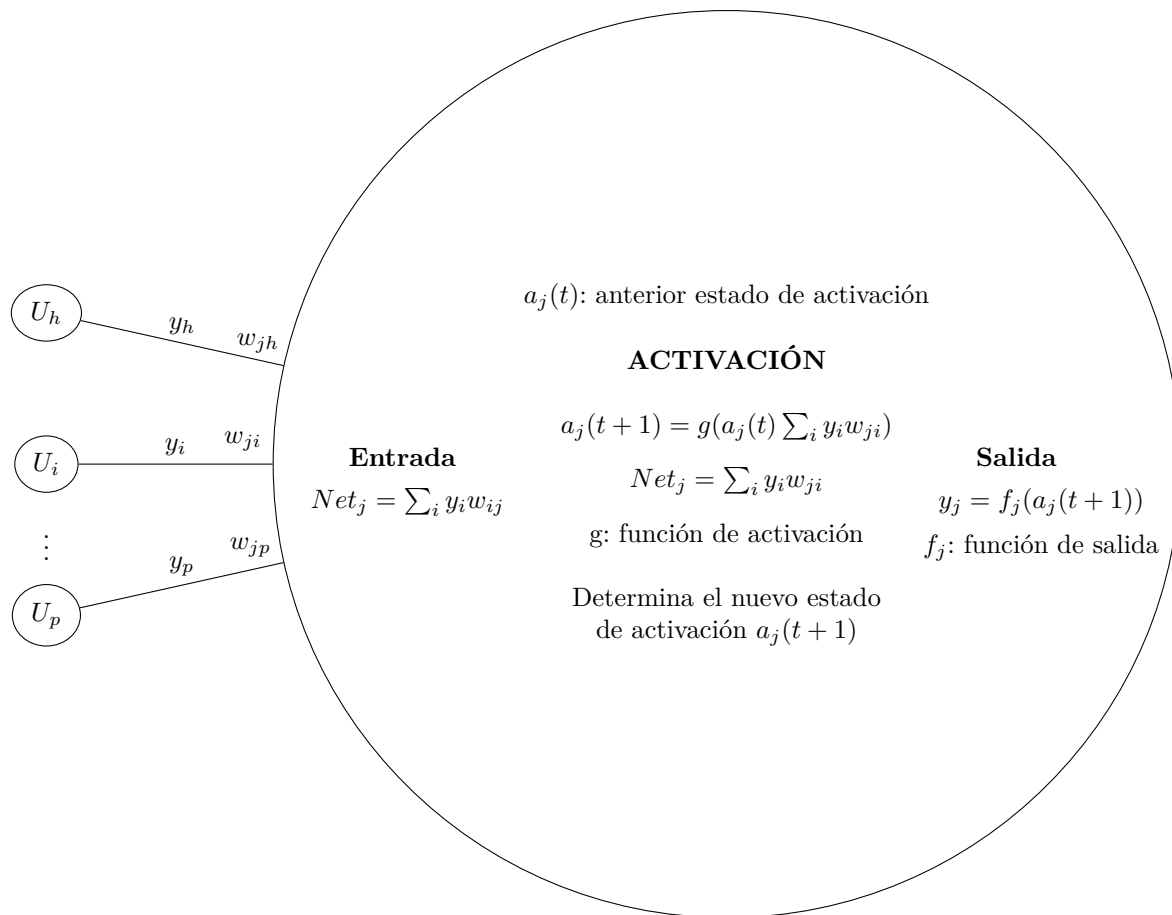


ENTRADAS Y SALIDAS DE UNA NEURONA

U_h, U_i, \dots, U_p : neuronas de entrada



1. ¿Qué estás viendo y para qué sirve?

El diagrama representa una **neurona artificial** que recibe señales de otras unidades, las pondera con **pesos**, actualiza un **estado interno** y produce una **salida**. Esta arquitectura permite modelar relaciones no lineales y, al incluir memoria, capturar dependencias temporales útiles en tareas de *clasificación, regresión y series de tiempo*.

2. Leyenda del diagrama (mapa de símbolos)

- **Entradas** $\{y_i\}$: salidas de neuronas anteriores (p. ej., y_h, y_i, y_p).
- **Pesos** w_{ji} : importancia de la entrada y_i en la neurona j (conexión de i a j).
- **Suma neta** Net_j : combinación lineal de entradas ponderadas

$$\text{Net}_j = \sum_i y_i w_{ji}.$$

- **Estado interno** $a_j(t)$: memoria de la neurona en el instante t .
- **Función de activación** $g(\cdot)$: actualiza el estado.
- **Función de salida** $f_j(\cdot)$: transforma el estado en salida observable y_j .

Nota de índices. Si encuentras w_{ij} en un material equivalente, mantén una convención coherente *origen* \rightarrow *destino*. En esta guía, w_{ji} significa «peso desde i hacia j ».

3. Construcción paso a paso (de izquierda a derecha)

3.1. Entradas \rightarrow pesado

Cada señal y_i llega a través de su conexión con peso w_{ji} . Un peso positivo *refuerza* la entrada; uno negativo la *inhibe*.

3.2. Integración lineal

Las contribuciones se suman en

$$\text{Net}_j = \sum_i y_i w_{ji},$$

que puede leerse como «evidencia total» *previa* a la no linealidad.

3.3. Memoria/Estado

El diagrama incorpora el **estado previo** $a_j(t)$ y sugiere una interacción *multiplicativa* con la evidencia actual:

$$a_j(t+1) = g(a_j(t) \text{Net}_j).$$

En otros textos verás variantes *aditivas*, por ejemplo $a_j(t+1) = g(\text{Net}_j + u_j a_j(t) + b_j)$, pero el espíritu es el mismo: la nueva activación depende de la información entrante y de la memoria.

3.4. No linealidad (activación)

La función g introduce no linealidad, controla escalas y previene saturación desbordada del estado. Opciones habituales: tanh, sigmoide, ReLU, GELU. La elección influye en *estabilidad temporal* y *capacidad de representación*.

3.5. Observación (salida)

La salida visible es

$$y_j = f_j(a_j(t+1)).$$

Casos típicos: identidad (regresión), sigmoide (binaria), y softmax a nivel de capa (multiclase).

4. Lectura interpretativa del diagrama

- **Rol de Net_j :** evidencia *actual* proveniente de las entradas.
- **Rol de $a_j(t)$:** *memoria*/inercia: el nuevo estado no parte de cero.
- **No linealidad g :** decide cómo se combinan memoria y evidencia; regula saturación y estabilidad.
- **Salida y_j :** lo que «ve» el resto de la red o el usuario.

5. Ecuaciones clave del modelo (según el gráfico)

$$\text{Net}_j = \sum_i y_i w_{ji}, \quad a_j(t+1) = g(a_j(t) \text{Net}_j), \quad y_j = f_j(a_j(t+1)).$$

Si se incluye un *sesgo* b_j , puede incorporarse en Net_j o dentro del argumento de g para desplazar la respuesta.

6. Ejemplo guiado (para comprender el flujo)

1. Define entradas $\mathbf{y} = (y_1, y_2, y_3)$.
2. Define pesos hacia j : $\mathbf{w}_{j\leftarrow} = (w_{j1}, w_{j2}, w_{j3})$.
3. Calcula $\text{Net}_j = \mathbf{y} \cdot \mathbf{w}_{j\leftarrow}$.
4. Toma el estado previo $a_j(t)$.
5. Aplica g : $a_j(t+1) = g(a_j(t) \text{Net}_j)$.
6. Obtén la salida: $y_j = f_j(a_j(t+1))$.

Lectura cualitativa: si $\text{Net}_j > 0$ y $a_j(t) > 0$, la activación tenderá a *reforzar* el estado; si tienen signos opuestos, tenderá a *debilitarlo* (según la forma de g).

7. Variantes y extensiones útiles

- **Sesgo b_j** : ajusta el «punto de encendido» de la neurona.
- **Normalización de entradas**: estabiliza aprendizaje y magnitudes del estado.
- **Recurrente aditiva**: $a_j(t+1) = g(\text{Net}_j + u_j a_j(t) + b_j)$, frecuente en RNN simples.
- **Capa de salida**: elegir f_j coherente con la tarea (regresión/probabilidades).

8. Errores típicos al estudiar este diagrama

- **Confundir índices** en los pesos (origen vs. destino).
- **Omitir el sesgo** en la implementación.
- **Elegir g incompatible** con la escala del problema (saturaciones prematuras).
- **No separar «estado» de «salida»**: a_j no es necesariamente la magnitud reportada al exterior.

Cierre

El diagrama describe una neurona **con dinámica**: integra señales actuales (Net_j), combina con su **estado previo** $a_j(t)$ mediante una **no linealidad** g , y entrega una **salida** y_j . Esta secuencia explica qué computa la neurona, por qué incluye memoria y cómo su diseño (elección de g y f_j) se alinea con la tarea.