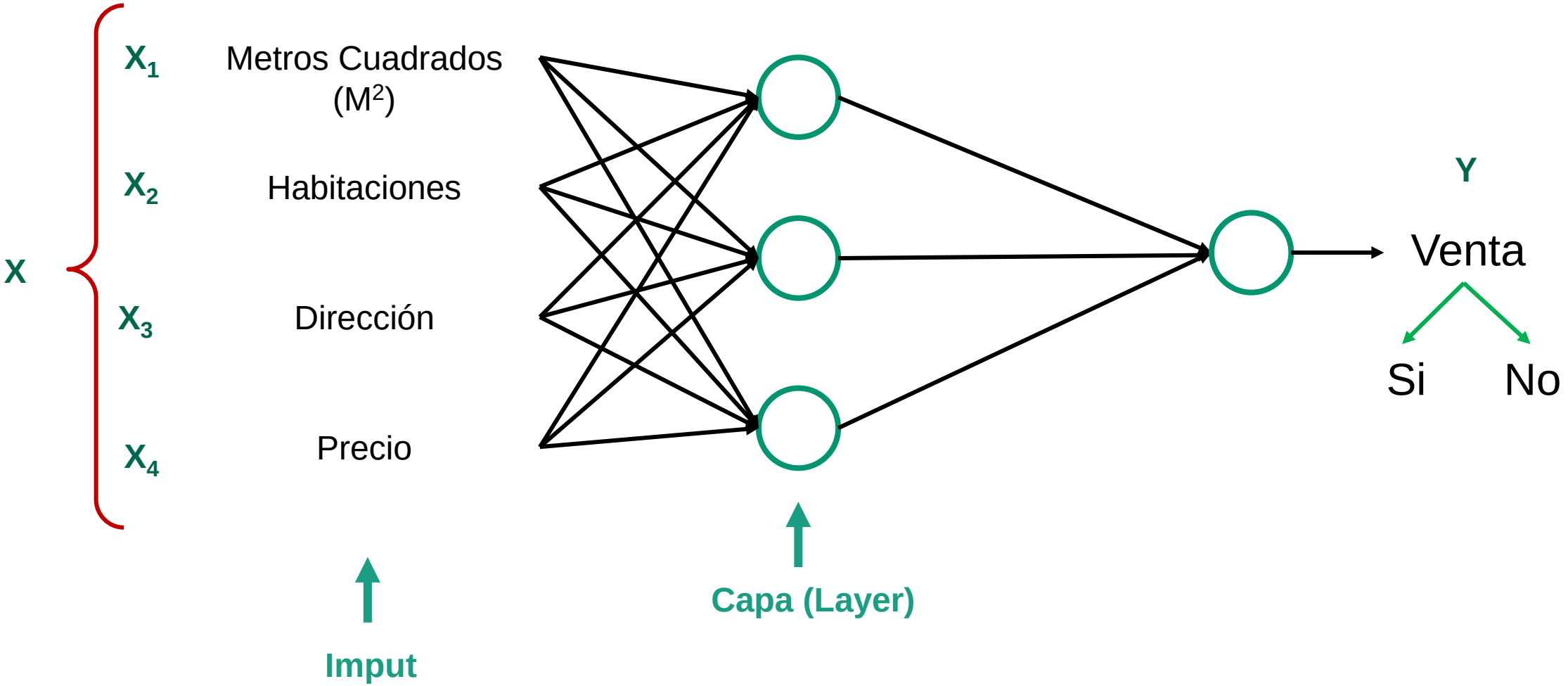
Neurona

Predecir el Precio de una Vivienda



Regresión Logística

Clasificador Binario



Regresión Logística

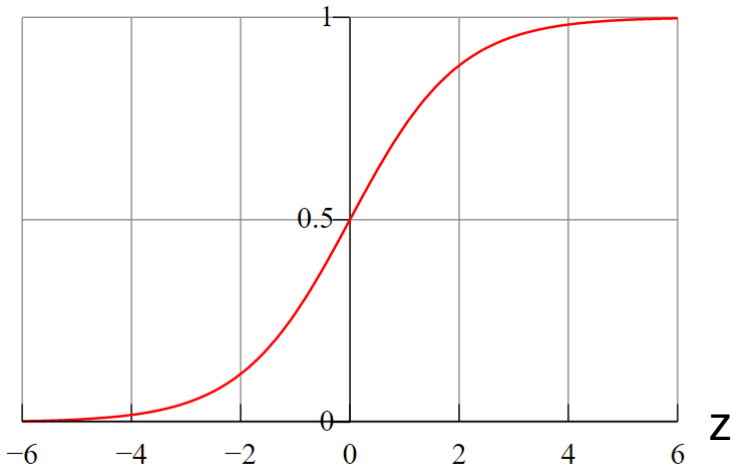
$$x * w + b = y \longrightarrow \text{¿ } \{0, 1\} \text{ ?}$$

$$\sigma(\underbrace{x * w + b}_{z}) = y$$

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

Si z es positivo y grande $\sigma(z) = \frac{1}{1 + 0} = 1$

Si z es negativo y grande $\sigma(z) = \frac{1}{1 + \text{grande}} = 0$



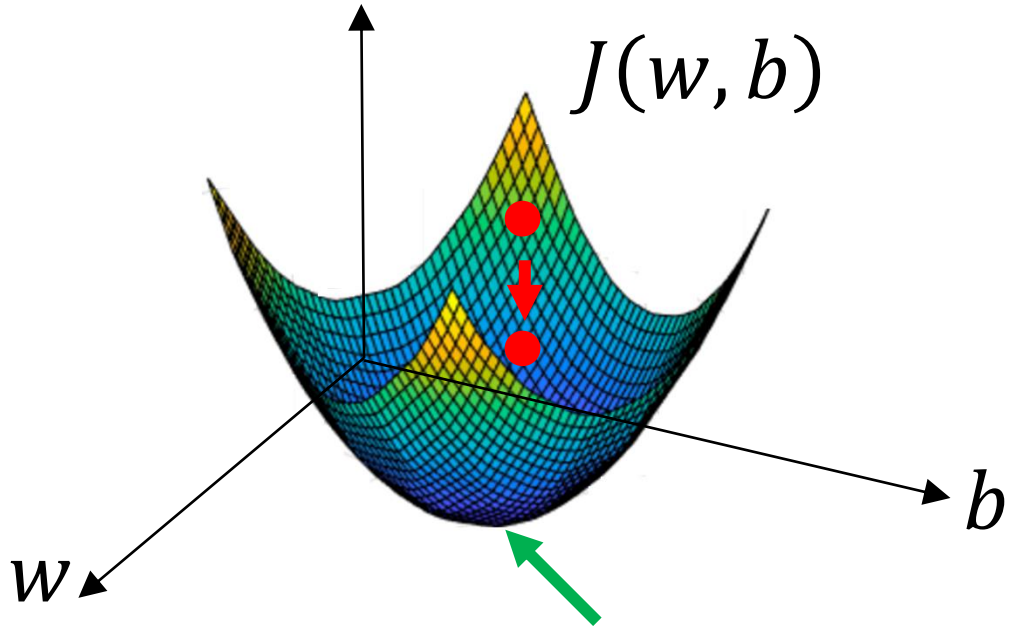
Pérdida Logarítmica (logloss, binary cross-entropy):

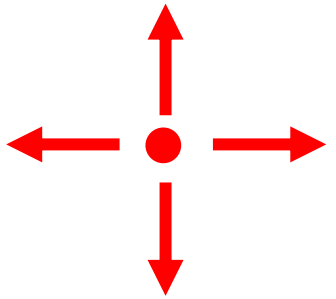
$$\mathcal{L}_i = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$$

Descenso del Gradiente

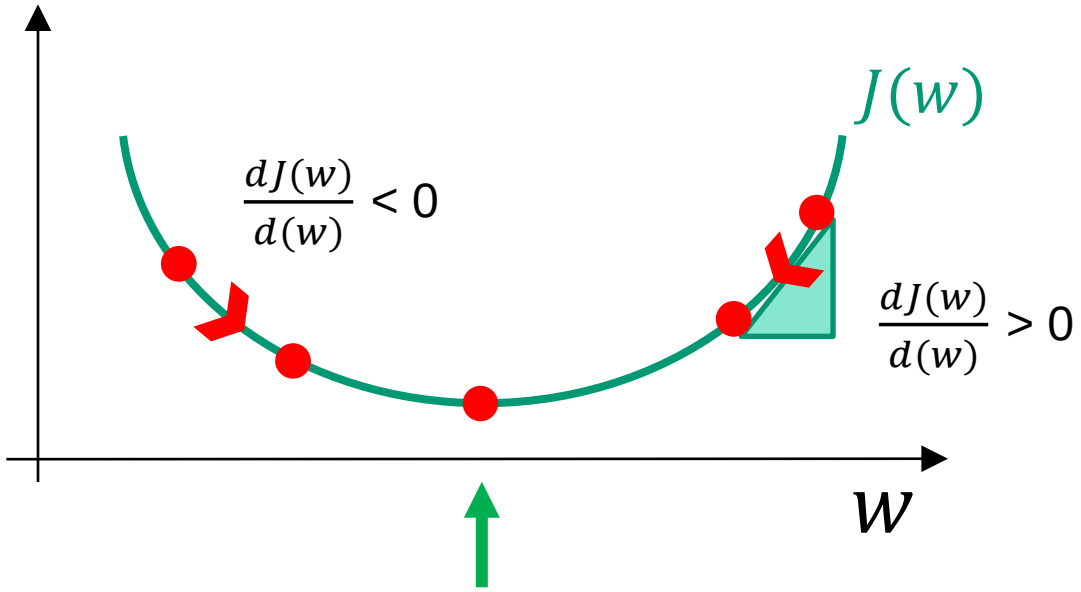
Descenso del Gradiente (Gradient Descent)

Función de Costo: $J(w,b) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\hat{y}_i, y_i)$





Descenso del Gradiente (Gradient Descent)



Ratio de Aprendizaje
(Learning Rate)

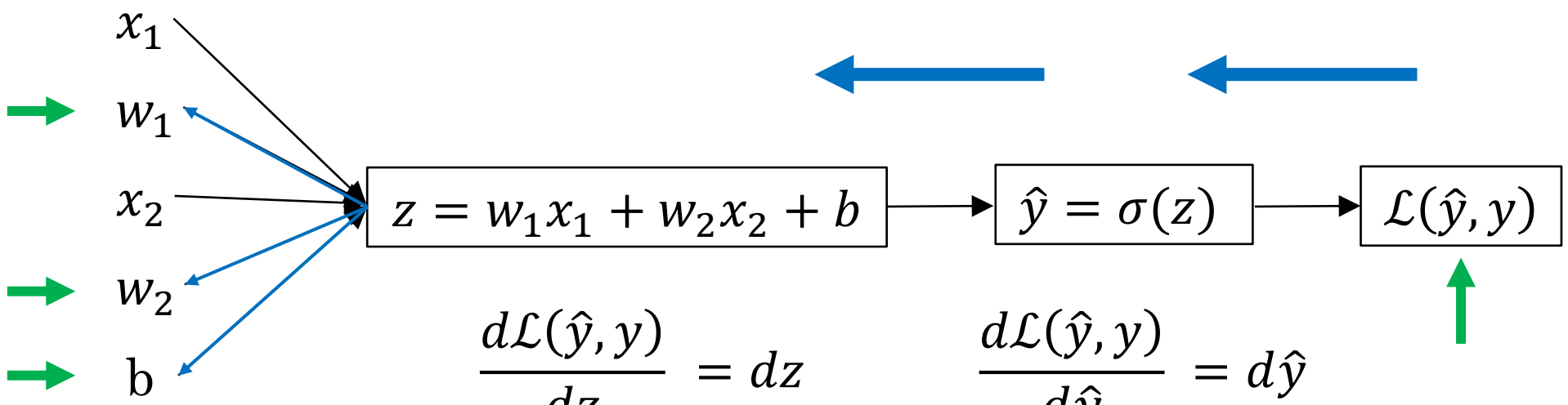
$$w := w - \alpha \underbrace{\frac{dJ(w)}{d(w)}}_{dw}$$

Iteramos:

$$w := w - \alpha \frac{dJ(w)}{d(w)}$$

Hasta minimizar

Calculo de las Derivadas en Regresión Logística



$$\frac{d\mathcal{L}(\hat{y},y)}{dz} = dz \qquad \frac{d\mathcal{L}(\hat{y},y)}{d\hat{y}} = d\hat{y}$$

$$\frac{d\mathcal{L}(\hat{y},y)}{dw_1} = dw_1 \qquad \frac{d\mathcal{L}(\hat{y},y)}{dw_2} = dw_2 \qquad \frac{d\mathcal{L}(\hat{y},y)}{db} = db$$

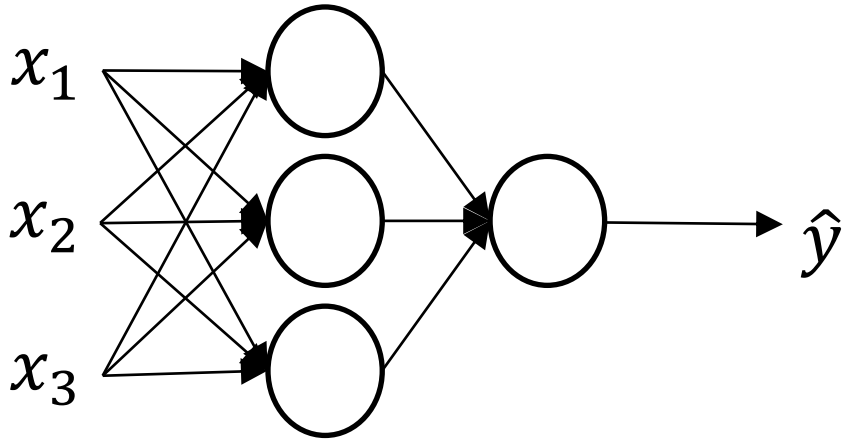
$$w_1 := w_1 - \alpha\, dw_1$$

$$w_2 := w_2 - \alpha\, dw_2$$

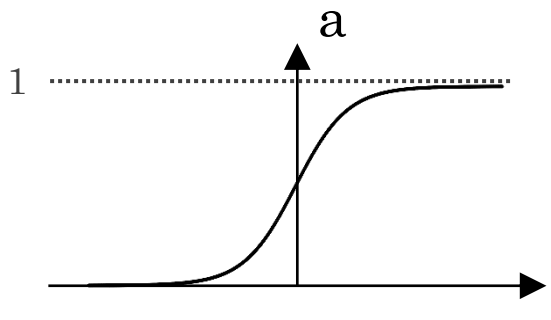
$$b := b - \alpha\, db$$

Funciones de Activación

Funciones de Activación

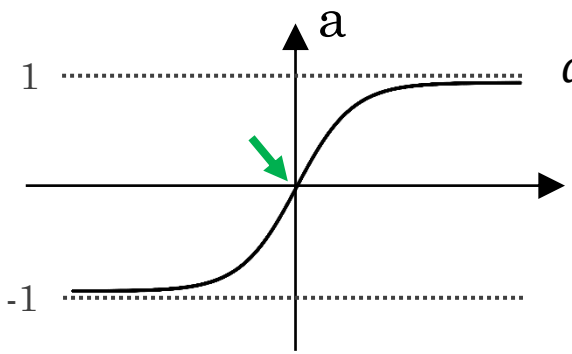


Sigmoid
e



$$a = \frac{1}{1 + e^{-z}}$$

Tanh



$$a = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

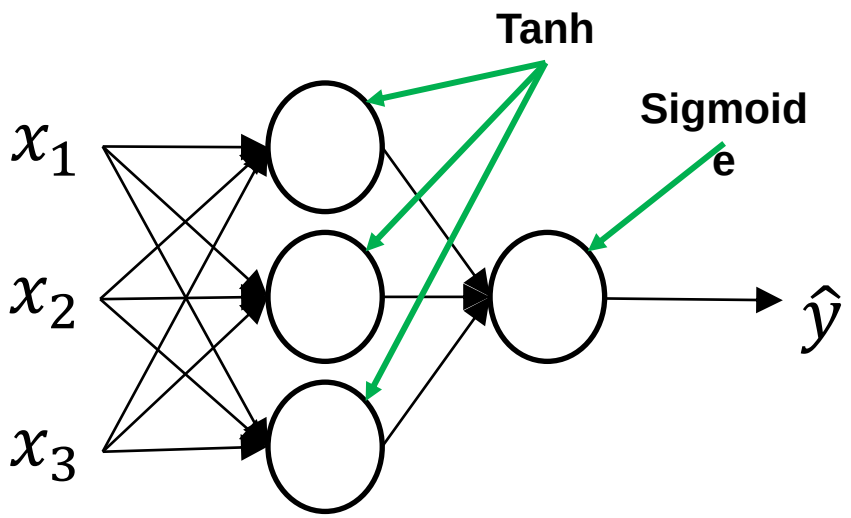
$$z^{[1]} = x^{t} W^{[1]} + b^{[1]}$$

$$a^{[1]} = \sigma(z^{[1]}) \quad g(z^{[1]})$$

$$z^{[2]} = a^{[1]} W^{[2]} + b^{[2]}$$

$$a^{[2]} = \sigma(z^{[2]}) \quad g(z^{[1]})$$

Funciones de Activación



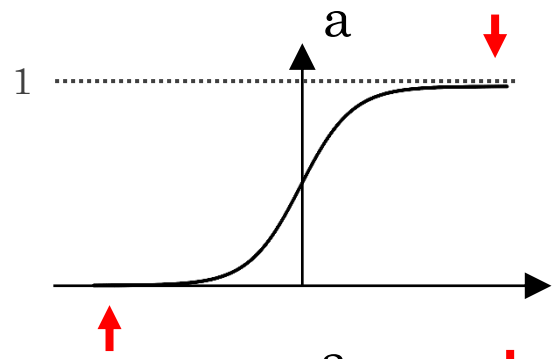
$$z^{[1]} = x^t W^{[1]} + b^{[1]}$$

$$a^{[1]} = \sigma(z^{[1]}) \quad g(z^{[1]})$$

$$z^{[2]} = a^{[1]} W^{[2]} + b^{[2]}$$

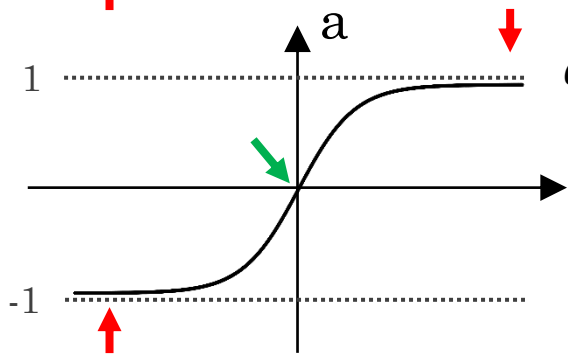
$$a^{[2]} = \sigma(z^{[2]}) \quad g(z^{[1]})$$

Sigmoid
e



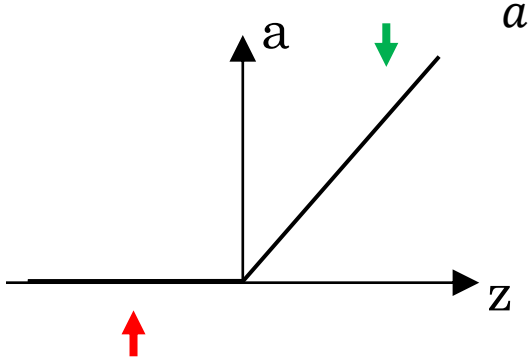
$$a = \frac{1}{1 + e^{-z}}$$

Tanh



$$a = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

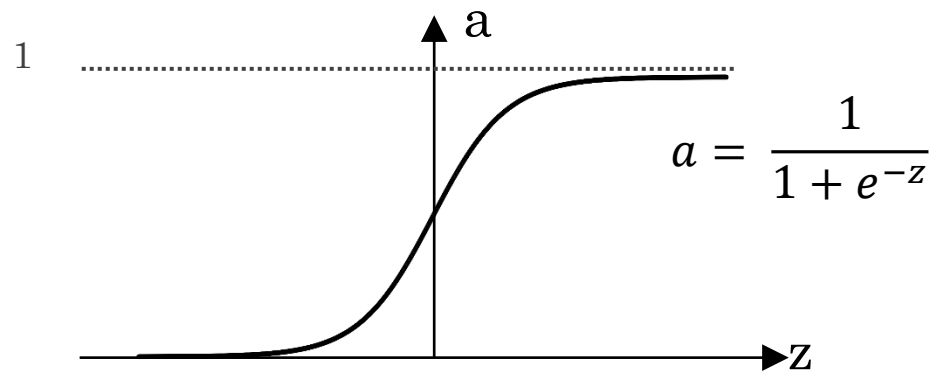
Relu



$$a = \max(0, z)$$

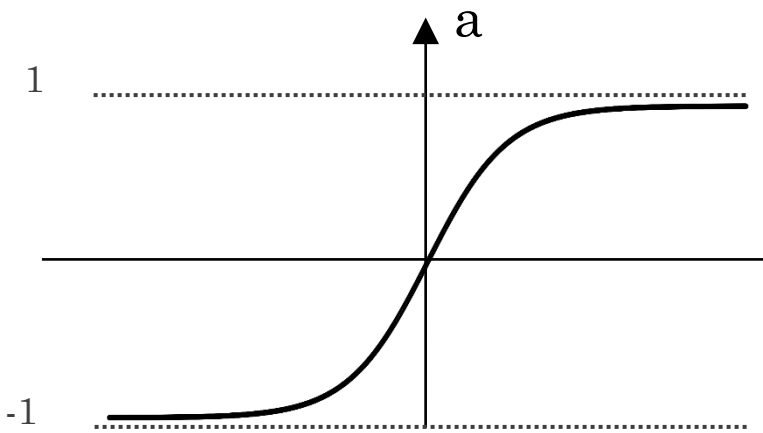
Repaso de las Funciones de Activación

Sigmoide (σ)



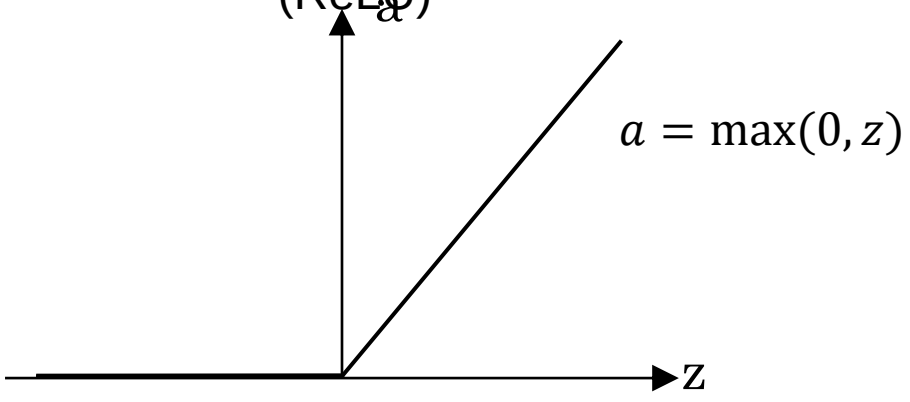
$$a = \frac{1}{1 + e^{-z}}$$

Tangente Hiperbólica (Tanh)



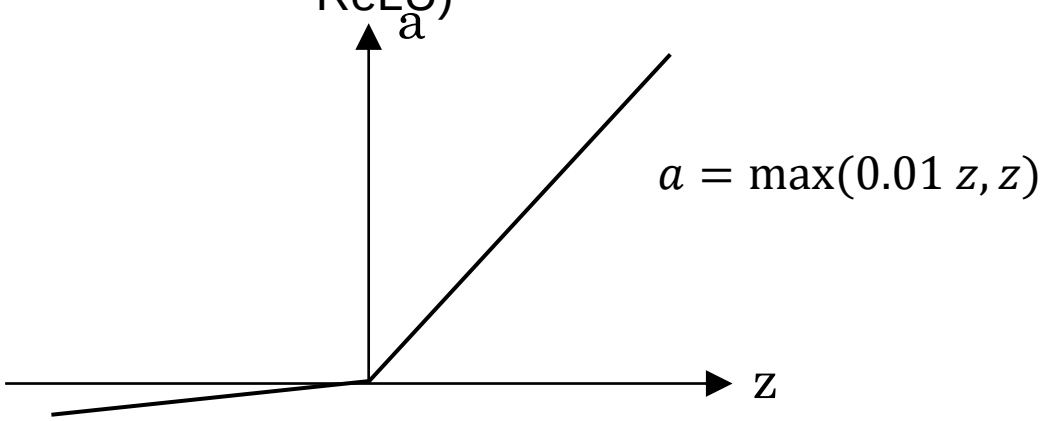
$$a = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Unidad Lineal Rectificada (ReLU)



$$a = \max(0, z)$$

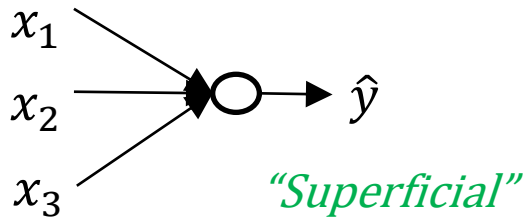
ReLU con “derrame” (leaky ReLU)



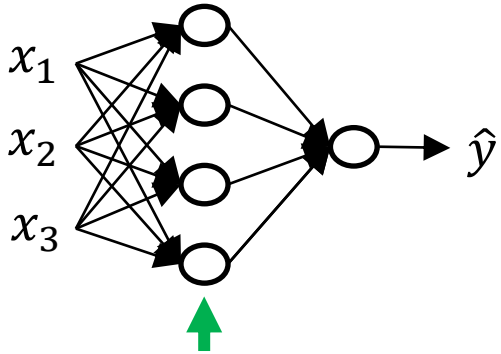
$$a = \max(0.01\ z, z)$$

Redes Neuronales Profundas

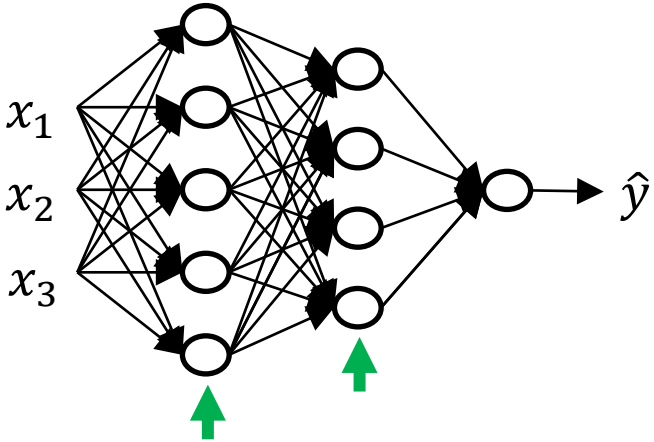
¿Qué son las Redes Neuronales Profundas?



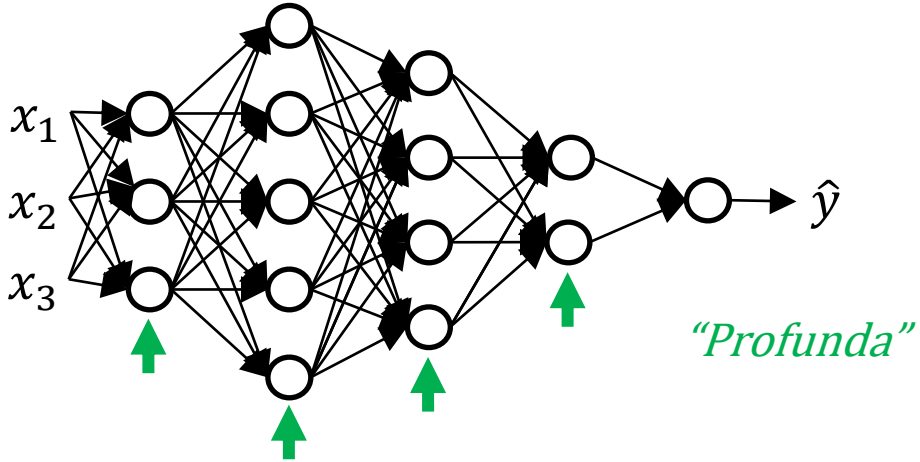
Regresión Logística



Red Neuronal de 1 Capa Oculta



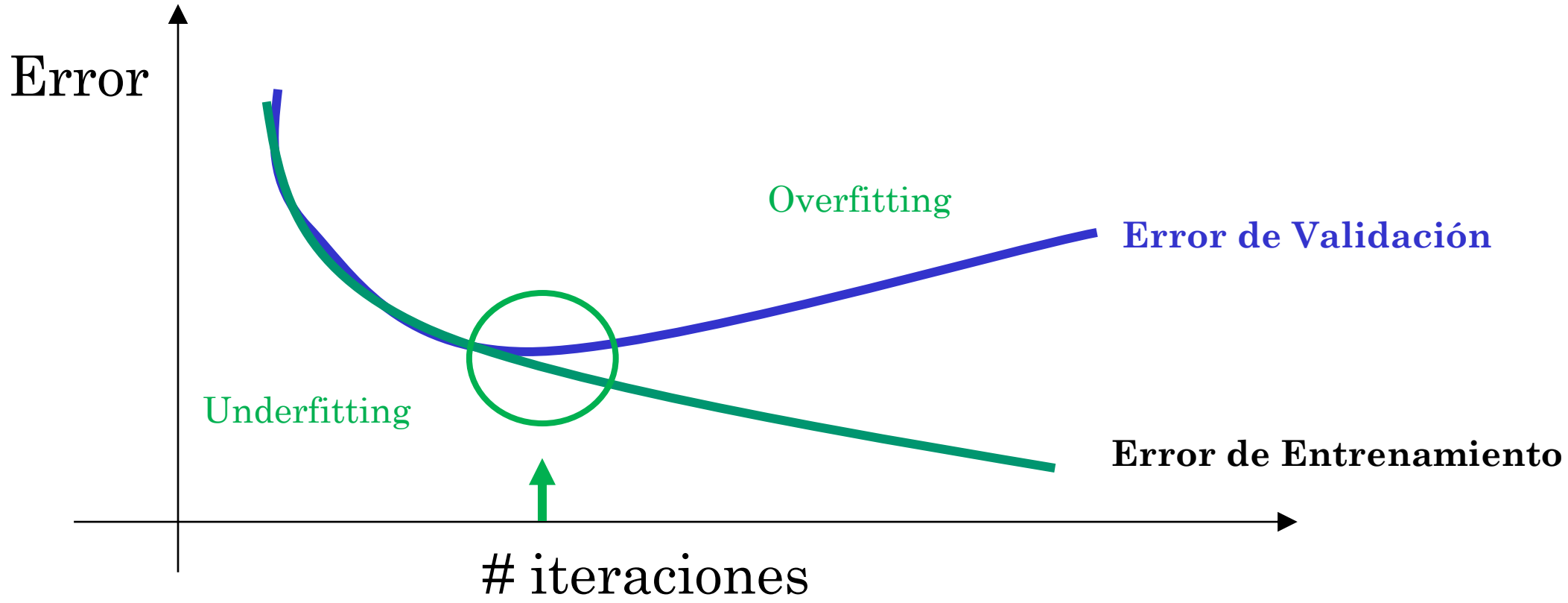
Red Neuronal de 2 Capa Oculta



Red Neuronal de 4 Capa Oculta

Regularización

- Parada Anticipada (*Early Stoping*)

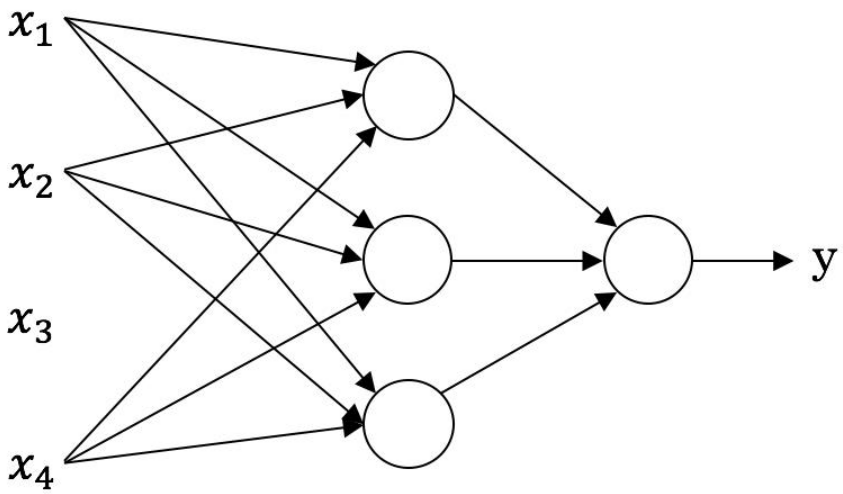


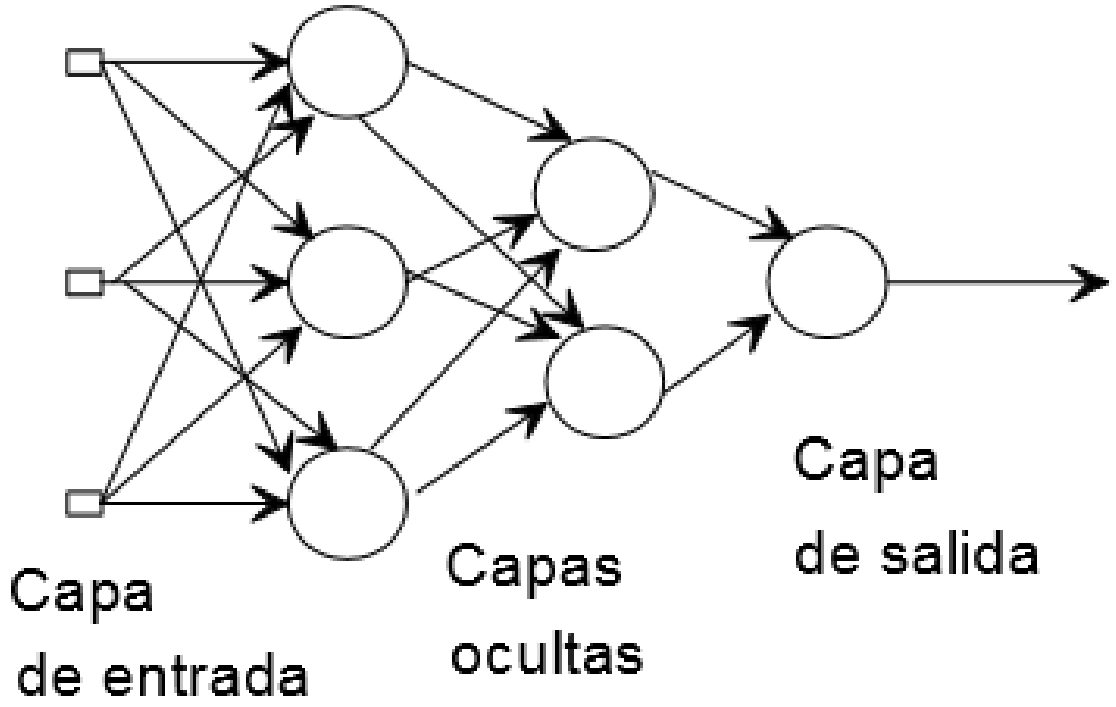
Aprendizaje Supervisado

Entrada (x)	Salida (y)	Aplicación	
Características de la Casa	Precio	Bienes Raíces	MLP o Red Neuronal Estándar
Info. de publicidad y usuarios	Click en ad.	Marketing	
Imagen	Objectos contenidos	Clasificación de imágenes	CNN / Transformers
Audio	Transcripción	Reconocimiento de voz	RNN / Transformers
Texto en inglés	Texto en español	Traducción	

Ejemplos de Arquitecturas

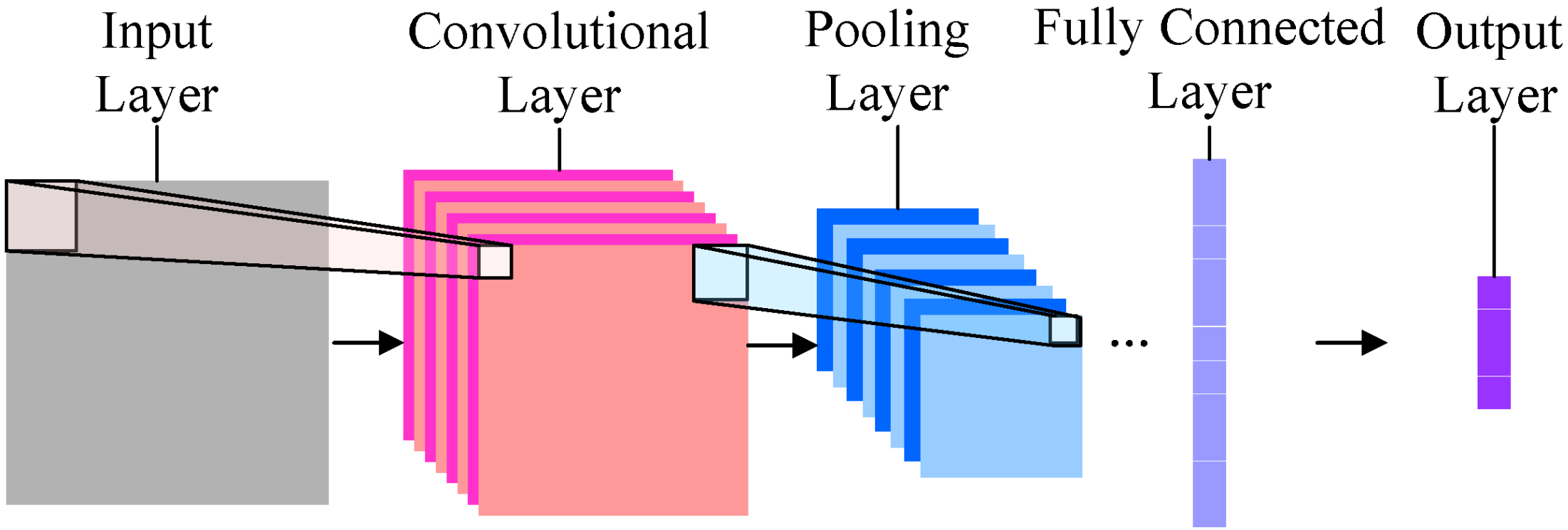
Multilayer Perceptron o Fully Connected Layer





Ejemplos de Arquitecturas

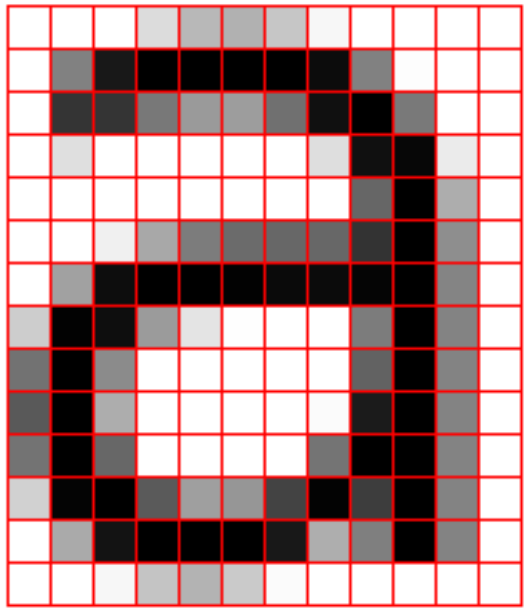
CNN (Redes Neuronales Convolucionales)



Ejemplos de Arquitecturas

CNN – El Input

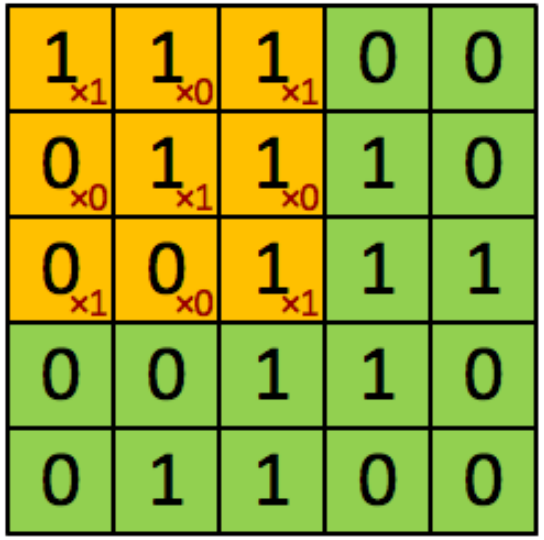
a



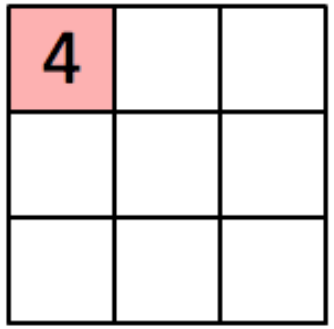
1.0	1.0	1.0	0.9	0.6	0.6	0.6	1.0	1.0	1.0	1.0	1.0
1.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0	1.0	1.0
1.0	0.2	0.2	0.5	0.6	0.6	0.5	0.0	0.0	0.5	1.0	1.0
1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.9	0.0	0.0	0.9	1.0
1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
1.0	1.0	1.0	0.5	0.5	0.5	0.5	0.5	0.4	0.0	0.5	1.0
1.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	1.0
0.9	0.0	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.6	1.0	1.0	1.0	1.0	1.0	0.5	0.0	0.5	1.0
0.5	0.0	0.7	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.5	1.0
0.6	0.0	0.6	1.0	1.0	1.0	1.0	0.5	0.0	0.0	0.5	1.0
0.9	0.1	0.0	0.6	0.7	0.7	0.5	0.0	0.5	0.0	0.5	1.0
1.0	0.7	0.1	0.0	0.0	0.0	0.1	0.9	0.8	0.0	0.5	1.0
1.0	1.0	1.0	0.8	0.8	0.9	1.0	1.0	1.0	1.0	1.0	1.0

Ejemplos de Arquitecturas

CNN – Kernel de Convolución



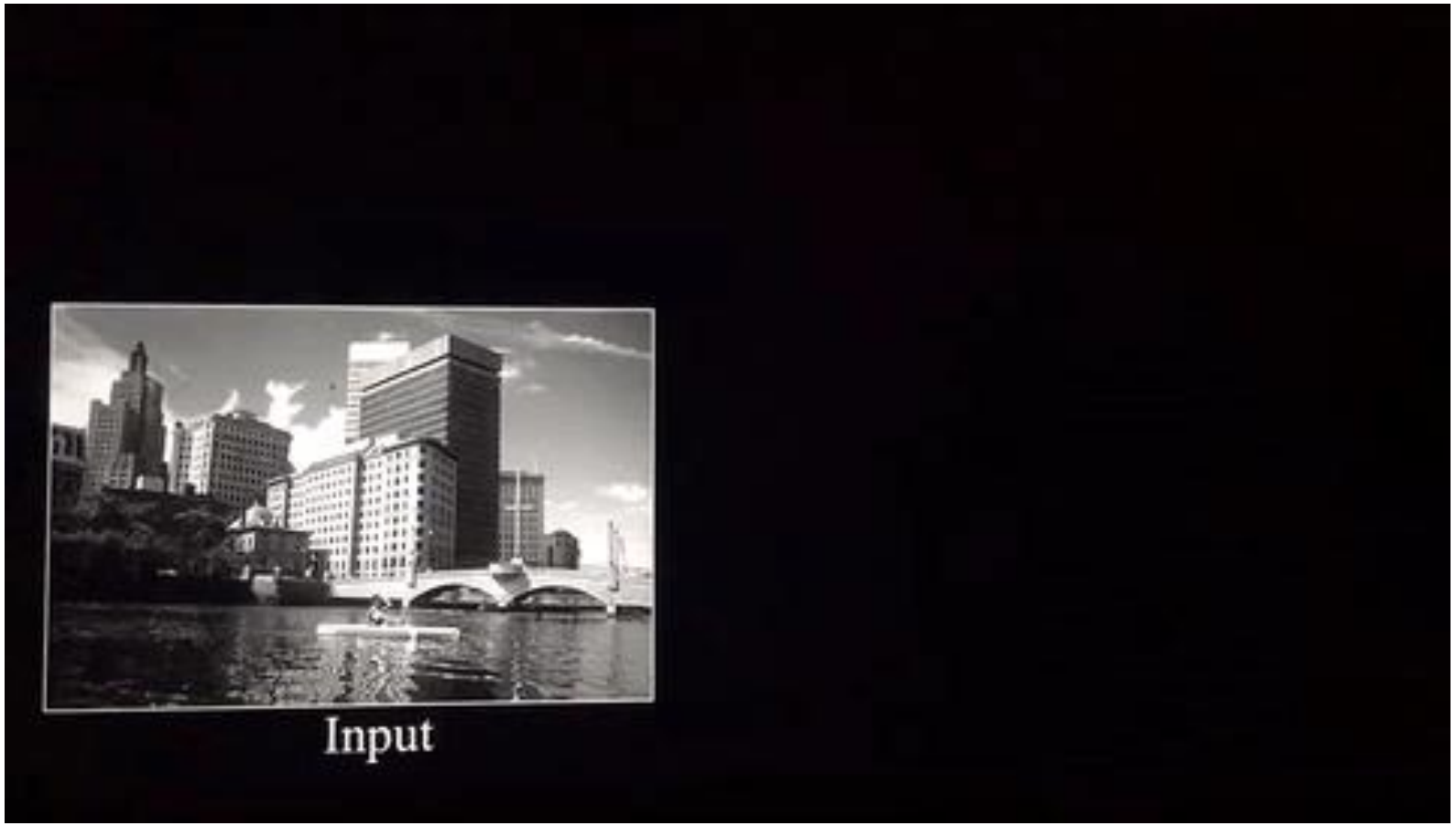
Image



Convolved Feature

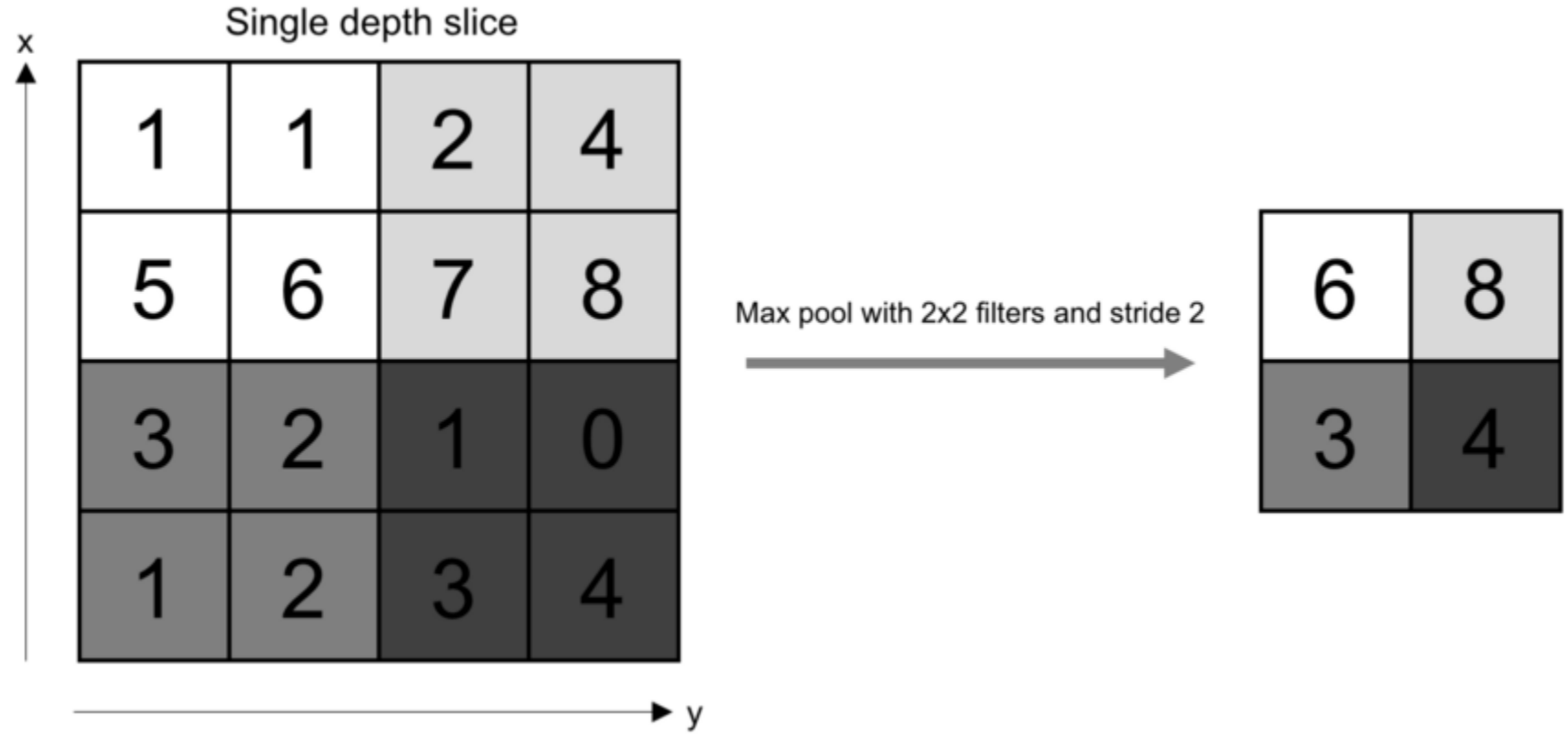
Ejemplos de Arquitecturas

CNN – Qué aprende la Convolución?



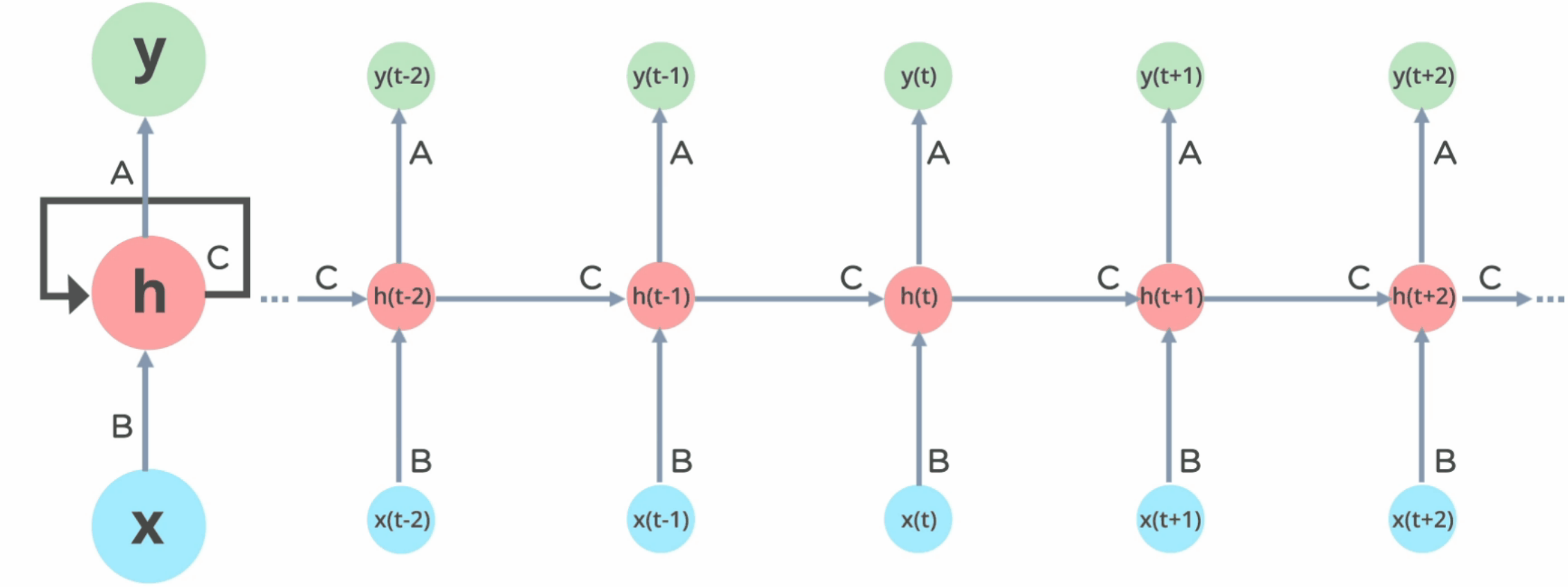
Ejemplos de Arquitecturas

CNN – Pooling



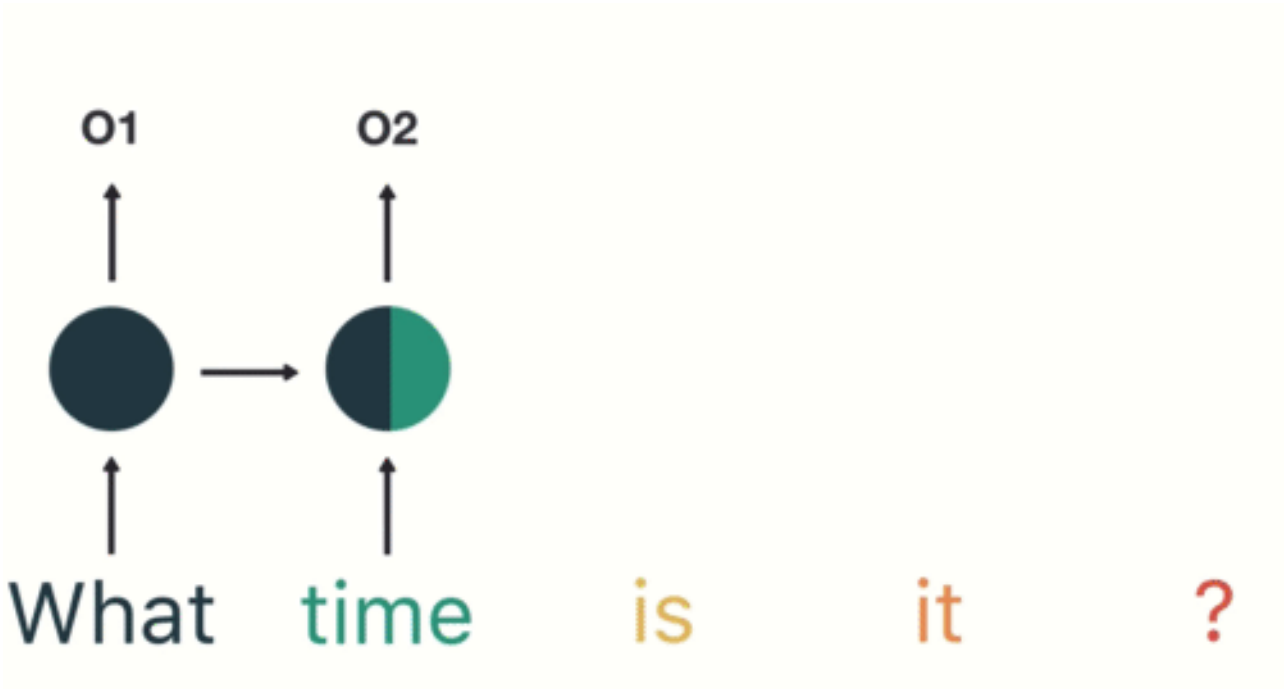
Ejemplos de Arquitecturas

RNN (Red Neuronales Recurrente)



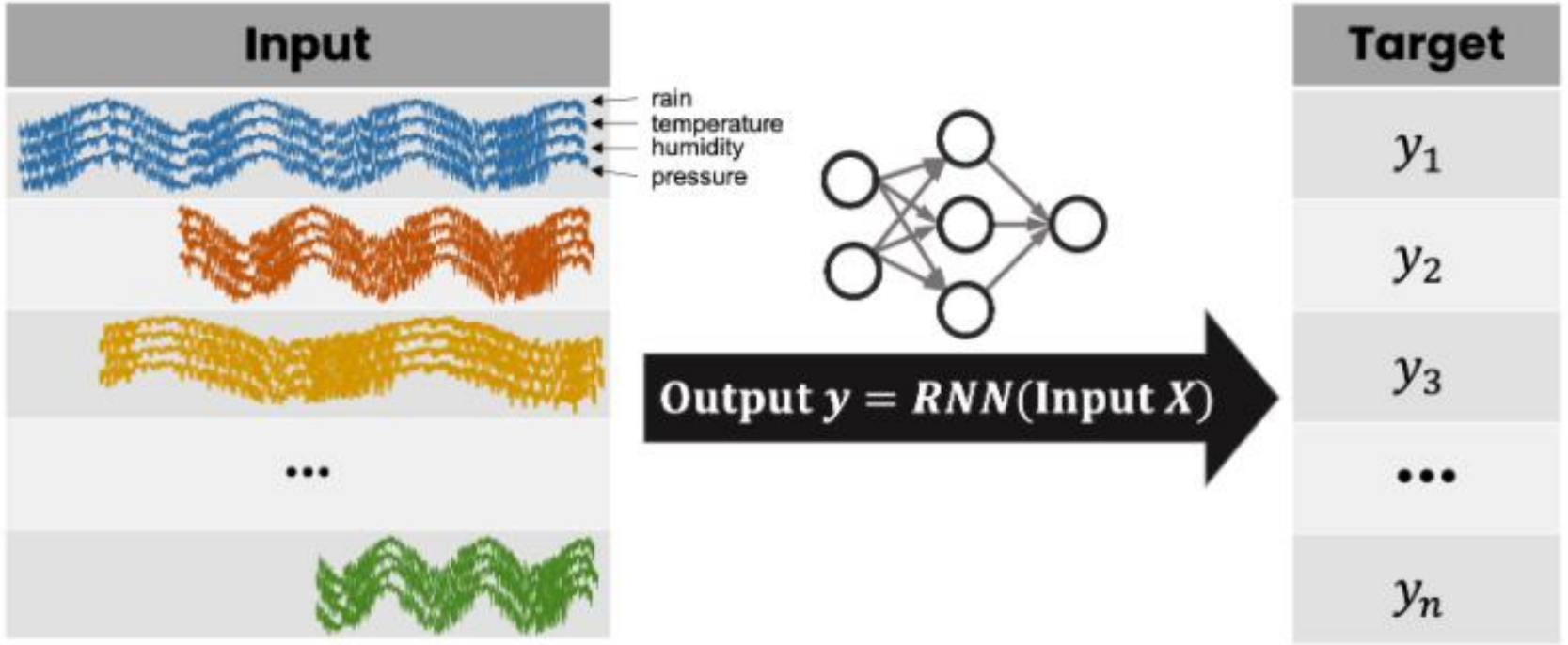
Ejemplos de Arquitecturas

RNN (Red Neuronales Recurrente)



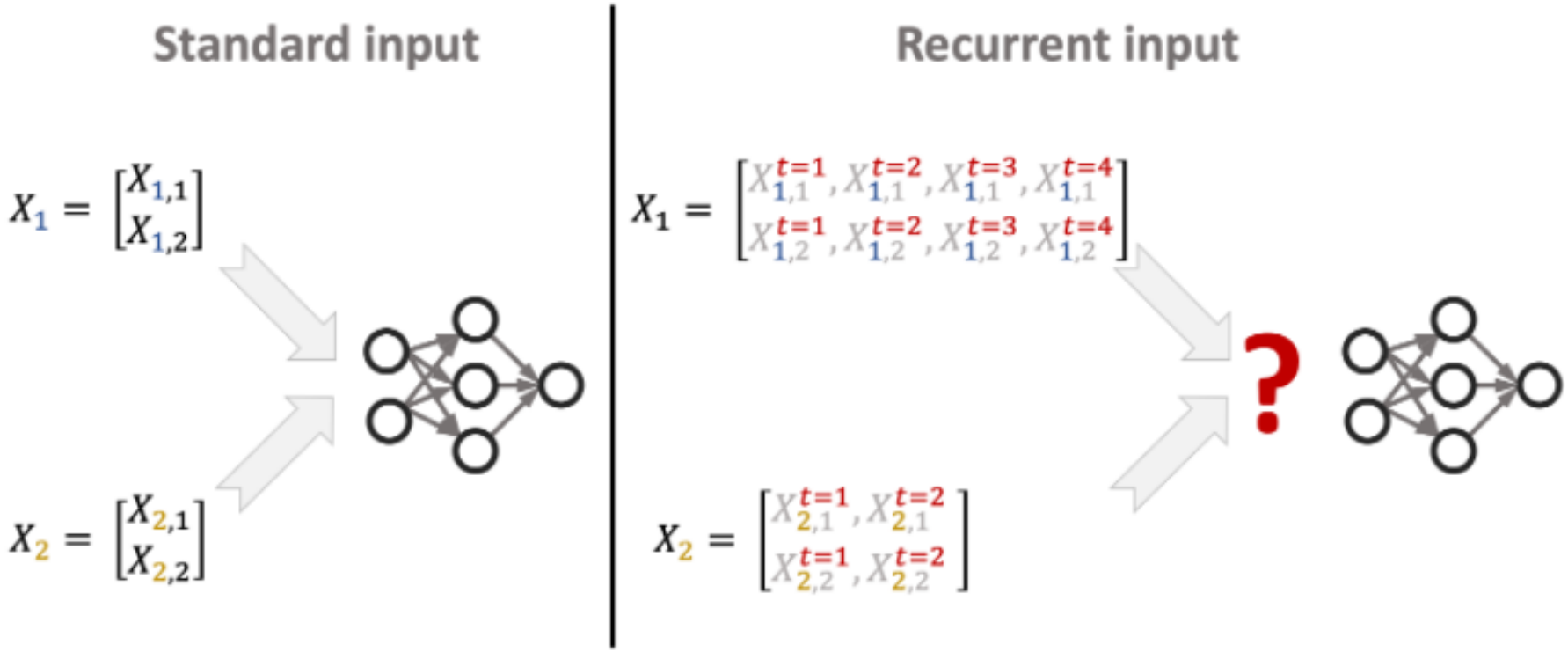
Ejemplos de Arquitecturas

RNN - Shape del Input



Ejemplos de Arquitecturas

RNN – Shape del Input

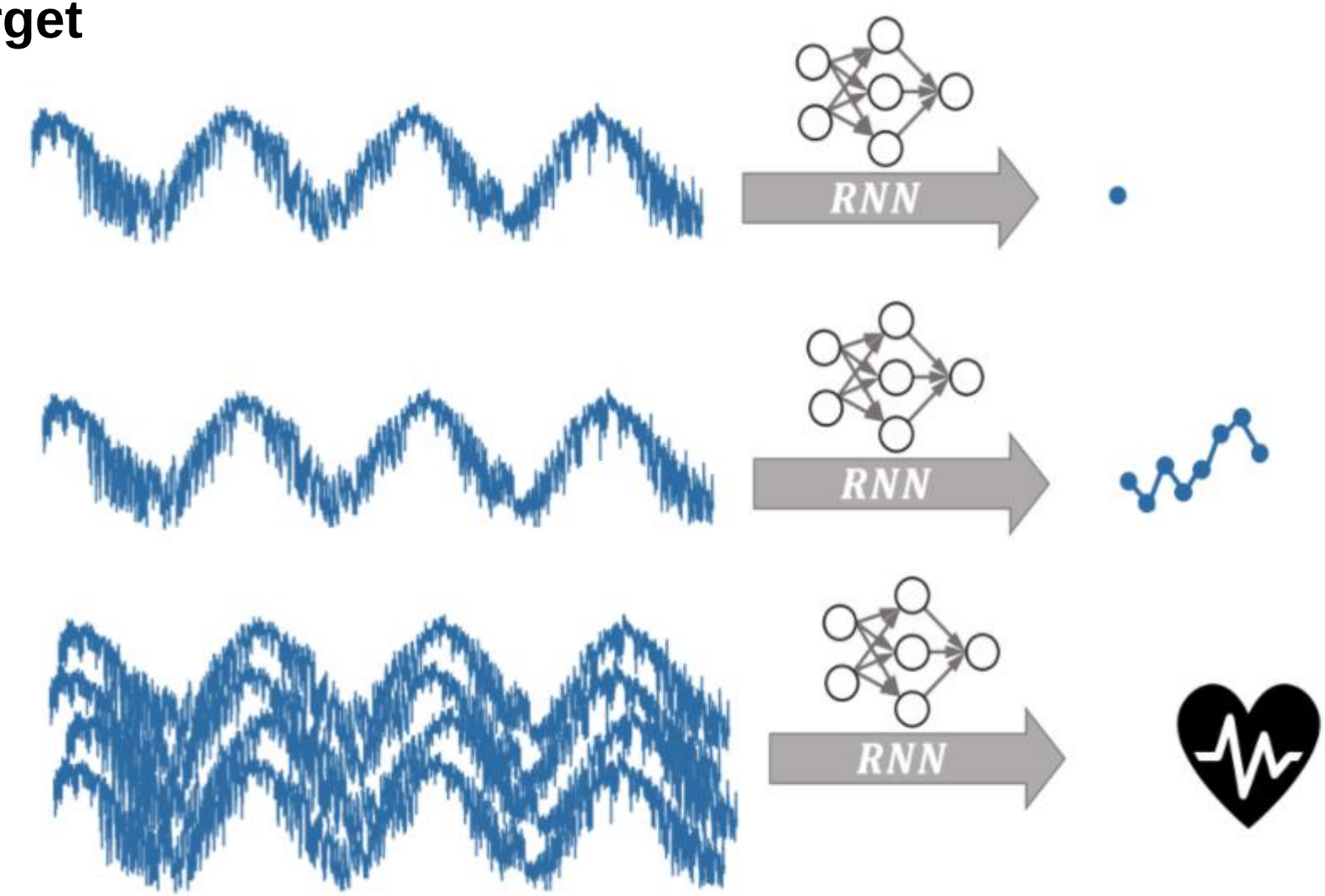


Ejemplos de Arquitecturas

RNN – El Target

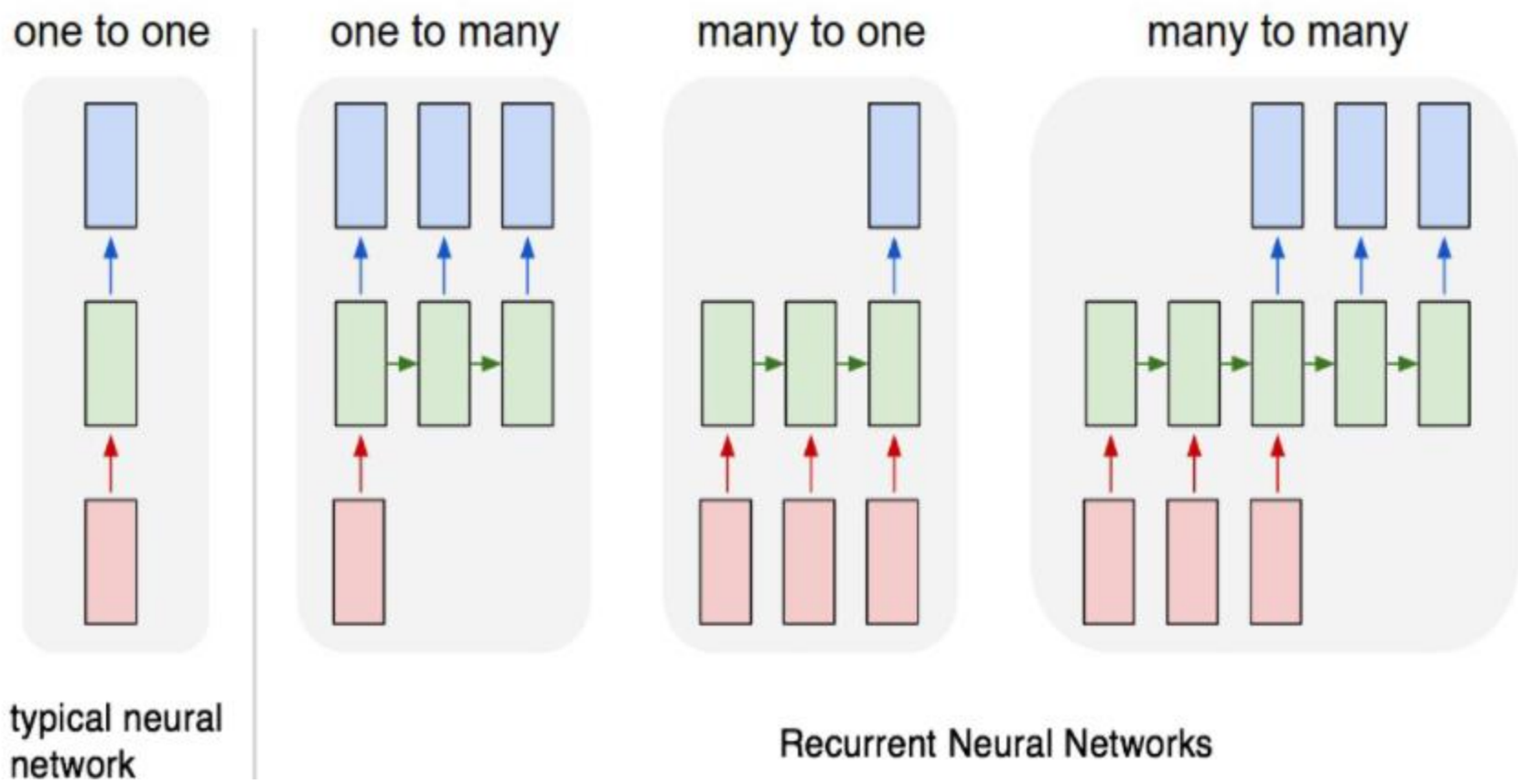
Regresión

Clasificación



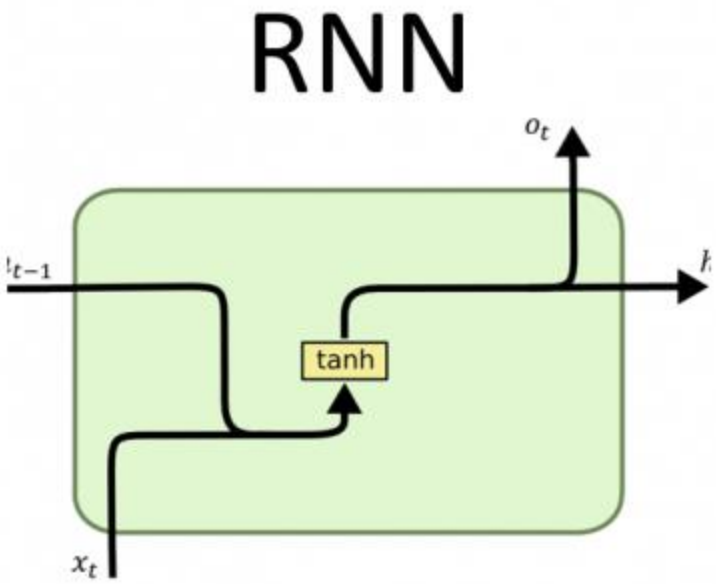
Ejemplos de Arquitecturas

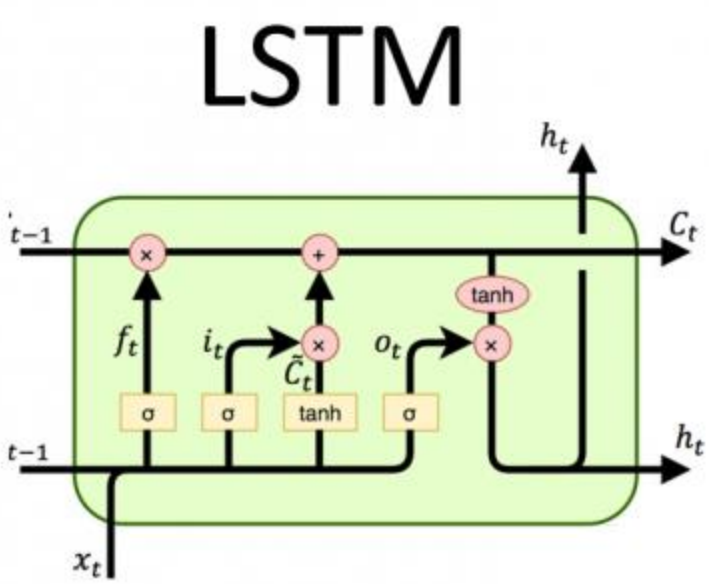
RNN – Estructura

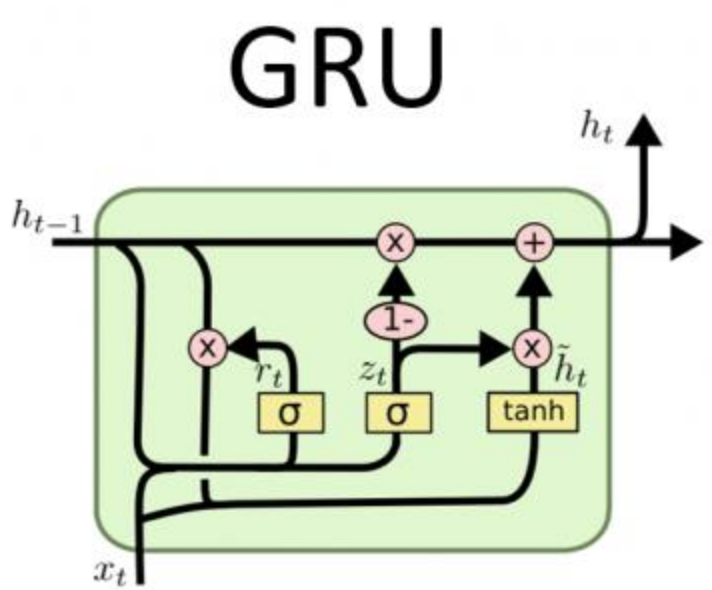


Ejemplos de Arquitecturas

RNN – Arquitecturas

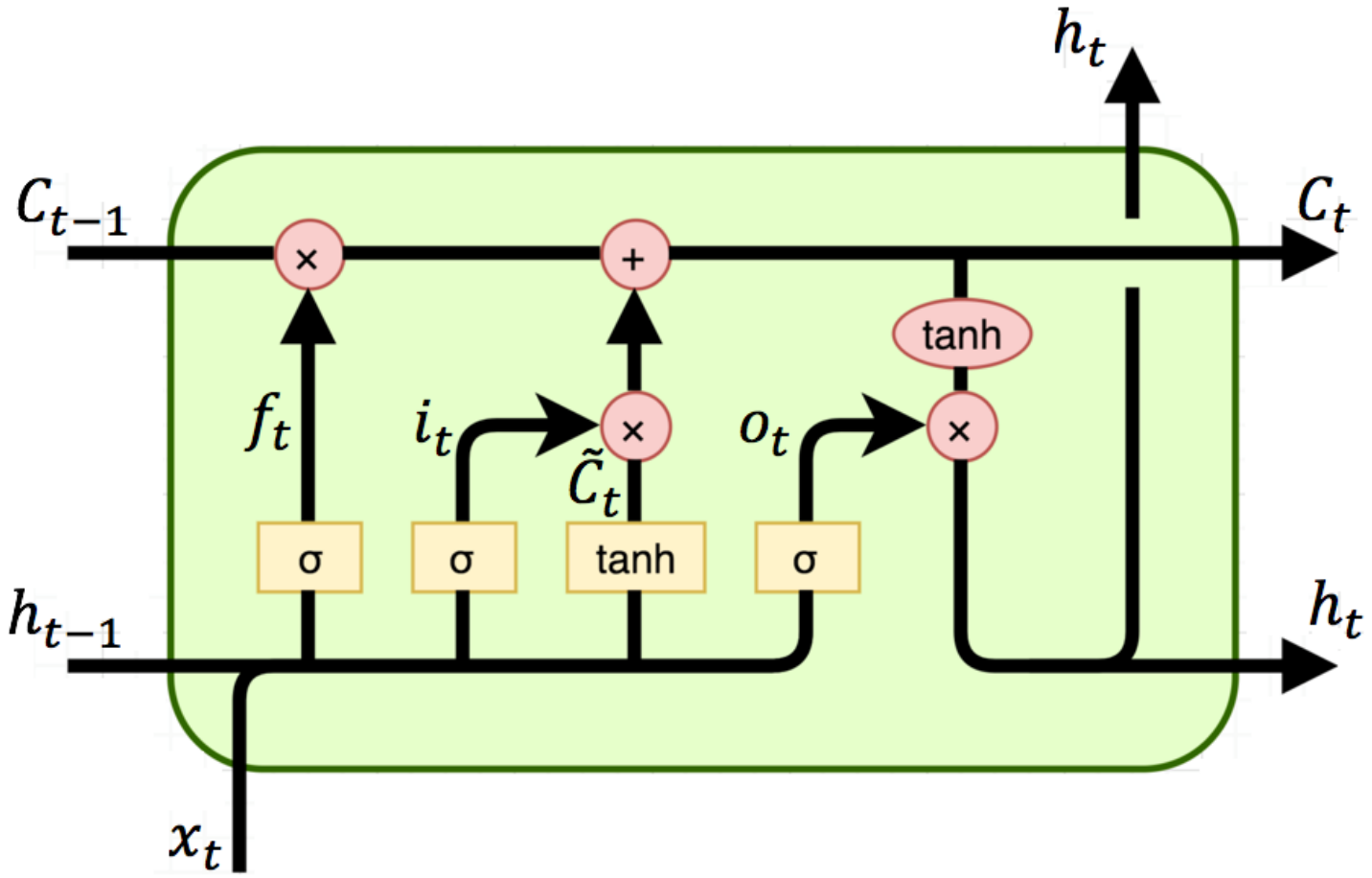






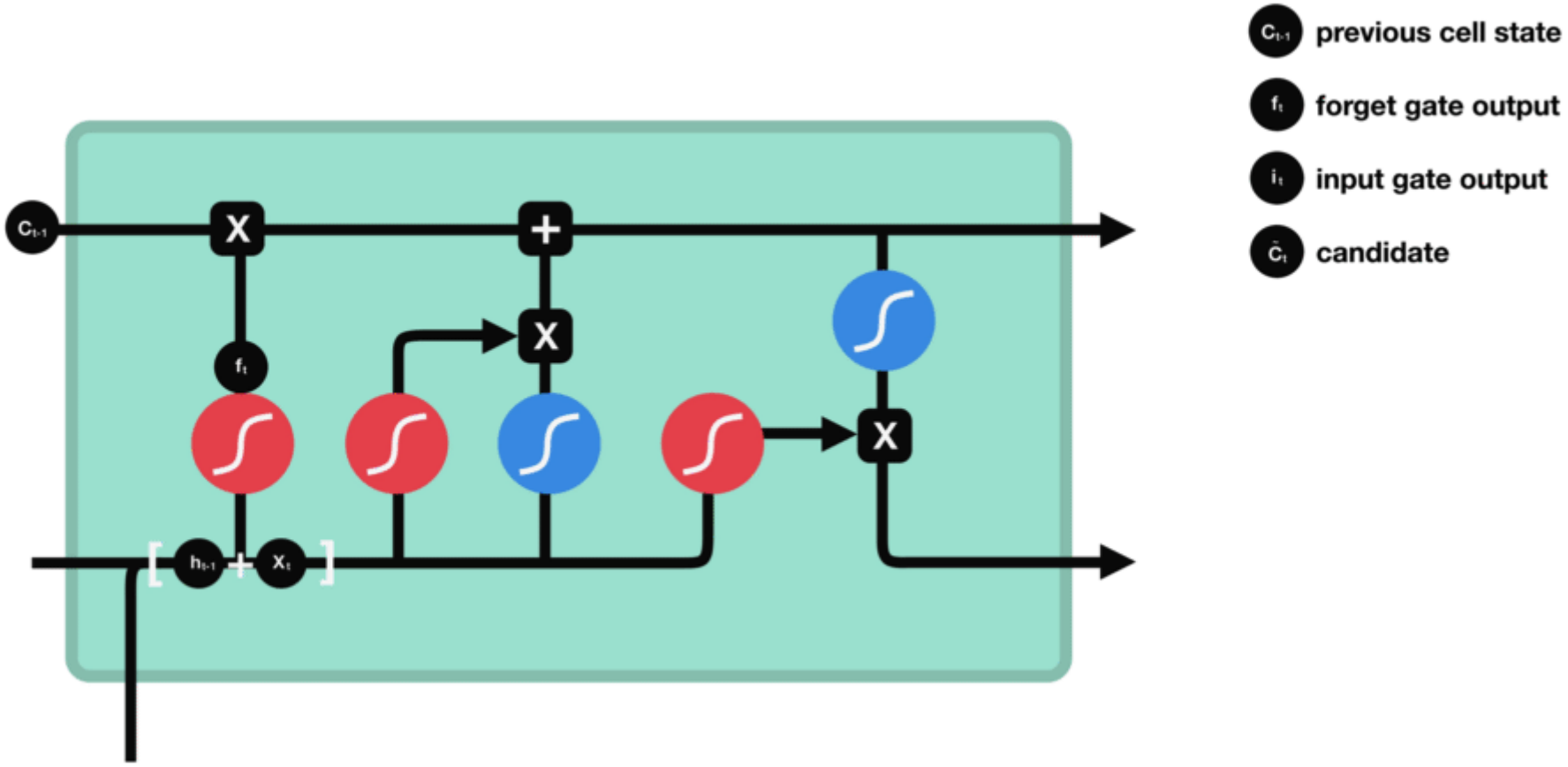
Ejemplos de Arquitecturas

RNN – LSTM



Ejemplos de Arquitecturas

RNN – LSTM



UCEMA

Muchas Gracias!