



UNIVERSIDAD NACIONAL DE MISIONES



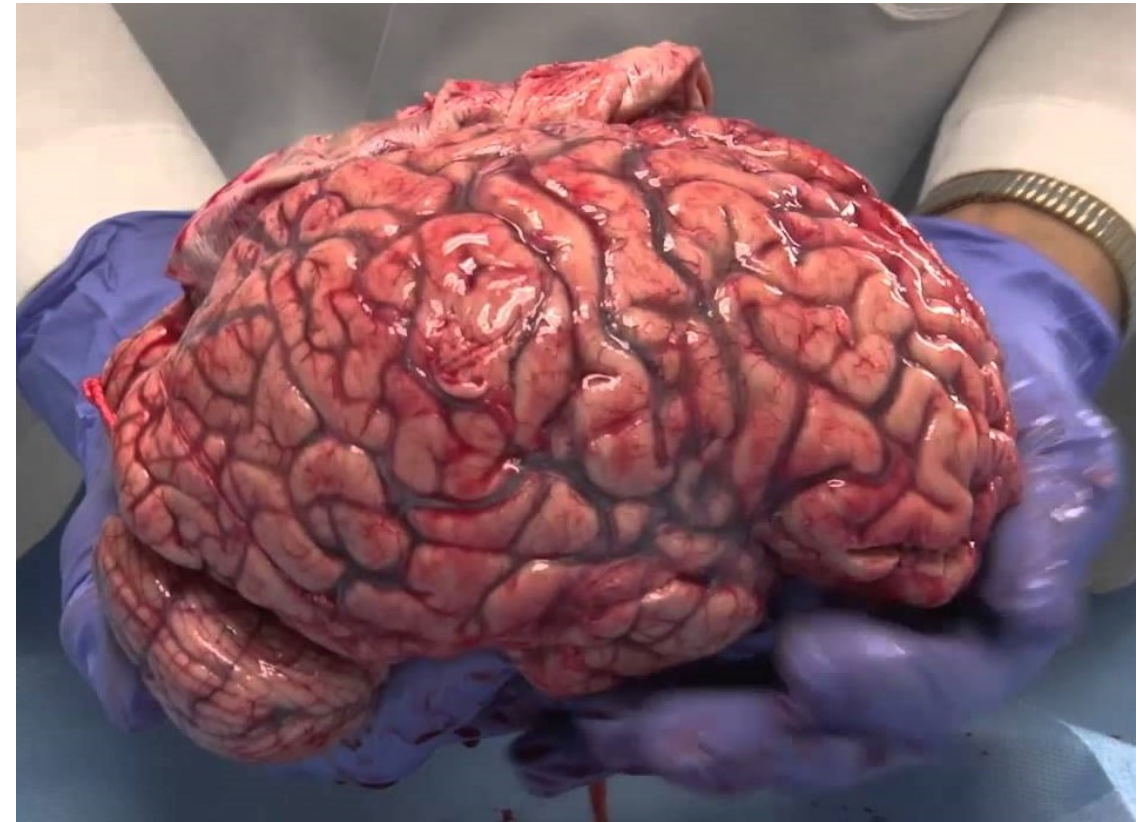
Facultad de Ingeniería

INTRODUCCIÓN A LAS REDES NEURONALES ARTIFICIALES

15/9/23 – Oberá, Misiones, Argentina

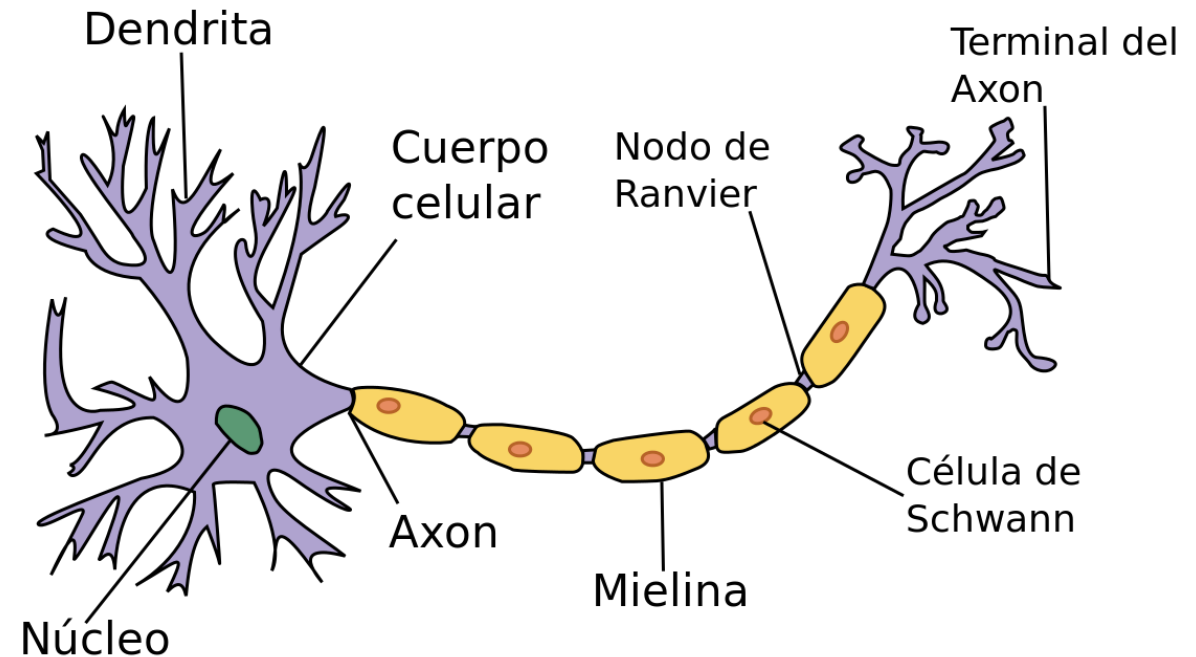
EL CEREBRO HUMANO

- $\approx 1,4$ kg, 1,13 litros (F) o 1,26 litros (H)
- Los pliegues (surcos) incrementan la superficie, que totaliza $\approx 0,25$ m²
- ≈ 80.000 M neuronas
- $\approx 1.000.000.000$ M sinapsis
- 200-400 kCal/día \rightarrow 10-20 Watts
- 1000 módulos, c/u con unas 500 NN
- Cómputo paralelo



LA NEURONA BIOLÓGICA

- Dendritas: entradas
- Axón: salida
- Un enlace axón-dendrita es una sinapsis.
- Una señal es transmitida al axón sólo cuando la neurona se excita; por eso esta puede inhibir o amplificar una señal.

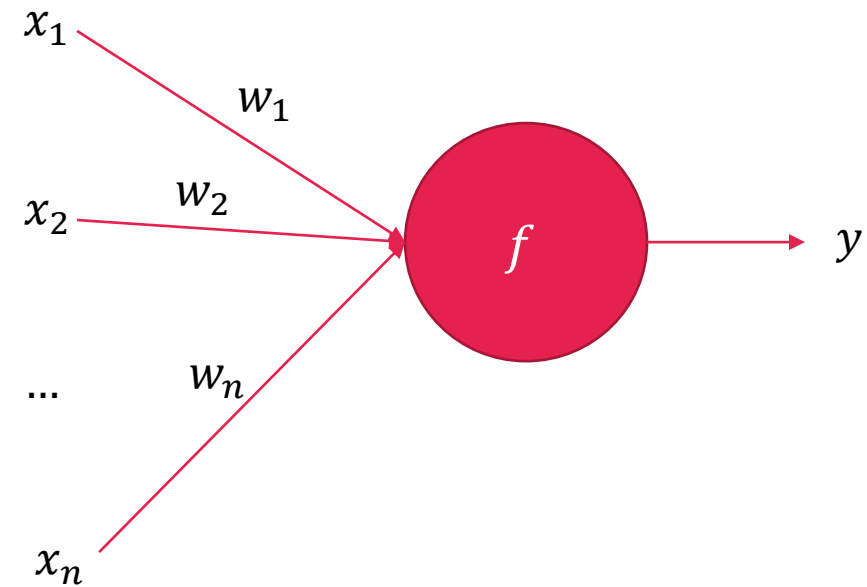


PERCEPTRÓN: NEURONA ARTIFICIAL

$$f(x) = \sum_{i=1}^n w_i \cdot x_i$$

$$f(x) = \mathbf{W} \cdot \mathbf{X}$$

$$y = \begin{cases} 1 & 0 < f(x) + b \\ 0 & \text{otro} \end{cases}$$

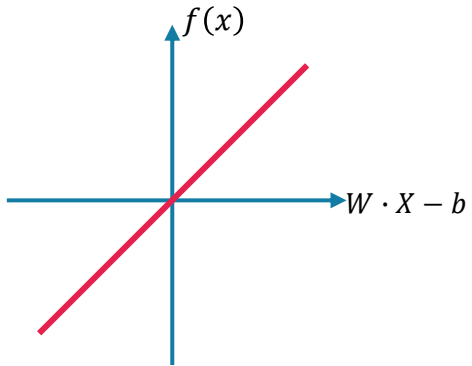


$$\mathbf{X} \sim [0,1]$$

$$y \sim [0,1]$$

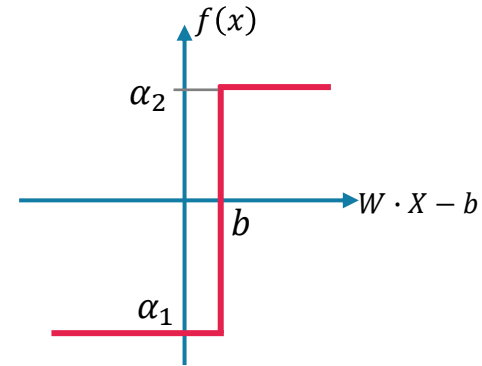
FUNCIONES DE ACTIVACIÓN

Lineal



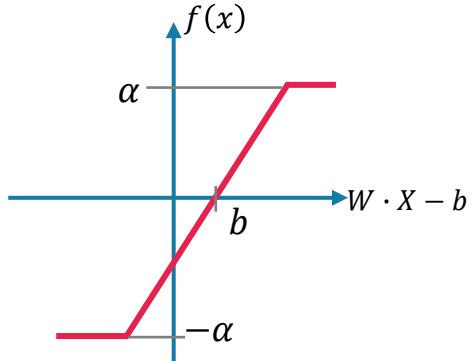
$$f(W \cdot X - b) = \lambda(W \cdot X - b)$$

Escalón



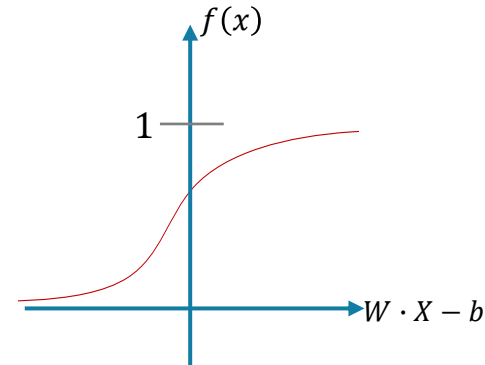
$$f(W \cdot X - b) = \begin{cases} \alpha_1 & b \leq W \cdot X \\ \alpha_2 & W \cdot X < b \end{cases}$$

Rampa



$$f(W \cdot X - b) = \begin{cases} -\alpha & -\epsilon \leq W \cdot X - b \\ W \cdot X - b & -\epsilon < W \cdot X - b < \epsilon \\ \alpha & \epsilon < W \cdot X - b \end{cases}$$

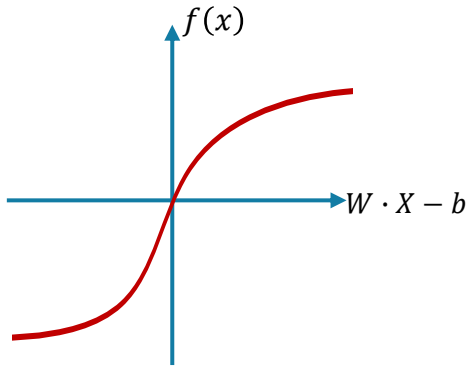
Sigmoidea



$$f(W \cdot X - b) = \frac{1}{1 + e^{-\lambda(W \cdot X - b)}}$$

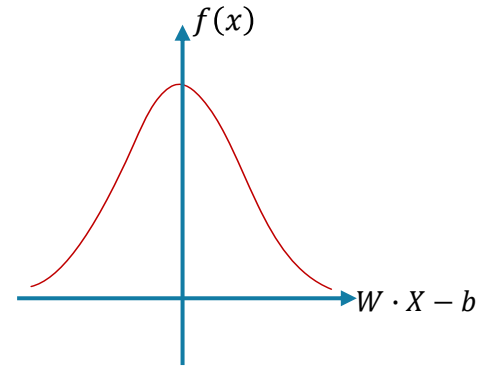
FUNCIONES DE ACTIVACIÓN

Tangente hiperbólica



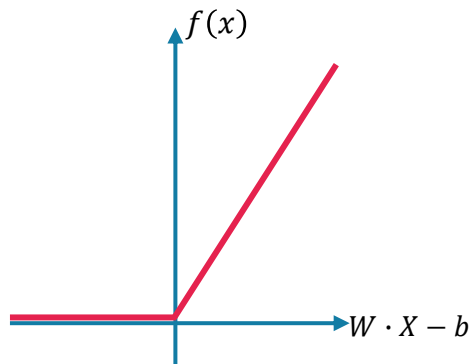
$$f(W \cdot X - b) = \frac{e^{(W \cdot X - b)} - e^{-(W \cdot X - b)}}{e^{(W \cdot X - b)} + e^{-(W \cdot X - b)}}$$

Gaussiana



$$f(W \cdot X - b) = e^{\frac{-(W \cdot X - b)^2}{\sigma^2}}$$

ReLU



$$f(W \cdot X - b) = \begin{cases} 0 & W \cdot X - b < 0 \\ W \cdot X - b & 0 < W \cdot X - b \end{cases}$$

FLUJO DE TRABAJO CON RNA



Preparación

- Importar y curar datos
- Análisis exploratorio
- Elegir topología
- Diseñar *input layer*
- Diseñar *output layer*
- Escoger validación



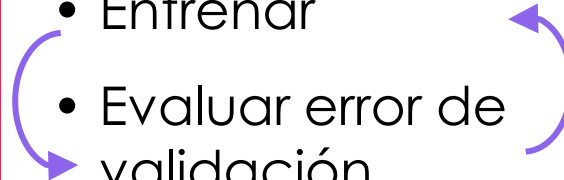
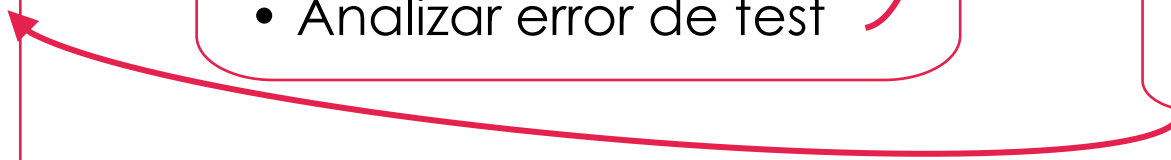
Entrenamiento

- Especificar topología
- Entrenar
- Evaluar error de validación
- Analizar error de test



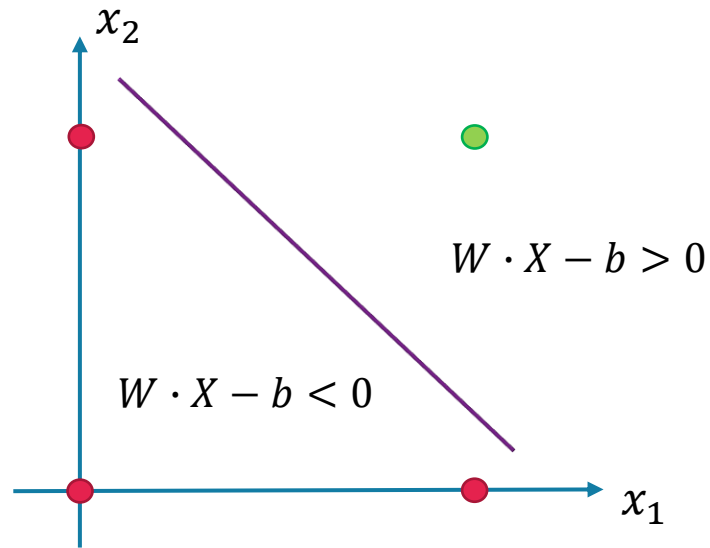
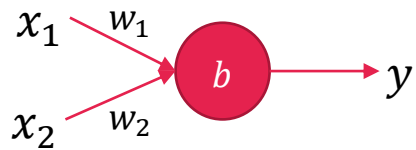
Implementación

- Exportar modelo
- Implementar consulta
- Analizar rendimiento y desempeño



APRENDIZAJE: LO MÁS ELEMENTAL

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1



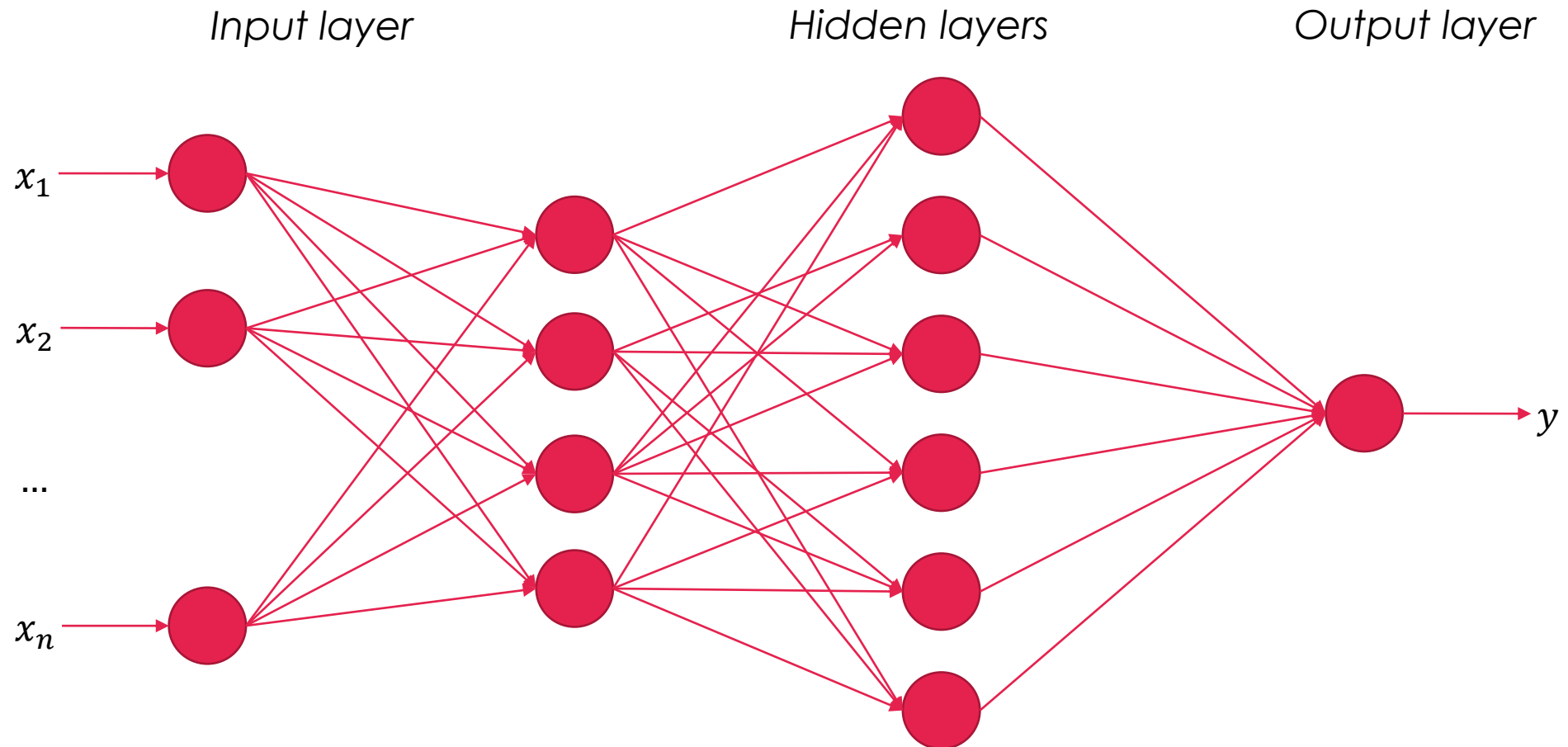
$$\begin{cases} w_1 \cdot 1 + w_2 \cdot 1 - b > 0 & |_{1,1} \\ w_1 \cdot 1 + w_2 \cdot 0 - b < 0 & |_{1,0} \\ w_1 \cdot 0 + w_2 \cdot 1 - b < 0 & |_{0,1} \end{cases}$$

$$\begin{Bmatrix} w_1 \\ w_2 \\ b \end{Bmatrix} \begin{Bmatrix} 1 \\ 1 \\ 2 \end{Bmatrix}$$



Inviabile en grandes dimensiones
y con múltiples neuronas

EN LA REALIDAD...

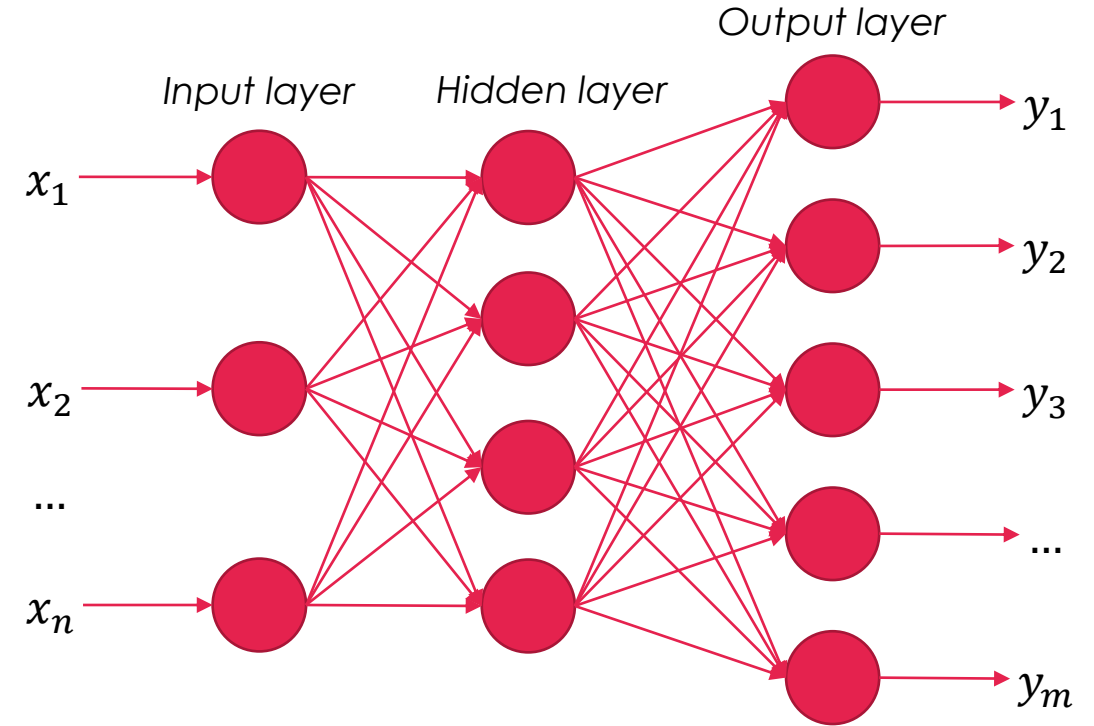


APRENDIZAJE SUPERVISADO EN RNA



FEEDFORWARD NN

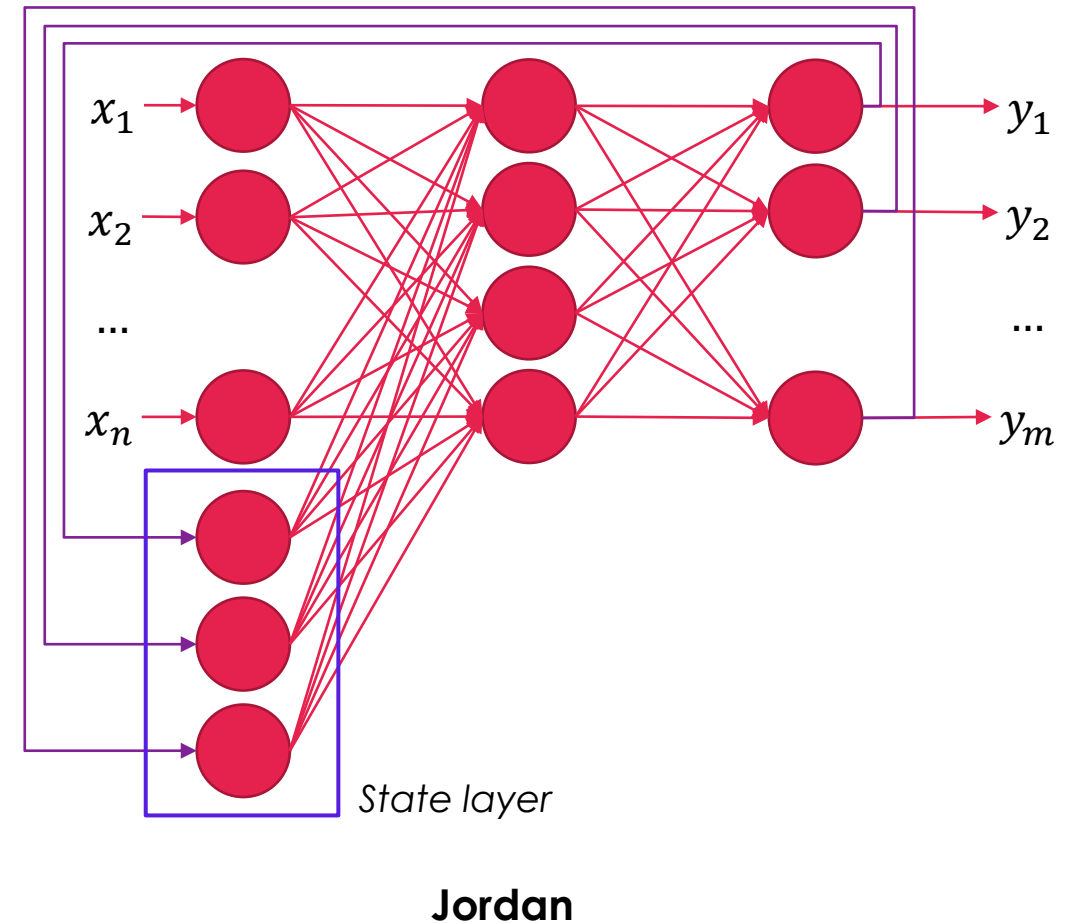
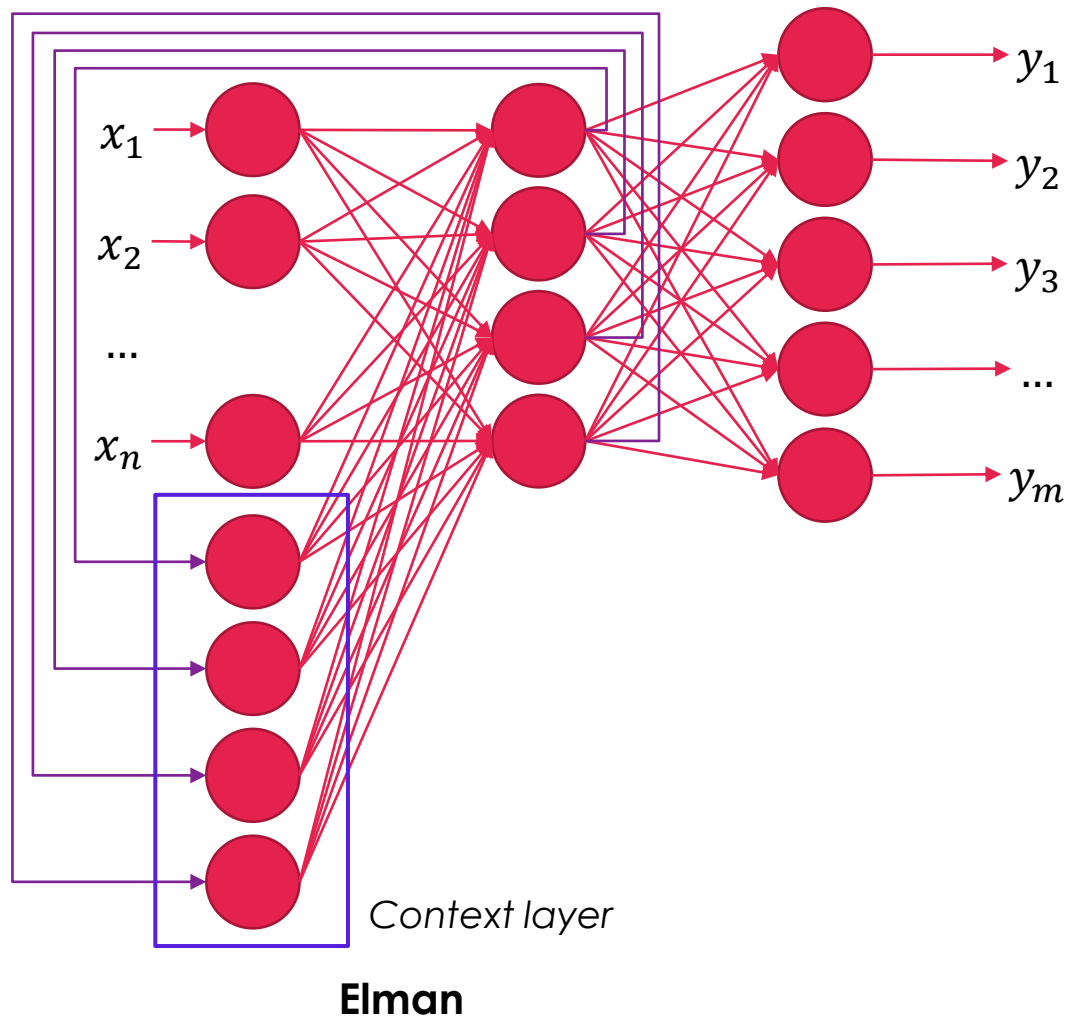
- Típicamente una sola capa oculta, con H neuronas, todo *full connected*.
- Cada función de activación puede ser diferente.
- Típicamente la capa de entrada tiene función de activación lineal.
- Puede aproximar cualquier función continua para la frontera de decisión con una sola capa si esta tiene suficientes neuronas.
- Si la capa de entrada tiene funciones de orden elevado (diferente de lineal), se llama *Functional Link NN*.



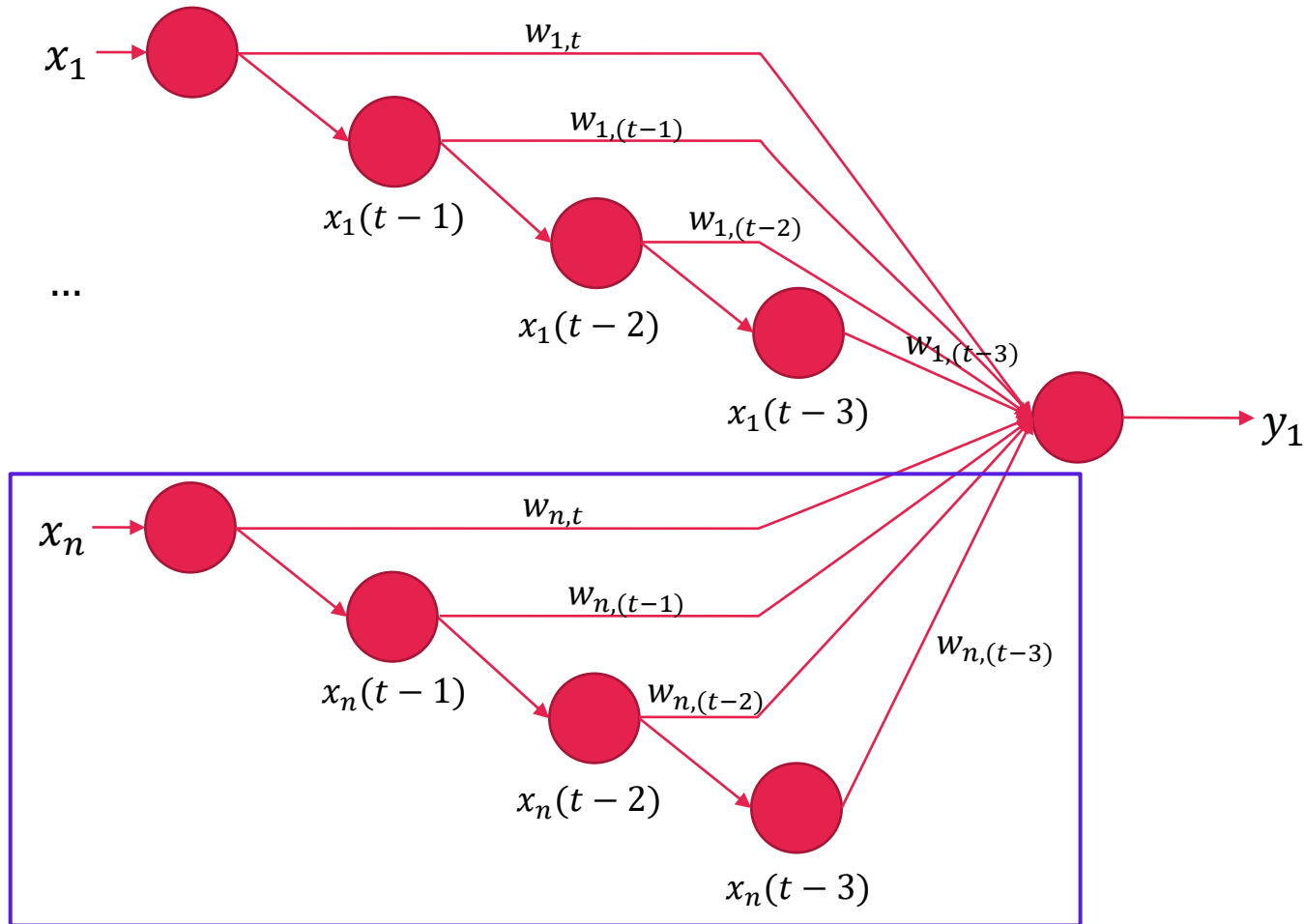
$$y'_{m,k} = f_{y_m} \left(\sum_{h=1}^H w_{m,h} \cdot f_{y_h} \left(\sum_{n=1}^N w_{h,n} \cdot x_{n,k} \right) \right)$$

SIMPLE RECURRENT NN

Incorpora retroalimentaciones que le confiere la habilidad de aprender características temporales de los datos.

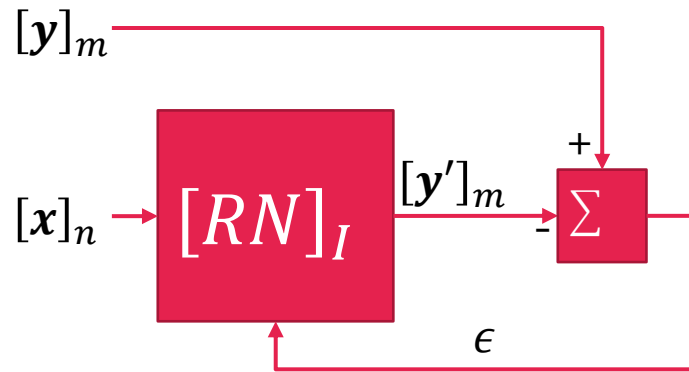


TIME DELAY NN



- El aprendizaje de características temporales de los datos se ve reforzado por múltiples secuencias de retardos, generalmente con pesos descendentes.
- El bloque de la neurona con sus retardos ponderados es la unidad básica para asociar *FFNN multilayer*.

DESCENSO DEL GRADIENTE



$$w_i(t) = w_i(t - 1) + \Delta w_i(t)$$

$$\Delta w_i(t) = \eta \left(-\frac{\delta \epsilon}{\delta w_i} \right) \quad \eta \text{ tasa de aprendizaje}$$

Regla generalizada Widrow-Hoff

$$\frac{\delta \epsilon}{\delta w_i} = -2(y_k - y'_k) \cdot x_{i,k}$$

$$w_i(t) = w_i(t - 1) + 2\eta(y_k - y'_k) \cdot x_{i,k}$$

K observaciones

$$\epsilon = \sum_{k=1}^K (y_k - y'_k)^2$$

Objetivo $\epsilon \rightarrow 0$

BACKPROPAGATION

Cada iteración de aprendizaje popularmente se denomina “época”, se suceden dos etapas:

1. Pasar *feedforward*: calcular la salida de la NN para cada observación presente en el *training set*.
2. *Backward propagation*: propagar desde las salidas hacia las entradas una señal de error.

En estocástico, los pesos son actualizados según:

$$w_{m,h}(t) += \Delta w_{m,h}(t) + \alpha \Delta w_{m,h}(t - 1)$$

α momento

DETENCIÓN ¿CUÁNDO PARAR?

- Máximo número de épocas.
- Cuando el error sobre el *training set* es aceptablemente pequeño, se suele utilizar el error cuadrático medio.
- Cuando hay sobre entrenamiento: el conjunto de entrenamiento empieza a ser memorizado; esto se verifica si el error de validación más la varianza es mayor que el promedio de errores de validación en las épocas anteriores.

$$\epsilon'_V > \overline{\epsilon_V} + \sigma_{\epsilon_V}$$