CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN - IPN

Modelos de neuronas y arquitectura de redes neuronales

Presentación a cargo de Miguel Angel Soto Hernandez

TEMAS PRINCIPALES

PUNTOS QUE SE ABORDARÁN

¿Qué es una red neuronal artificial?

Modelos de neuronas

- Neurona de una sola entrada
- Funciones de transferencia
- Neurona de Multiples entradas

Arquitectura de redes neuronales

- Una capa de neuronas
- Multiples capas de neuronas
- Redes recurrentes

Introducción a las redes neuronales artificiales (RNA)

¿Qué es una red neuronal artificial?

NEURONA BIOLÓGICA

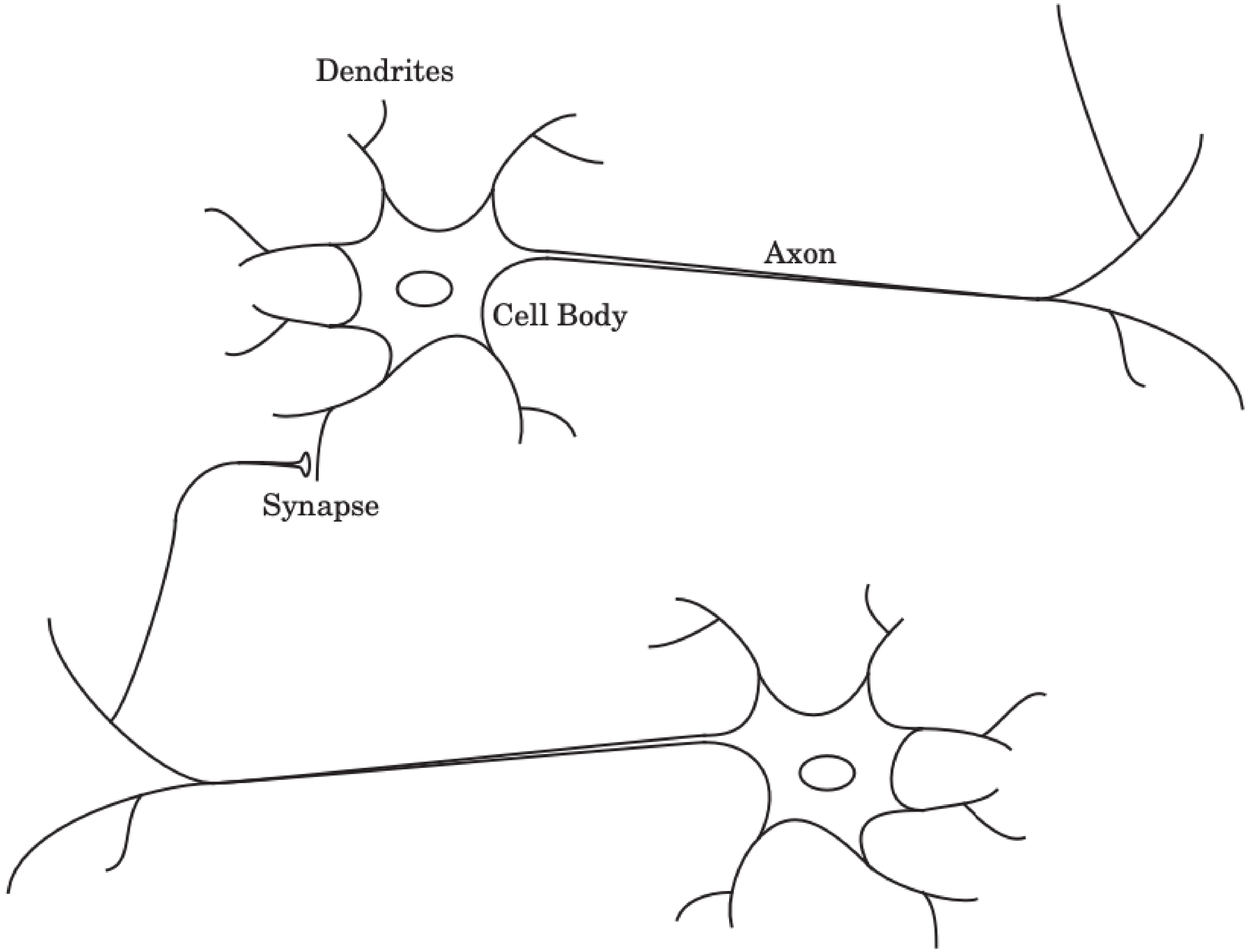


Figure 1.1 Schematic Drawing of Biological Neurons

Modelos de Neuronas

NEURONA ARTIFICIAL DE UNA SOLA ENTRADA

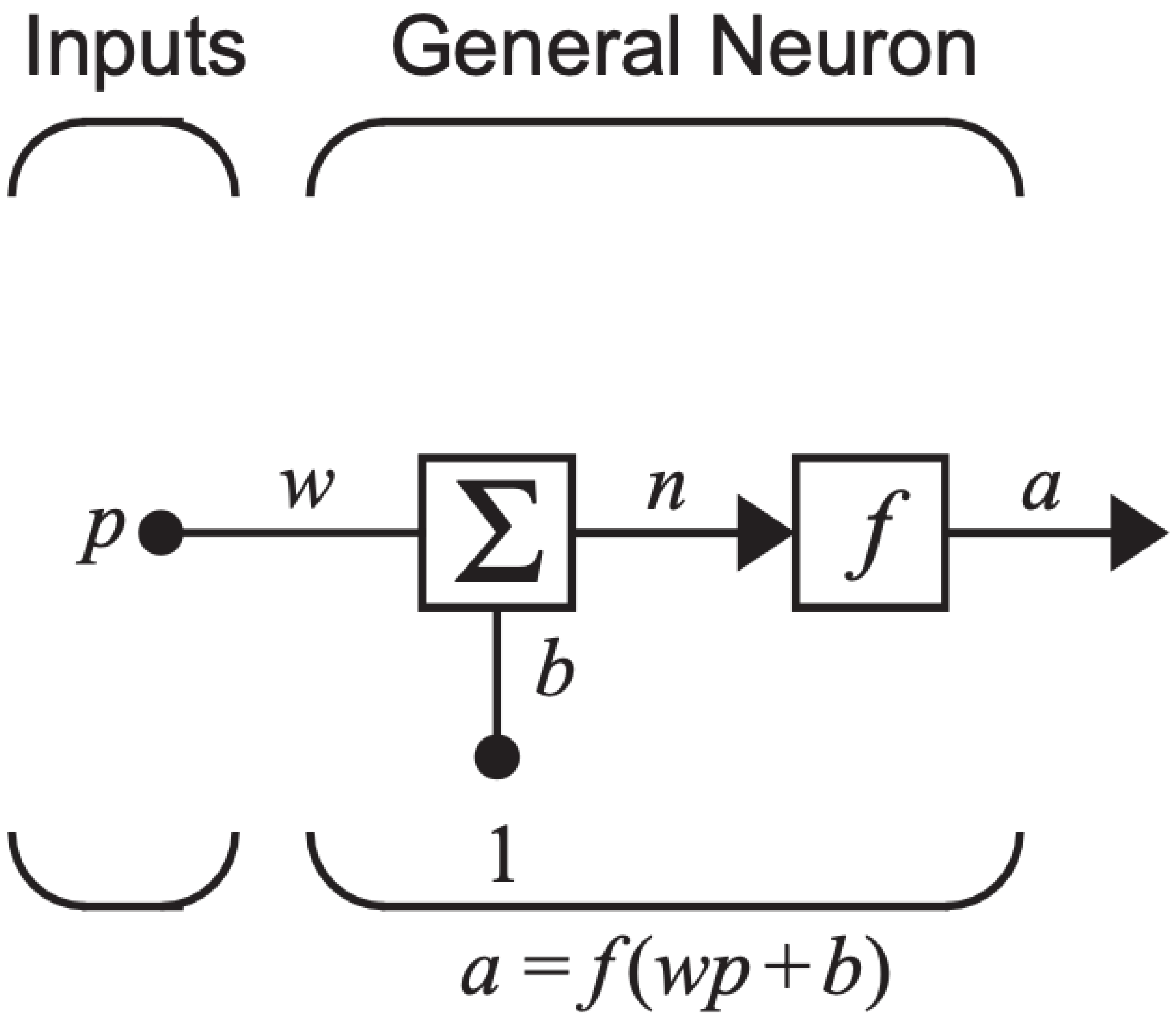


Figure 2.1 Single-Input Neuron

FUNCIÓN DE TRANSFERENCIA DE LÍMITE DURO

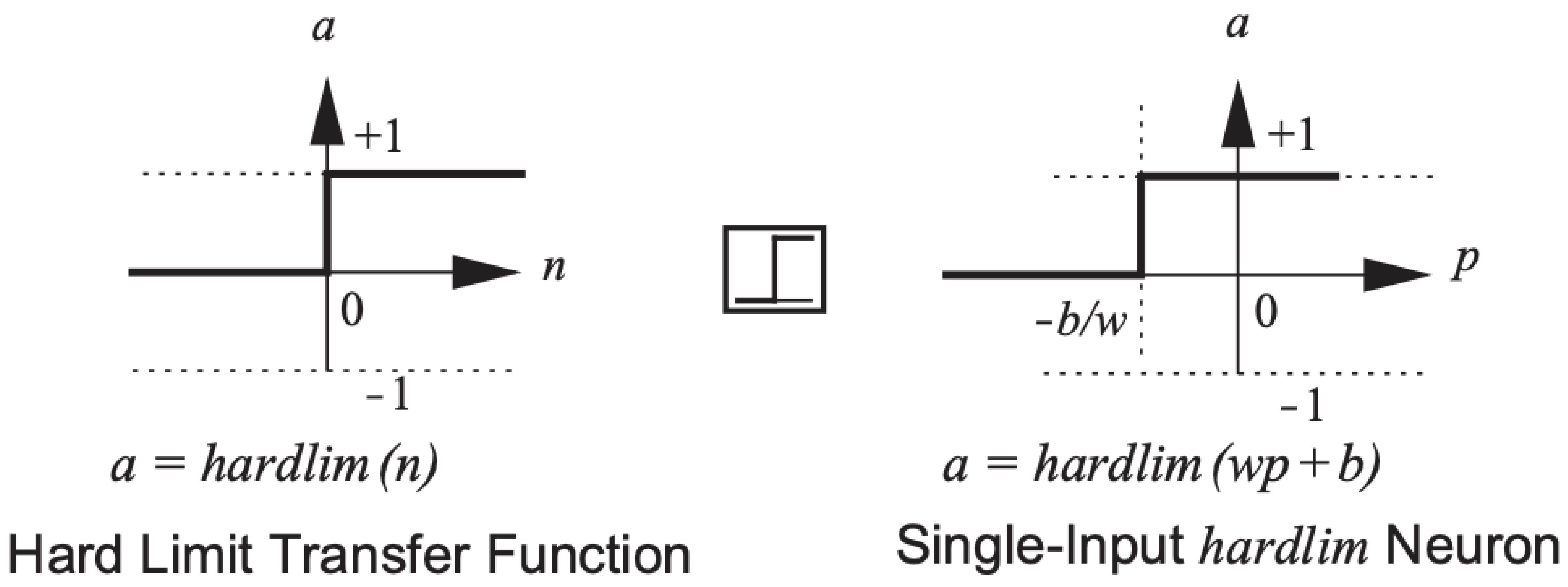
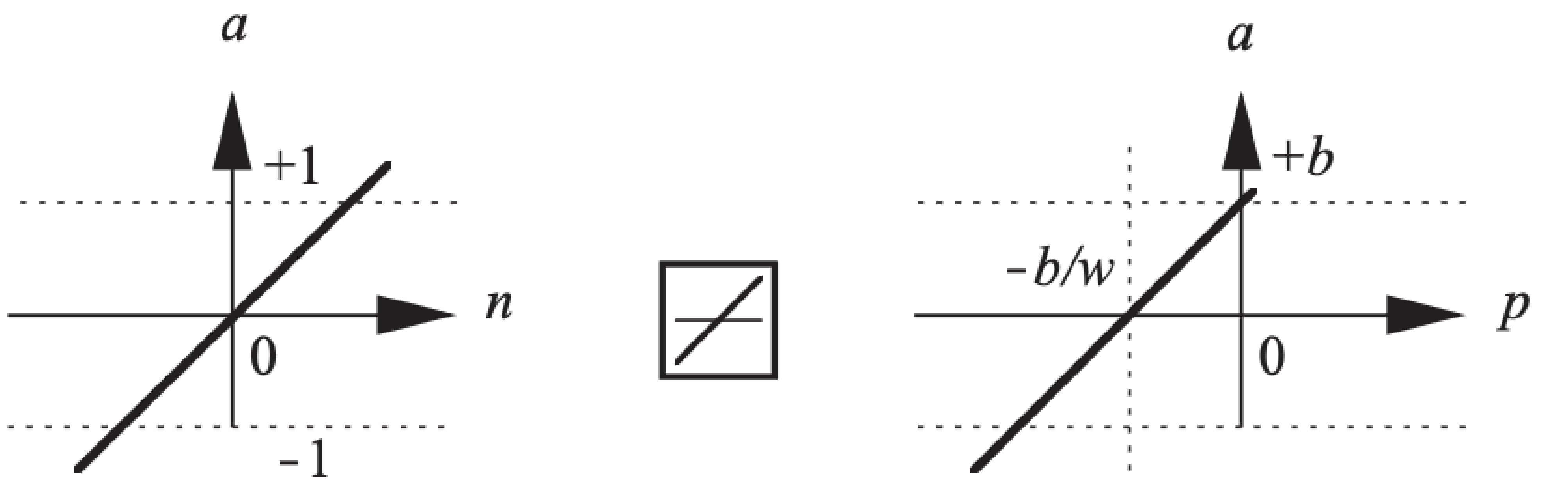


Figure 2.2 Hard Limit Transfer Function

FUNCIÓN DE TRANSFERENCIA LINEAR



$a = purelin(n)$ — Linear Transfer Function

$a = purelin(wp + b)$ — Single-Input *purelin* Neuron

Figure 2.3 Linear Transfer Function

FUNCIÓN DE TRANSFERENCIA SIGMOIDE LOGARÍTMICA

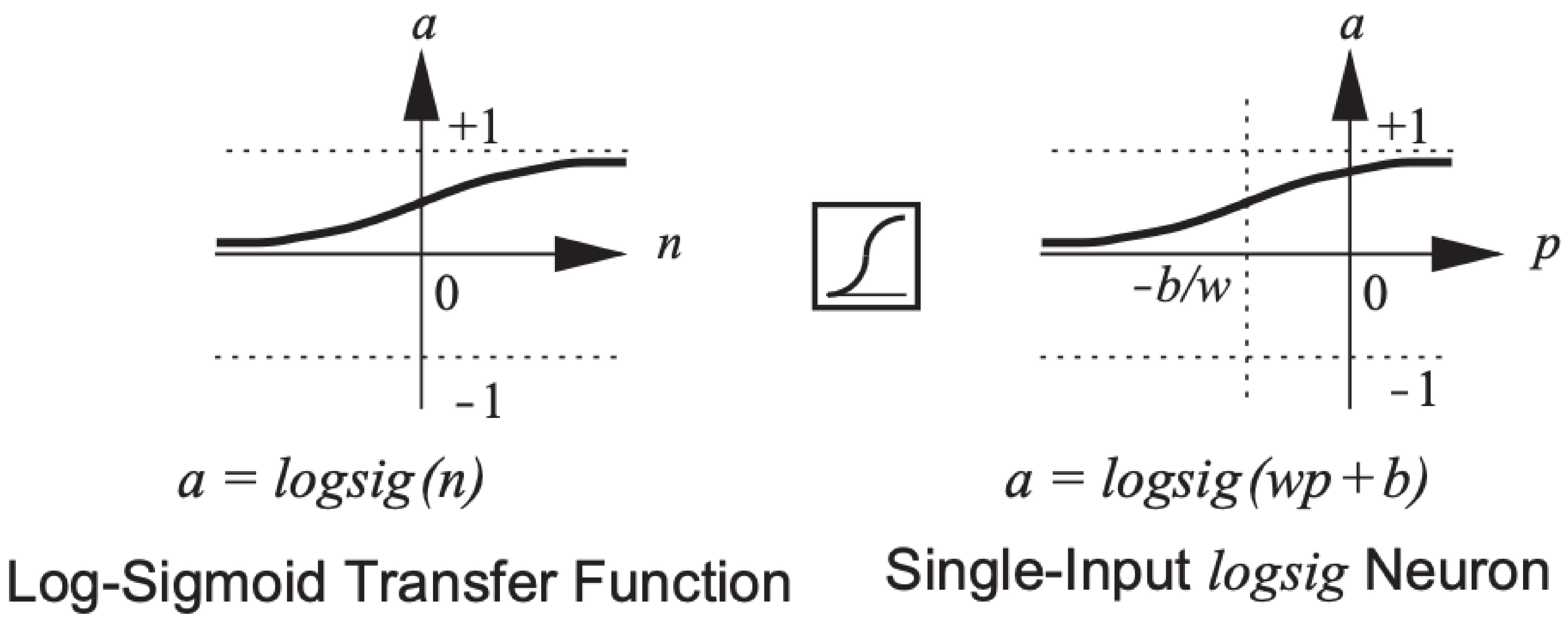


Figure 2.4 Log-Sigmoid Transfer Function

Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \ge 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \ge 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \le n \le 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \le n \le 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \dfrac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \dfrac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \le n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons	$\boxed{\mathbf{C}}$	compet

Table 2.1 Transfer Functions

NEURONA ARTIFICIAL DE MÚLTIPLES ENTRADAS

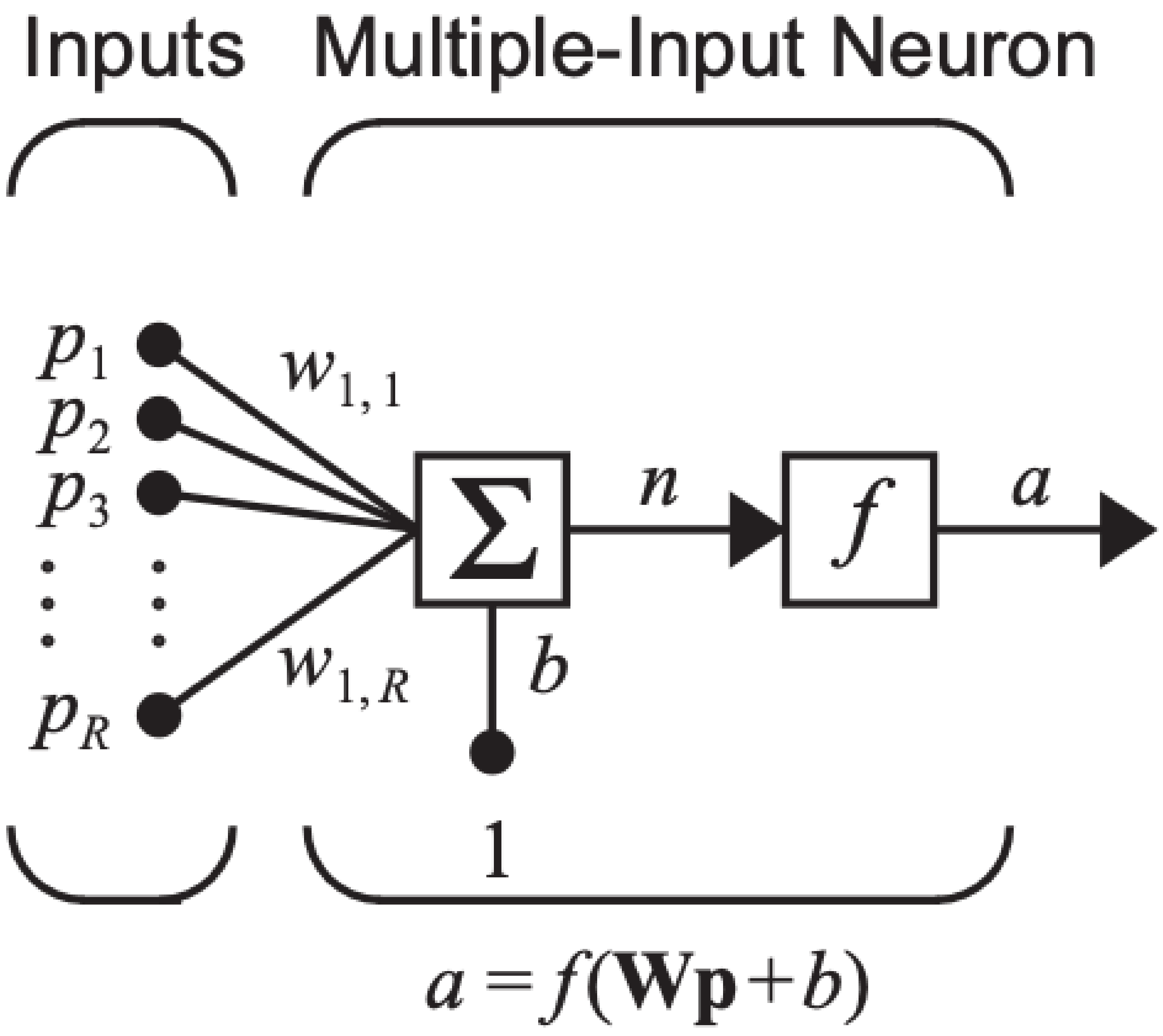


Figure 2.5 Multiple-Input Neuron

NEURONA ARTIFICIAL DE MÚLTIPLES ENTRADAS, NOTACIÓN REDUCIDA

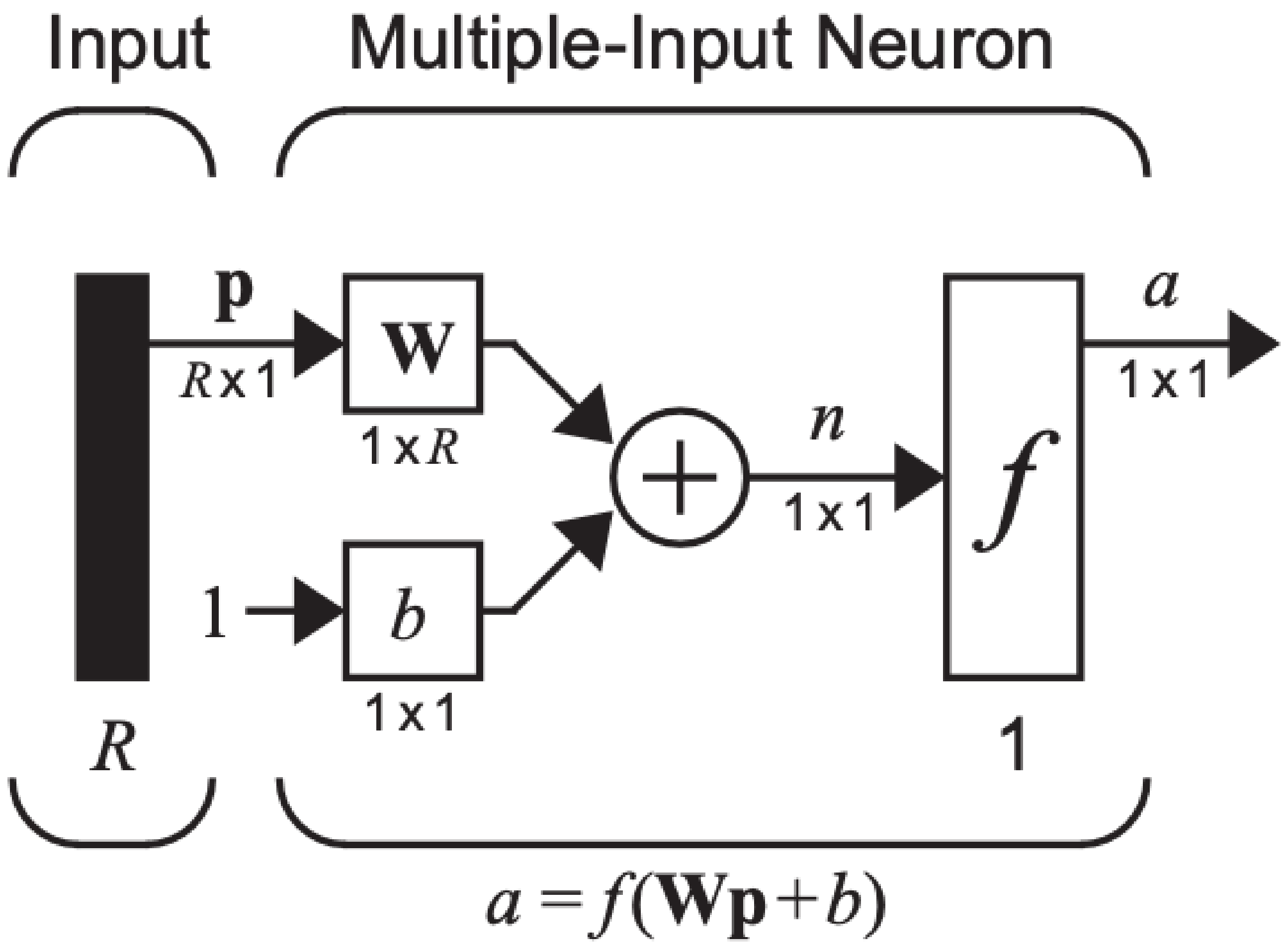


Figure 2.6 Neuron with R Inputs, Abbreviated Notation

Modelos matemáticos generales

a = f(Wp + b)

S*1 S*R R*1 S*1

UNA NEURONA + UNA ENTRADA

R = 1

S = 1

a = f(Wp + b)

1*1 1*1 1*1 1*1

MÚLTIPLES NEURONAS + UNA ENTRADA

R = 1

S > 1 **->** S = 6

a = f(Wp + b)

6*1 6*1 1*1 6*1

Modelos matemáticos generales

$a = f(Wp + b)$

S*1 S*R R*1 S*1

UNA NEURONA + MÚLTIPLES ENTRADAS

R> 1 **->** R = 3

S = 1

$\mathbf{a = f(Wp + b)}$

1*1 1*3 3*1 1*1

MÚLTIPLES NEURONAS + MÚLTIPLES ENTRADAS

R> 1 **->** R = 5

S > 1 **->** S = 4

$\mathbf{a = f(Wp + b)}$

4*1 4*5 5*1 4*1

Arquitectura de redes neuronales artificiales

UNA CAPA DE NEURONAS

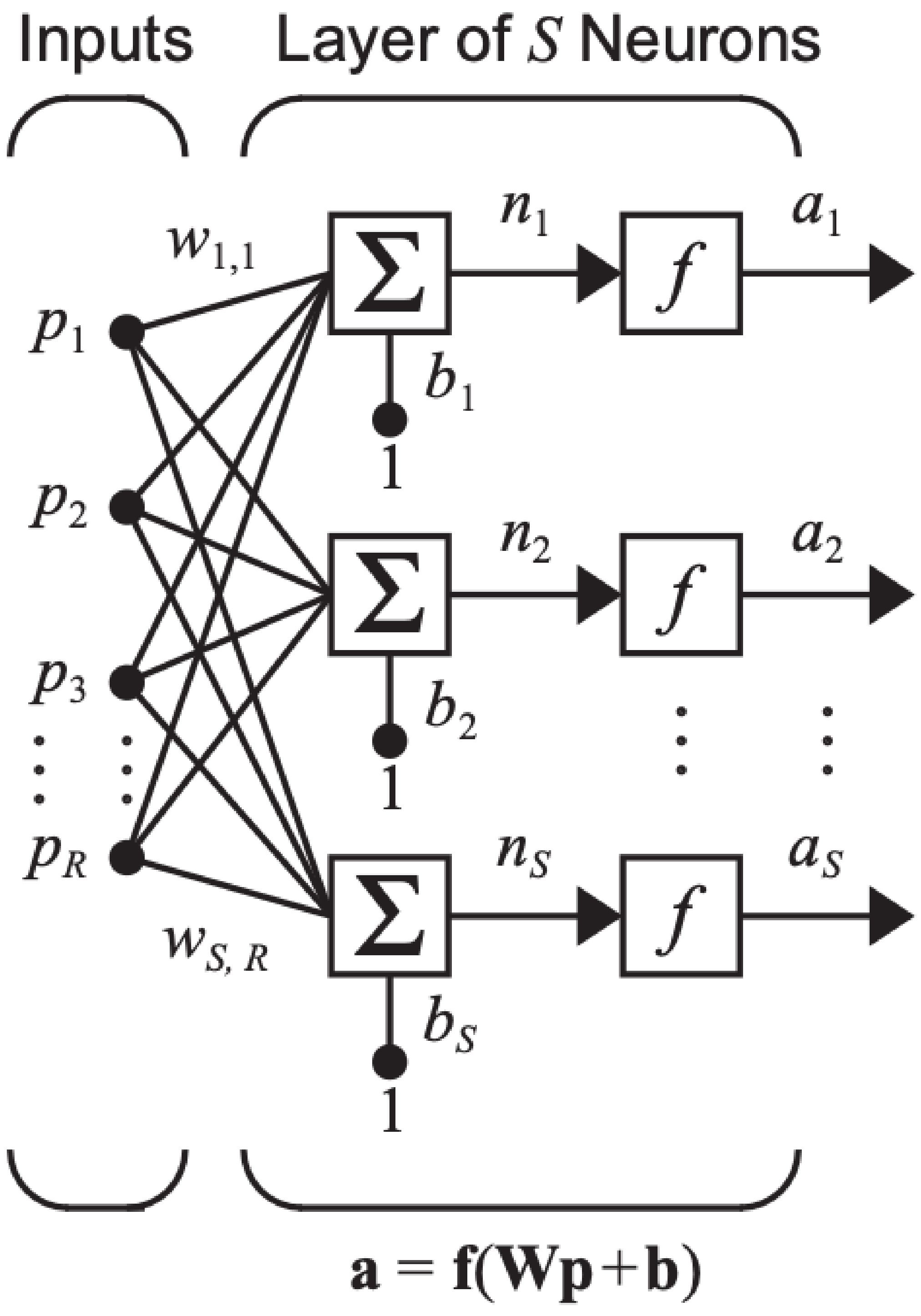


Figure 2.7 Layer of *S* Neurons

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \ldots & w_{1,R} \\ w_{2,1} & w_{2,2} & \ldots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \ldots & w_{S,R} \end{bmatrix}.$$

UNA CAPA DE NEURONAS, NOTACIÓN ABREVIADA

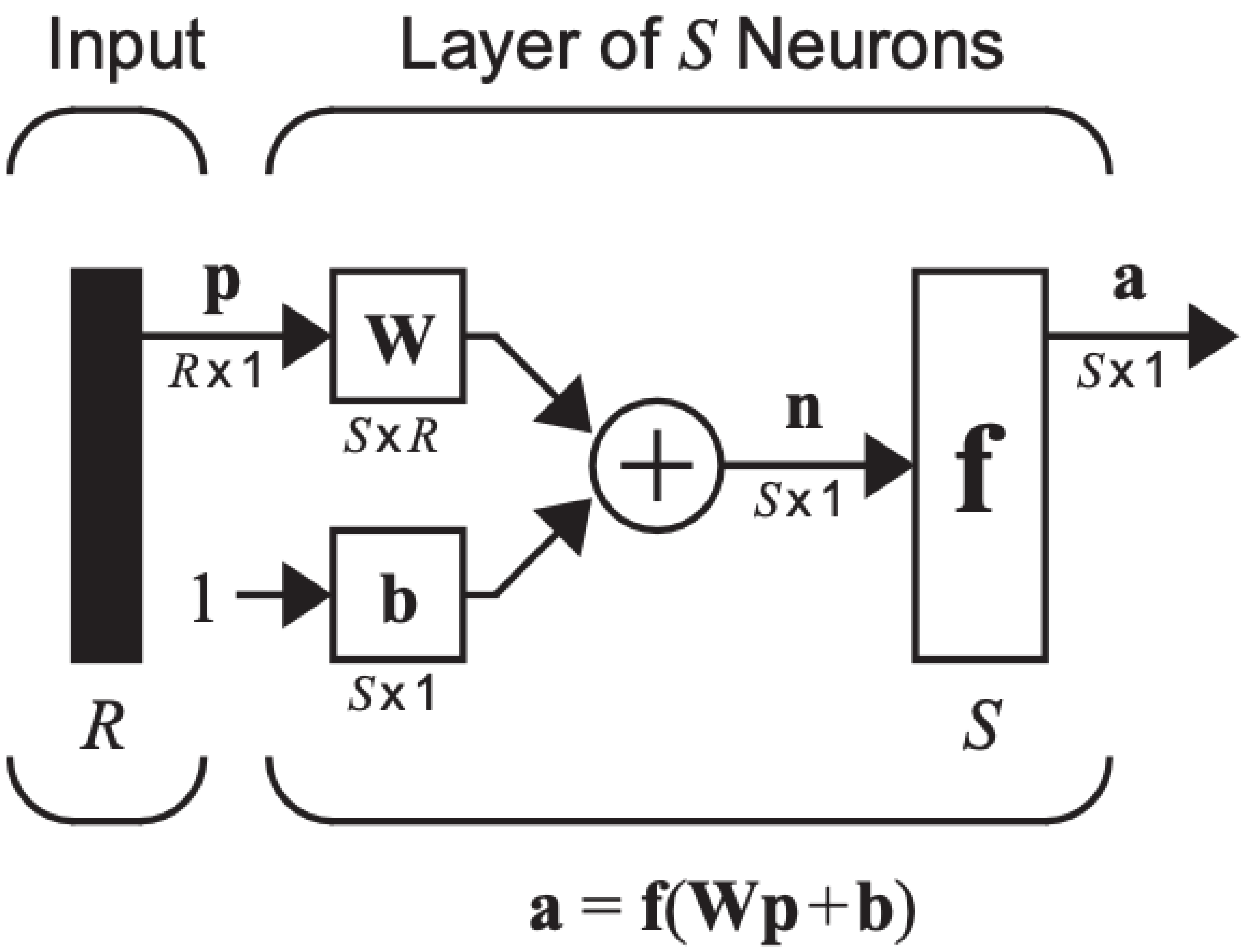
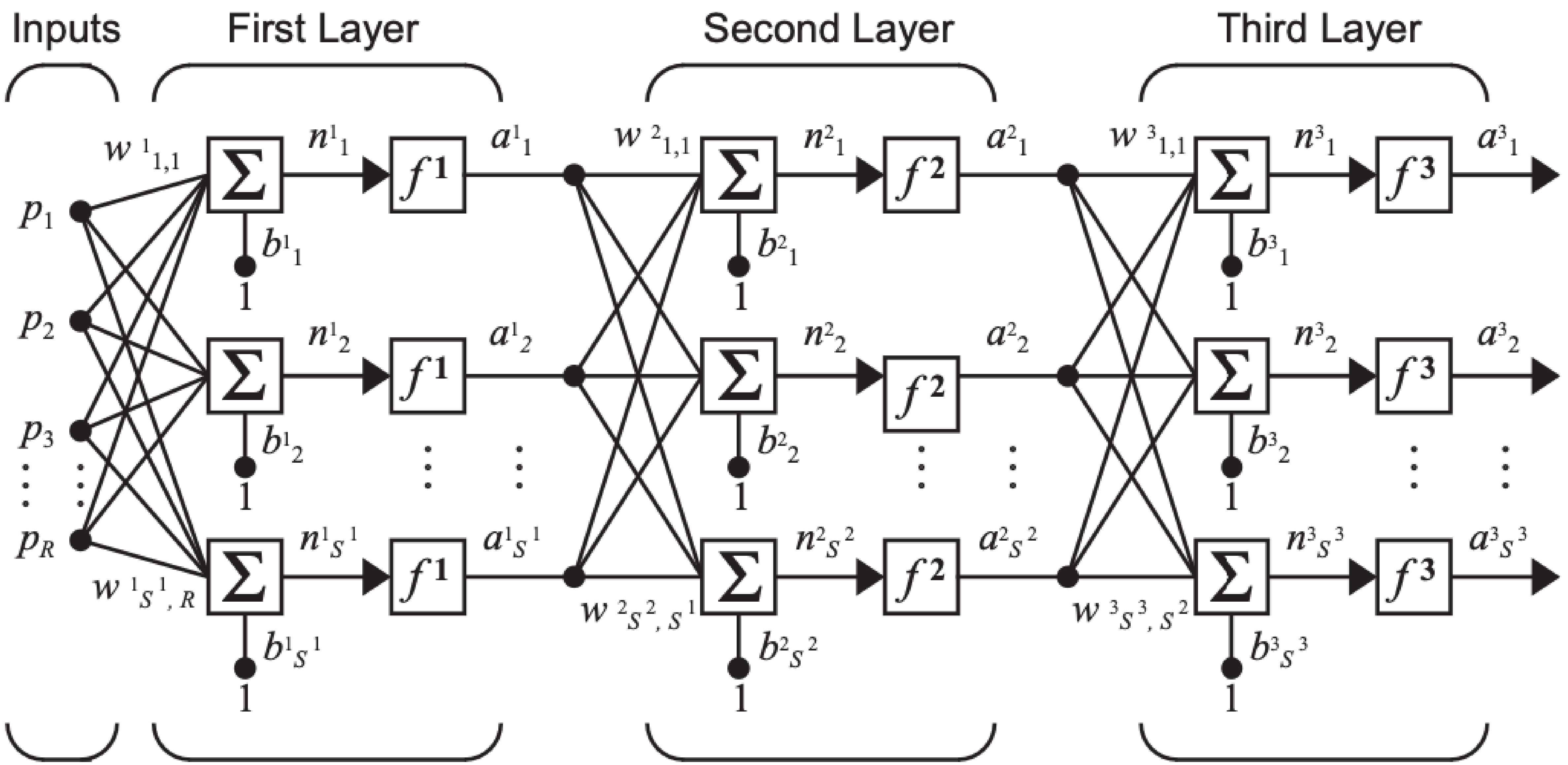


Figure 2.8 Layer of S Neurons, Abbreviated Notation

MÚLTIPLES CAPAS DE NEURONAS

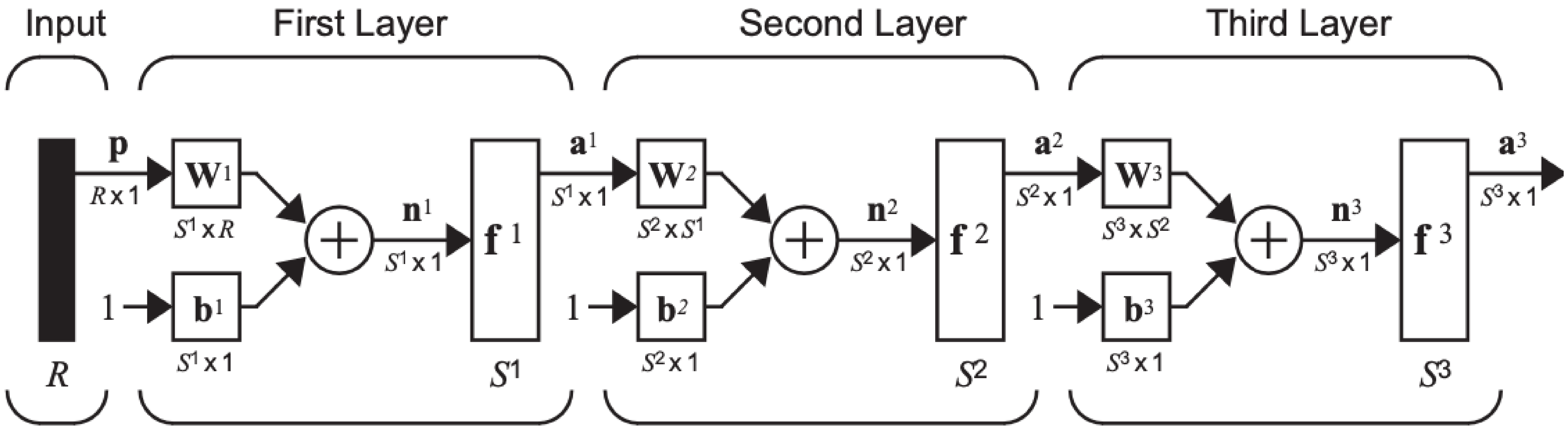


$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{W}^2\mathbf{a}^1+\mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{a}^2+\mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$$

Figure 2.9 Three-Layer Network

MÚLTIPLES CAPAS DE NEURONAS, NOTACIÓN ABREVIADA



$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)$ $\mathbf{a}^2 = \mathbf{f}^2(\mathbf{W}^2\mathbf{a}^1+\mathbf{b}^2)$ $\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{a}^2+\mathbf{b}^3)$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$$

Figure 2.10 Three-Layer Network, Abbreviated Notation

Modelos matemáticos generales

$$a = f(Wp + b)$$

S*1 S*R R*1 S*1

PERCEPTRÓN MULTICAPA

$$V_1 = [1\ 3\ 4\ 2]$$
$$V_1 = [R\ S^1\ S^2\ S^3]$$

1ra capa:

$$a^1 = f^1(W^1 p + b^1)$$

3*1 3*1 1*1 3*1

2da capa:

$$a^2 = f^2(W^2 a^1 + b^2)$$

4*1 4*3 3*1 4*1

3ra capa:

$$a^3 = f^3(W^3 a^2 + b^3)$$

2*1 2*4 4*1 2*1

REDES RECURRENTES

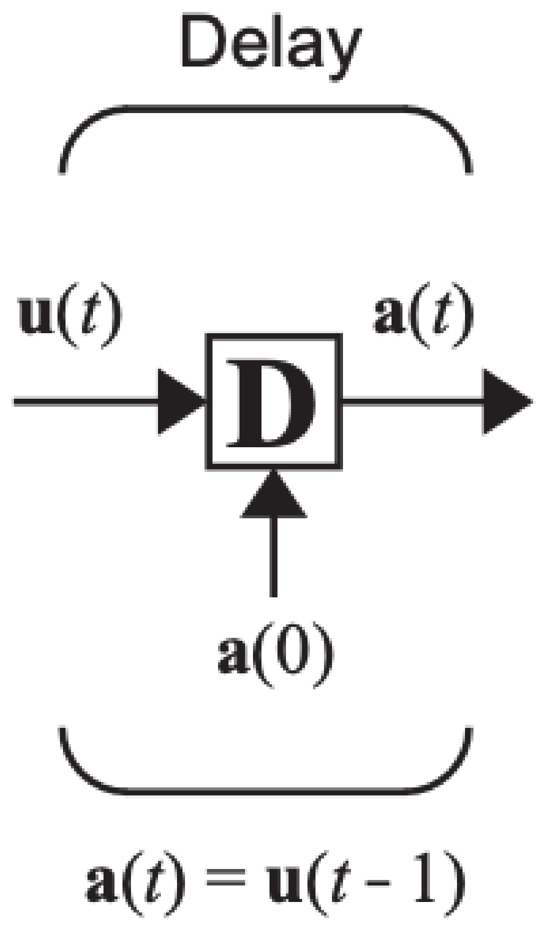


Figure 2.11 Delay Block

EJEMPLO:

t = 0; **a(0)** = 7 ; **u(t)** = [3 4 8]

t = 0 -> a(0) = (0 – 1) = u(–1)

Como u(-1) no está definido, tenemos que usar la condición inicial, tal que:

t = 1 -> a(1) = (1 – 1) = u(3)

t = 2 -> a(2) = (2 – 1) = u(4)

t = 3 -> a(3) = (3 – 1) = u(8)

a(t) = [7 3 4 8]

REDES RECURRENTES

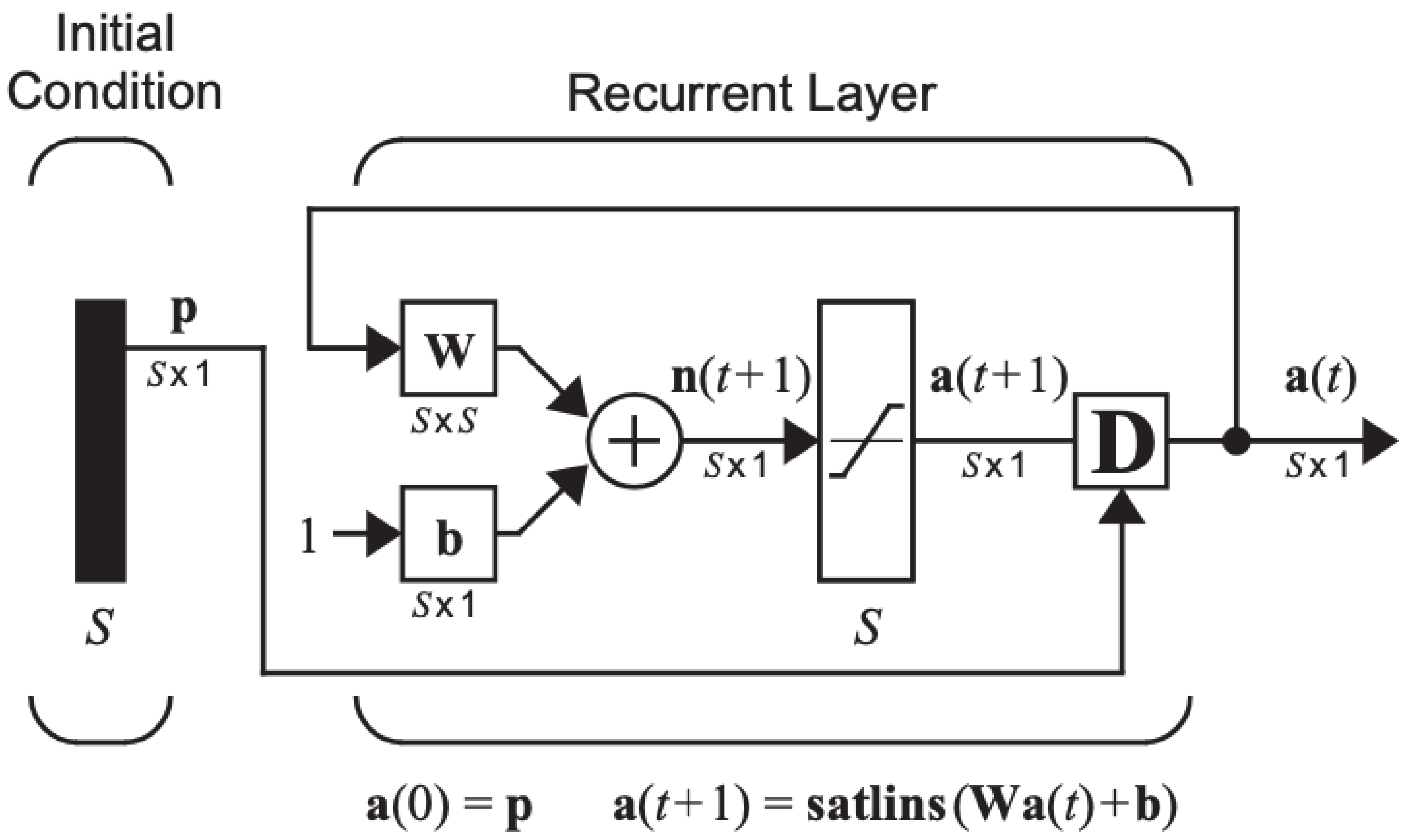


Figure 2.13 Recurrent Network

ENTONCES, ¿CÓMO ELEGIMOS UNA ARQUITECTURA DE RED?

NÚMERO DE ENTRADAS DE LA RED = NÚMERO DE PROBLEMAS DE ENTRADA

NÚMERO DE NEURONAS EN LA CAPA DE SALIDA = NÚMERO DE SALIDAS DE LA RED

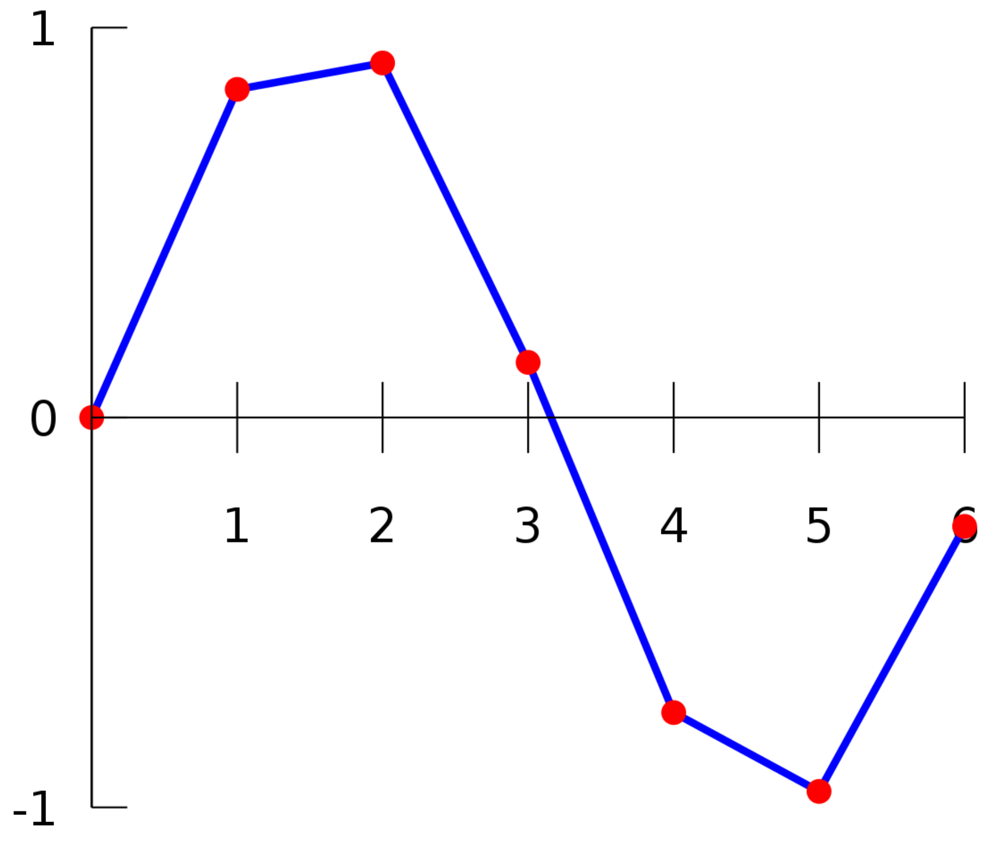
LA ELECCIÓN DE LA FUNCIÓN DE TRANSFERENCIA DE LA CAPA DE SALIDA ESTÁ DETERMINADA, AL MENOS EN PARTE, POR LA ESPECIFICACIÓN DEL PROBLEMA DE LAS SALIDAS.

Introducción a las redes neuronales artificiales

¿Cómo funciona una RNA?

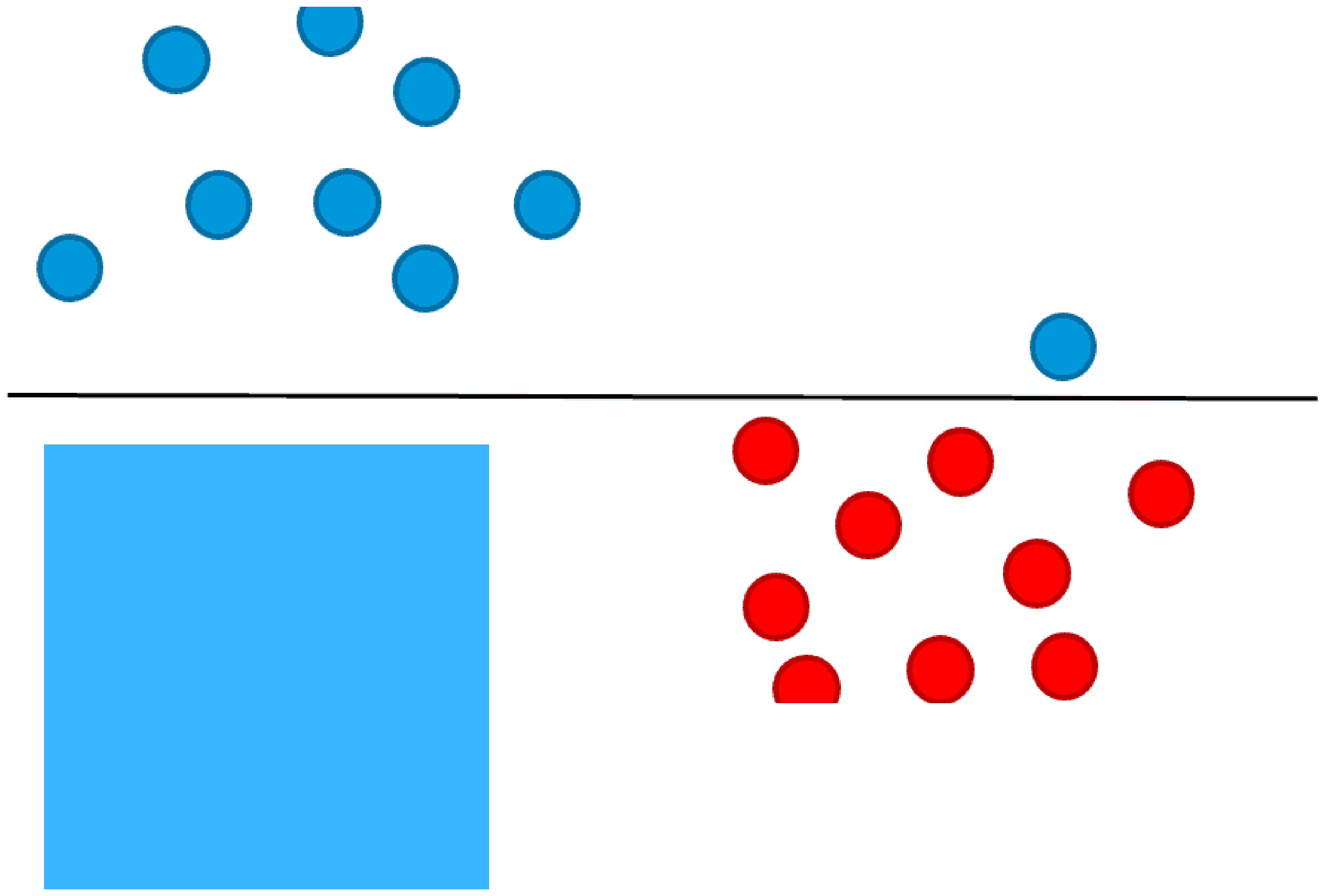
REGRESOR

Ajusta a los datos



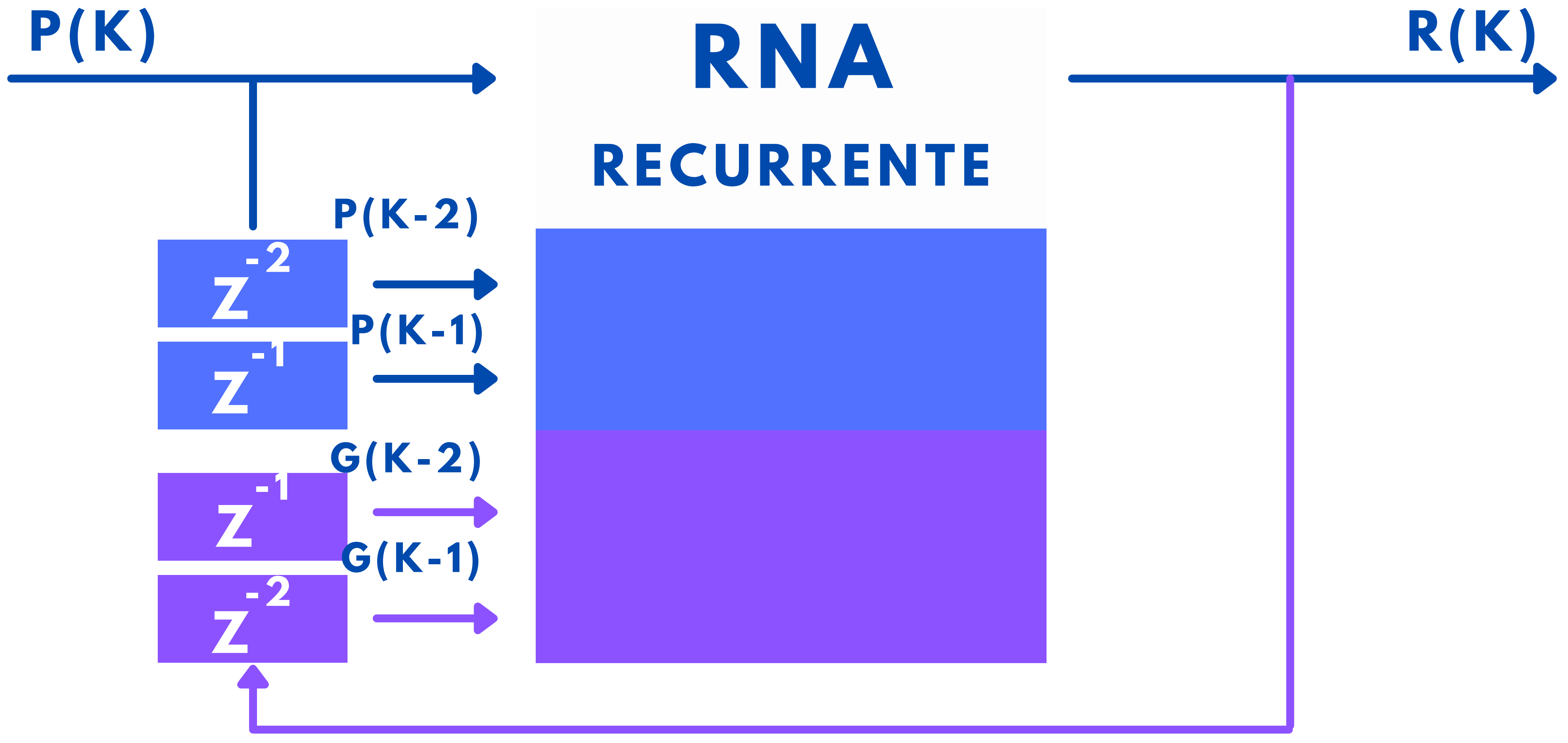
CLASIFICADOR

Separa los datos y dibuja una frontera entre ellos

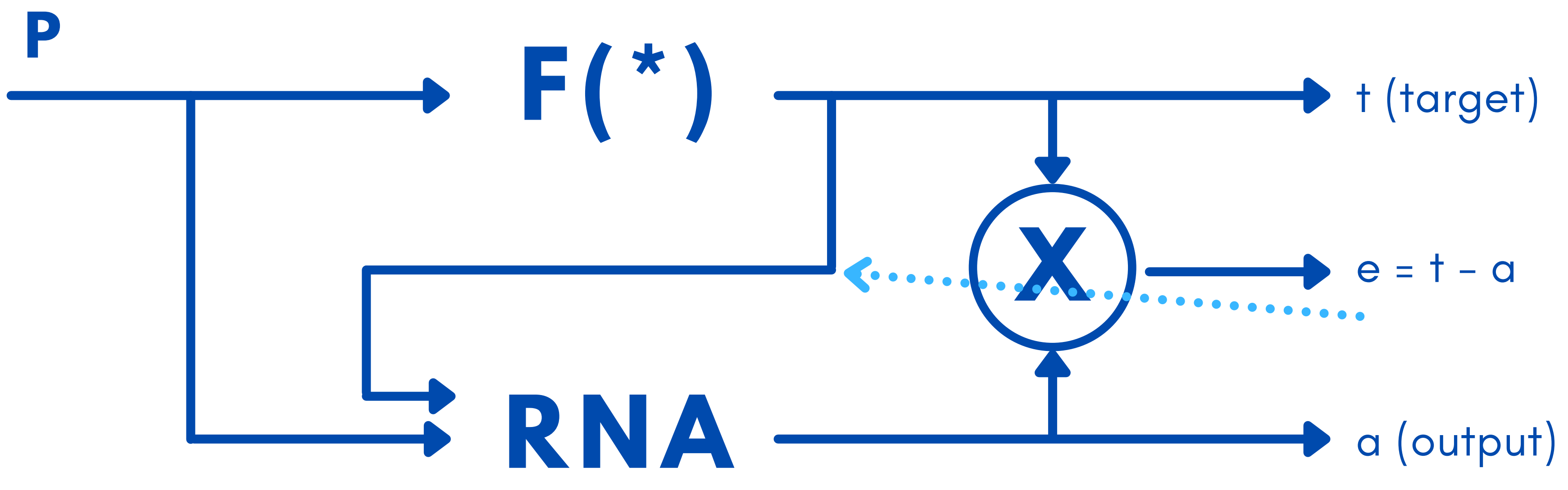


RNA RECURRENTES

Se usan para pronóstico de valores futuros (extrapolación).



AJUSTE DE FUNCIÓN



TIPOS DE APRENDIZAJE

SUPERVISADO

- Pares de objetos.
- Etiquetados.

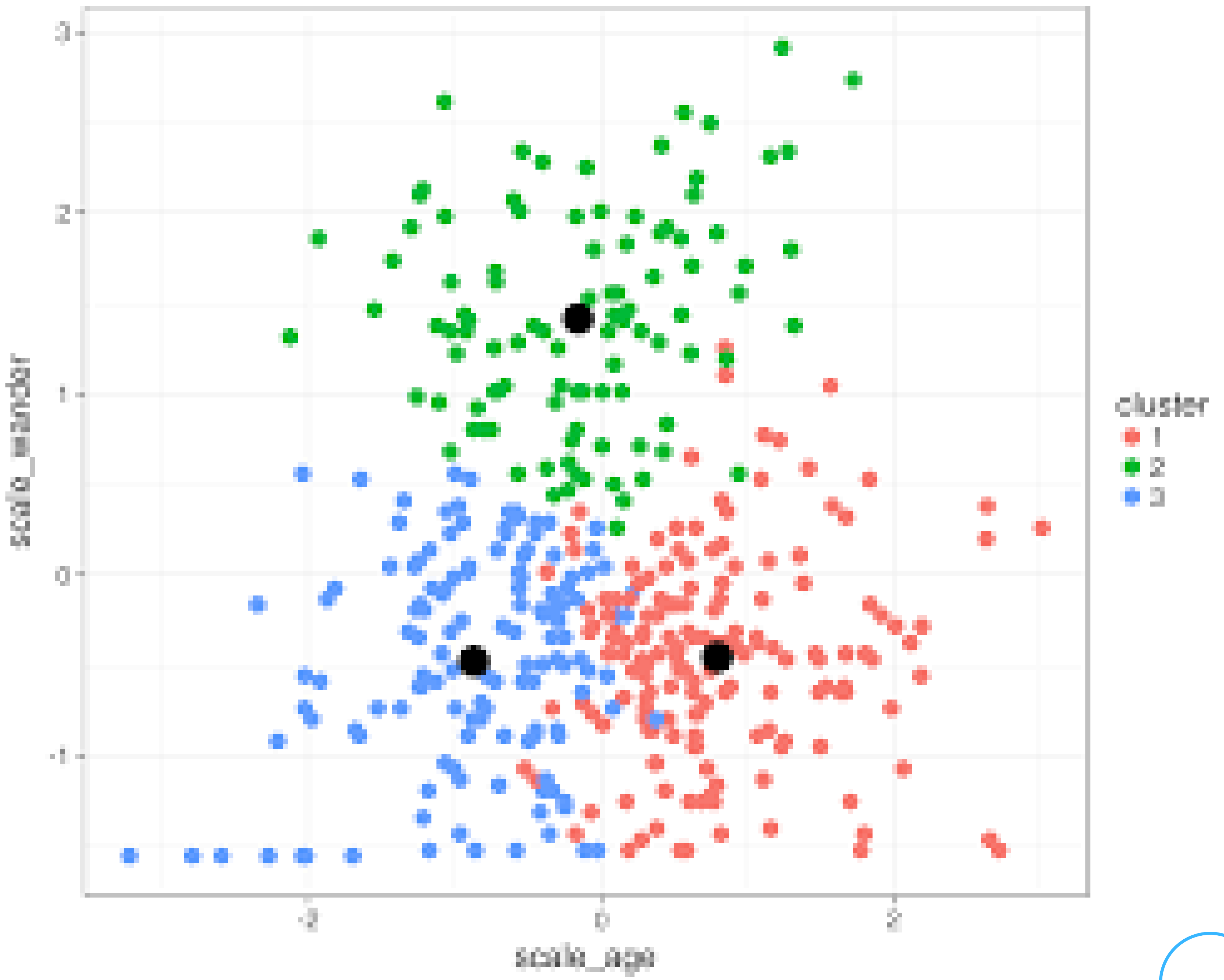
REFORZADO

- Prueba y error.
- Pérdida y recompensa.

NO SUPERVISADO

- Auto-organizado.

K-MEDIAS O K-MEANS



RNA

TEOREMA DE CYBENKO

$$|f(*) - \hat{f}(*)| < \varepsilon$$

MODO CLASIFICADOR

REFERENCIAS

Demuth, H. B., Beale, M. H., Jesús, D. O., & Hagan, M. T. (2014). Neural Network Design (2nd Edition) (2nd ed.). Martin Hagan.