

El Plano Computacional

Anatomía, Ingeniería y Diagnóstico de las Redes Neuronales Artificiales

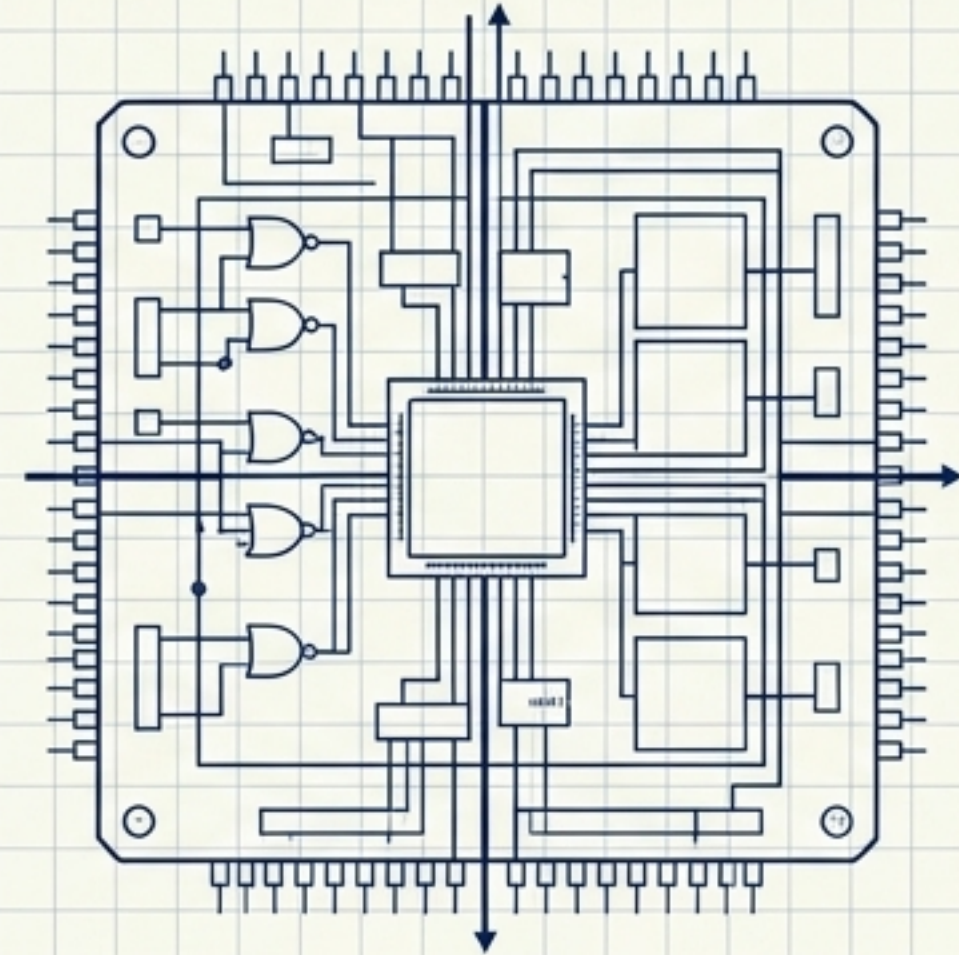
Síntesis de Arquitectura de Sistemas y Optimización Heurística

El Paradigma Biológico: Eficiencia y Paralelismo

El Cerebro



Circuito Tradicional



Velocidad: Lenta (10^{-3} s)

Arquitectura: Masivamente paralela
($\approx 10^{11}$ neuronas, $\approx 10^4$
conexiones/neurona)

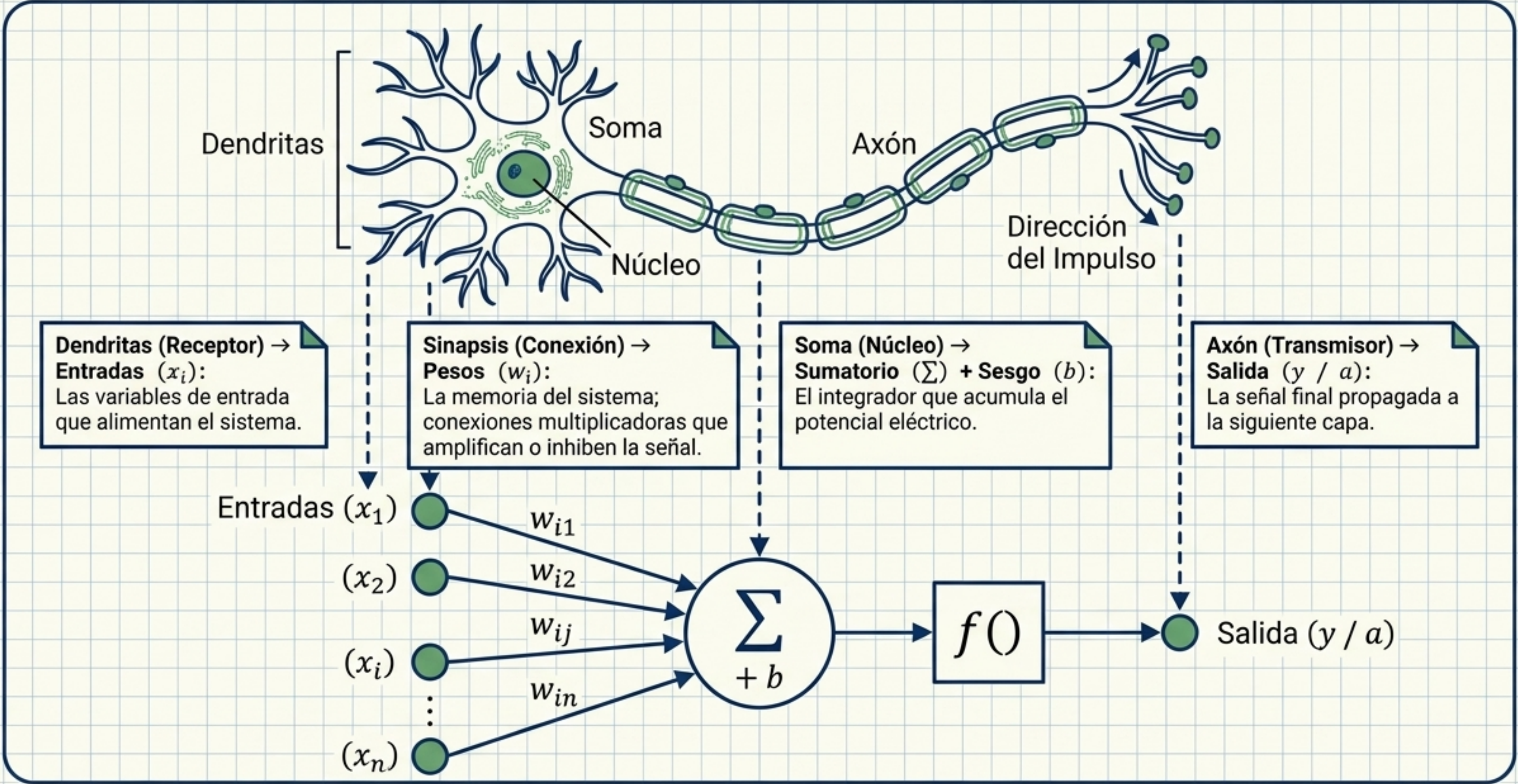
Ventajas: Tolerante a fallos, flexible (aprende sin programación explícita), maneja ruido e información cualitativa, bajo consumo (20% de la energía corporal).

Velocidad: Rápida (10^{-9} s)

Arquitectura: Secuencial,
rígidamente programada.

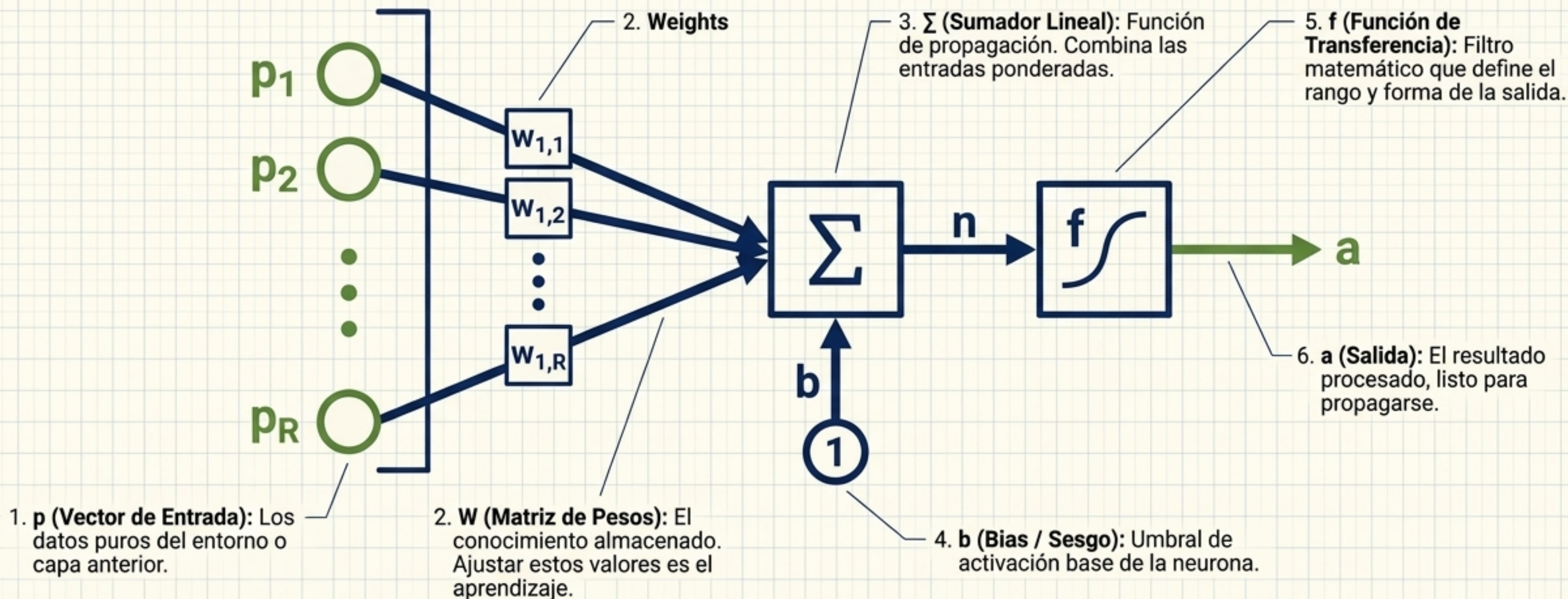
Desventajas: Falla ante daños estructurales, incapacidad innata para manejar ruido o ambigüedad sin reglas explícitas.

Diccionario de Traducción Biológica-Matemática

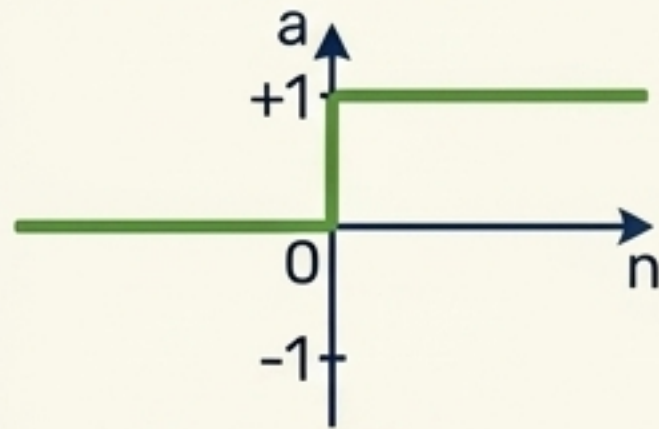


Arquitectura de la Unidad Básica (La Neurona Artificial)

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



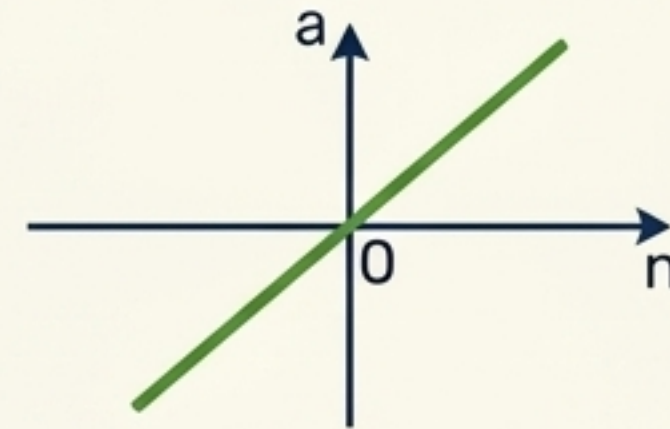
Galería de Funciones de Transferencia



Limitador Fuerte (hardlim)

Rango: $[0, 1]$

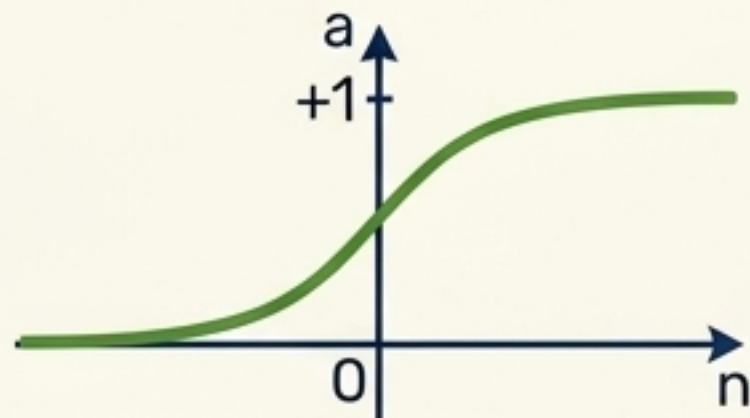
Uso: Funciones de discriminación binaria o decisiones tipo interruptor.



Lineal (purelin)

Rango: $[-\infty, +\infty]$

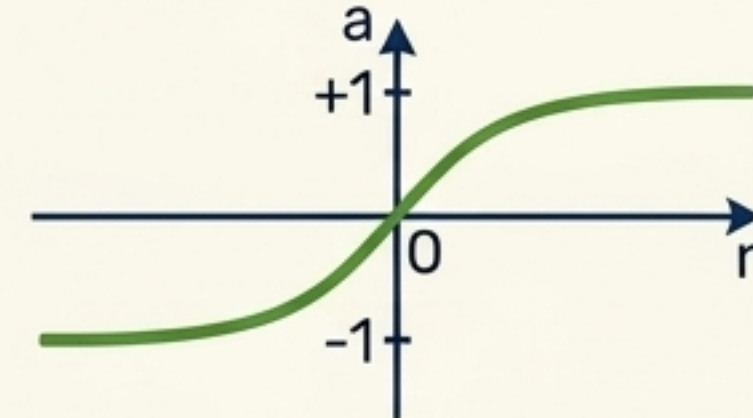
Uso: Aproximación lineal y capas de salida donde no se requiere acotar el valor.



Sigmoideal Logarítmico (logsig)

Rango: $[0, 1]$

Uso: Compresión de señales asimétricas, mapeo de probabilidades.



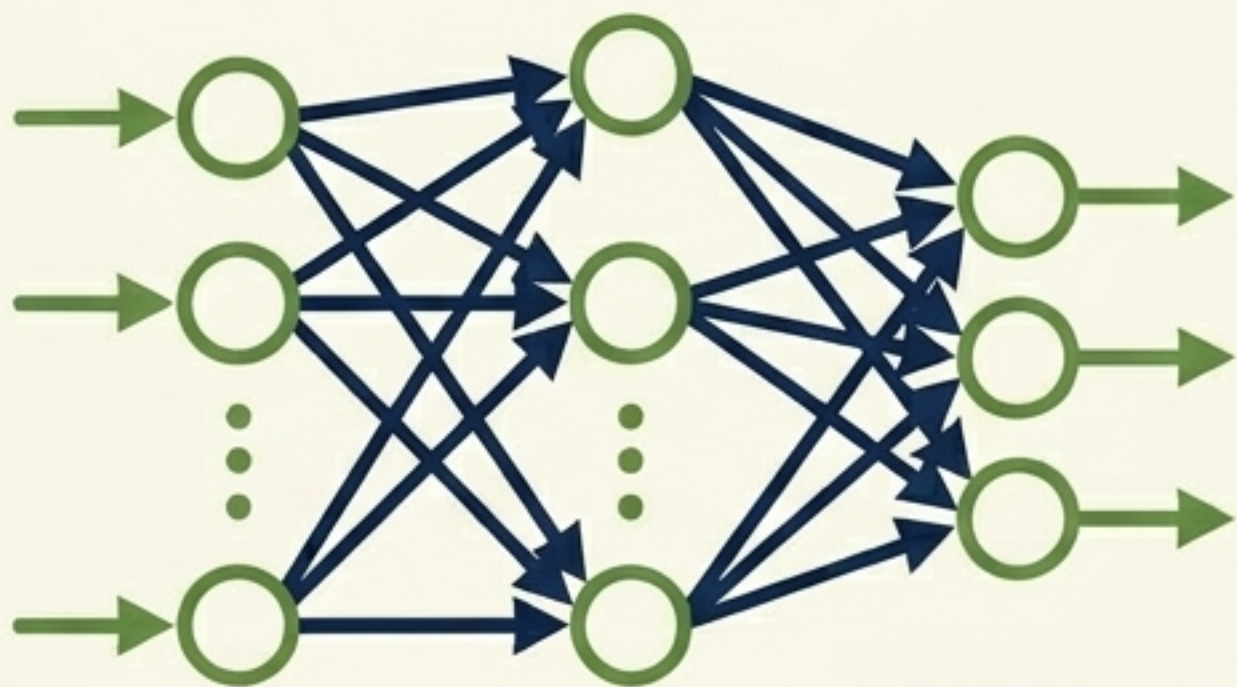
Tangente Sigmoidea (tansig)

Rango: $[-1, 1]$

Uso: Ideal para datos centrados en cero, mayor gradiente que logsig.

Topología del Sistema: Unidireccional vs. Recurrente

Redes Hacia Adelante (Feed-forward)

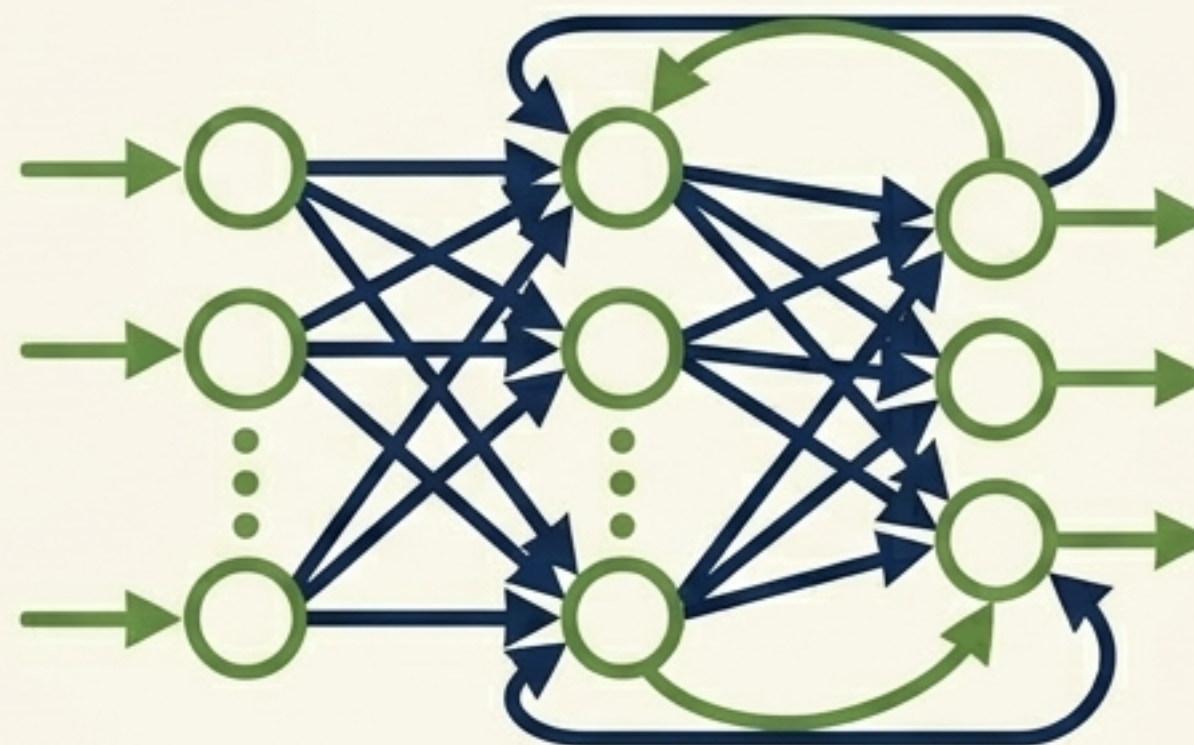


Flujo: La información viaja en un solo sentido (Entrada → Capas Ocultas → Salida).

Dependencia: La salida actual solo depende de la entrada actual.

Aplicación: Aproximación de funciones estáticas, reconocimiento de patrones.

Redes Recurrentes



Flujo: Contiene ciclos de retroalimentación; la información retorna a capas anteriores.

Dependencia: La salida depende de la entrada actual y del estado interno previo (memoria).

Aplicación: Series temporales, secuencias dinámicas, procesamiento de lenguaje.

El Motor de Aprendizaje: Minimización del Error

1. Propagación

Los datos atraviesan la red generando una salida predictiva.

2. Cálculo de Coste

Se mide el **Error Cuadrático Medio** (MSE) entre la predicción y el dato real (t).

$$MSE = \frac{\sum(\text{prediction} - t)^2}{N}$$

4. Actualización

Se ajustan los **pesos** (w) y **sesgos** (b) y se repite la iteración hasta cumplir el criterio de parada.

3. Optimización

Búsqueda del mínimo de la función. Dos enfoques principales:

Gradiente:

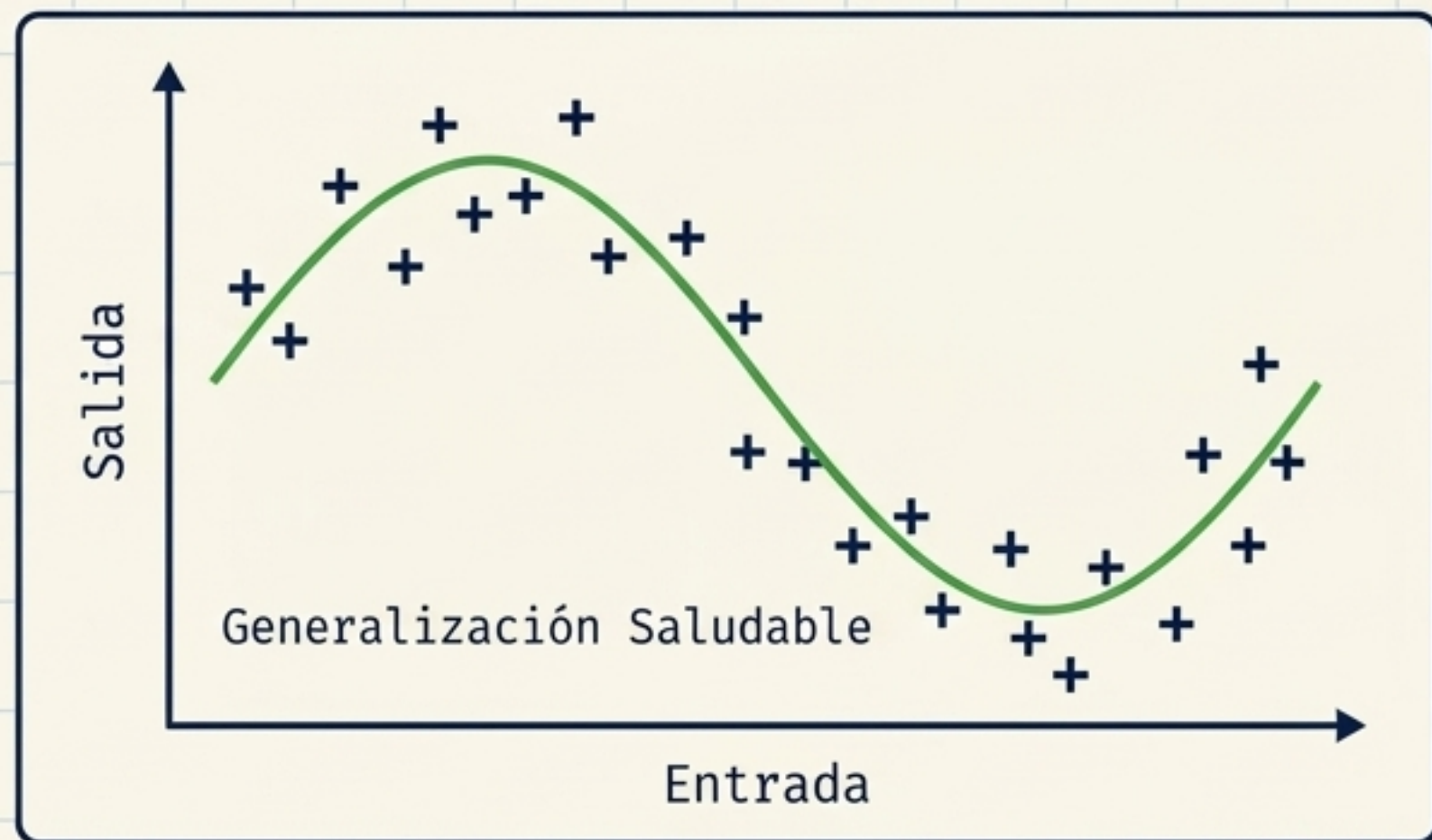
Derivada multidimensional; ajuste en serie (búsqueda local). ⚠️

Algoritmos Genéticos:

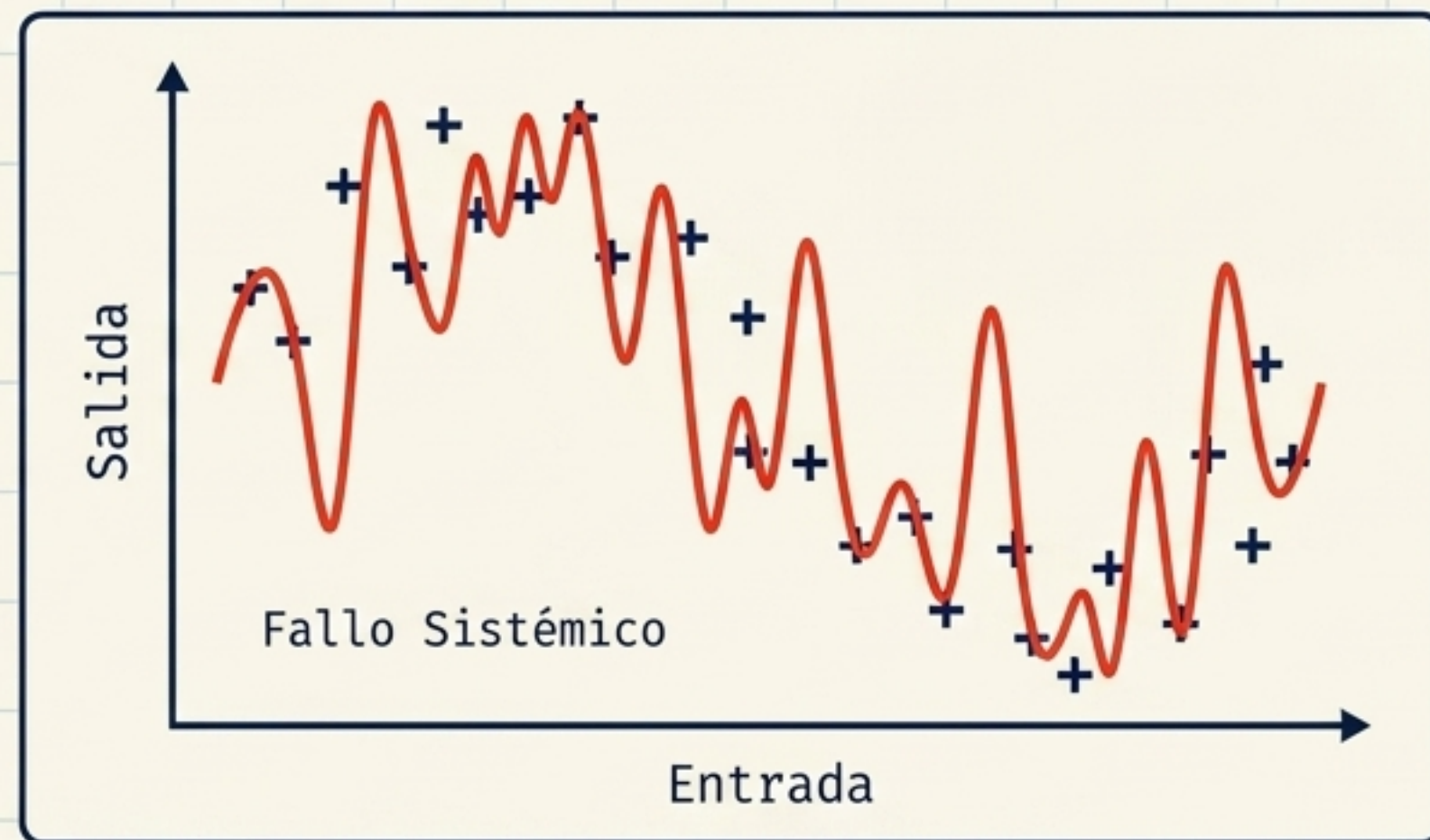
Mutación de parámetros; búsqueda en paralelo (sobrevive el hijo con **menor error**). ⚠️

La Paradoja del Aprendizaje: El Peligro del Sobreaprendizaje

Ajuste Óptimo



Sobreaprendizaje (Overfitting)

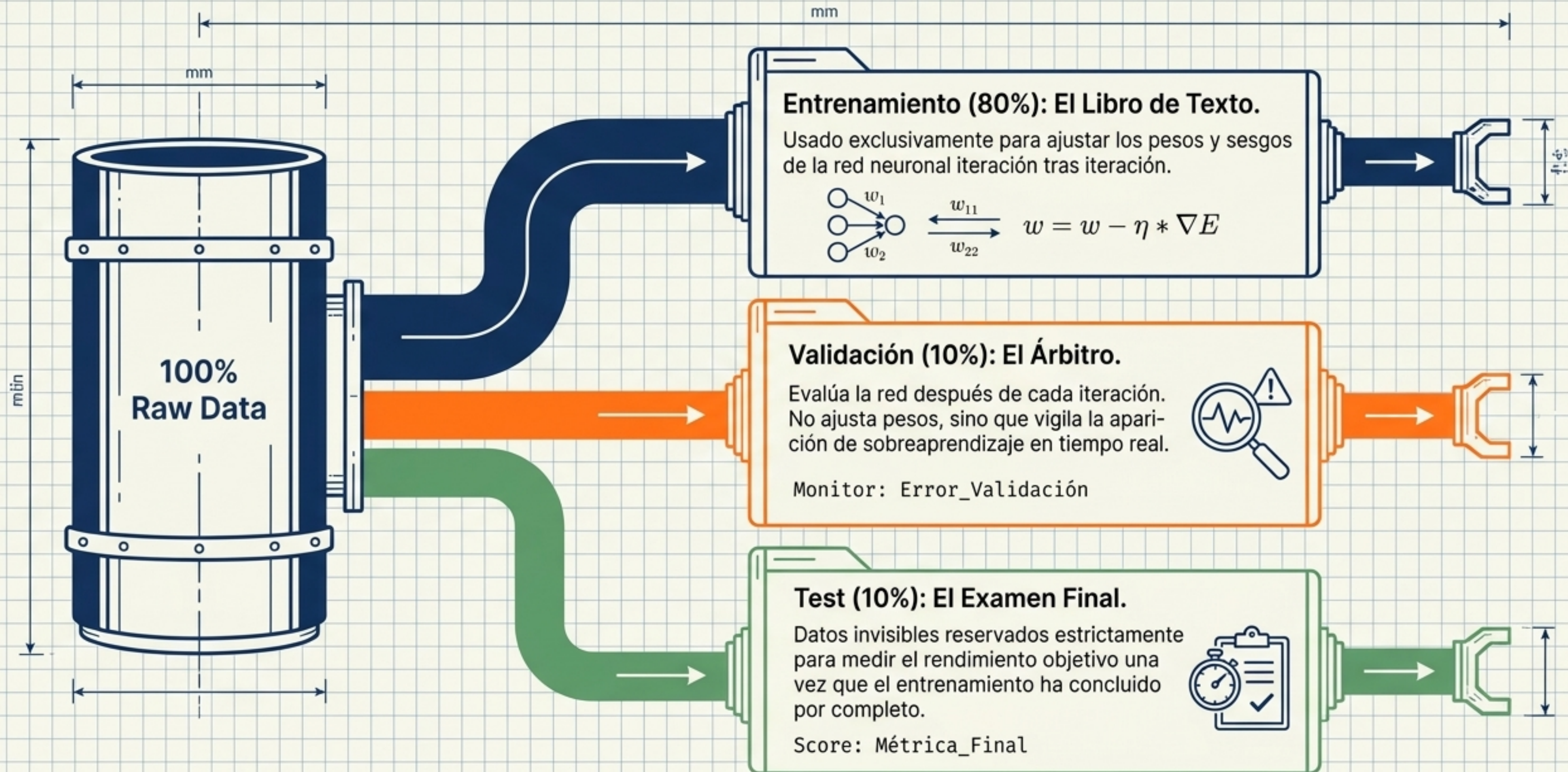


Un error de **entrenamiento de cero** puede ser el síntoma de una **red colapsada**.

Explanation: El Sobreaprendizaje (Pérdida de Generalización) ocurre cuando una red dispone de demasiados parámetros y comienza a memorizar el ruido y los errores de los datos, en lugar de extraer el patrón matemático subyacente.

Consequence: Ajuste perfecto en laboratorio, fracaso absoluto ante datos nuevos del mundo real.

Estrategia de Contención: División de Datos



Panel de Diagnóstico I: Early-Stopping

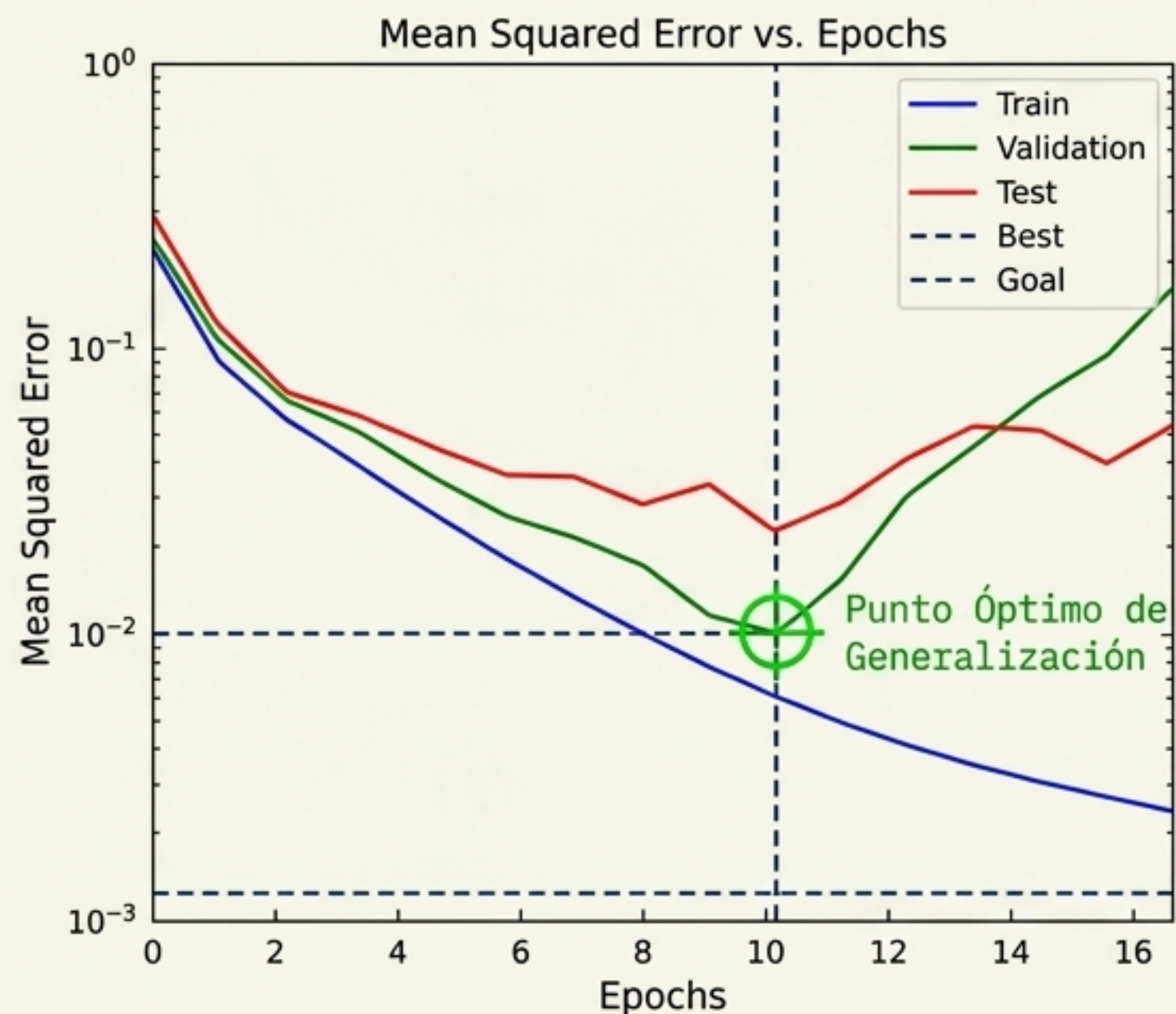
Diagnostico Dossier Carder

El Síntoma:

Durante las iteraciones sucesivas, el error de entrenamiento sigue disminuyendo, pero el error de validación comienza a ascender bruscamente.

La Intervención:

Se ignora el objetivo de MSE mínimo del entrenamiento. El proceso se detiene inmediatamente en el punto de inflexión donde las curvas divergen. La red revierte a los pesos exactos de ese punto óptimo de generalización.



Panel de Diagnóstico II: Regularización (La Navaja de Ockham)

$$MSE = MSW + MSE_{data}$$

El Principio:

A igualdad de condiciones, la explicación más simple es la correcta. Redes más simples son más robustas frente al ruido.

La Intervención:

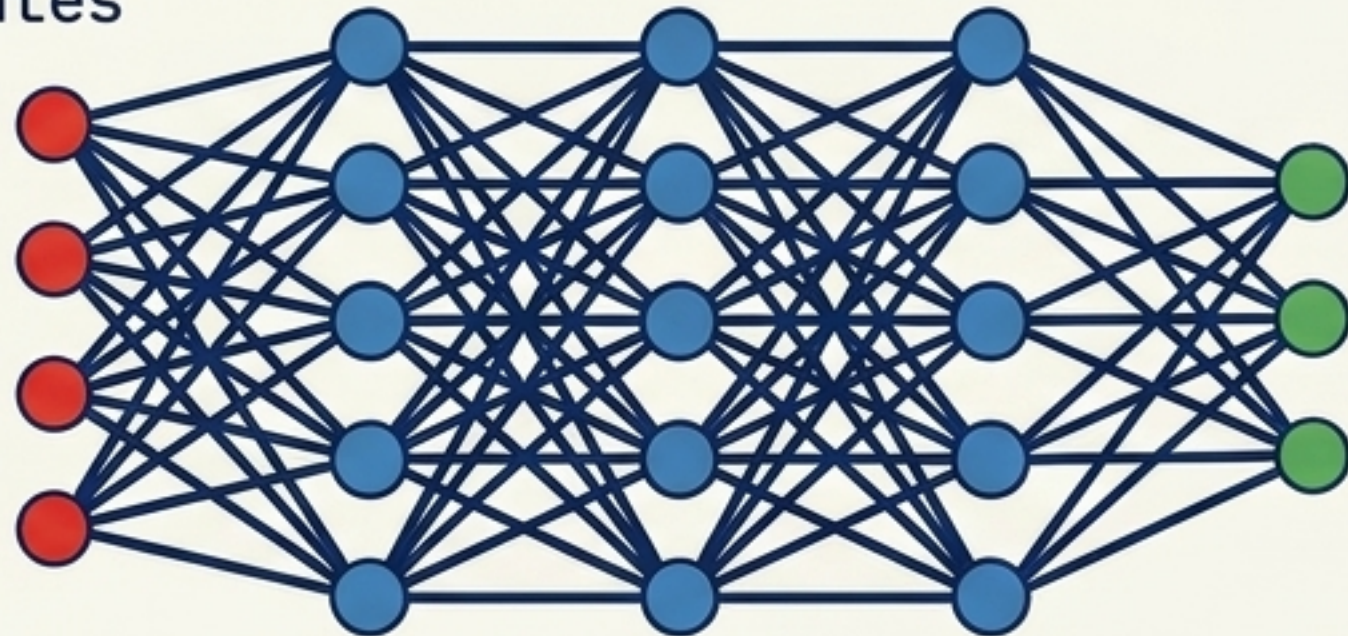
Modificar la función de coste tradicional. En lugar de minimizar solo el error (MSE), se agrega un sumatorio ponderado del tamaño de los pesos de la red (MSW).

El Resultado:

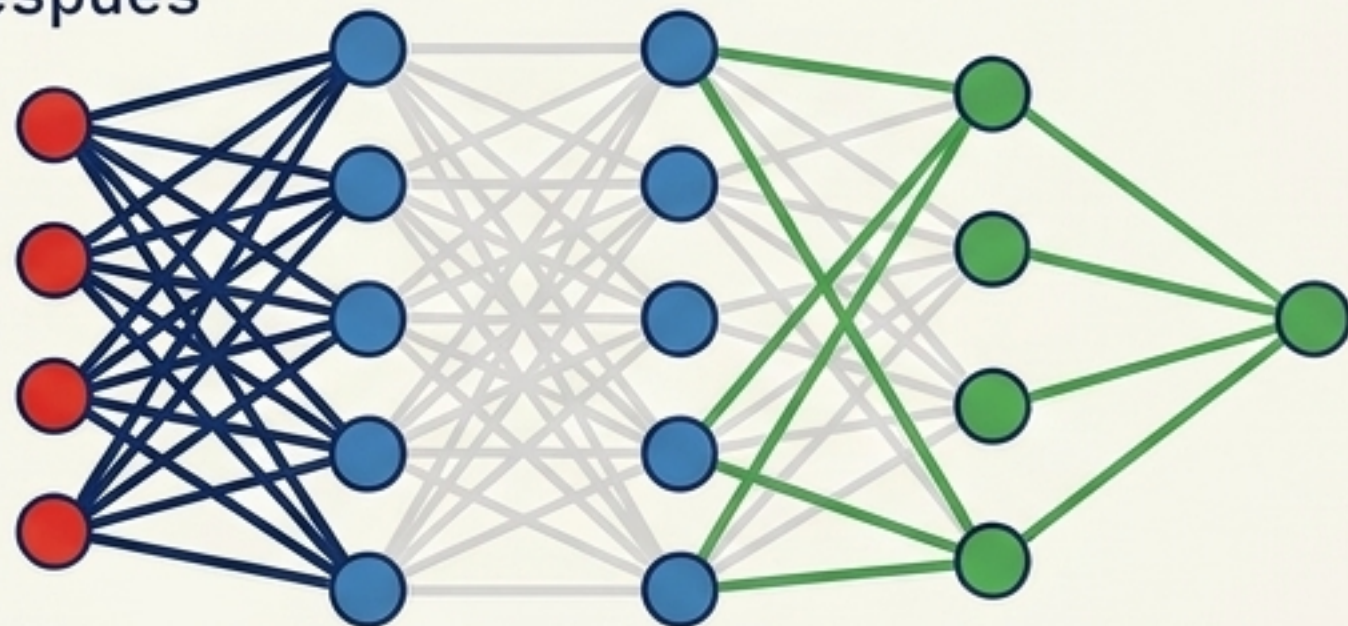
Si dos redes tienen el mismo margen de error, el algoritmo penalizará la que tenga pesos grandes y caóticos, favoreciendo la arquitectura con valores más pequeños y estables.

Panel de Diagnóstico III: Redes Podadas (Pruning)

Antes



Después



El Síntoma:

Variables de entrada redundantes o poco relevantes generan complejidad innecesaria.

La Intervención:

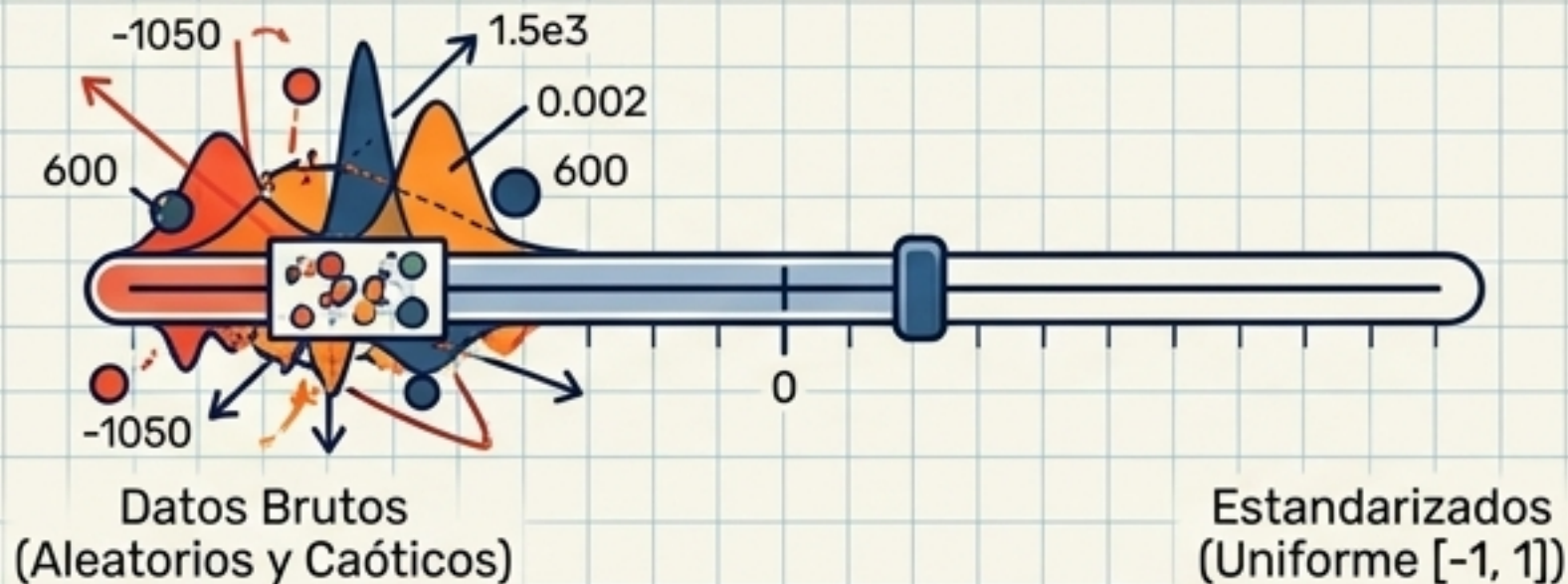
Desconexión física de neuronas o eliminación de entradas. Se emplea la función de coste *PSE* (*Predicted Squared Error*), la cual castiga matemáticamente a la red basándose en su número total de parámetros en relación con la cantidad de datos disponibles.

El Resultado:

Una red con grados de libertad estrictamente limitados, forzada a extraer solo las características esenciales.

Calibración Previa: Estandarización y Dimensionamiento

Panel 1: Estandarización de Datos

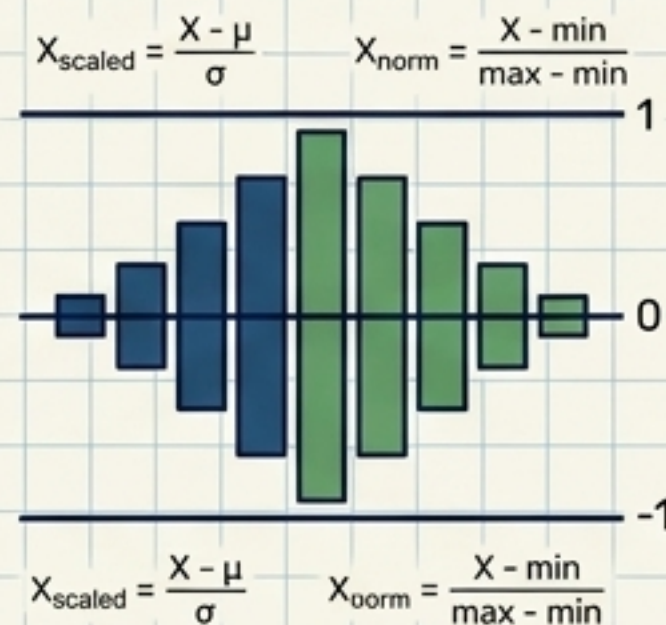


El Problema:

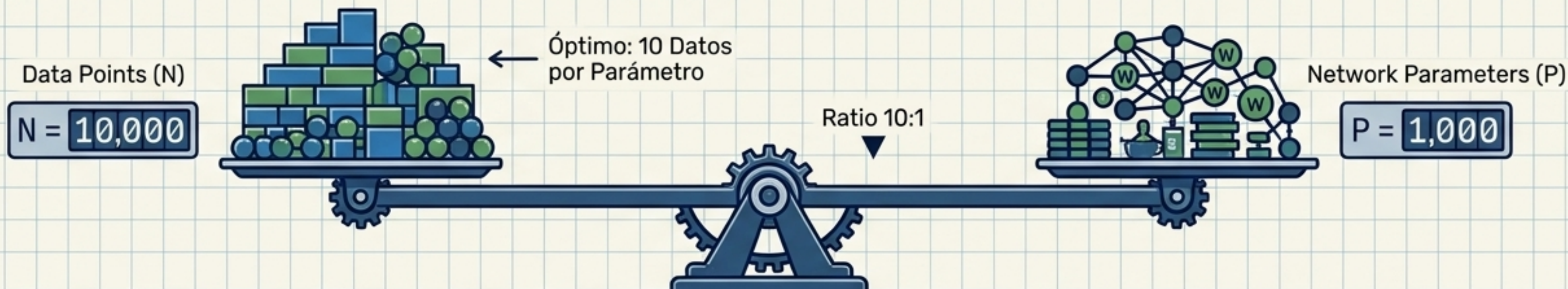
Los pesos se inician aleatoriamente con valores pequeños; datos no escalados bloquean el entrenamiento.

La Solución:

Forzar todas las variables a un rango uniforme (ej. [-1, 1]) o centrar en 0 con desviación típica 1. Funciona exponencialmente mejor si se linealizan distribuciones no normales (ej. $Y = \ln(X)$).



Panel 2: Regla de Dimensionamiento

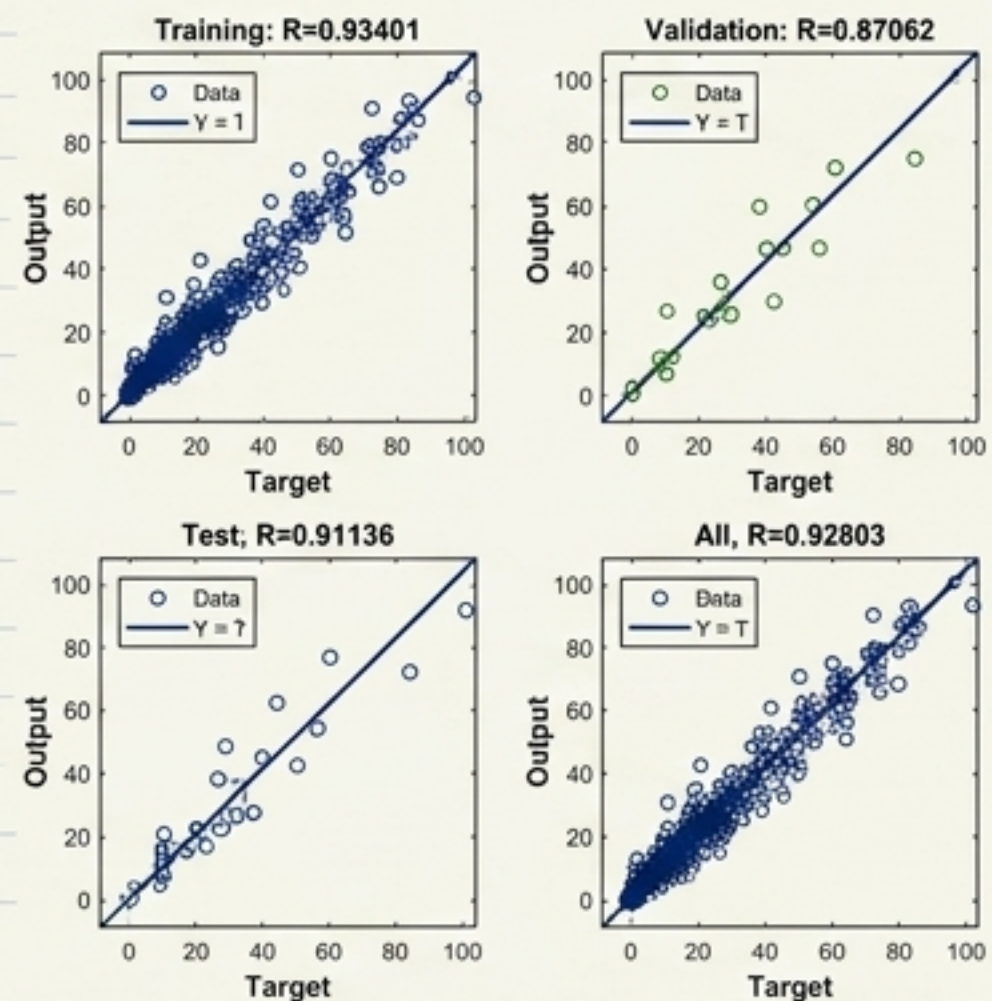


El Principio: El número de parámetros (grados de libertad) no debe superar la cantidad de datos.
Regla general óptima: 10 datos disponibles por cada 1 parámetro de la red.

$$N \geq 10 * P$$

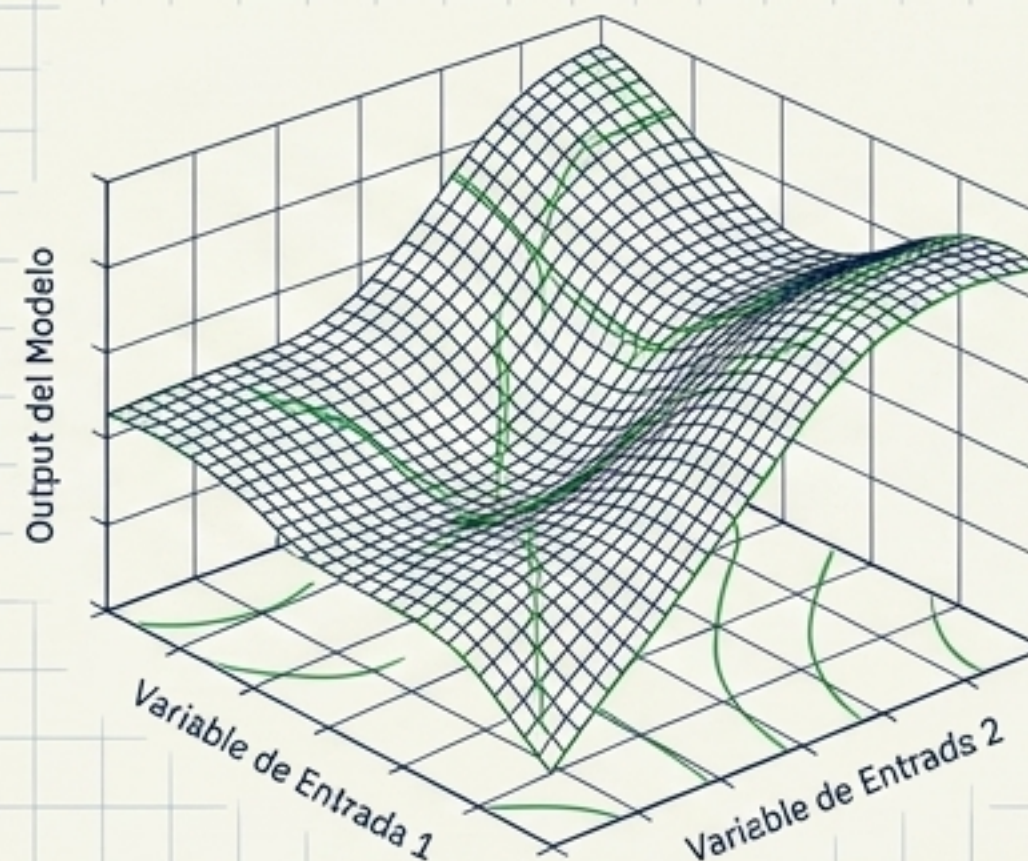
Simulación: El Laboratorio Virtual

Validación Cruzada



Validación Cruzada: Comprobación gráfica de la bondad de ajuste comparando la salida de la red contra el objetivo real (Target). En un modelo perfecto, todos los puntos descansan exactamente sobre la recta $Y=T$.

Curvas de Diseño



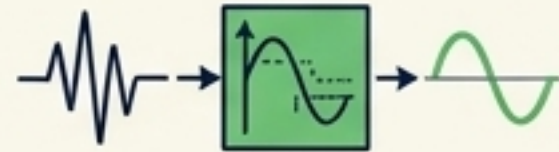
Curvas de Diseño: Una vez validada sin sobreaprendizaje, la red genera proyecciones instantáneas. Permite fijar todas las variables excepto una, observando en tiempo real la influencia aislada de entradas específicas sin necesidad de simulaciones físicas costosas.

Balance del Sistema: Ventajas y Desafíos



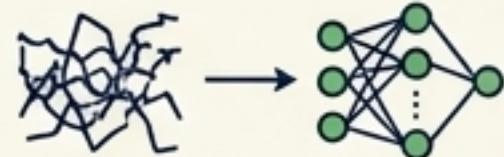
Ventajas (Pros):

Alta robustez frente a ruido en datos de entrada/salida.



SIGNAL_INTEGRITY: ROBUST (>98%).

Independencia entre la complejidad del problema y el diseño de la red.



DESIGN_FLEXIBILITY: MAXIMAL.
COMPLEXITY \neq NETWORK_STRUCTURE.

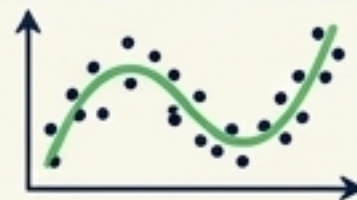
Rapidez extrema en la fase de simulación (laboratorio virtual).



EXECUTION_SPEED: EXTREME (ms).

SIMULATION: VIRTUAL_LAB

Capacidad de aproximar variables universales altamente no-lineales.



APPROXIMATION: UNIVERSAL_NONLINEAR_FUNCTIONS

Executive Scorecard



Desafíos (Cons):

Riesgo constante de convergencia en mínimos locales (soluciones subóptimas).



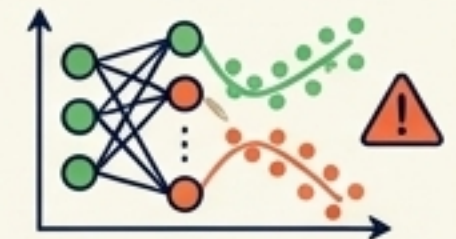
CONVERGENCE_RISK: LOCAL_MINIMA (SUBOPTIMAL).

Comportamiento de Caja Negra (opacidad en el razonamiento lógico).



LOGIC_TRANSPARENCY: OPAQUE
(BLACK_BOX_MODEL).

Peligro crítico de pérdida de generalización (sobreaprendizaje).



CRITICAL_RISK: OVERFITTING
(POOR_GENERALIZATION).

Incapacidad absoluta para extrapolar de manera segura fuera del rango de los datos de entrenamiento.



SAFE_EXTRAPOLATION: IMPOSSIBLE
(OUT_OF_RANGE_DATA).

EXTRAPOLATION: UNSAFE