



Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riit.com.mx

Aplicaciones didácticas de redes neuronales mediante Excel® para Ingeniería Química y Bioquímica

Educational applications of neural networks using Excel® for Chemical Engineering and Biochemistry

Calderón-Ramírez, M.^a, Gaitán-Vélez, V.^b, Quemada-Villagómez, M.L.^b, Martínez-González, G.M.^c, Jiménez-Islas, H.^{b*}

^a Departamento de Diseño y Desarrollo de Equipo, Tecnológico Nacional de México, CP:38020, CRODE Celaya, Gto.

^b Departamento de Ingeniería Bioquímica y Ambiental, Tecnológico Nacional de México, CP: 38010, Celaya, Gto.

^c Departamento de Ingeniería Química, Tecnológico Nacional de México, CP: 38010, Celaya, Gto.

mario.calderon@itcelaya.edu.mx; bvgv25@gmail.com; miriam.quemada@itcelaya.edu.mx;

gloriam@iqcelaya.itc.mx; hugo.jimenez@itcelaya.edu.mx*

Área de aplicación: Ingeniería Química y Bioquímica.

Innovación tecnológica: Uso de hoja electrónica para entrenamiento de redes neuronales.

Recibido: 12 junio 2024

Aceptado: 10 octubre 2025

Abstract

Currently, applications of artificial neural networks can be found in various fields due to the rapid development of information technology, ushering in a new era of System Identification (SI) that encompasses every aspect of our society. This paper aims to give students an overview of general methodologies to help them understand System Identification and its potential applications in chemical and biochemical engineering. So, it explains the basics of Artificial Neural Network algorithms using spreadsheet software (Microsoft Excel®). The handling of experimental results, the construction models with algebraic equations, use of mathematical minimization functions and models, and the use of the optimization tool (SOLVER™) were employed to convey the essential concepts. This procedure was applied in a didactic manner to various case studies using static-feedforward neural networks, dynamic-feedback neural networks, continuous differential dynamic neural networks (black box model), and complementary hybrid neural networks (grey box models).

These models require information through input data and generate an output that describes complex and nonlinear behaviors. This work introduces students to the use of basic artificial intelligence models applied to chemical and biochemical engineering.

Keywords: Applied Artificial Intelligence, Neural Network Models, Optimization in Engineering, Algebraic Models.

Resumen

Actualmente se pueden encontrar aplicaciones de redes neuronales artificiales en diversos campos debido al rápido desarrollo de la tecnología de la información, abriendo paso a una nueva era de la Identificación de Sistemas (IS) que abarca cada aspecto de nuestra sociedad. El objetivo del presente trabajo es proporcionar a la comunidad estudiantil una visión de las metodologías generales con el fin de proporcionar a los estudiantes un panorama de la Identificación de Sistemas y sus aplicaciones potenciales en ingeniería química y bioquímica. Por lo tanto, se expone el fundamento de los algoritmos de Redes Neuronales Artificiales utilizando un software de hoja de cálculo electrónica (Microsoft Excel®). El manejo de resultados experimentales, construcción de modelos con ecuaciones algebraicas, uso de funciones y modelos matemáticos de minimización, así como el manejo de la herramienta de optimización (SOLVER™) se utilizaron para expresar los conceptos esenciales. Este procedimiento se aplicó, de forma didáctica, a diferentes casos de estudio utilizando redes neuronales *feedforward*, redes neuronales *Dynamic-feedback*, redes neuronales dinámicas continuas diferenciales (modelos de caja negra) y redes neuronales híbridas complementarias (Cajas Gris o modelos híbridos). Estos modelos requieren información a través de datos de entrada y generan una salida que predice comportamientos complejos y no lineales. Este trabajo proporciona una introducción al estudiante para que pueda usar modelos básicos de inteligencia artificial aplicados a la ingeniería química y bioquímica.

Palabras clave: Inteligencia Artificial Aplicada, Modelos de Redes Neuronales, Optimización en Ingeniería, Modelos algebraicos.

I. Introducción

Desde sus primeros semestres los estudiantes de licenciatura realizan experimentos de laboratorio y obtienen numerosos datos que deben interpretar. Sin embargo, a menudo ellos no reconocen que estos datos pueden representarse mediante modelos matemáticos. La Identificación de Sistemas (IS) [1] es la interacción inicial con los fundamentos de la Inteligencia Artificial (IA), que se logra a través de modelos matemáticos adaptativos y que es esencial en procesos de ingeniería tanto académicos como industriales. Adaptar a los estudiantes a

esta era digital es crucial, lo que se logra mediante prácticas de laboratorio y el uso de herramientas computacionales como las Redes Neuronales Artificiales (RNA) [1].

El uso de RNA permite representar modelos no lineales de manera flexible, robusta al ruido experimental y con capacidad para describir fenómenos con fuerte carga aleatoria [2]. Estas redes proveen modelos empíricos útiles en aplicaciones como detección de errores, procesamiento de señales, modelado y control de procesos. Aunque requieren ser validadas y someterse a

un sobreajuste (*overfit*), las ventajas significativas son su menor tiempo de respuesta y su capacidad de mejora continua [3].

En ingeniería química y bioquímica, el uso de inteligencia artificial y aprendizaje automático ha crecido sustancialmente, con las redes neuronales desempeñando un papel importante [4]. Permiten analizar grandes cantidades de datos, aplicándose en tareas como modelado, clasificación y predicción, describiendo fenómenos complejos a partir de datos experimentales [5]. Algunos ejemplos incluyen el flujo turbulento en tuberías, curvas de adsorción y procesos de cristalización, ya que son excelentes ejemplos de modelos de alta complejidad no lineal [6].

Las RNA son útiles en sistemas no lineales multiparamétricos, donde los modelos fenomenológicos son complejos de desarrollar y a menudo imprecisos [7]. Son esenciales en el control de sistemas químicos y en bioprocesos, donde se necesitan evaluación en tiempo real y adaptabilidad a cambios ambientales [8]. Además, permiten el análisis estadístico multivariable y la construcción de superficies de respuesta complejas, superando las limitaciones de modelos polinomiales en sistemas altamente no lineales [9].

Las redes neuronales artificiales (RNA) representan una de las herramientas más poderosas en el ámbito del modelado y la optimización de procesos complejos, especialmente en disciplinas como la ingeniería química, donde los sistemas suelen ser altamente no lineales, multivariables y difíciles de describir mediante modelos deterministas. Entre las arquitecturas más relevantes se encuentran las redes neuronales feedforward, las redes neuronales dinámicas y las redes neuronales diferenciales, cada una con características particulares que las hacen útiles en distintos contextos.

Aunque las redes neuronales funcionan como una caja negra, se han desarrollado metodologías para identificar la predominancia de factores de peso, facilitando la interpretación de variables en la predicción de procesos [10]. Las redes feedforward, también conocidas como perceptrones multicapa, son las más básicas y utilizadas. Se caracterizan por un flujo de información unidireccional, desde la capa de entrada hasta la capa de salida, pasando por capas ocultas que transforman los datos mediante funciones de activación. Estas redes carecen de memoria o retroalimentación, lo que las hace más apropiadas para problemas estáticos. En ingeniería química se aplican ampliamente en la predicción de propiedades fisicoquímicas de compuestos y mezclas, en la estimación de parámetros de calidad de productos en procesos de destilación, polimerización o fermentación, y en la optimización de condiciones de operación, reduciendo la necesidad de experimentos costosos o simulaciones complejas.

Por su parte, las redes neuronales dinámicas incorporan retroalimentación y permiten capturar la evolución temporal de los procesos, lo que resulta fundamental en sistemas que no son estáticos. Ejemplos de estas arquitecturas son las redes recurrentes (RNN) y sus variantes modernas como LSTM o GRU, que han demostrado gran eficacia para trabajar con series temporales. En ingeniería química se utilizan para el control predictivo de procesos donde la temperatura, la presión o la concentración cambian con el tiempo, en la predicción de comportamientos futuros de variables de operación a partir de datos históricos, y en la simulación de fenómenos transitorios como reacciones exotérmicas, donde predecir el comportamiento dinámico es crucial para la seguridad y eficiencia del sistema. Muchos procesos químicos y bioquímicos son dinámicos, requiriendo redes neuronales recursivas [11].

Las redes neuronales diferenciales, conocidas como *Neural ODEs*, constituyen un enfoque emergente que combina la flexibilidad del aprendizaje automático con la solidez del modelado matemático tradicional. En este caso, la evolución de un sistema se plantea como una ecuación diferencial ordinaria parametrizada por una red neuronal, lo que permite integrar el conocimiento basado en primeros principios con la capacidad predictiva de las RNA. Este enfoque ha demostrado ser especialmente valioso en la ingeniería química para el modelado híbrido de procesos complejos como la fermentación o la cristalización, donde los mecanismos físicos no siempre están completamente descritos; también para la identificación de cinéticas de reacción sin necesidad de suponer modelos mecanísticos *a priori*, y en la simulación de procesos continuos como el flujo en reactores tubulares o la difusión en sistemas de separación. Estas redes permiten describir fenómenos a través de ecuaciones diferenciales, siendo especialmente útiles en sistemas complejos [12].

En conjunto, estas tres arquitecturas muestran la versatilidad de las redes neuronales como herramientas aplicadas a la ingeniería química. Mientras que las feedforward resultan adecuadas para el modelado estático y la predicción de propiedades, las dinámicas ofrecen soluciones efectivas para el análisis de procesos transitorios, y las diferenciales establecen un puente entre la física tradicional y la inteligencia artificial. Gracias a ello, hoy en día las RNA contribuyen a diseñar procesos más seguros, eficientes y sostenibles, lo que las convierte en un recurso estratégico en el desarrollo de la industria química moderna y por lo tanto es esencial su comprensión y aprendizaje.

El uso de hojas de cálculo, como Excel®, para construir y entrenar RNA es una alternativa didáctica y asequible para introducir a los estudiantes a la IA. Esta

herramienta facilita la visualización del proceso de modelado neuronal, ajuste de datos y generación de respuestas, complementando la formación académica y desmitificando la aparente complejidad del uso de estos métodos [13-14]. La implementación de esta estrategia con Excel® ha demostrado ser viable, proporcionando resultados que respaldan su uso para la construcción de redes neuronales [15-16].

Excel® es una herramienta ampliamente utilizada en la enseñanza de la ingeniería, lo que permite a los estudiantes familiarizarse con el proceso de planteamiento de modelos neuronales, ajustando datos y generando respuestas de manera clara y didáctica [17]. Por lo tanto, el objetivo de este trabajo es emplear hojas de cálculo de Excel® como estrategia didáctica en la enseñanza de RNA en el aula, que se enfocará en el manejo de datos experimentales, la construcción de modelos con ecuaciones algebraicas, el uso de funciones y métodos matemáticos de minimización, y la utilización de herramientas de optimización (SOLVER™), para mostrar una visión didáctica del uso de esta herramienta en ingeniería química y bioquímica [18], aunque hay otras alternativas para hacer estimación de parámetros como MATLAB, MAPLE o códigos computacionales en lenguajes de alto nivel disponibles en Internet. Es importante mencionar que la aplicación de RNA se ha extendido en ciencia e ingeniería, por lo que se ha creado software especializado que esencialmente es una caja negra, a la cual se alimentan datos esperando una respuesta sin conocer los fundamentos matemáticos y metodológicos empleados [19]. En contraste, las hojas de Excel® permiten comprender estos fundamentos, facilitando su uso con conocimiento elemental que permite conocer los conceptos básicos del algoritmo que posteriormente podrá ser trasladado a plataformas multilenguaje (R *Statistical*

Computing, MATLAB + Simulink, Python, Symbolic Math Computing, entre otros).

En este trabajo se propone utilizar el fundamento de los algoritmos de Redes Neuronales Artificiales mediante el uso de una hoja de cálculo para manipular datos numéricos (Microsoft Excel® y la herramienta de optimización SOLVER™). El método se aplicará a diferentes casos de estudio que involucren redes neuronales estacionarias, redes neuronales recursivas, redes neuronales dinámicas continuas diferenciales y redes neuronales híbridas complementarias (Caja Gris, modelos híbridos), lo que presentará una visión general y didáctica del uso de esta herramienta para las ingenierías de química y bioquímica principalmente, aunque los conceptos teóricos y formas de resolución de los casos de estudio tratados son extrapolables a cualquier área de la ciencia.

II. Materiales y métodos

Las redes neuronales fueron establecidas para replicar el comportamiento del cerebro, una serie de neuronas (nodos) interrelacionados (parámetros de peso) que se interconectan entre sí de forma similar a la fuerza sináptica y que tienen una función de transferencia que envía una señal de salida (impulso nervioso). La construcción de una red neuronal comienza con la definición de la arquitectura, que incluye la cantidad de capas y el número de neuronas en cada capa. Las redes neuronales suelen estar compuestas por una capa de entrada, una o más capas ocultas y una capa de salida, como se describe en la Figura 1. Cada neurona en una capa recibe una señal de las neuronas de la capa anterior, la procesa mediante una función de activación y transmite la salida hacia las neuronas de la siguiente capa. La elección de la función de activación o de transferencia depende de su aplicación [5, 9].

El siguiente paso en la construcción de una red neuronal es la inicialización de los pesos y sesgos, que son los parámetros ajustables del modelo. Estos parámetros se inicializan con pequeños valores aleatorios. Posteriormente, la red se entrena utilizando un conjunto de datos experimentales, donde se ajustan los pesos y sesgos para minimizar la diferencia entre las predicciones de la red y los valores reales. Este proceso de ajuste se realiza mediante un algoritmo de optimización, usualmente el gradiente de máximo descenso, aunque puede utilizarse el método de Levenberg-Marquardt, método de Luus-Jaakola, entre diversos algoritmos de optimización, mediante una función de minimización que mide el error de las predicciones. Durante el entrenamiento, el algoritmo de retropropagación se utiliza para calcular los gradientes de la función de minimización con respecto a los pesos y sesgos, permitiendo que los parámetros se actualicen de manera eficiente, hasta alcanzar la tolerancia numérica establecida *a priori* para definir que se ha alcanzado convergencia. Esta estructura es la forma genérica de una red; a continuación, se mostrarán las formas matemáticas de las redes a utilizar en este trabajo.

Red estática (*Feedforward*)

La red estática es un sistema simple que consiste en: una capa de entrada, la cual incluye los datos de entrada que alimentan la red; una capa oculta, que permite las relaciones complejas entre la entrada y la salida; y la capa de salida, con la que obtiene los resultados y la salida de la red (Figura 1). Esta configuración, aun siendo simple, permite lograr resultados muy significativos [20]. Este tipo de red se conoce como red de retroalimentación (*Feedforward*).

El estado de cada neurona v en la primera capa oculta está definido como la suma de los productos de cada entrada x_j con su correspondiente factor w_j y la b que

corresponde al sesgo del proceso de aprendizaje de la red (bias), ver Ec. (1).

$$v = \sum_{j=1}^m w_j x_j + b = \sum_{j=0}^m w_j x_j \quad (\text{Ec. 1})$$

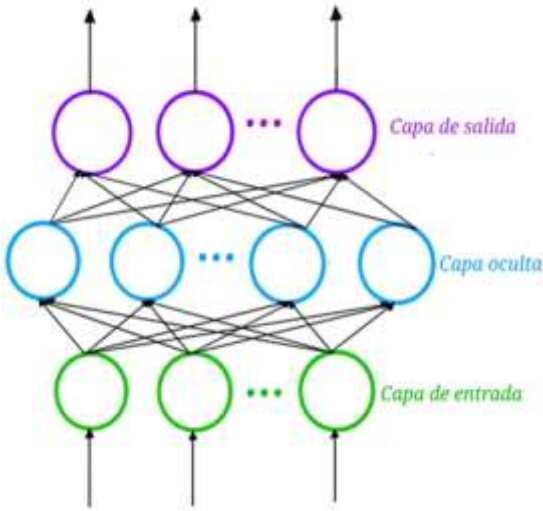


Figura 1. Construcción de una Red Neuronal Artificial simple [3].

En la ecuación 1 existen dos posibles escenarios para definir la ecuación, un caso donde se presente el sesgo (bias) por modelado representado por b o el caso que no se tenga este término, por lo que el índice de la sumatoria se ve modificada, en los casos de estudio empleados en este trabajo se utiliza por default las ecuaciones que incluyen los términos bias.

Adicionalmente a los factores de peso, las redes neuronales son determinadas por las funciones de activación por cada nodo, a excepción de los nodos de entrada. Hay diferentes funciones de activación o de transferencia según las aplicaciones, pero en general se tienen las funciones lineales, sigmoidea, tangente hiperbólica y muchas más que permiten definir el rango de salida que se busca (Ecs. (2-6)).

$$f(x) = cx \quad (\text{Ec. 2})$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad (\text{Ec. 3})$$

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (\text{Ec. 4})$$

$$f(x) = \frac{1}{2}(1 + \tanh(x)) \quad (\text{Ec. 5})$$

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (\text{Ec. 6})$$

Una característica importante de las funciones de activación no lineales es que, independientemente de la magnitud de las entradas, la función entrega una salida en un rango pequeño de valores, lo que permite modular órdenes de magnitud grandes entre los valores de las neuronas de entrada. Por último, las neuronas de salida hacen la predicción de las variables de respuesta de la red en función de la información recibida en las neuronas de entrada. Una vez que se tiene la estructura general de la red neuronal, es

necesario entrenarla, es decir, a través de observaciones o datos experimentales se realiza un ajuste de parámetros, en donde los factores de peso van cambiando, disminuyendo el error relativo de la predicción por parte de la red neuronal y del dato experimental. Existen múltiples algoritmos de aprendizaje; la más común es la retropropagación (*backpropagation*), en la cual la diferencia entre el valor deseado y el obtenido se calcula por cada salida de la neurona. Esta técnica se considera un

aprendizaje supervisado donde a cada iteración en que se vayan ajustando los valores se le denomina época (*epoch*). Por cuestión didáctica, en el uso de esta herramienta, se utiliza SOLVER™ de Excel® para realizar este ajuste de parámetros (Técnica de regresión no lineal o mínimos cuadrados).

Una vez entrenada la red, ésta debe ser alimentada con nuevos datos para verificar, examinar y determinar si se obtiene una predicción fidedigna de la información. Este enfoque permite a las redes neuronales aprender patrones complejos y realizar predicciones precisas en una variedad de aplicaciones. La cuestión de decidir cuantas

neuronas o capas ocultas debe llevar la red es una tarea heurística y dependerá de la complejidad del problema a resolver y la disponibilidad de datos experimentales. En este trabajo, los casos de estudio se realizarán en Excel®, por lo tanto, no se usarán más de 10 neuronas, con una sola capa oculta. En algunos casos donde los factores de peso tienden a cero, se pueden eliminar estas conexiones. Los datos se suelen preprocesar en un rango de 0 a 1 debido a que las salidas de las funciones de transferencia, como la función logarítmica sigmoidea o la tangente hiperbólica, tienen la salida en ese rango y se suelen normalizar los datos según la Ec. (7) u (8).

$$A = \frac{V - V_{min}}{V_{max} - V_{min}} \quad (\text{Ec. 7})$$

$$A = \left(\left(\frac{0.8}{V_{max} - V_{min}} \right) V + \left(0.1 - \frac{0.8V_{min}}{V_{max} - V_{min}} \right) \right) \quad (\text{Ec. 8})$$

Donde A es el valor adimensional, V es la variable original y V_{max} , V_{min} son los valores máximo y mínimo de los datos. Generalmente, la Ec. (7) es una adimensionalización general con rango de 0 a 1 mientras que la Ec. (8) tiene un grado de amortiguamiento que se puede considerar según el valor numérico, en este caso, estaría entre los valores 0.1 y 0.8, la razón de estas formas de adimensionalización dependerá de las funciones de transferencia que se utilicen.

Red recursiva

La red recursiva tiene una estructura similar a la red estándar estática, con nodos conectados a través de conexiones de alimentación hacia adelante. Tiene la característica de tener conexiones añadidas recurrentes, que pueden tener un retraso en el tiempo como se muestra en la Figura 2 [8].

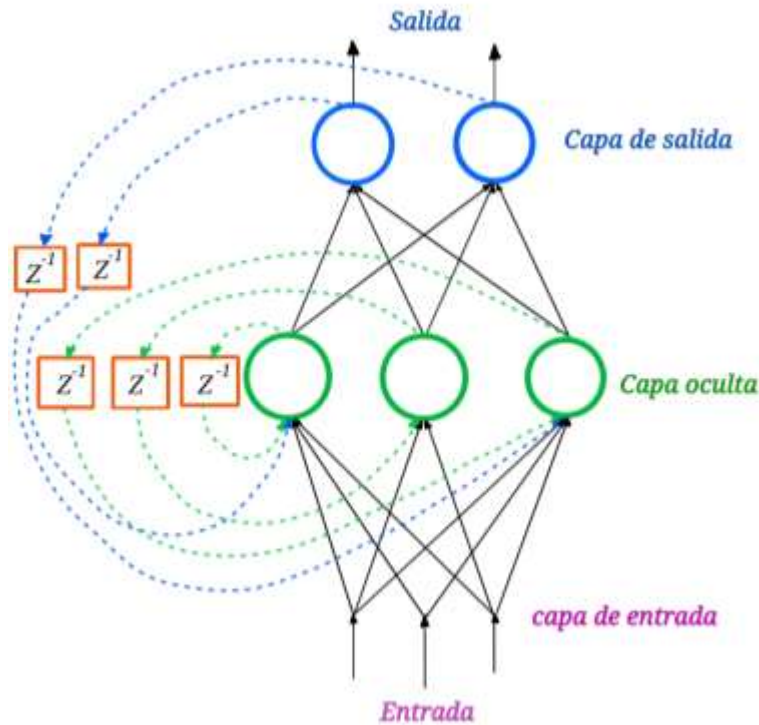


Figura 2. Estructura básica de una red recursiva con conexiones recurrentes [3].

Los modelos de redes neuronales recurrentes tienen la misma relación para realimentar la red como respuesta de impulso infinito autorregresivo. Los modelos tienen que ser modelos de respuesta de impulso finito de valores con respuesta promedio, lo que permite describir la evolución de las variables definidas del modelo en caso de presentar un cambio abrupto sin modificaciones de la trayectoria de la función. Lo anterior se permite por sus conexiones de retroalimentación, las variables recursivas pueden ser del orden que se requiera de acuerdo con las necesidades de las ventanas de datos de entradas retrasadas. Es una herramienta utilizada principalmente para describir sistemas dinámicos no periódicos. Este tipo de red neuronal presenta una validez en rangos con poca extrapolación.

Red dinámica diferencial

Las redes dinámicas surgieron como una necesidad de realizar descripciones temporales que puedan identificar

comportamientos en tiempos mayores que las redes recursivas. Las redes de dinámica diferencial han mostrado la capacidad de describir bifurcaciones. Las redes de dinámica diferencial son capaces de representar dinámicas completas, incluyendo bifurcaciones de estado estable, por lo que este tipo de configuración de red neuronal se le conoce como modelo de continuo en el tiempo [13].

La red dinámica diferencial es una herramienta para el análisis de series de tiempo (*Time Series Analysis*), que permite describir atractores en soluciones diferenciales de periodo cíclico en tiempo prolongado. La construcción de funciones diferenciales continuas implica usar la red neuronal en forma de una ecuación diferencial, tal como se muestra en la Ec. (9).

$$\frac{dy}{dt} = RNA(y, t, a) \quad (\text{Ec. 9})$$

El uso de esta función diferencial permite la construcción de atractores en espacio fase. El entrenamiento de este tipo de red se realiza con la formulación de una derivación numérica tipo Euler (Ec. (10)) o Runge-Kutta de 4° orden (Figura 3), método numérico utilizado para la solución de ecuaciones diferenciales, para obtener los datos deseados en el tiempo. El problema más evidente de esta aproximación de cálculo es la sensibilidad en la estimación de las derivadas numéricas, especialmente cuando se trate de derivadas de alto grado y con presencia de ruido.

$$y_{i+1} = y_i + h * RNA(y, t, a) \quad (\text{Ec. 10})$$

Como se observa en la Ec. (10) el integrador de Euler se utiliza para la obtención de datos, los cuales se comparan en el integrador hasta que el error disminuya, este tipo de aproximación es la que puede generar mayor error al ser un integrador de primer orden. Para los casos de estudio en Excel® se utilizará este método y se utilizará *Visual Basic for Applications* de Excel (VBA) para el integrador tipo Runge-Kutta de 4° orden que se muestra en la Figura 3. Este tipo de red neuronal puede ampliarse o ser la base de construcciones más complejas como ecuaciones diferenciales parciales, utilizando una ampliación de esta metodología de acuerdo con la bibliografía [21].

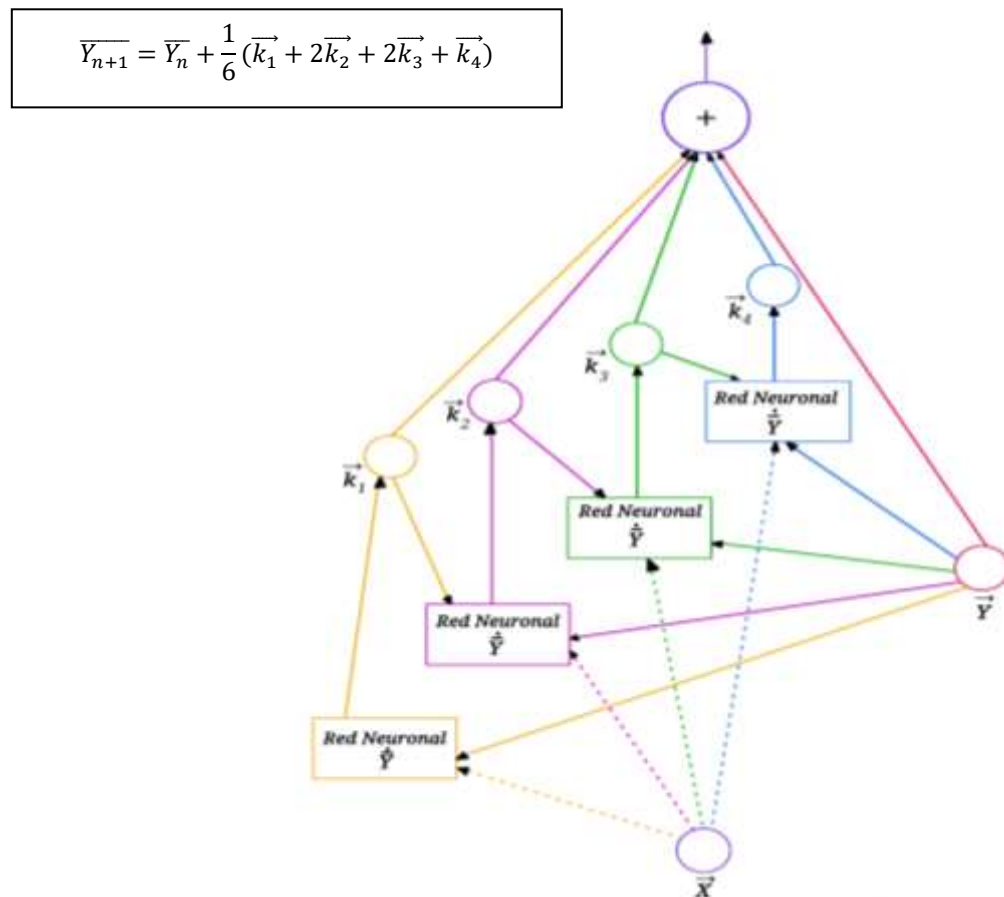


Figura 3. Representación esquemática de la implementación de un esquema de cuarto orden tipo Runge-Kutta [13].

Red híbrida (modelo de la caja gris)

Las redes neuronales pueden utilizarse acopladas a funciones que representen la fenomenología y las variaciones en parámetros que las ecuaciones fundamentales no son capaces de describir por completo en su comportamiento. Los modelos híbridos combinan el conocimiento parcial de un proceso con las redes neuronales para capturar la dinámica restante. Una característica que tiene este tipo de metodología es que permite reducir el tamaño de la red neuronal requerida para capturar la dinámica del problema [14].

Generalmente los modelos híbridos contienen un modelo fenomenológico básico que describe la parte bien conocida del proceso y una parte de la red que permite la corrección de errores que afectan la parte compleja o la parte desconocida del fenómeno. Los dos grandes beneficios de los modelos híbridos consisten en que el proceso se mantiene descrito fenomenológicamente y el carácter empírico de la red permite la flexibilidad en la precisión de los cálculos obtenidos [6].

La forma general de este tipo de funciones estacionarias tiene la estructura de la forma fenomenológica de un sistema representado por una función y un complemento de vector de probabilidades que se acopla algebraicamente en la ecuación, un ejemplo de este tipo de funciones se puede ver en la Ec. (11).

$$F(x) = g(x; p) + RNA(x; p) \quad (\text{Ec. 11})$$

En esta expresión $F(x)$ representa la función de salida, $g(x; p)$ representa la parte fenomenológica de la función, el término $RNA(x; p)$ representa la parte descrita por la red neuronal, x representa la variable independiente y p el parámetro. Es importante comentar que la red (RNA) podría estar dentro de la función como elemento

multiplicativo, un ejemplo de esto es la ecuación de cinética biológica representada por una red, en el caso de ser sistemas complejos y altamente no lineales.

La metodología para el análisis de modelos continuos en el tiempo, para la predicción de dinámicas prolongadas cíclicas de sistemas lineales, también puede describirse por modelos fenomenológicos acoplados a redes neuronales que permiten una descripción más completa. Esta metodología tiende a la caracterización parcial de la fenomenología fundamental conocida y complementada con la información de las redes neuronales [22].

En el caso de las aproximaciones de caja gris, se conoce parte del elemento fenomenológico de la ecuación, explícitamente calculado y la red neuronal, la cual aproxima solamente los elementos desconocidos con información probabilística, un ejemplo de este caso de estudio se presenta en la Ec. (12).

$$\frac{dy}{dt} = g(x, t; p) + RNA(x, t; p) \quad (\text{Ec. 12})$$

Donde $\frac{dy}{dt}$ representa la función diferencial, $g(x, t; p)$ es el elemento fenomenológico de la ecuación y $RNA(x, t; p)$ el elemento correspondiente a la red neuronal, que se considera como la forma general de la estructura del modelo híbrido para sistemas continuos.

Uso de hojas de cálculo (Excel®)

El uso de hojas de cálculo proporciona una alternativa popular y de costo asequible para enseñar técnicas de modelado y análisis en la toma de decisiones, por lo que es una alternativa que permite a los estudiantes comprender y aplicar los fundamentos numéricos y matemáticos de un sistema. En este trabajo se utilizan las hojas de cálculo para estructurar los modelos matemáticos y el procedimiento numérico paso a paso, con el

fin de que los estudiantes puedan disponer de una metodología didáctica de aplicación de las redes neuronales en diferentes casos de estudio.

Se estructuraron modelos de dos formas: la primera con fórmulas directas de la red aplicadas en las celdas, como un modelo matemático; la segunda forma utiliza una forma genérica de red con programación en VBA. La intención de manejar estas dos formas es plantear los modelos de manera que se pueda avanzar del modelo básico de la red hacia la construcción avanzada de modelos, tanto en la hoja de trabajo, como programados con VBA.

El siguiente paso es el uso de herramientas para el ajuste de parámetros (entrenamiento de la red neuronal), para lo cual se utilizó la herramienta SOLVER™ de Excel®, que permite la minimización de la función

objetivo (Figura 4). Existen múltiples metodologías para la minimización de funciones, como son los métodos de Levenberg-Marquardt, Gauss-Newton, gradiente del máximo descenso, etc., sin embargo, no se abordarán en este trabajo. SOLVER™ incorpora el uso de una herramienta GRG no lineal (*Generalized Reduced Gradient*) para la optimización de las funciones, de una forma didáctica y práctica, que facilita el entrenamiento de las redes neuronales.

Una práctica muy común al trabajar los datos de la red neuronal es el proceso de adimensionalización, que requiere el uso de una ecuación matemática que transforme las funciones en un rango de 0 a 1 para que la red pueda procesar la información más fácilmente, posteriormente la ecuación se regresará a sus variables originales.

Parámetros de Solver

Establecer objetivo: SAS1

Para: ☒ Máx ☐ Min ☐ Valor de: 0

Cambiando las celdas de variables:

Sujeto a las restricciones:

Convertir variables sin restricciones en no negativas

Método de resolución: GRG Nonlinear

Método de resolución

Seleccione el motor GRG Nonlinear para problemas de Solver no lineales suavizados. Seleccione el motor L.P Simplex para problemas de Solver lineales, y seleccione el motor Evolutionary para problemas de Solver no suavizados.

Ayuda Resolver Cerrar

Figura 4. Ventana ilustrativa de los parámetros que solicita SOLVER™ para el ajuste de parámetros [15].

En el Anexo A se describe la metodología para el entrenamiento de una red neuronal estática sencilla para el caso de la simulación del balance de materia estacionario en un tanque con dos corrientes de entrada y una corriente de salida.

III. Resultados

En esta sección se muestran ejemplos del uso de cada una de las configuraciones de las redes propuestas. Se utilizan datos de problemas de ingeniería química o bioquímica para mostrar su aplicación de forma didáctica.

Casos de estudio de red estática

Caso 1: Oxidación catalítica del 3-hexanol

Se presenta el caso de estudio de la oxidación catalítica del compuesto 3-hexanol con peróxido de hidrógeno presentado por Gallort y col. 1998. Los datos muestran el efecto del solvente (CH_3OH) en la conversión parcial del peróxido de hidrógeno usando una red neuronal donde la entrada de la red es el tiempo y la salida es la concentración del solvente [23].

La Tabla 1 muestra los datos de la oxidación del 3-hexanol respecto de la conversión parcial y su relación respecto de la concentración de solvente.

Tabla 1. Datos de la oxidación catalítica del 3-Hexanol, Gallort y col. 1998 [23].

Tiempo modificado de reacción (t) (h kg/kmol)	Conversión parcial (y)	
	0.75g de CH_3OH	1.3g de CH_3OH
3	0.055	0.040
6	0.090	0.070
13	0.120	0.100
18	0.150	0.130
26	0.165	0.150
28	0.175	0.160

Se utilizó una red hacia adelante de 5 neuronas ocultas programada de acuerdo con los conceptos básicos de las redes neuronales, este programa se muestra en el Anexo B. En dicha sección se muestran las diferentes etapas del código, el primer segmento es la definición de las variables y la lectura de factores de peso donde se usan valores aleatorios entre 0 y 1; el siguiente segmento se relaciona con los nodos de entrada de acuerdo con la Ec. (1). Posteriormente se conectan estos resultados con las funciones de transferencia que se pueden seleccionar de las ecuaciones 2 a 6. Estas funciones de transferencia tienen una aplicación dependiendo de los rangos de datos que se elijan. Finalmente, la función de salida, en este caso de estudio, se tiene solamente una salida, por lo que es una variable, en el caso

de que sean salidas múltiples, se requiere que sea una salida del tipo matriz.

Los datos de la red se ajustaron hasta una suma de errores al cuadrado de 0.004532 utilizando una red sin adimensionalizar, con un error porcentual aproximado de 19.2484% entre los datos experimentales y los datos de la red. El error absoluto medio (MAE) es de 0.01701, mientras que el error cuadrático medio (MSE) y su raíz (RMSE) son de 0.00038 y 0.01943, respectivamente. Finalmente, el coeficiente de determinación calculado fue de 0.80184, esto indica un buen ajuste de los datos de la red con una baja variabilidad. En la Figura 5 se muestran los resultados de una simulación usando la red con un parámetro de masa de solvente de 1 g de CH_3OH .

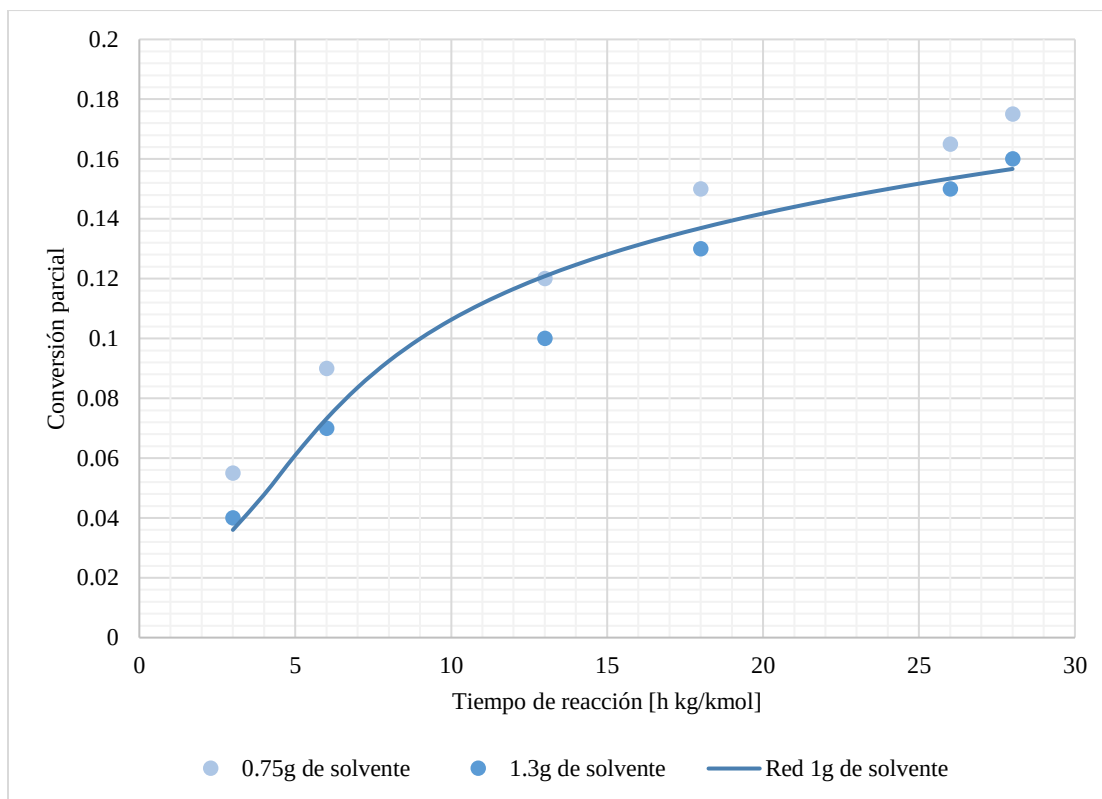


Figura 5. Conversión parcial de H_2O_2 a través de reacción de oxidación 3 hexanol, simulación usando la red con 1 g de solvente Gallort y col. 1998 [23].

En la simulación de la red generada se puede observar el comportamiento intermedio de los datos de validación y las diferentes características de interpolación que presentan las redes neuronales. Este tipo de modelos son excelentes en la predicción de datos intermedios y, aunque su capacidad para extrapolar es limitada, son muy útiles para la construcción de modelos de interpolación complejos no lineales y de múltiples variables.

Caso 2: Producción de Escleroglucano

En este caso de estudio se utilizó la producción fermentativa de escleroglucano,

los datos experimentales fueron tomados de Desai y col. 2008 [11], donde se tienen cuatro variables de entrada y una variable de salida. Las entradas corresponden a la sacarosa, el extracto de levadura, el K_2HPO_4 y el $MgSO_4$. La salida es la producción de escleroglucano, medido en g/L. Se utilizaron 21 datos experimentales de entrada y salida para el entrenamiento de la red y 9 datos de validación seleccionados de forma aleatoria, ver Tabla 2 y 3. También en este caso de estudio se utilizó una capa oculta con 5 neuronas.

Tabla 2. Datos para el entrenamiento de la red neuronal estática *feedforward*, Desai y col. 2008 [11].

NO	Entrada				Salida
	Sacarosa	Extracto de levadura	K ₂ HPO ₄	MgSO ₄	Escleroglucano
1	35	1.0	1.25	0.50	7.97
2	65	1.0	1.25	0.50	15.22
3	35	2.0	1.25	0.50	9.53
4	65	1.0	1.75	0.50	13.16
5	35	2.0	1.75	0.50	9.81
6	65	2.0	1.75	0.50	11.65
7	35	1.0	1.25	1.00	8.47
8	65	1.0	1.25	1.00	14.05
9	35	1.0	1.75	1.00	7.09
10	65	1.0	1.75	1.00	11.10
11	35	2.0	1.75	1.00	6.69
12	65	2.0	1.75	1.00	10.44
13	20	1.5	1.50	0.75	6.29
14	50	2.5	1.50	0.75	10.38
15	50	1.5	1.00	0.75	10.65
16	50	1.5	2.00	0.75	9.50
17	50	1.5	1.50	0.25	10.16
18	50	1.5	1.50	0.75	11.50
19	50	1.5	1.50	0.75	11.10
20	50	1.5	1.50	0.75	10.99
21	50	1.5	1.50	0.75	10.98

Los datos de validación se pueden utilizar para que se evite el sobreentrenamiento de la red. Este fenómeno se presenta cuando se trata de sobre ajustar los factores de peso a los datos, de tal forma que pierde capacidad de interpolación, por lo tanto, se pierde el control buscando el punto en el que el número de iteraciones proporcione un mínimo en el error

promedio de los datos de validación y un valor bajo en la función objetivo. En algunos programas comerciales se presenta una gráfica que muestra el mínimo error en los valores de validación, donde se determinan los mínimos conjuntos entre los valores de entrenamiento junto con los valores de validación.

Tabla 3. Datos usados para la validación, no usados en el entrenamiento, Desai y col. 2008 [11].

NO	Entrada				Salida
	Sacarosa	Extracto de levadura	K_2HPO_4	$MgSO_4$	Escleroglucano
1	65	2.0	1.25	0.50	12.60
2	35	1.0	1.75	0.50	9.44
3	80	1.5	1.50	0.75	14.88
4	50	0.5	1.50	0.75	12.25
5	50	1.5	1.50	1.25	10.65
6	50	1.5	1.50	0.75	11.13
7	50	1.5	1.50	0.75	11.03
8	35	2.0	1.25	1.00	9.50
9	65	2.0	1.25	1.00	11.70

Los datos de las Tablas 2 y 3 se adimensionalizan utilizando las Ec. (7) y (8). Se utiliza cada conjunto de datos para que la red sea entrenada y converja más fácilmente. Todos los datos adimensionalizados quedan en el rango de 0 y 1 si se usa la Ec. (7); y entre 0.1 y 0.9 usando la Ec. (8), en este caso de estudio se utilizó la Ec. (8), tal como se muestra en la Tabla 4.

Con los datos del modelo generado por el macro de Excel® se realiza un ajuste de mínimos cuadrados para obtener los parámetros que mejor se adaptan a la red utilizando la herramienta SOLVER™. De los datos resultantes se obtiene el valor del error promedio en relación con los datos experimentales, tal como se muestra en la Tabla 5.

Tabla 4. Extracto de datos con los factores adimensionales, datos adimensionales y la salida de la red ya ajustada.

NO	Entrada				Salida
	Sacarosa	Extracto de levadura	K_2HPO_4	$MgSO_4$	Escleroglucano
1	0.36666	0.10000	0.3	0.36666	0.25050
2	0.90000	0.10000	0.3	0.36666	0.90000
3	0.36666	0.63333	0.3	0.36666	0.39025
4	0.90000	0.10000	0.7	0.36666	0.71545
5	0.36666	0.63333	0.7	0.36666	0.41534
6	0.90000	0.63333	0.7	0.36666	0.58017
7	0.36666	0.10000	0.3	0.90000	0.29529
8	0.90000	0.10000	0.3	0.90000	0.79518
9	0.36666	0.10000	0.7	0.90000	0.17166
10	0.90000	0.10000	0.7	0.90000	0.53090

Fuente: Elaboración propia, basada en los datos de Desai y col. [11].

Al realizar este ejercicio se registraron datos cada 5 iteraciones y se observó el comportamiento de los errores. Se pudo observar que el error cuadrático de los datos de entrenamiento es 0.10008 y disminuye progresivamente, el error promedio

porcentual del valor de la red respecto del valor real de los datos de validación es de 15.66919 y también disminuye progresivamente. En un punto, las iteraciones empiezan a crecer, esto quiere decir que en este punto el modelo comienza a

sobreentrenarse, este ejemplo se puede utilizar para mostrar paso a paso el uso de la red neuronal y los detalles importantes a considerar. El error absoluto medio (MAE) es de 0.05805, mientras que el error cuadrático medio (MSE) y su raíz (RMSE) son de 0.00477 y 0.06903, respectivamente. Finalmente, el coeficiente de determinación calculado fue de 0.87745.

Este tipo de red neuronal es la más común y simple que se puede encontrar. Si se desea mayor precisión, se pueden incrementar el

número de neuronas y el número de capas, sin embargo, entre más complejo sea el modelo, se requerirá una capacidad computacional mayor. Para este trabajo se utilizarán redes pequeñas ya que para una red más robusta y grande se ocupa un software especializado como MATLAB, o bien, que ésta sea programada en FORTRAN, Python o algún otro lenguaje de alto nivel. También pueden usarse con algoritmos de ajuste y mayor enfoque en la velocidad, como es el caso de Levenberg-Marquardt o algoritmos híbridos.

Tabla 5. Ajuste de mínimos cuadrados y el cálculo del error promedio de los datos experimentales.

Experimental adimensional	ANN adimensional	Mínimos Cuadrados	Resultados ANN	
Escleroglucano	Escleroglucano		Escleroglucano	Error %
0.25050	0.33594	0.00729	8.92371	11.96637
0.90000	0.80036	0.00992	14.10780	7.30742
0.39025	0.37546	0.00021	9.36485	1.73288
0.71545	0.68926	0.00068	12.86765	2.22143
0.41534	0.27937	0.01848	8.29222	15.47176
0.58017	0.61016	0.00089	11.98474	2.87337
0.29529	0.26715	0.00079	8.15585	3.70885
0.79518	0.71984	0.00567	13.20906	5.98526
0.17166	0.15922	0.00015	6.95109	1.95913
0.53090	0.61067	0.00636	11.99036	8.02134
0.13583	0.20960	0.00544	7.51350	12.30953
0.47178	0.52844	0.00321	11.07255	6.05893
0.10000	0.06413	0.00128	5.88966	6.36467
0.46640	0.40118	0.00425	9.65194	7.01400
0.49059	0.59856	0.01165	11.8552	11.31665
0.38756	0.39191	1.88E-05	9.54846	0.51012
0.44669	0.57190	0.01567	11.55767	13.75663
0.56674	0.49413	0.00527	10.68948	7.04794
0.53090	0.49413	0.00135	10.68948	3.69832
0.52105	0.49413	0.00072	Suma	Error promedio
0.52015	0.49413	0.00067		
		0.10007	10.68948	2.73443
			10.68948	2.64584
				6.41452 %

Fuente: Elaboración propia, basada en los datos de Desai y col. [11].

Caso 3: Catálisis lipídica de tensioactivos en esteramidas

Para este caso de estudio se utilizó el diseño experimental para la optimización de una

reacción enzimática en la preparación de tensioactivos basados en esteramidas. Se emplearon cinco variables de proceso: cantidad de enzima, tiempo de reacción, temperatura de reacción, relación molar de

substratos y velocidad de agitación. Este problema es reportado por Masoumi y col. 2014 [24].

Para implementar la red neuronal en Excel® se adimensionalizaron los datos de entrada y el dato de salida utilizando la Ec. (7), se utilizó el mismo código mostrado en el Anexo B con una red neuronal de 5 neuronas ocultas; los datos para entrenarla fueron 30, con un error real promedio del 7.87%. Posteriormente se tomaron datos para validación con un error del 6.6% de error real. Se realizaron iteraciones individuales para observar el comportamiento de los errores, se pudo observar que, después de la iteración 35, comienza un cambio en la relación del error y el ajuste del error de los datos de validación comienza a subir proporcionalmente. Mientras tanto, el error de los datos usados en el ajuste disminuye, esto demuestra el efecto de la divergencia, por lo que, en cuanto al entrenamiento de la red neuronal, se debe evitar la divergencia. En la Tabla 6 se muestra un segmento de los datos adimensionalizados

utilizados en el entrenamiento de la red. Se realizó un análisis estadístico donde se pudo observar que el error cuadrático de los datos experimentales con los datos de red es 661.06769 y disminuye progresivamente. El error promedio porcentual del valor de la red respecto del valor real de los datos de validación es de 7.87522 mientras que el error absoluto medio (MAE) es de 3.72744. Por otro lado, el error cuadrático medio (MSE) y su raíz (RMSE) son de 22.03559 y 4.69421, respectivamente. Finalmente, el coeficiente de determinación calculado fue de 0.83660.

Con los resultados generados por la red se regresaron los valores adimensionales a sus valores de variables originales. Con estos valores se realizaron unas gráficas de ajuste entre el valor de conversión real (experimental) contra el valor de conversión predicho por la red (Figura 6), gráficas similares se encuentran reportadas en la bibliografía de referencia, Masoumi y col. 2014 [24].

Tabla 6. Segmento de datos adimensionales usados en el modelo de la red neuronal.

Cantidad de enzima (w/w%)	Tiempo de reacción (h)	Temperatura de reacción (°C)	Tasa molar de sustratos (moles)	Velocidad de agitación (r.p.m)	conversión (%)
0.50000	0.50000	0.63636	0.36363	0.50000	0.67860
0.50000	0.50000	0.636363	0.36363	0.00000	0.49169
0.21428	0.78571	0.272727	0.00000	0.78571	0.50118
0.214285	0.78571	1.00000	0.00000	0.78571	0.83229
0.50000	0.50000	0.00000	0.36363	0.50000	0.31426
0.50000	0.50000	0.63636	0.36363	0.50000	0.56356
0.21428	0.21428	0.27272	0.00000	0.21428	0.41959
0.50000	1.00000	0.63636	0.36363	0.50000	0.75545
0.78571	0.21428	0.27272	0.72727	0.21428	0.00000
0.21428	0.21428	0.27272	0.72727	0.21428	0.01208
0.50000	0.50000	0.63636	0.36363	1.00000	0.45801
0.78571	0.78571	1.00000	0.00000	0.21428	0.82732
0.78571	0.21428	0.27272	0.72727	0.78571	0.18951
1.00000	0.50000	0.63636	0.36363	0.50000	0.50593
0.50000	0.50000	0.63636	1.00000	0.50000	0.28059
0.78571	0.78571	0.27272	0.72727	0.21428	0.22296
0.50000	0.50000	0.63636	0.36363	0.50000	0.80811

0.50000	0.50000	0.63636	0.363636	0.50000	0.71228
0.50000	0.50000	0.63636	0.363636	0.50000	0.65465
0.21428	0.21428	1.00000	0.727272	0.78571	0.42931

Fuente: Elaboración propia, basada en los datos de Masoumi y col. [24].

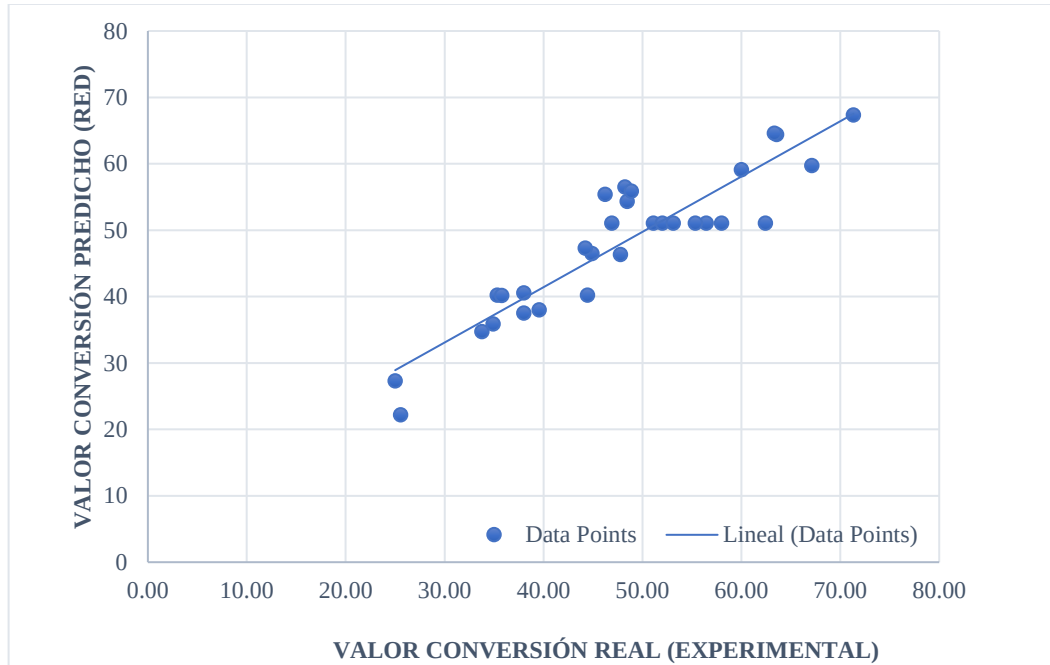


Figura 6. Relación lineal de los valores de conversión reales contra los valores predichos por la red, de los datos entrenados, Masoumi y col. 2014 [24].

Se puede observar un ajuste admisible empleando la red, además, se muestra que la red es la mejor opción para la construcción de modelos no lineales con parámetros múltiples utilizados en la optimización de procesos. En este caso de estudio se demuestra que este tipo de herramientas es susceptible al sobreentrenamiento y, por lo tanto, es necesario tener una metodología para evitarlo.

Caso de estudio de red recursiva

Caso 4: Modelo diferencial de reacción química

Para este caso de estudio con una red recursiva se utilizó una ecuación diferencial con solución conocida, Dua, 2011 [25], el modelo matemático propuesto es descrito por la Ec. (13a) y (13b). Este es un modelo general de dos ecuaciones diferenciales con

dos incógnitas, z_1 y z_2 , respecto de la variable t , de tal forma que dispongan de una solución analítica.

$$\frac{dz_1}{dt} = -5z_1 \quad (\text{Ec. 13a})$$

$$\frac{dz_2}{dt} = 5z_1 - z_2 \quad (\text{Ec. 13b})$$

Este modelo tiene condiciones iniciales $z(t = 0) = [1, 0]$ y una solución analítica mostrada en la Ec. (14a) y (14b).

$$z_1(t) = e^{-5t} \quad (\text{Ec. 14a})$$

$$z_2(t) = \frac{5}{4} e^{-5t} (-1 + e^{4t}) \quad (\text{Ec. 14b})$$

Un ejemplo en la ingeniería química de modelos con comportamiento dinámico y no cíclico son los fenómenos de cinética oxidativa, los cuales son susceptibles para el

uso de este tipo de redes. Éstas se han desarrollado para analizar series de tiempo basadas en información obtenida con anterioridad para predecir o especular los datos en el tiempo futuro. Para este caso de estudio se presentan datos puntuales representados por puntos y la simulación de la red neuronal recurrente representada como una línea continua, ver Figura 7. La red se estructuró con la lectura de dos datos

anteriores de cada una de las variables utilizadas, es decir, con cuatro datos de entrada a la red. Estos datos se ajustaron usando mínimos cuadrados, para hacer más fácil el entrenamiento, lo ideal es usar un rango de valores que se encuentre entre 0 y 1. En el Anexo C se muestra la modificación que se hizo para que se obtengan 2 variables de salida de la red generada.

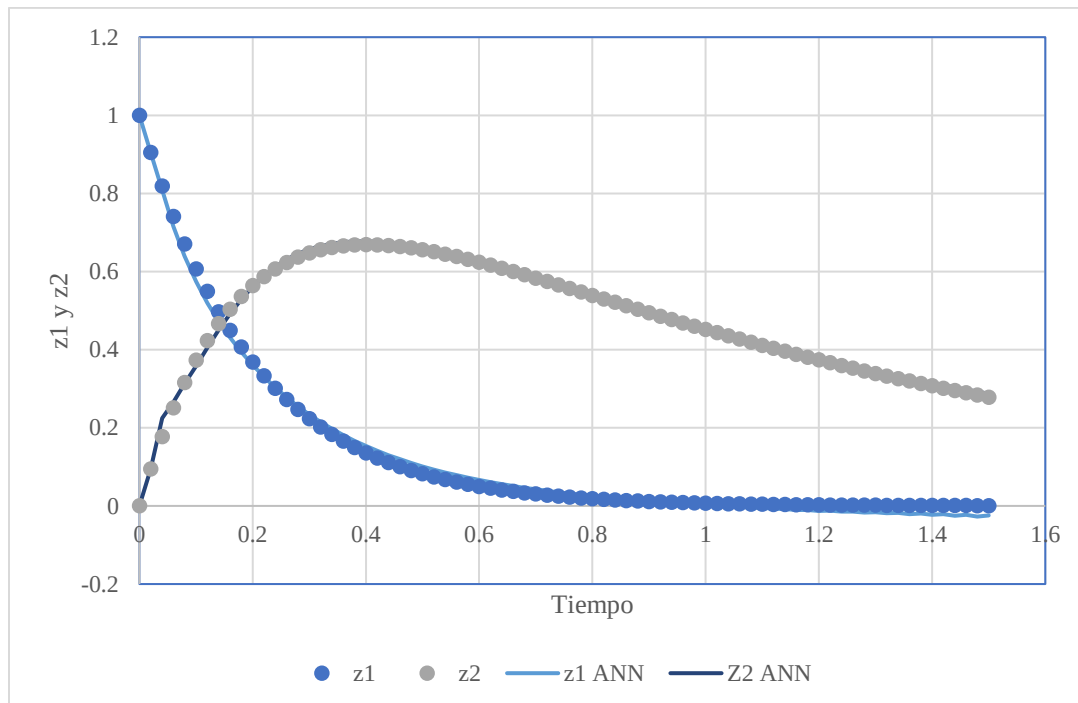


Figura 7. Representación gráfica de los resultados analíticos y la simulación de la red neuronal recursiva, Dua, 2011 [25].

En la Tabla 7 se muestran los datos introducidos en Excel® que utiliza la función de la red neuronal como función recurrente retroalimentada, para predecir el comportamiento en el tiempo del sistema de reacción. Esta estrategia es muy útil para el control de reactores y para el seguimiento de comportamientos secuenciales, o bien, para describir sistemas espacio-estado no lineales o con perturbaciones, temas muy amplios que no se abordarán en los ejemplos mostrados en este estudio.

En este caso de estudio se utiliza una red de 5 neuronas por cuestión didáctica, sin embargo, para problemas de mayor complejidad se puede incrementar el número de nodos hasta lograr la descripción de sistemas más complejos. Además, debido a las características de este tipo de red, se observa que es dependiente de la secuencia previa de datos y, por lo tanto, muy susceptible a generar distorsiones de la dinámica. Este tipo de dificultades se resuelve con redes diferenciales continuas, las cuales se presentarán a continuación.

Tabla 7. Datos de Excel® con la red recurrente y el ajuste por mínimos cuadrados.

t	z1	z2	t	z1 ANN	z2 ANN	Mínimos Cuadrados			z1 ANN	z2 ANN
0	1	0	0	1	0				1	
0.02	0.90483	0.09420	0.02	0.90483	0.09420			<u>Sumatoria</u>	0.90483	0.09420
0.04	0.81873	0.17757	0.04	0.80838	0.22484	0.00010	0.00223	<u>0.02437</u>	0.80838	0.22484
0.06	0.74081	0.25118	0.06	0.71567	0.26534	0.00063	0.00020		0.71567	0.26534
0.08	0.67032	0.31599	0.08	0.63889	0.31293	0.00098	9.38E-06		0.63889	0.31293
0.1	0.60653	0.37288	0.1	0.57443	0.35854	0.00103	0.00020		0.57443	0.35854
0.12	0.54881	0.42263	0.12	0.51975	0.40506	0.00084	0.00030		0.51975	0.40506
0.14	0.49658	0.46596	0.14	0.47259	0.44985	0.00057	0.00025		0.47259	0.44985
0.16	0.44932	0.50351	0.16	0.43111	0.49154	0.00033	0.00014		0.43111	0.49154
0.18	0.40656	0.53587	0.18	0.39420	0.52896	0.00015	4.70E-05		0.39420	0.52896
0.2	0.36787	0.56356	0.2	0.36084	0.56156	4.94E-05	3.97E-06		0.36084	0.56156
0.22	0.33287	0.58705	0.22	0.33064	0.58923	4.93E-06	4.72E-06		0.33064	0.58923
0.24	0.30119	0.60679	0.24	0.30303	0.61218	3.38E-06	2.91E-05		0.30303	0.61218
0.26	0.27253	0.62314	0.26	0.27792	0.63082	2.90E-05	5.89E-05		0.27792	0.63082
0.28	0.24659	0.63648	0.28	0.25487	0.64562	6.85E-05	8.36E-05		0.25487	0.64562
0.3	0.22313	0.64711	0.3	0.23394	0.65706	0.00011	9.90E-05		0.23394	0.65706
0.32	0.20189	0.65531	0.32	0.21470	0.66556	0.00016	0.00010		0.21470	0.66556
0.34	0.18268	0.66135	0.34	0.19726	0.67153	0.00021	0.00010		0.19726	0.67153
0.36	0.16529	0.66547	0.36	0.18119	0.67530	0.00025	9.66E-05		0.18119	0.67530
0.38	0.14956	0.66786	0.38	0.16665	0.67714	0.00029	8.61E-05		0.16665	0.67714
0.4	0.13533	0.66873	0.4	0.15319	0.67733	0.00031	7.39E-05		0.15319	0.67733

Fuente: Elaboración propia, basada en los datos de Dua, [25].

En el análisis estadístico se pudo observar que el error cuadrático de los datos de entrenamiento es 0.01033 mientras que el error promedio porcentual del valor de la red respecto del valor real de los datos de validación es de 3.69596. El error absoluto medio (MAE) es de 0.01220, mientras que el error cuadrático medio (MSE) y su raíz (RMSE) son de 0.00025 y 0.01561, respectivamente. Finalmente, el coeficiente de determinación calculado fue de 0.99531.

Caso de estudio de red dinámica diferencial

Caso 5: Modelo FitzHugh-Nagumo

Para el caso de red dinámica se utilizó el modelo *FitzHugh-Nagumo*, el cual describe un prototipo excitable, siendo este un modelo bidimensional simplificado del modelo Hodgkin-Huxley de generación de impulsos eléctricos en el cerebro de calamares gigantes (*Architeuthis*). Esta formulación matemática

representa un oscilador excitable donde, si un estímulo externo (I_{ext}) sobrepasa cierto valor, el sistema presentará características de espacio fase. Este modelo se eligió por presentar comportamientos cíclicos que permiten mostrar las capacidades de simulación de las redes neuronales dinámicas.

Las ecuaciones de este modelo se muestran en la Ec. (15a) y (15b):

$$\frac{dv}{dt} = v - \frac{v^3}{3} - w + I_{ext} \quad (\text{Ec. 15a})$$

$$\tau \frac{dw}{dt} = v + a - bw \quad (\text{Ec. 15b})$$

Donde los términos v representarían el potencial de la membrana, w es la variable de recuperación $a = 0.7$, $b = 0.8$, $\tau = 0.08$,

$I_{ext} = 0.5$ son constantes y generan una función con características oscilatorias.

La red dinámica utilizada en este caso de estudio está basada en un modelo tipo Euler de primer orden, para que su uso en el software Excel® se pueda mostrar de forma sencilla y didáctica, lo cual es adecuado para explicar este tipo de configuración de red neuronal. Para la descripción diferencial con mayor precisión se recomienda utilizar órdenes superiores como el propuesto por la bibliografía [13]. Se realizó una red de doble capa oculta con 3 neuronas ocultas, 2 neuronas de entrada y 2 de salida. Este tipo de sistemas tienen una estructura más compleja, por lo tanto, la red se programa en ciclos, tal

como se presenta en el Anexo D, y la estructura utilizada corresponde la forma de la ecuación 9 donde la red se acopla a una formulación diferencial, esto significa que la red está embebida en la parte diferencial de la ecuación diferencial.

El procedimiento para encontrar la solución del problema es similar a lo que se ha realizado en el uso de las redes anteriores. Sin embargo, a diferencia de las redes anteriores, se adimensionalizan las ecuaciones, no los datos, esto con el fin de facilitar la convergencia (Ec. 16). Adicionalmente, para este caso, se usan más datos para lograr una mejor definición. Debido a ello, tiene la desventaja de aumentar el tiempo de procesamiento computacional.

$$\begin{aligned} \frac{\tau}{a_w} \frac{d\bar{w}}{dt} &= \left(\frac{\bar{v} - b_v}{a_v} \right) + a - b \left(\frac{\bar{w} - b_w}{a_w} \right) \\ \frac{d\bar{w}}{dt} &= \frac{a_w}{a_v \tau} (\bar{v} - b_v) + \frac{a}{\tau} a_w - \frac{b}{\tau} (\bar{w} - b_w) \end{aligned} \quad (\text{Ec. 16})$$

En este caso de estudio didáctico se usaron 21 datos en un tiempo de simulación de 100 segundos con pasos de intervalos de 0.5 segundos. Posteriormente, se realizó una

suma de cuadrados en la cual se utilizó SOLVER™ para ajustar los factores de peso de la red, ver Tabla 8.

Tabla 8. Datos de Excel® con la red dinámica y el ajuste por mínimos cuadrados.

	Datos Experimentales		Datos de Modelo		Mínimos Cuadrados		Sumatoria
1	0.63430	0.25203	0.57035	0.26995	0.00408	0.00032	0.04642
5	0.84114	0.55439	0.81498	0.53378	0.00068	0.00042	
10	0.74853	0.82078	0.77267	0.80183	0.00058	0.00035	
15	0.39498	0.88562	0.34852	0.88392	0.00215	2.91E-06	
20	0.16214	0.52216	0.19568	0.55336	0.00112	0.00097	
25	0.21747	0.27522	0.19809	0.27435	0.00037	7.60E-07	
30	0.27327	0.14396	0.24851	0.12889	0.00061	0.00022	
35	0.34144	0.10007	0.34235	0.10278	8.25E-07	7.36E-06	
40	0.62126	0.16675	0.53971	0.24181	0.00665	0.00563	
45	0.83790	0.56877	0.83846	0.56634	3.11E-07	5.91E-06	
50	0.74363	0.82795	0.74758	0.82257	1.56E-05	2.90E-05	
55	0.34278	0.87893	0.32135	0.87124	0.00045	5.90E-05	
60	0.16440	0.50945	0.19270	0.52683	0.00080	0.00030	
65	0.21968	0.26794	0.20069	0.25850	0.00036	8.91E-05	

70	0.27562	0.14062	0.25384	0.12233	0.00047	0.00033
75	0.34509	0.10000	0.35307	0.10670	6.37E-05	4.50E-05
80	0.66090	0.17603	0.56176	0.26176	0.00982	0.00734
85	0.83467	0.58270	0.85797	0.59524	0.00054	0.00015
90	0.73860	0.83478	0.72642	0.84264	0.00014	6.17E-05
95	0.28469	0.87050	0.30162	0.86040	0.00028	0.00010
100	0.16669	0.49686	0.19071	0.50686	0.00057	1.00E-04

Fuente: Elaboración propia, basada en los datos de Rico-Martínez y col. [13].

En este caso de estudio se genera un error de desviación debido a la acumulación del error cuadrático medio mínimo (MSE) obtenido por la formulación en el transcurso del tiempo. Esto se debe a que, al discretizar las derivadas, una pequeña variación en los datos experimentales genera una gran distorsión con mínimos cambios en las variables. En la Figura 8 se muestra la gráfica comparativa de los datos experimentales con respecto a la simulación, así como una extrapolación de la simulación.

En cuanto al análisis estadístico, se observa que el error cuadrático de los datos de entrenamiento es 0.04642 y el error promedio porcentual del valor de la red respecto del valor real de los datos de validación es de 7.55269. El error absoluto medio (MAE) es de 0.02326, mientras que el error cuadrático medio (MSE) y su raíz (RMSE) son de 0.00111 y 0.03290, respectivamente. Finalmente, el coeficiente de determinación calculado fue de 0.98358.

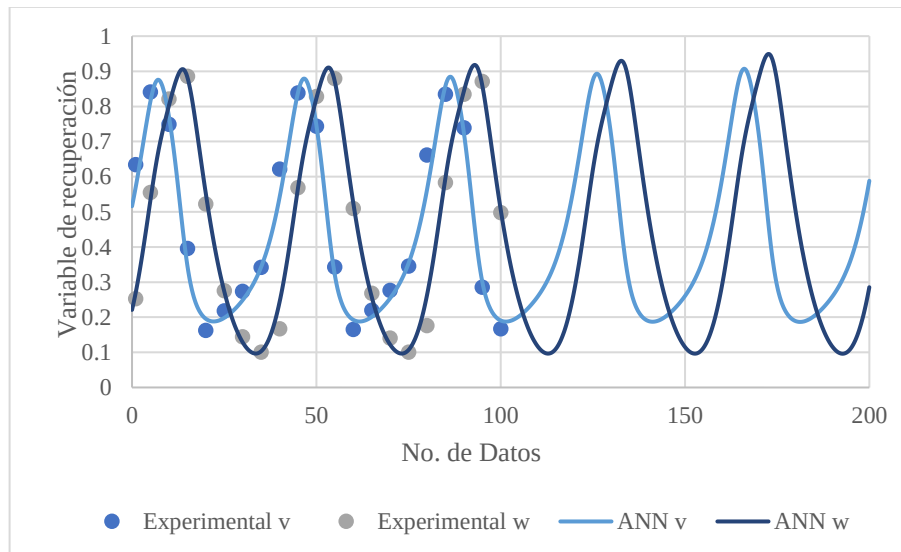


Figura 8. Representación gráfica de los resultados analíticos y la simulación de la red neuronal dinámica.

Fuente: Elaboración propia.

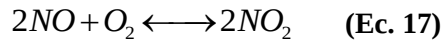
Las redes neuronales generalmente se utilizan para predicciones de datos en tiempos cortos, sin embargo, las redes neuronales dinámicas tienen la capacidad de realizar extrapolaciones en el futuro (*forecasting*) y su uso radica en casos de comportamiento

complejo periódico y de modelado de funciones continuas.

Caso de estudio de red híbrida (modelo de caja gris)

Caso 6: Modelo de reacción gaseosa Bodenstein-Linder

En este caso de estudio se utilizó el modelo de una reacción en fase gaseosa de NO con O_2 , basado en el modelo Bodenstein-Linder publicado por Bellman y col. 1967 [26]. La reacción se muestra en la Ec. (17).



El comportamiento de este sistema de reacción se describe por la Ec. (18).

$$\frac{dx}{dt} = k_1(\alpha - x)(\beta - x)^2 - k_2x^2 \quad (\text{Ec. 18})$$

Donde x representa la concentración de NO_2 , $\alpha = 126.2$ y $\beta = 91.9$. Se realizó un ajuste del modelo con datos obtenidos de la bibliografía [26], se puede observar que al usar SOLVER™ los datos k_1 y k_2 son parámetros condicionados. Esto quiere decir que su relación es muy distante, por lo que, para ajustar correctamente el parámetro k_1 , se requirió un escalado del factor. Se construyó un sistema híbrido, con el fin de comprobar el buen funcionamiento de la estrategia híbrida de caja gris para el ajuste de la primera

sección de la Ec. (18). Esto se puede ver en la Ec. (19), la cual es una red que complementa la ecuación fenomenológica, esta característica intrínseca de un modelo de caja gris.

$$\frac{dx}{dt} = RNA(x) - k_2x^2 \quad (\text{Ec. 19})$$

En la Tabla 9 se muestran los datos obtenidos en el procedimiento ejecutado en una hoja de cálculo de Excel®. En una parte se tiene el ajuste del parámetro k_1 y k_2 de forma tradicional con un factor de escala para poder resolverlo; además, se muestra el uso de un modelo híbrido para el ajuste de la Ec. (19). En el análisis estadístico se observó que el error cuadrático de los datos es 4.45501 y el error promedio porcentual del valor de la red respecto del valor real de los datos es de 3.36618. El error absoluto medio (MAE) es de 0.47974, mientras que el error cuadrático medio (MSE) y su raíz (RMSE) son de 0.29700 y 0.54498, respectivamente. Finalmente, el coeficiente de determinación calculado fue de 0.99851.

Tabla 9. Datos de Excel® y el ajuste por mínimos cuadrados de la ecuación 18 y el modelo híbrido de caja gris.

t	Concentración NO_2	Modelo	Mínimos cuadrados		Modelo híbrido	Mínimos cuadrados	
0	0	0	0	Sumatoria	0	0	Sumatoria
1	1.4	4.58090	10.11813	19.02716	3.97627	6.63721	10.13555
2	6.3	8.20115	3.61437		7.47011	1.36915	
3	10.5	11.76261	1.59419		11.02948	0.28035	
4	14.2	14.91743	0.51471		14.27480	0.00559	
5	17.6	17.99372	0.15502		17.51200	0.00774	
6	21.4	20.48636	0.83473		20.17818	1.49282	
7	23	22.73082	0.07245		22.60364	0.15709	
9	27	26.59966	0.16026		26.81286	0.03501	
11	30.5	29.79987	0.49017		30.28598	0.04580	
14	34.4	33.64598	0.56854		34.38207	0.00032	
19	38.8	38.25388	0.29824		39.02686	0.05146	
24	41.6	41.31227	0.08278		41.81871	0.04783	
29	43.5	43.46760	0.00104		43.56222	0.00387	
39	45.3	46.02282	0.52246		45.26464	0.00124	

Fuente: Elaboración propia, basada en los datos de Bellman y col. [26].

El modelo híbrido aplicado utilizó una red neuronal programada en VBA, en dicho programa se pueden incrementar el número

de neuronas ocultas, ver Anexo E. Los resultados obtenidos de la red híbrida se graficaron en la Figura 9.

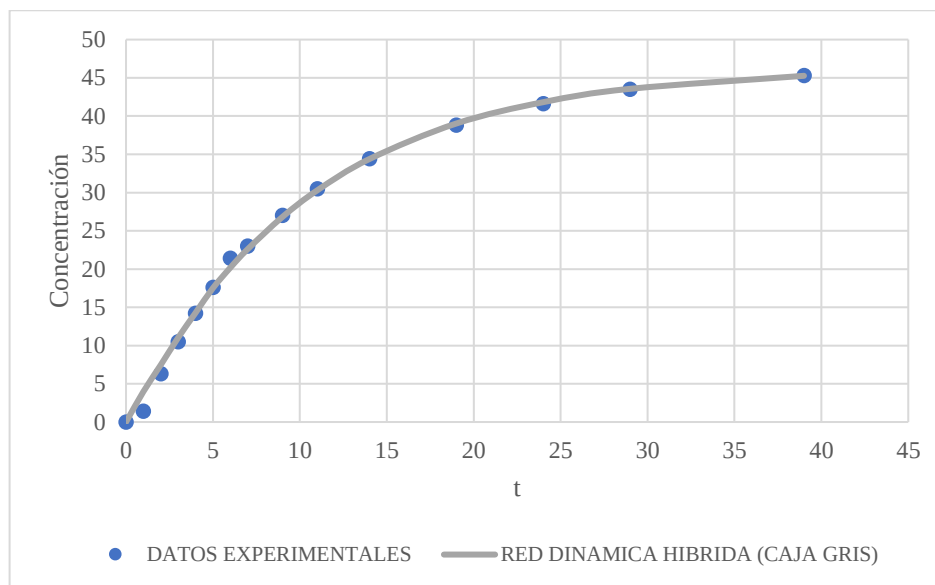


Figura 9. Representación gráfica de los resultados experimentales y la simulación de la red neuronal híbrida (caja gris).

Fuente: Elaboración propia.

IV. Discusión

Los casos de estudio tratados son una representación general de las metodologías más comúnmente usadas en ingeniería química que se relacionan con el uso de redes neuronales, las redes utilizadas en estos casos de estudio engloban las redes neuronales *feedforward*, las más comúnmente utilizadas para la descripción de fenómenos no lineales en estado estacionario, en este tipo de redes se presentan algunos ejemplos de aplicación en ingeniería química-bioquímica, estas redes se utilizan para descripción de sistemas, construcción de modelos empíricos que tienen múltiples interacciones que son difíciles de identificar. Las redes recursivas, explicadas también en este estudio permiten la descripción en periodos cortos de tiempo de sistemas dinámicos que tienen una relación directa con los datos anteriores, este tipo de redes se utiliza mucho para sistemas de reacciones y sistemas de comportamiento dinámico. La red neuronal de dinámica

diferencial es un tipo especial de red neuronal que representa comportamientos dinámicos más complejos y es especialmente útil en sistemas cíclicos; en este caso la función de la ecuación diferencial se sustituye por una red neuronal, la cual al realizar el ajuste de datos permite capturar el comportamiento global del sistema. Finalmente, las redes híbridas que combinan el comportamiento dinámico empírico con una función fenomenológica conocida, la parte de la red neuronal captura la información que modifique el comportamiento fenomenológico con una función empírica.

V. Conclusiones

El propósito principal de este trabajo fue mostrar de manera concisa y ejemplificada las estructuras y metodologías del uso de redes neuronales utilizando un software que facilite la comprensión didáctica de la estructura de una red neuronal y sus diferentes aplicaciones. Las formas

analizadas en este trabajo se centran en redes estáticas, redes recursivas, redes dinámicas y redes híbridas (caja gris), permitiendo observar claramente la pedagogía en el uso de las redes neuronales. Este artículo introduce los conceptos básicos para el uso de redes programadas con métodos más avanzados, y con base en este conocimiento, amplía su aplicación a redes dinámicas para describir modelos de ecuaciones diferenciales parciales, con discretizaciones de diferencias finitas o métodos de colocación ortogonal.

Se concluye que, con ejercicios simples y programación en VBA, es posible programar redes neuronales didácticas usando herramientas de hoja de cálculo. Además, se observa que las diferentes metodologías de redes neuronales tienen aplicaciones específicas que permiten el análisis de diversos problemas en ingeniería química y bioquímica. Por ejemplo, las redes estáticas son excelentes para describir fenómenos en estado estacionario o ecuaciones algebraicas con una fuerte influencia de ruido. Las redes recursivas son útiles para modelos en estado dinámico con comportamiento monotónico y predicciones en rangos específicos. Por otro lado, las redes dinámicas recrean modelos continuos y cíclicos, permitiendo un rango de extrapolación y capturando comportamientos de bifurcación que otros modelos de redes no pueden realizar. Finalmente, los modelos híbridos combinan redes neuronales con ecuaciones fenomenológicas para mejorar la predicción de los datos. Se espera que las herramientas presentadas en este artículo permitan mejorar la comprensión en el uso de redes neuronales para los ingenieros químicos y bioquímicos interesados en la caracterización de modelos complejos o con la presencia de ruido, haciendo hincapié en que las metodologías planteadas son extrapolables fácilmente a otras ramas de la ciencia e ingeniería.

VI. Agradecimientos

Los autores agradecen al Tecnológico Nacional de México su agenda para fomentar la investigación científica y tecnológica.

VII. Conflictos de intereses

Los autores manifiestan que no existen conflictos de interés.

VIII. Referencias

- [1] E. Esche, J. Weigert, G. Brand Rihm, J. Göbel, and J.-U. Repke, "Architectures for neural networks as surrogates for dynamic systems in chemical engineering", *Chemical Engineering Research and Design*, vol. 177, pp. 184–199, Jan. 2022, doi: 10.1016/j.cherd.2021.10.042.
- [2] M. Mowbray *et al.*, "Machine learning for biochemical engineering: A review", *Biochem Eng J*, vol. 172, no. May, p. 108054, 2021, doi: 10.1016/j.bej.2021.108054.
- [3] D. M. Himmelblau, "Applications of artificial neural networks in chemical engineering". *Korean Journal of Chemical Engineering*, vol. 17, no. 4, pp. 373–392, Jul. 2000, doi: 10.1007/BF02706848.
- [4] J. Panerati, M. A. Schnellmann, C. Patience, G. Beltrame, and G. S. Patience, "Experimental methods in chemical engineering: Artificial neural networks-ANNs", *Canadian Journal of Chemical Engineering*, vol. 97, no. 9, pp. 2372–2382, 2019, doi: 10.1002/cjce.23507.
- [5] A. Sato, Z. Sha, and S. Palosaari, "Neural networks for chemical engineering unit operations", *Chem Eng Technol*, vol. 22, no. 9, pp. 732–739, 1999, doi: 10.1002/(SICI)1521-4125(199909)22:9<732::AID-CEAT732>3.0.CO;2-1.
- [6] M. Pirdashti, S. Curteanu, M. H. Kamangar, M. H. Hassim, and M. A.

- Khatami, "Artificial neural networks: Applications in chemical engineering", *Reviews in Chemical Engineering*, vol. 29, no. 4, pp. 205–239, 2013, doi: 10.1515/revce-2013-0013.
- [7] J. S. Almeida, "Predictive non-linear modeling of complex data by artificial neural networks", *Curr Opin Biotechnol*, vol. 13, no. 1, pp. 72–76, 2002, doi: 10.1016/S0958-1669(02)00288-4.
- [8] D. M. Himmelblau, "Accounts of experiences in the application of artificial neural networks in chemical engineering", *Ind Eng Chem Res*, vol. 47, no. 16, pp. 5782–5796, 2008, doi: 10.1021/ie800076s.
- [9] A. Aglodiya, "Application of Artificial Neural Network (ANN) in Chemical Engineering: A Review", *International Journal of Advance Research and Innovative Ideas in Education*, vol. 3, no. 2, pp. 5322–5328, 2017, [Online]. Available: http://ijariie.com/AdminUploadPdf/Application_of_Artificial_Neural_Network__ANN__in_Chemical_Engineering_A_Review_ijariie5013.pdf
- [10] M. N. Karim and S. L. Rivera, "Artificial neural networks in bioprocess state estimation". *Adv Biochem Eng Biotechnol*, vol. 46, pp. 1–33, 1992, doi: 10.1007/bfb0000703.
- [11] K. M. Desai, S. A. Survase, P. S. Saudagar, S. S. Lele, and R. S. Singhal, "Comparison of artificial neural network (ANN) and response surface methodology (RSM) in fermentation media optimization: Case study of fermentative production of scleroglucan". *Biochem Eng J*, vol. 41, no. 3, pp. 266–273, 2008, doi: 10.1016/j.bej.2008.05.009.
- [12] J. D. Olden and D. A. Jackson, "Illuminating the 'black box': a randomization approach for understanding variable contributions in artificial neural networks", *Ecol Modell*, vol. 154, no. 1–2, pp. 135–150, Aug. 2002, doi: 10.1016/S0304-3800(02)00064-9.
- [13] R. Rico-Martínez, K. Krischer, I. G. Kevrekidis, M. C. Kube, and J. L. Hudson, "Discrete- vs. Continuous-time nonlinear signal processing of Cu electrodisolution data", *Chem Eng Commun*, vol. 118, no. 1, pp. 25–48, Nov. 1992, doi: 10.1080/00986449208936084.
- [14] Q. Xiong and A. Jutan, "Grey-box modelling and control of chemical processes", *Chem Eng Sci*, vol. 57, no. 6, pp. 1027–1039, 2002, doi: 10.1016/S0009-2509(01)00439-0.
- [15] T. Muneer and S. Ivanova, "Excel-VBA", in *Springer Nature*, vol. 4, no. 1, Cham: Springer International Publishing, 2022, pp. 143–160. doi: 10.1007/978-3-030-94085-0.
- [16] W. Fone, "Using a familiar package to demonstrate a difficult concept", *ACM SIGCSE Bulletin*, vol. 33, no. 3, pp. 165–168, Sep. 2001, doi: 10.1145/507758.377675.
- [17] B. E. Segee and M. D. Amos, "Artificial Neural Networks using Microsoft Excel for Windows 95", *ASEE Annual Conference Proceedings*, 1997, doi: 10.18260/1-2-6424.
- [18] S. Demir, N. M. Demir, A. Karadeniz, and H. C. Yörüklü, "Implementation of an MS Excel Tool for Backpropagation Neural Network Algorithm in Environmental Engineering Education". *Sigma: Journal of Engineering & Natural Sciences / Mühendislik ve Fen Bilimleri Dergisi*, vol. 36, no. 1, pp. 251–260, 2018, [Online]. Available: <https://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=133819222&site=ehost-live&scope=site>

- [19] J. García *et al.*, “Hojas de cálculo para la simulación de redes neuronales artificiales (RNA)”, *Qüestió: Quaderns d’Estadística i Investigació Operativa*, vol. 26, no. 1–2, pp. 289–305, 2002, [Online]. Available: <https://idus.us.es/handle/11441/41001>
- [20] T. Muneer and S. Ivanova, “Artificial Neural Networks in Excel”, in *Excel-VBA*, Cham: Springer International Publishing, 2022, pp. 143–162. doi: 10.1007/978-3-030-94085-0_9.
- [21] R. González-García, R. Rico-Martínez, and I. G. Kevrekidis, “Identification of distributed parameter systems: A neural net based approach”, *Comput Chem Eng*, vol. 22, pp. S965–S968, Mar. 1998, doi: 10.1016/S0098-1354(98)00191-4.
- [22] R. Rico-Martinez, J. S. Anderson, and I. G. Kevrekidis, “Continuous-time nonlinear signal processing: a neural network based approach for gray box identification”, in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, IEEE, 1994, pp. 596–605. doi: 10.1109/NNSP.1994.366006.
- [23] J. E. Gallot, M. P. Kapoor, and S. Kaliaguine, “Kinetics of 2-hexanol and 3-hexanol oxidation reaction over TS-1 catalysts”, *AIChE Journal*, vol. 44, no. 6, pp. 1438–1454, Jun. 1998, doi: 10.1002/aic.690440621.
- [24] Fard Masoumi, H. R., Basri, M., Kassim, A., Abdullah, D. K., Abdollahi, Y., Gani, S. S. A., & Rezaee, M., “Optimization of process parameters for lipase-catalyzed synthesis of esteramines-based esterquats using wavelet neural network (WNN) in 2-liter bioreactor”, *Journal of Industrial and Engineering Chemistry*, vol. 20, no. 4, pp. 1973–1976, Jul. 2014, doi: 10.1016/j.jiec.2013.09.019.
- [25] V. Dua, “An Artificial Neural Network approximation based decomposition approach for parameter estimation of system of ordinary differential equations”, *Comput Chem Eng*, vol. 35, no. 3, pp. 545–553, Mar. 2011, doi: 10.1016/j.compchemeng.2010.06.005.
- [26] R. Bellman, J. Jacquez, R. Kalaba, and S. Schwimmer, “Quasilinearization and the estimation of chemical rate constants from raw kinetic data”, *Math Biosci*, vol. 1, no. 1, pp. 71–76, Mar. 1967, doi: 10.1016/0025-5564(67)90027-2.

ANEXO A

Se tiene un tanque agitado perfectamente en estado estacionario como se ilustra en la Figura 1A, para demostrar la robustez de las redes neuronales en la predicción en ingeniería.

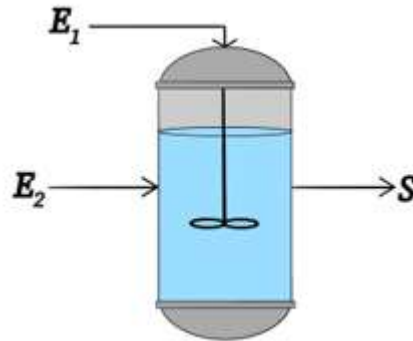


Figura 1A. Esquema del balance de masa en un tanque agitado.

El balance de masa total en estado estacionario es $E_1 + E_2 = S$, se va a construir una red neuronal simple con 2 neuronas de entrada E_1 y E_2 , una capa con 5 neuronas ocultas (O) y una neurona de salida S , cuya estructura se muestra en la Figura 2A.

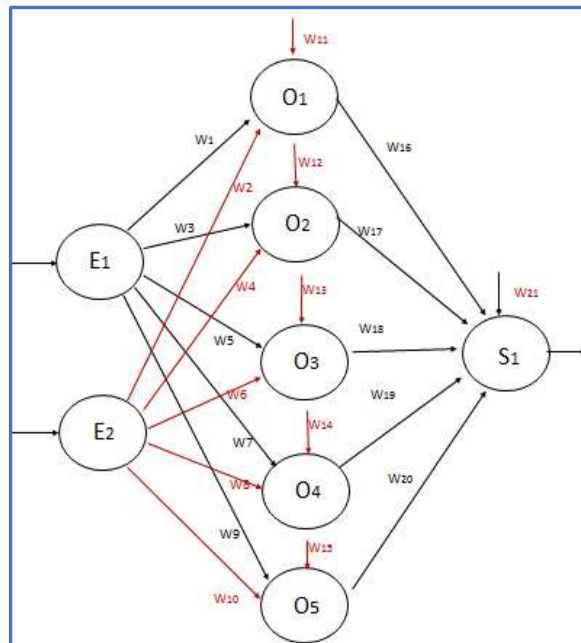


Figura 2A. Esquema de la red neuronal estática para simular el balance de masa en el tanque agitado.

Las ecuaciones de la red neuronal de la Figura 2A se muestran en la Tabla A1, donde se han separado las ecuaciones para cada neurona oculta (O), la posterior activación (A) con la función *Tanh* y la suma de activaciones para construir la salida de la red neuronal. Los factores de peso (w_1 - w_{10} ; w_{16} - w_{20}) y los factores de sesgo (bias) (w_{11} - w_{15} ; w_{21}), son los parámetros indeterminados del sistema a modelar.

Tabla A1. Ecuaciones algebraicas para la formación del modelo matemático.

Neuronas ocultas	Activación	Salida red neuronal
$O_1 = w_1 * E_1 + w_2 * E_2 + w_{11}$ $O_2 = w_3 * E_1 + w_4 * E_2 + w_{12}$ $O_3 = w_5 * E_1 + w_6 * E_2 + w_{13}$ $O_4 = w_7 * E_1 + w_8 * E_2 + w_{14}$ $O_5 = w_9 * E_1 + w_{10} * E_2 + w_{15}$	$A_1 = Tanh(O_1)$ $A_2 = Tanh(O_2)$ $A_3 = Tanh(O_3)$ $A_4 = Tanh(O_4)$ $A_5 = Tanh(O_5)$	$S = w_{16} * A_1 + w_{17} * A_2 +$ $w_{18} * A_3 + w_{19} * A_4 +$ $w_{20} * A_5 + w_{21}$

La ecuación completa que representa la red neuronal que va a representar el balance de masa en el tanque agitado es:

$$S = w_{16} * (Tanh(w_1 * E_1 + w_2 * E_2 + w_{11})) + w_{17} * (Tanh(w_3 * E_1 + w_4 * E_2 + w_{12})) + w_{18} * (Tanh(w_5 * E_1 + w_6 * E_2 + w_{13})) + w_{19} * (Tanh(w_7 * E_1 + w_8 * E_2 + w_{14})) + w_{20} * (Tanh(w_9 * E_1 + w_{10} * E_2 + w_{15})) + w_{21} \quad \text{Ec. A1}$$

El siguiente paso es entrenar la red neuronal, para lo cual se manufacturaron 30 datos (E_1 , E_2 y S que están en la Tabla A2), teniendo en mente que, entre más datos experimentales se dispongan, se pueden construir redes neuronales más complejas que van a tener una mejor capacidad predictiva. A continuación, se construye un esquema de mínimos cuadrados, proporcionando valores aleatorios entre 0 y 1 al vector de pesos y sesgos (w_1 - w_{21}). Mediante la herramienta SOLVERTM de EXCEL[®], se minimiza la suma cuadrática de residuos, cambiando los valores del vector w_i . Cuando se logra convergencia, el vector w_1 - w_{21} adquiere los valores reales de los factores de peso y sesgo, que se muestran en la Tabla A3. Cabe mencionar que pueden existir varios vectores solución para w_i , por ser ecuaciones no lineales, situación que debe discutirse en cursos avanzados. La Tabla A2 también muestra los valores de salida S predichos por la red neuronal y el correspondiente error relativo, donde se aprecia la excelente capacidad de predicción de las redes neuronales, lo que es crucial cuando se simulan fenómenos complejos.

Tabla A2. Ecuaciones algebraicas para la formación del modelo matemático.

E_1	E_2	S	Red neuronal (ec. A1)	Residuos cuadráticos	Error relativo (%)
0.6	0.2	0.8	0.80004	0.00000	0.0053
0.5	0.3	0.8	0.80004	0.00000	0.0055
0.4	0.4	0.8	0.80005	0.00000	0.0057
0.5	0.1	0.6	0.59999	0.00000	0.0009
0.6	0.3	0.9	0.90004	0.00000	0.0046
0.3	0.55	0.85	0.85005	0.00000	0.0063
0.7	0.3	1	0.99998	0.00000	0.0017
0.3	0.33	0.66	0.63000	0.00090	4.7619
0.45	0.25	0.7	0.70002	0.00000	0.0027
0.5	0.2	0.7	0.70002	0.00000	0.0026
0.2	0.4	0.6	0.59999	0.00000	0.0010
0.3	0.31	0.61	0.61000	0.00000	0.0007
0.4	0.6	1	0.99999	0.00000	0.0006

E ₁	E ₂	S	Red neuronal (ec. A1)	Residuos cuadráticos	Error relativo (%)
0.5	0.45	0.95	0.95003	0.00000	0.0027
0.3	0.7	1	1.00000	0.00000	0.0003
0.4	0.23	0.63	0.63000	0.00000	0.0000
0.25	0.25	0.5	0.50000	0.00000	0.0005
0.6	0.12	0.72	0.72002	0.00000	0.0032
0.4	0.12	0.52	0.51999	0.00000	0.0010
0.45	0.12	0.57	0.56999	0.00000	0.0014
0.8	0.13	0.93	0.93003	0.00000	0.0027
1	0.1	1.1	1.09983	0.00000	0.0154
0.9	0.1	1	0.99997	0.00000	0.0025
0.5	0.6	1.1	1.09986	0.00000	0.0132
0.5	0.23	0.78	0.73003	0.00250	6.8453
0.4	0.12	0.52	0.51999	0.00000	0.0010
0.05	0.9	0.95	0.95004	0.00000	0.0040
0.14	0.15	0.29	0.29022	0.00000	0.0765
0.76	0.14	0.9	0.90004	0.00000	0.0041
0.33	0.67	1	1.00000	0.00000	0.0004
					0.3925
Suma de Residuos cuadráticos				0.0033974	

Tabla A3. Valores de los factores de peso y sesgo al concluir el entrenamiento.

Wi	1	2	3	4	5	6	7
valor	0.3044415	0.5275673	0.1188476	0.1191174	0.1188042	0.1120577	0.1201172
Wi	8	9	10	11	12	13	14
valor	0.1327205	0.1259825	0.1256501	0.1915595	-0.1647334	0.0797581	0.0817723
Wi	15	16	17	18	19	20	21
valor	0.0238920	-0.0010635	5.4016611	1.2049950	0.5784136	1.2521969	0.7104792

Al tener la red entrenada, se debe valorar la capacidad predictiva de la Ec. A1, algunos ejemplos se muestran en la Tabla A4, donde se aprecia la excelente capacidad de predicción, que es importante como paso para el control y diseño de procesos en ingeniería. Inclusive, algunos datos están fuera del rango de valores de E₁, E₂ que se utilizaron en el entrenamiento, obteniéndose errores relativos aceptables. Esta capacidad predictiva depende del sistema a modelar, de la red elegida, de la cantidad de datos experimentales disponibles y del entrenamiento de la red, por lo que debe valorarse para cada situación en particular.

Tabla A4. Capacidad predictiva de la red neuronal.

E ₁	E ₂	Experimental	Predicción de la red entrenada	Error Relativo (%)
0.22	0.11	0.33	0.33015	0.04546
0.51	0.22	0.73	0.73003	0.00371
0.22	0.1	0.32	0.32017	0.05222
1.34	0.78	2.12	2.10738	0.59547
2.11	0.44	2.55	2.52178	1.10670
3.47	0.99	4.46	4.23263	5.09802
1.43	1.65	3.08	3.02034	1.93716
0.02	0.05	0.07	0.07100	1.43341

Repitiendo el proceso de entrenamiento para 3 neuronas ocultas (que producen 13 factores de peso) y utilizando los mismos datos de la Tabla A4, la capacidad predictiva de la red neuronal disminuye, dando un error relativo promedio de 12.39096%. También se puede observar que la capacidad de extrapolación de la red disminuye significativamente, como es de esperarse.

Tabla A5. Capacidad predictiva de la red neuronal con 3 neuronas ocultas.

E ₁	E ₂	Experimental	Predicción de la red entrenada	Error Relativo (%)
0.22	0.11	0.33	0.32846	0.46653
0.51	0.22	0.73	0.73492	0.67363
0.22	0.1	0.32	0.31803	0.61584
1.34	0.78	2.12	1.92130	9.37279
2.11	0.44	2.55	2.17959	14.52599
3.47	0.99	4.46	2.73314	38.71893
1.43	1.65	3.08	2.39122	22.36299
0.02	0.05	0.07	0.06541	6.56112
Promedio				12.39096

Ahora se evalúa con 7 neuronas ocultas (que producen 29 factores de peso) y se vuelve a entrenar la red con los 30 datos de la Tabla A2. La capacidad predictiva de la red se muestra en la Tabla A6, donde se observa una ligera disminución del error relativo promedio, que se explica dado que los factores de peso son casi iguales al número de datos para entrenamiento, lo que produce un ajuste deficiente.

Tabla A6. Capacidad predictiva de la red neuronal con 7 neuronas ocultas.

E ₁	E ₂	Experimental	Predicción de la red entrenada	Error Relativo (%)
0.22	0.11	0.33	0.32805	0.59049
0.51	0.22	0.73	0.73509	0.69737
0.22	0.1	0.32	0.31760	0.75001
1.34	0.78	2.12	1.92359	9.26468
2.11	0.44	2.55	2.17785	14.59409
3.47	0.99	4.46	2.78756	37.49856
1.43	1.65	3.08	2.41753	21.50866
0.02	0.05	0.07	0.06314	9.80589
Promedio				12.12912

Por último, si se entrena la red neuronal con las 7 neuronas ocultas y empleando 100 datos experimentales (incluidos los 30 datos originales), los resultados se muestran en la Tabla A7.

Tabla A7. Capacidad predictiva de la red neuronal con 7 neuronas ocultas y 100 datos de entrada.

E_1	E_2	Experimental	Predicción de la red entrenada	Error Relativo (%)
0.22	0.11	0.33	0.32998	0.00653
0.51	0.22	0.73	0.73003	0.00401
0.22	0.1	0.32	0.31998	0.00735
1.34	0.78	2.12	2.11908	0.04326
2.11	0.44	2.55	2.54334	0.26110
3.47	0.99	4.46	4.14008	7.17299
1.43	1.65	3.08	3.04955	0.98876
0.02	0.05	0.07	0.07004	0.06069
			Promedio	1.21200

Donde se ha recuperado la capacidad predictiva de la red neuronal, aunque la red más eficiente para este caso de estudio sería la de 5 neuronas ocultas. Análisis similares deberán hacerse para cada problema en particular para diseñar una red neuronal óptima.

ANEXO B

Option Explicit

Option Base 1

Public Const aa As Integer = 31 'número de pesos

Function redsim(xp1 As Double, xp2 As Double, xp3 As Double, xp4 As Double)

Application.Volatile

Dim k As Integer

Dim A(aa) As Double

Dim O(10) As Double

Dim T(10) As Double

'Lectura de parámetros

For k = 1 To aa

A(k) = Cells(k, 1)

Next k

'Operaciones de la red

'Operaciones de interconexión en la entrada

O(1) = xp1 * A(1) + xp2 * A(2) + xp3 * A(3) + xp4 * A(4) + A(5)

O(2) = xp1 * A(6) + xp2 * A(7) + xp3 * A(8) + xp4 * A(9) + A(10)

O(3) = xp1 * A(11) + xp2 * A(12) + xp3 * A(13) + xp4 * A(14) + A(15)

O(4) = xp1 * A(16) + xp2 * A(17) + xp3 * A(18) + xp4 * A(19) + A(20)

O(5) = xp1 * A(21) + xp2 * A(22) + xp3 * A(23) + xp4 * A(24) + A(25)

'Operaciones con la función de transferencia

T(1) = 1 / (1 + Exp(-O(1)))

T(2) = 1 / (1 + Exp(-O(2)))

T(3) = 1 / (1 + Exp(-O(3)))

T(4) = 1 / (1 + Exp(-O(4)))

T(5) = 1 / (1 + Exp(-O(5)))

'Función de salida

redsim = T(1) * A(26) + T(2) * A(27) + T(3) * A(28) + T(4) * A(29) + T(5) * A(30) + A(31)

End Function

ANEXO C

$$S(1) = T(1) * A(26) + T(2) * A(27) + T(3) * A(28) + T(4) * A(29) + T(5) * A(30) + A(31)$$

$$S(2) = T(1) * A(32) + T(2) * A(33) + T(3) * A(34) + T(4) * A(35) + T(5) * A(36) + A(37)$$

redsim = S 'REGRESO DE LA RED COMO UN ARREGLO USAR SELECCIONAR, F2 Y
CONTRO+SHIFT+ENTER (COMO LA MATRIZ).

ANEXO D

Option Explicit 'DECLARAR TODAS LAS VARIABLES

Option Base 1 'EL INDICE INICIA EN 1

Public Const no As Integer = 2 'NUMERO DE NEURONAS DE SALIDA

Public Const nn As Integer = 3 'NUMERO DE NEURONAS EN LAS CAPAS OCULTAS

Public Const ni As Integer = 2 'NUMERO DE NEURONAS DE ENTRADA

Function REDNE2(xp1 As Double, xp2 As Double, tipo As Integer)

Application.Volatile 'INSTRUCCION IMPORTANTE PARA ACTUALIZAR

Dim t(3, nn, nn) As Double

Dim teta(3, nn) As Double

Dim xp(4, nn) As Double

Dim anet(3, nn) As Double

Dim xi(1, ni) As Double

Dim i, j, k, jp, lp, l, kp, cont, ncg As Integer

Dim HTan, temp As Double

ncg = 2 * nn + no + ((ni + nn + no) * nn)

'LECTURA DE LOS FACTORES DE PESO

cont = 0

For i = 1 To 3

If (i = 3) Then

jp = no

Else

jp = nn

End If

For j = 1 To jp

If (i = 1) Then

kp = ni

Else

kp = nn

End If

For k = 1 To kp

cont = cont + 1

temp = Cells(cont, 1)

temp = Range("A" + Trim(Str(cont))).Value

t(i, j, k) = temp

Next k

cont = cont + 1

temp = Cells(cont, 1)

teta(i, j) = Range("A" + Trim(Str(cont))).Value

teta(i, j) = temp

Next j

Next i

'BLOQUE DE LA RED NEURONAL'

xp(1, 1) = xp1 'LECTURA DE LA ENTRADA

xp(1, 2) = xp2

For k = 2 To 4

 If (k = 4) Then

 jp = no

 Else

 jp = nn

 End If

 For j = 1 To jp

 If (k = 2) Then

 lp = ni

 Else

 lp = nn

 End If

 xp(k, j) = teta((k - 1), j)

 For l = 1 To lp

 xp(k, j) = xp(k, j) + t((k - 1), j, l) * xp((k - 1), l)

 Next l

 If (k <> 4) Then

 anet((k - 1), j) = xp(k, j)

 'FUNCION EMPLEADA EN ESTE CASO TANGENTE HIPERBÓLICA

HTan = (Exp(anet((k - 1), j)) - Exp(-anet((k - 1), j))) / (Exp(anet((k - 1), j)) + Exp(-anet((k - 1), j)))

 'HTan = 1 / (1 + Exp(-anet((k - 1), j)))

 xp(k, j) = HTan

 End If

 Next j

Next k

'REGRESO DEL VALOR OBTENIDO POR LA RED

If (tipo = 1) Then

 REDNE2 = xp(4, 1)

 ElseIf (tipo = 2) Then

 REDNE2 = xp(4, 2)

 Else

 REDNE2 = 0

End If

End Function

ANEXO E

Option Explicit

Option Base 1

Public Const aa As Integer = 31 'numero de pesos $NI*NN + NN*NS + NN + NS$

Public Const nn As Integer = 10 'numero de nodos de la red

Public Const ni As Integer = 1 'numero de nodos de entrada

Public Const ns As Integer = 1 'numero de nodos salida

Function redsim(xp1 As Double)

Application.Volatile

Dim k, i, j As Integer

Dim A(aa) As Double

Dim O(nn) As Double

Dim T(nn) As Double

Dim cont As Integer

Dim suma1 As Double

Dim ymodel(ns) As Double

Dim x(ni) As Double

'Lectura de parámetros

For k = 1 To aa

A(k) = Cells(k, 1)

Next k

'Neuronas de entrada igual al número ni

x(1) = xp1

'Operaciones de la red

cont = 1

For i = 1 To nn

suma1 = 0

For j = 1 To ni

suma1 = suma1 + x(j) * A(cont)

cont = cont + 1

Next j

O(i) = suma1

Next i

For i = 1 To nn

O(i) = O(i) + A(cont)

'T(i) = 1 / (1 + Exp(-O(i)))

'T(i) = 2 / (1 + Exp(-2 * O(i))) - 1

T(i) = (Exp(O(i)) - Exp(-O(i))) / (Exp(O(i)) + Exp(-O(i)))

cont = cont + 1

Next i

suma1 = 0

```
For i = 1 To ns
    For j = 1 To nn
        suma1 = suma1 + T(j) * A(cont)
        cont = cont + 1
    Next j
    ymodel(i) = suma1
Next i
```

```
For i = 1 To ns
    ymodel(i) = ymodel(i) + A(cont)
    cont = cont + 1
Next i
```

```
'Función de salida
redsim = ymodel(1)
End Function
```