

Recepción: 18.02.2026

Revisión: 31.03.2026

Publicación: 08.05.2026

<https://orcid.org/0000-0003-0527-5815>

<https://orcid.org/0000-0003-1541-1454>

<https://orcid.org/0000-0001-6660-469X>

<https://orcid.org/0000-0003-3090-7933>

REDES NEURONALES ARTIFICIALES: MODELADO, ENTRENAMIENTO Y VALIDACIÓN

ARTIFICIAL NEURAL NETWORKS: MODELING, TRAINING, AND VALIDATION.

Verónica Santacruz Vázquez

Guadalupe Cuahuizo Huitzil

Claudia Santacruz Vázquez

Santa Toxqui López

Facultad de Ingeniería Química

Av. San Claudio y 18 Sur,

Col. Jardines de San Manuel, CP 72570

Puebla, Pue., México

Benemérita Universidad Autónoma de Puebla (BUAP)

Correos:

veronica.santacruz@correo.buap.mx

guadalupe.cuahuizo@alumno.buap.mx

claudia.santacruz@correo.buap.mx

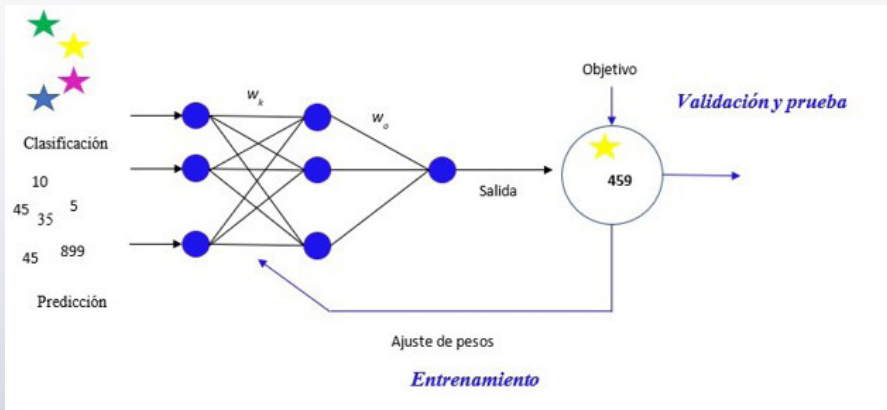
santa.toxqui@correo.buap.mx

Resumen

El documento aborda de manera integral el concepto, funcionamiento y aplicación de las Redes Neuronales Artificiales (RNA), destacándose como una herramienta fundamental de la inteligencia artificial inspirada en el sistema nervioso biológico. Las RNA tienen la capacidad de aprender a partir de datos, reconocer patrones complejos y apoyar la toma de decisiones, lo que ha impulsado su uso en diversos sectores. Se describen sus principales aplicaciones en áreas como salud y biotecnología, industria y automatización, alimentos y biotecnología alimentaria, tecnologías de la información, finanzas y educación, resaltando su utilidad en diagnóstico, predicción, optimización de procesos y análisis de grandes volúmenes de datos.

El documento explica cómo funciona un RNA, detallando el modelo de neurona artificial basado en entradas, pesos sinápticos, suma ponderada y funciones de activación. Asimismo, se presentan los tipos de aprendizaje automático: supervisado, no supervisado y por refuerzo, señalando sus características, ventajas y limitaciones. Como conclusión, se enfatiza que las RNA son sistemas de procesamiento de información altamente versátiles, cuya principal fortaleza es su capacidad de aprendizaje a partir de datos mediante el ajuste de sus conexiones internas.

Palabras clave: Redes neuronales artificiales, entrenamiento, validación, prueba.



Abstract

The document comprehensively addresses the concept, functioning, and application of Artificial Neural Networks (ANNs), highlighting them as a fundamental tool of artificial intelligence inspired by the biological nervous system. ANNs have the ability to learn from data, recognize complex patterns, and support decision-making, which has driven their use across various sectors. Their main applications are described in areas such as health and biotechnology, industry and automation, food and food biotechnology, information technologies, finance, and education, emphasizing their usefulness in diagnosis, prediction, process optimization, and the analysis of large volumes of data. The document explains how an ANN works, detailing the artificial neuron model based on inputs, synaptic weights, weighted summation, and activation functions. Likewise, the types of machine learning—supervised, unsupervised, and reinforcement learning—are presented, outlining their characteristics, advantages, and limitations. In conclusion, it is emphasized that ANNs are highly versatile information-processing systems whose main strength lies in their ability to learn from data through the adjustment of their internal connections.

Keywords: Artificial neural networks, training, validation, testing.

Introducción

El interés por las redes neuronales artificiales (RNA) entre los jóvenes estudiantes ha crecido de manera notable en los últimos años, impulsado principalmente por el avance acelerado de la inteligencia artificial y su presencia en la vida cotidiana. Este interés se manifiesta tanto en el ámbito académico como en el personal y profesional.

En primer lugar, las RNA resultan atractivas para los estudiantes porque permiten comprender y desarrollar tecnologías actuales como los asistentes virtuales, los sistemas de recomendación, el reconocimiento de imágenes y voz, y las aplicaciones de aprendizaje automático. Al observar que estas herramientas están integradas en plataformas digitales, redes sociales y dispositivos inteligentes, los jóvenes perciben su estudio como útil, actual y con impacto real. Asimismo, el aprendizaje de RNA fomenta el desarrollo del pensamiento lógico, la resolución de problemas, el análisis de datos y la creatividad. Para muchos estudiantes, trabajar con modelos de inteligencia artificial representa un reto intelectual estimulante que combina matemáticas, programación y ciencias aplicadas, lo cual incrementa su motivación y participación. Otro factor importante es la alta demanda laboral asociada a áreas como la ciencia de datos, la inteligencia artificial y la ingeniería en sistemas. Los jóvenes reconocen que el dominio de conceptos relacionados con RNA puede mejorar sus oportunidades

profesionales, lo que refuerza su interés por aprender sobre estas tecnologías desde etapas tempranas de su formación.

La incorporación de metodologías activas en el aula, como proyectos prácticos, simulaciones y uso de *software* especializado, ha contribuido a que las RNA sean percibidas como un tema accesible y atractivo, incluso para estudiantes sin una formación técnica avanzada.

¿Qué son las redes neuronales artificiales?

Las *redes neuronales artificiales* (RNA) son una herramienta clave de la *inteligencia artificial*, impulsando la innovación tecnológica y mejorando la eficiencia en distintos sectores. Su capacidad para aprender y adaptarse las convierte en un pilar fundamental del desarrollo tecnológico actual y futuro. Las RNA son un *modelo computacional* inspirado en la estructura y el funcionamiento del *sistema nervioso biológico*, específicamente en la manera en que las neuronas en el cerebro humano procesan información. Estas redes han demostrado ser especialmente eficientes en tareas de *reconocimiento de patrones*, *aprendizaje automático* y toma de decisiones (Krogh, 2008). Las RNA simulan el mecanismo de aprendizaje de las neuronas biológicas del cerebro humano. Mismas que están conectadas entre sí mediante axones y dendritas, y la región de conexión entre estos dos componentes se denomina sinapsis Figura 1a (Aggarwall et al., 2018).

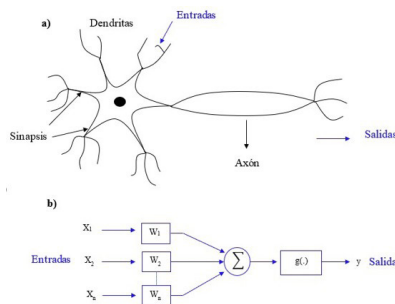


Figura 1. a) Neurona biológica b) Neurona artificial.

El uso de las RNA en la tecnología ha transformado de manera significativa múltiples sectores, ya que permiten a las máquinas aprender de los datos, reconocer patrones complejos y tomar

decisiones de forma similar al cerebro humano. En la Tabla 1, se describen sus principales aplicaciones:

Tabla 1. Aplicaciones de las RNA.

Área	Usos y aplicaciones
Salud y biotecnología (Wu et al., 2018)	Diagnóstico médico mediante análisis de imágenes (rayos X, resonancias, tomografías). Predicción de enfermedades y apoyo a la medicina personalizada. Desarrollo de fármacos y análisis de datos genómicos.
Industria y automatización (Petrova et al., 2021)	Control de procesos industriales y mantenimiento predictivo. Optimización de la producción y reducción de errores. Robots inteligentes capaces de adaptarse a su entorno.
Alimentos y biotecnología alimentaria (Huang et al., 2007)	Control de calidad y detección de defectos. Optimización de formulaciones y procesos productivos. Análisis de propiedades fisicoquímicas y sensoriales.
Tecnología de la información y comunicación (Wu et al., 2018)	Reconocimiento de voz e imagen (asistentes virtuales, reconocimiento facial). Traducción automática y procesamiento del lenguaje natural. Sistemas de recomendación en plataformas digitales
Finanzas y comercio (Pallathadka et al., 2023)	Detección de fraudes y análisis de riesgos. Predicción de mercados y comportamiento del consumidor. Automatización de servicios financieros.
Educación e investigación (Dharmasaroja et al., 2016)	Sistemas de aprendizaje personalizado. Análisis de grandes volúmenes de datos científicos. Simulación y modelado de fenómenos complejos.

¿Cómo funciona una RNA?

Una red neuronal artificial (RNA) funciona como un sistema de capas interconectadas que procesan información y aprende ajustando sus conexiones internas (pesos) a partir de datos. Durante el proceso de aprendizaje en

una *neurona artificial*, las *sinapsis* se modelan mediante un peso, de manera que cada entrada ($X_1, X_2 \dots X_m$) se multiplica por su peso ($Wk_1, Wk_2 \dots Wkm$) antes de ser enviada a la *neurona artificial* (Figura 2).

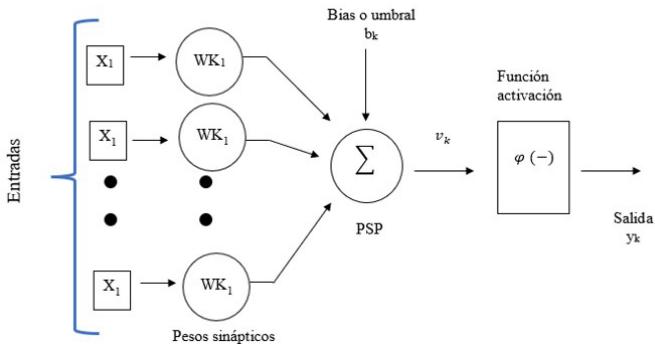


Figura 2. Neurona artificial (Adaptado de Gupta, 2013).

Aquí, las señales ponderadas se suman para proporcionar la entrada, luego se calcula una *función de activación*, que puede ser: *función escalonada*, *tangente hiperbólica*, *sigmoidea*, etc. Y finalmente se devuelve un valor como salida (Hajian & Styles, 2018, Gurney, 2018).

Entre las diferentes propiedades de las *RNA*, se encuentra la propiedad para “*aprender*” a partir de su entorno por medio de un *proceso iterativo* que le permite ajustar los *pesos sinápticos* y los *umbrales* aplicados a la red (Alarcón, 2008). El *aprendizaje automático* es una rama de la *inteligencia artificial* que permite a los sistemas aprender a partir de datos y mejorar su desempeño sin ser programados explícitamente. Dentro de este campo se distinguen distintos enfoques, entre los que destacan el *aprendizaje supervisado* y el *aprendizaje no supervisado* (Amrutha & Ajai, 2018).

El *aprendizaje automático supervisado* utiliza datos previamente etiquetados, donde cada entrada tiene asociada una salida conocida. El modelo aprende comparando sus resultados con los valores reales y ajusta sus parámetros para minimizar el error, por lo que se emplea principalmente en tareas de clasificación y predicción (Di Franco & Santurro, 2021). El *aprendizaje supervisado* se utiliza en una variedad de aplicaciones, como clasificación, regresión y *reconocimiento de patrones*, y es eficaz cuando se dispone de un conjunto de datos etiquetado. Este modelo puede proporcionar información precisa sobre la relación entre las entradas y las salidas esperadas (Haykin, 2009).

Por su parte, el *aprendizaje automático no supervisado* trabaja con datos no etiquetados. En este caso, el modelo identifica patrones, agrupamientos o relaciones entre los datos, sin contar con una referencia previa. En este *algoritmo* se proporciona como entrada a la *RNA*, una gran cantidad de datos y características de cada observación, para que la red genere los datos de salida. Generalmente se emplea el *aprendizaje no supervisado* para clasificar y diferenciar imágenes en función de algunas características

como el color, tamaño, forma, para la clasificación y separación en grupos de los datos (Dike et al., 2018).

Y finalmente el *Aprendizaje por refuerzo*, en esta técnica se considera una variación de las técnicas de aprendizaje supervisado, cuyo proceso de aprendizaje se realiza mediante prueba y error, generando una respuesta satisfactoria o insatisfactoria por parte de la *RNA* para cada entrada. Si es satisfactorio, los pesos y *umbrales sinápticos* se incrementan gradualmente para reforzar (recompensar) esta condición de comportamiento involucrada con el sistema (Da Silva et al., 2017).

Funciones de activación

Durante el *proceso de aprendizaje* se aplican las *funciones de activación* que se definen como *ecuaciones matemáticas* que procesan los datos alimentados a la *neurona*.

Algunas *funciones de activación* comunes utilizadas en *redes neuronales* son la *función Sigmoidea*, *función Tangente Hiperbólica (tanh)*, *Rectified Linear Unit (ReLU)*, *Leaky rectified linear Unit (Leaky ReLU)*, entre otras (Liew et al., 2016). La elección de la *función de activación* depende del tipo de problema que se está abordando y puede requerir *experimentación* para determinar cuál funciona mejor para cada caso específico. Las *funciones de activación* se pueden clasificar en:

Funciones de activación discretas - La salida de la neurona solo puede tomar un conjunto finito de valores. Normalmente, se utilizan dos valores 0 o 1, por lo que se consideran *neuronas binarias*, este tipo de *funciones de activación discretas* facilitan y reducen la complejidad de las implementaciones de *redes neuronales* en *hardware* (Plagianakos et al., 2001).

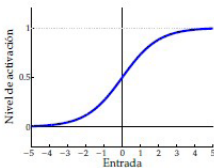
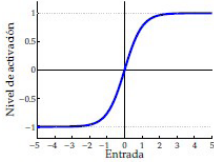
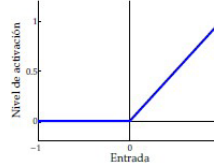
Funciones de activación continuas - La salida de la neurona puede tomar cualquier valor dentro de un intervalo. Generalmente, el rango de ese intervalo está limitado, [0, 1] o bien al *intervalo* [-1, 1] (Berzal, 2018).

La *función de activación* juega un papel importante en el entrenamiento de *redes neuronales*, ya que proporciona la *no linealidad* necesaria del modelo para poder aprender *relaciones complejas* entre los *datos de entrada y salida* (Rasamoelina et al., 2020). Dentro de las *fun-*

ciones de activación continuas se pueden utilizar *funciones de activación lineales* (*función identidad*) como *funciones no lineales*. En la Tabla 2, se presentan algunas de las funciones de activación más empleadas en la construcción de las *RNA*.

Tabla 2. Tipos de funciones de activación (Liew et al., 2016).

	Función Activación	Rango	Grafica
Lineal	$y = \text{sgn}(x)$	$[-\infty, +\infty]$	
Función umbral	$y = u(x)$	$\begin{cases} 1 & \text{si } z \geq 0 \\ 0 & \text{si } z < 0 \end{cases}$	
Función signo	$y = \text{sgn}(x)$	$\begin{cases} 1 & \text{si } z \geq 0 \\ -1 & \text{si } z < 0 \end{cases}$	
Saturación binario	$y = \begin{cases} 1 & \text{si } z > 1 \\ z & \text{si } 0 \leq z \leq 1 \\ 0 & \text{si } z < 0 \end{cases}$	$[0, +1]$	
Saturación simétrica	$y = \begin{cases} 1 & \text{si } z > 1 \\ z & \text{si } -1 \leq z \leq 1 \\ -1 & \text{si } z < -1 \end{cases}$	$[-1, +1]$	

Sigmoidal	$y = \frac{1}{1 + e^{-z}}$	$[0, +1]$	
Tangente hiperbólica	$y = \tanh \tanh(z)$ $y = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$[-1, +1]$	
Función lineal rectificadora	$y = f_{relu}(z)$	$\begin{cases} z & \text{si } z \leq 0 \\ 0 & \text{si } z < 0 \end{cases}$	

Arquitectura de las RNA

La estructura básica de un *RNA* puede expresarse como *Capa de Entrada* - *Capas Ocultas* - *Capa de Salida*. Cada conexión entre *nod*os tiene un peso asociado que se ajusta durante el *entrenamiento* de la red para aprender la representación y la relación entre las entradas y las salidas (Essa et al., 2020, Al-Shathr et al., 2021, Asha et al., 2021). Las capas de una *red multicapa* se dividen en dos categorías: *capas visibles* y *capas ocultas*;

las *capas visibles* son la de entrada y de salida de la *red neuronal*. Todas las capas intermedias diferentes a las *capas de entrada* y de *salida* reciben el nombre de *capas ocultas* (Figura 3). El número de *capas ocultas* y su respectivo número de *neuronas* dependen de la naturaleza y complejidad del problema que está mapeando la red, así como de la cantidad y calidad de los datos disponibles sobre el problema (Da Silva et al., 2017).

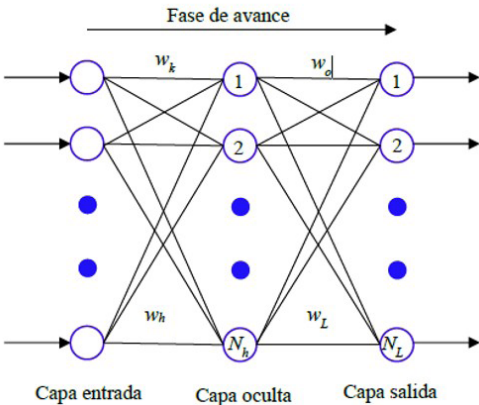


Figura 3. Red neuronal feed forward (Hemeida et al., 2020).

Clasificación por su conexión

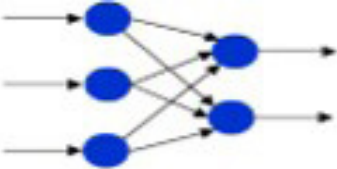
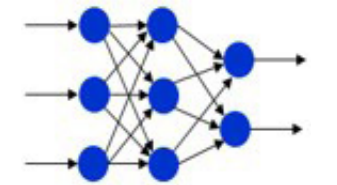
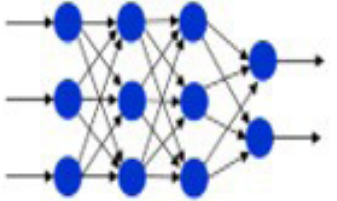
La configuración de una *RNA* se refiere a su estructura y diseño, incluyendo la disposición de las capas, el número de *neuronas* en cada capa, la conexión entre ellas y otros aspectos específicos. Las *RNA* se pueden clasificar según su conexión en; *función de base radial*, *redes recurrentes*, *feed forward*, y otras (Hemeida et al., 2020).

Redes neuronales de función base radial. - Se consideran una forma especial de *redes neuronales multicapa* que contienen solo una *capa oculta* con *funciones de activación* basadas en *Gauss*, se han aplicado en la clasificación de fallas de procesos, control no lineal y pronósticos de series de tiempo (Aljarah et al., 2018).

Redes recurrentes. - Las conexiones entre las *neuronas* forman ciclos, permitiendo que la información se *retroalimente* en la red (Salinas et al., 2020).

Finalmente, las *redes neuronales feedforward* se componen: una *capa de entrada*, *capas ocultas* y una *capa de salida*. Las distintas capas se conectan entre sí de tal forma que la salida de la capa *i* se utiliza como entrada en la capa *i + 1*. Las salidas de las *neuronas* en la *capa de salida* representan la salida de la red (Hemeida et al., 2020). En una *red neuronal* de tipo *feed-forward*, podemos encontrar varias arquitecturas, dependiendo del número de *capas ocultas* que se utilicen; ver Tabla 3

Tabla 3. Configuraciones de redes neuronales (Da Silva et al., 2017).

Tipo	Características	Características
Redes simples o unicapa	Las neuronas de la capa de entrada, se limitan a recibir las señales de entrada provenientes del exterior, redistribuyen esas entradas a las neuronas de la capa de salida.	
Redes multicapa, con una capa oculta	Si se añaden nuevas capas intermedias, estas capas ya no serán visibles desde el exterior, lo que obligará a utilizar algoritmos como backpropagation para ajustar sus parámetros internos.	
Redes profundas, con varias capas ocultas	Las capas ocultas le permiten a la red neuronal construir un modelo interno de la forma en que los patrones de datos de entrada están relacionados con las salidas deseadas.	

El desempeño de la *RNA* se ve afectado por la *función de transferencia* y la configuración (Hemeida et al., 2020). Sin embargo, no es posible deter-

minar teóricamente cuántas capas o *neuronas ocultas* se necesitan para cada problema (Benardos & Vosniakos, 2007; Rezakazemi et al., 2011).

Etapas para construir un modelo RNA

Para la construcción de una RNA se consideran los procesos de *entrenamiento*, validación y prueba. El proceso de *entrenamiento* consiste en elegir un *modelo* que permita aprender patrones y relaciones a partir de los datos de entrada, donde los pesos del modelo se ajustan iterativamente mediante la *retropropagación* y el *descenso del gradiente* para minimizar la *función de costo* en este conjunto (Haykin, 2009).

El proceso de validación se utiliza para decidir cuándo detener el *entrenamiento* y con ello evitar el *sobreajuste* y finalmente el proceso de prueba permite evaluar la capacidad predictiva y proporcionar una estimación objetiva de cómo el *modelo* generaliza a datos no vistos y no utilizados en el proceso de *entrenamiento* (Dayhoff et al., 2001).

Inicialmente los datos de entrada se dividen en tres conjuntos distintos para realizar los procesos de *entrenamiento*, validación y prueba. Es necesario asegurarse que el valor máximo y el valor mínimo de cada variable de entrada se

encuentren presente en el conjunto de *entrenamiento*, de lo contrario, si dichos valores se asignan en el conjunto de prueba, la *red de perceptrón multicapa (MLP)* podría generar errores significativos (Zhang et al., 1999). Khatti et al. (2019) propusieron una división aleatoria de los datos experimentales en una proporción de 70, 15 y 15% para los tres conjuntos, mientras que Kalantary et al. (2019) propusieron una división aleatoria de 60, 20 y 20% para el conjunto de *entrenamiento*, validación y prueba respectivamente. Por otro lado, Solis-Rios et al. (2023) propone una división 75, 15 y 10 para los conjuntos de *entrenamiento*, validación y prueba respectivamente.

Profundizando sobre el proceso de *entrenamiento* de una red, este consta de varias etapas, la primera es el ajuste de los *pesos sinápticos* de la *capa de salida*, seguido del ajuste de los *pesos sinápticos* de las *capas intermedias* y finalmente el ajuste de los *pesos sinápticos* de la primera *capa oculta*, los cuales se describen en seguida.

Parte I: ajuste de los pesos sinápticos de la capa de salida.

La *función de error cuadrático medio* de la ecuación 1. se emplea para medir el desempeño asociado con los resultados producidos por las *neuronas* de salida con respecto a la muestra dada, es decir:

$$E(k) = \frac{1}{2} \sum_{i=1}^{n_3} \left(d_j(k) - Y_j^{(3)}(k) \right)^2 \quad \text{ec. 1.}$$

Donde $Y_j^{(3)}(k)$ es el valor producido por la *j-ésima neurona* de salida de la red para la *k-ésima muestra* de *entrenamiento*, mientras que $d_j(k)$ es el valor deseado.

El objetivo del proceso de *entrenamiento* de la *capa neuronal de salida* consiste en ajustar la matriz de peso $W_{ji}^{(3)}$ para minimizar el error entre las salidas producidas por la red con respecto a las salidas deseadas.

$$\nabla E^{(3)} = \frac{\partial E}{\partial W_{ji}^{(3)}} = \frac{\partial E}{\partial Y_j^{(3)}} \cdot \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} \cdot \frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} \quad \text{ec. 2}$$

$$\frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} = Y_i^{(2)} \quad \text{ec. 3} \quad \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} = g'(I_j^{(3)}) \quad \text{ec. 4} \quad \frac{\partial E}{\partial Y_j^{(3)}} = -(d_j - Y_j^{(3)}) \quad \text{ec. 5}$$

Dónde $g'(\cdot)$ denota la *derivada* de primer orden de la *función de activación* empleada. Reemplazando (3), (4) y (5) en (2), obtenemos ecuación 6:

$$\frac{\partial E}{\partial W_{ji}^{(3)}} = -(d_j - Y_j^{(3)}) \cdot g'(I_j^{(3)}) \cdot Y_i^{(2)} \quad \text{ec. 6}$$

El ajuste de la *matriz de pesos* $W_{ji}^{(3)}$ debe realizarse en la dirección opuesta al gradiente para minimizar el error, como lo muestra la ecuación 7:

$$\Delta W_{ji}^{(3)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(3)}} \Leftrightarrow \Delta W_{ji}^{(3)} = \eta (d_j - Y_j^{(3)}) \cdot g'(I_j^{(3)}) \cdot Y_i^{(2)} \quad \text{ec. 7}$$

Donde η es la *tasa de aprendizaje* del algoritmo de *retropropagación*.

Parte II: Ajuste de los pesos sinápticos de las capas intermedias.

A diferencia de las *neuronas* que pertenecen a la *capa de salida* de la *red MLP*, las *neuronas* de las *capas intermedias* no tienen acceso a los valores deseados de las salidas. El ajuste de sus *pesos sinápticos* se realiza mediante estimaciones de los errores de salida causados por aquellas *neuronas* de la *capa posterior*, que ya han sido ajustadas.

$$\nabla E^{(2)} = \frac{\partial E}{\partial W_{ji}^{(2)}} = \frac{\partial E}{\partial Y_j^{(2)}} \cdot \frac{\partial Y_j^{(2)}}{\partial I_j^{(2)}} \cdot \frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} \quad \text{ec. 8}$$

De las definiciones anteriores se obtienen las ecuaciones 9, 10 y 11:

$$\frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} = Y_i^{(1)} \quad \text{ec. 9} \quad \frac{\partial Y_j^{(2)}}{\partial I_j^{(2)}} = g'(I_j^{(2)}) \quad \text{ec. 10} \quad \frac{\partial E}{\partial Y_j^{(2)}} = - \sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \quad \text{ec. 11}$$

Sustituyendo las ecuaciones 9, 10 y 11 en 8, obtenemos la ecuación 12:

$$\frac{\partial E}{\partial W_{ji}^{(2)}} = - \left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \right) \cdot g'(I_j^{(2)}) \cdot Y_i^{(1)} \quad \text{ec. 12}$$

Dónde $\delta_j^{(2)}$ se define como el *gradiente local* relacionado con la *j-ésima neurona* en la *capa de salida*, y está dado por la ecuación 13:

$$\delta_j^{(2)} = (d_j - Y_j^{(2)}) \cdot g'(I_j^{(2)}) \quad \text{ec. 13}$$

Por tanto, el ajuste de la *matriz de pesos* $W_{ji}^{(2)}$ debe realizarse en la dirección opuesta al *gradiente* para minimizar el error como lo muestra la ecuación 14:

$$\Delta W_{ji}^{(2)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(2)}} \Leftrightarrow \Delta W_{ji}^{(2)} = \eta \cdot \left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \right) \cdot g'(I_j^{(2)}) \cdot Y_i^{(1)} \quad \text{ec. 14}$$

Ajustar los pesos sinápticos de la primera capa oculta

$$\nabla E^{(1)} = \frac{\partial E}{\partial W_{ji}^{(1)}} = \frac{\partial E}{\partial Y_j^{(1)}} \cdot \frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} \cdot \frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}} \quad \text{ec. 15}$$

De las definiciones anteriores se obtiene ecuación 1.16, 1.17 y 1.18:

$$\frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}} = x_i \quad \text{ec. 16} \quad \frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} = g'(I_j^{(1)}) \quad \text{ec. 17} \quad \frac{\partial E}{\partial Y_j^{(1)}} = - \sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \quad \text{ec. 18}$$

Sustituyendo ecuaciones 16, 17 y 18 en 15, obtenemos la ecuación 19:

$$\frac{\partial E}{\partial W_{ji}^{(1)}} = - \left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \right) \cdot g'(I_j^{(1)}) \cdot x_i \quad \text{ec. 19}$$

Dónde $\delta_j^{(2)}$ se define como el *gradiente particular* con respecto a la *j-ésima neurona* de la *segunda capa intermedia*, ecuación 20:

$$\delta_j^{(2)} = - \left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \right) \cdot g'(I_j^{(2)}) \quad \text{ec. 20}$$

Por tanto, el ajuste de la *matriz de pesos* $W_{ji}^{(1)}$ debe realizarse en la dirección opuesta al *gradiente* para minimizar el error, empleando la ecuación 21:

$$\Delta W_{ji}^{(1)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(1)}} \Leftrightarrow \Delta W_{ji}^{(1)} = \eta \cdot \left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \right) \cdot g'(I_j^{(1)}) \cdot x_i \quad \text{ec. 21}$$

Proceso de validación y prueba de las RNA

El proceso de validación se utiliza para decidir cuándo detener el *entrenamiento* y con ello evitar el *sobreajuste* que ocurre cuando el modelo se ha *sobreentrenado*. En la Figura 3 se describe dicho comportamiento, se ilustra el *error cuadrático medio* (MSE) de un conjunto de *entrenamiento* y prueba, en función del número de *iteraciones*. Se puede observar que el MSE en el conjunto de *entrenamiento* y en el conjunto de prueba disminuye a medida que se incrementa hasta cierto número de *iteraciones sucesivas* (n^1), posteriormente el MSE del conjunto de prueba comienza a aumentar mientras que el MSE del conjunto de *entrenamiento* continúa disminuyendo, esta

última región presenta un *sobreentrenamiento*. Para evitar dicho fenómeno, se suspende el *entrenamiento* cuando el MSE en el conjunto

de prueba es mínimo, lo que ocurre en n^1 *iteraciones* como se muestra en la Figura 4 (Dayhoff et al., 2001).

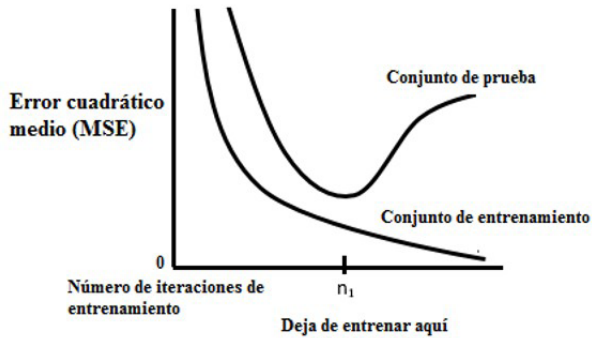


Figura 4. Comportamiento del sobreentrenamiento (Dayhoff et al., 2001).

La división aleatoria de una muestra en conjuntos de *entrenamiento* y prueba puede introducir sesgos en la selección y evaluación del *modelo*, en el sentido de que las características de la prueba pueden ser muy diferentes a las del *entrenamiento*, especialmente cuando se trata de muestras de tamaño pequeño. Por esta razón, es necesario emplear la *validación cruzada* para describir con precisión el *rendimiento* predictivo de las RNA (Zhang et al., 1999, Karimi et al., 2015)

La *validación cruzada* es una técnica basada en la división de datos para realizar evaluaciones predictivas de *modelos estadísticos*; consiste en dividir los datos disponibles en un conjunto de *entrenamiento* y uno de prueba, donde el modelo se ajusta a los datos de *entrenamiento*

y posteriormente se evalúa en función de sus predicciones sobre los datos de prueba. Al repetir este proceso para muchas divisiones diferentes de los datos, se estima el *rendimiento* predictivo promedio de uno o más *modelos* (Yates et al., 2023).

El aprendizaje de la red está relacionado con su *topología* o arquitectura, es decir, del número de capas, del número de *neuronas* en cada capa y del esquema de conexiones. El desempeño de la RNA se ve afectado por la *función de transferencia* y la arquitectura (Hemeida et al., 2020). Sin embargo, no es posible determinar teóricamente cuántas capas o *neuronas* ocultas se necesitan para cada problema (Benardos & Vosniakos, 2007; Rezakazemi et al., 2011).

Conclusión

Como conclusión se puede resumir que las RNA son sistemas de procesamiento de información, inspirados en el *modelo biológico* del sistema nervioso, y que tienen una característica fundamental referente a su habilidad para aprender a partir de datos externos o datos de entrada, mediante el ajuste de sus *interconexiones*.

Conflicto de intereses

Los autores de este manuscrito declaran no tener ningún conflicto de interés.

Declaración de privacidad

Los datos de este artículo, así como los detalles técnicos para la realización del escrito, se pueden compartir a solicitud directa del autor de correspondencia.

Los datos personales facilitados por los autores a RD-ICUAP se usarán exclusivamente para los fines declarados por la misma, no estando disponibles para ningún otro propósito ni proporcionados a terceros.

Descargo de Responsabilidad (Inteligencia Artificial) Los autores declaran por la presente que NO se han utilizado tecnologías de IA generativa, tales como modelos de lenguaje grandes (ChatGPT, COPILOT, etc.) y generadores de texto a imagen, durante la redacción o edición de este manuscrito.

Agradecimientos

Guadalupe Cuahuizo-Huitzil agradece a la Secretaría de Ciencia, Humanidades Tecnología e Innovación (SECITHI) por la beca de estancia posdoctoral.

Referencias

- Krogh, A. (2008). What are artificial neural networks?. *Nature biotechnology*, 26(2), 195-197.
- Aggarwal, C. C. (2018). *Neural networks and deep learning* 10, No (978). Cham: springer.
- Wu, Y. C., & Feng, J. W. (2018). Development and application of artificial neural network. *Wireless Personal Communications*, 102(2), 1645-1656.
- Petrova, A. K., & Sinitca, A. M. (2021). Application of Neural Networks in the Tasks of Automation at Industrial Enterprises. In 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus) (pp. 577-581). IEEE.
- Huang, Y., Kangas, L. J., & Rasco, B. A. (2007). Applications of artificial neural networks (ANNs) in food science. *Critical reviews in food science and nutrition*, 47(2), 113-126.
- Pallathadka, H., Ramirez-Asis, E. H., Loli-Poma, T. P., Kaliyaperumal, K., Ventayen, R. J. M., & Naved, M. (2023). Applications of artificial intelligence in business management, e-commerce and finance. *Materials Today: Proceedings*, 80, 2610-2613.
- Dharmasaroja, P., & Kingkaew, N. (2016). Application of artificial neural networks for prediction of learning performances. In 2016 12th international conference on natural computation, fuzzy systems and knowledge discovery (icnc-fskd) pp. 745-751. IEEE.
- Gupta, N. (2013). Artificial neural network. *Network and Complex Systems*, 3(1), 24-28.
- Hajian & Styles. (2018). *Application of Soft Computing and Intelligent Methods in Geophysics*, Springer International Publishing AG, part of Springer Nature.
- Gurney, K. (2018). *An introduction to neural networks*. CRC press.
- Alarcón Celis Jaime. (2008). *Arquitectura inteligente basada en lógica reconfigurable aplicada a redes neuronales artificiales para la clasificación de información*. Tesis doctorado, Universidad Politécnica de Madrid.
- Amrutha, J., & Ajai, A. R. (2018). Performance analysis of backpropagation algorithm of artificial neural networks in verilog. In 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 1547-1550.
- Di Franco, G., & Santurro, M. (2021). Machine learning, artificial neural networks and social research. *Quality & quantity*, 55(3), 1007-1025.
- Haykin, S. (2009). *Neural networks and learning machines*, 3/E. Pearson Education India.
- Dike, H. U., Zhou, Y., Deveerasetty, K. K., & Wu, Q. (2018). Unsupervised learning based on artificial neural network: A review. In 2018 IEEE International Conference on Cyborg and Bionic Systems (CBS), 322-327. IEEE.
- Da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., dos Reis Alves, S. F., da Silva, I. N., ... & dos Reis Alves, S. F. (2017). Artificial neural network architectures and training processes, 21-28. Springer International Publishing.
- Liew, S. S., Khalil-Hani, M., & Bakhteri, R. (2016). Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing*, 216, 718-734.
- Berzal, F. (2018). *Redes Neuronales & Deep Learning*, Granada.
- Plagianakos, V. P., Magoulas, G. D., Nouis, N. K., & Vrahatis, M. N. (2001). Training multilayer networks with discrete activation functions. *IJCNN'01. International Joint Conference on Neural Networks. Proceedings*, 4, 2805-2810). IEEE.
- Rasamoelina, A. D., Adjailia, F., & Sincák, P. (2020). A review of activation function for artificial neural network. In 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 281-286. IEEE.
- Essa, F. A., Abd Elaziz, M., & Elsheikh, A. H. (2020). An enhanced productivity prediction model of active solar still using artificial neural network and Harris Hawks optimizer. *Applied Thermal Engineering*, 170, 115020.
- Al-Shathir, A., Shakor, Z. M., Majdi, H. S., AbdulRazak, A. A., & Albayati, T. M. (2021). Comparison between artificial neural network and rigorous mathematical model in simulation of industrial heavy naphtha reforming process. *Catalysts*, 11(9), 1034.
- Asha, R. B., & KR, S. K. (2021). Credit card fraud detection using artificial neural network. *Global Transitions Proceedings*, 2(1), 35-41.

- Da Silva, I. N., Hernane Spatti, D., Andrade Flauzino, R., Liboni, L. H. B., dos Reis Alves, S. F., da Silva, I. N., ... & dos Reis Alves, S. F. (2017). Artificial neural network architectures and training processes, 21-28. Springer International Publishing.
- Hemeida, A. M., Hassan, S. A., Mohamed, A. A. A., Alkhalaf, S., Mahmoud, M. M., Senju, T., & El-Din, A. B. (2020). Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research. *Ain Shams Engineering Journal*, 11(3), 659-675.
- Aljarah, I., Faris, H., Mirjalili, S., & Al-Madi, N. (2018). Training radial basis function networks using biogeography-based optimizer. *Neural Computing and Applications*, 29, 529-553.
- Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3), 1181-1191.
- Benardos, P. G., & Vosniakos, G. C. (2007). Optimizing feedforward artificial neural network architecture. *Engineering applications of artificial intelligence*, 20(3), 365-382.
- Rezakazemi, M., Razavi, S., Mohammadi, T., & Nazari, A. G. (2011). Simulation and determination of optimum conditions of pervaporative dehydration of isopropanol process using synthesized PVA-APTEOS/TEOS nanocomposite membranes by means of expert systems. *Journal of Membrane Science*, 379(1-2), 224-232.
- Dayhoff, J. E., & DeLeo, J. M. (2001). Artificial neural networks: opening the black box. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 91(S8), 1615-1635.
- Khatti, T., Naderi-Manesh, H., & Kalantar, S. M. (2019). Application of ANN and RSM techniques for modeling electrospinning process of polycaprolactone. *Neural Computing and Applications*, 31, 239-248.
- Kalantary, S., Jahani, A., Pourbabaki, R., & Beigzadeh, Z. (2019). Application of ANN modeling techniques in the prediction of the diameter of PCL/gelatin nanofibers in environmental and medical studies. *RSC advances*, 9(43), 24858-24874.
- Solis-Rios, D., Villarreal-Gómez, L. J., Goyes, C. E., Fonthal Rico, F., Cornejo-Bravo, J. M., Fong-Mata, M. B., ... & Mejía-Medina, D. A. (2023). A Neural Network Approach to Reducing the Costs of Parameter-Setting in the Production of Polyethylene Oxide Nanofibers. *Micromachines*, 14(7), 1410.
- Zhang, G., Hu, M. Y., Patuwo, B. E., & Indro, D. C. (1999). Artificial neural networks in bankruptcy prediction: General framework and cross-validation analysis. *European journal of operational research*, 116(1), 16-32.
- Dayhoff, J. E., & DeLeo, J. M. (2001). Artificial neural networks: opening the black box. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 91(S8), 1615-1635.
- Karimi, M. A., Pourhakkak, P., Adabi, M., Firoozi, S., Adabi, M., & Naghibzadeh, M. (2015). Using an artificial neural network for the evaluation of the parameters controlling PVA/chitosan electrospun nanofibers diameter. *e-Polymers*, 15(2), 127-138.
- Yates, L. A., Aandahl, Z., Richards, S. A., & Brook, B. W. (2023). Cross validation for model selection: a review with examples from ecology. *Ecological Monographs*, 93(1), e1557.