



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO

FACULTAD DE CIENCIAS DE LA INGENIERÍA

CARRERA DE INGENIERÍA EN ELECTRICIDAD

**Proyecto de investigación previo
a la obtención del título de
Ingeniero Eléctrico**

Título del Proyecto de Investigación:

**“PREDICCIÓN DE DEMANDA ELÉCTRICA UTILIZANDO REDES NEURONALES
ARTIFICIALES PARA CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO”**

Autores:

Alvaro Leonel Mora Silva

Stalin Fabian Toaquiza Chango

Director del Proyecto de Investigación:

Ing. Diego Patricio Peña Banegas

Quevedo - Los Ríos - Ecuador

2021

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS



Yo, Stalin Fabian Toaquiza Chango, declaro que la investigación aquí descrita es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este documento, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

f. _____
Stalin Fabian Toaquiza Chango
C.C. # 050425942-5

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS



Yo, Alvaro Leonel Mora Silva, declaro que la investigación aquí descrita es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Universidad Técnica Estatal de Quevedo, puede hacer uso de los derechos correspondientes a este documento, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

f. _____
Alvaro Leonel Mora Silva
C.C. # 050394650-1

CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN



El suscrito, Ing. Diego Patricio Peña Banegas, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el estudiante Stalin Fabian Toaquiza Chango, realizó el Proyecto de Investigación de grado titulado “PREDICCIÓN DE DEMANDA ELÉCTRICA UTILIZANDO REDES NEURONALES ARTIFICIALES PARA CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO”, previo a la obtención del título de Ingeniero Eléctrico, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

.....

Ing. Diego Patricio Peña Banegas

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN



El suscrito, Ing. Diego Patricio Peña Banegas, Docente de la Universidad Técnica Estatal de Quevedo, certifica que el estudiante Alvaro Leonel Mora Silva, realizó el Proyecto de Investigación de grado titulado “PREDICCIÓN DE DEMANDA ELÉCTRICA UTILIZANDO REDES NEURONALES ARTIFICIALES PARA CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO”, previo a la obtención del título de Ingeniero Eléctrico, bajo mi dirección, habiendo cumplido con las disposiciones reglamentarias establecidas para el efecto.

.....

Ing. Diego Patricio Peña Banegas

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO

Sr. Ing. Washington Chiriboga Casanova, MSc.

DECANO DE LA FACULTAD CIENCIAS DE LA INGENIERÍA

En su despacho.

De mi consideración.

En calidad de director del Trabajo de investigación titulado: “**PREDICCIÓN DE DEMANDA ELÉCTRICA UTILIZANDO REDES NEURONALES ARTIFICIALES PARA CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO.**”, me permito manifestar a usted lo siguiente:

Los señores **Alvaro Leonel Mora Silva** y **Stalin Fabian Toaquiza Chango**; estudiantes de la carrera de Ingeniería en Electricidad modalidad presencial, han cumplido con las correcciones pertinentes e ingresado su trabajo de investigación al sistema URKUND, tengo a bien certificar que el informe refleja un porcentaje favorable del 2%, cumpliendo con el reglamento de graduación de Estudiantes de Pregrado y la Normativa establecida por la Universidad Técnica Estatal de Quevedo.

Por la aprobación que se sirva de dar a la presente, quedo ante usted muy agradecido.



Documento	tesis 12.0.docx (D101550447)
Presentado	2021-04-13 22:43 (-05:00)
Presentado por	dpena@uteq.edu.ec
Recibido	dpena.uteq@analysis.arkund.com
Mensaje	Tesis Mostrar el mensaje completo

2% de estas 33 páginas, se componen de texto presente en 1 fuentes.

Atentamente



Firmado electrónicamente por:

DIEGO
PATRICIO
PENA
BANEGAS

Ing. Diego Patricio Peña Banegas, MSc.

DIRECTOR DEL PROYECTO DE INVESTIGACIÓN

CERTIFICADO DE APROBACIÓN POR TRIBUNAL DE SUSTENTACIÓN



UNIVERSIDAD TÉCNICA ESTATAL DE QUEVEDO FACULTAD DE CIENCIAS DE LA INGENIERÍA CARRERA DE INGENIERÍA EN ELECTRICIDAD

PROYECTO DE INVESTIGACIÓN

Título:

“PREDICCIÓN DE DEMANDA ELÉCTRICA UTILIZANDO REDES NEURONALES ARTIFICIALES PARA CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO”

Presentado al consejo directivo como requisito previo a la obtención del título de Ingeniero Eléctrico.

Aprobada por:

PRESIDENTE DEL TRIBUNAL

ING. WILLIAM MENDIETA

MIEMBRO DEL TRIBUNAL

ING. OSMAR VIERA

MIEMBRO DEL TRIBUNAL

ING. YADIRA ORTIZ

Quevedo – Los Ríos – Ecuador
2021

AGRADECIMIENTO

Agradecer a Dios por cuidarnos siempre y proteger a nuestras familias durante esta pandemia.

A mis padres que me han ayudado en todo este proceso de estudio, permitiéndome alcanzar una meta más en la vida.

A mis hermanos que siempre han estado presentes para ayudarme a seguir adelante.

A todas y todos los docentes que a lo largo de esta etapa universitaria han puesto dedicación al impartir sus conocimientos, valores y experiencias que nos forjan a ser buenos profesionales.

Alvaro Leonel Mora Silva.

AGRADECIMIENTO

En primer lugar, quiero agradecer a Dios por todas las bendiciones que ha puesto en mi camino durante esta etapa de mi vida, mostrándome que el esfuerzo, compañerismo y constancia tienen su recompensa.

Agradecer a mis padres por todo el sacrificio que día a día han hecho por mí para darme la oportunidad de convertirme en un profesional con principios de moral y ética.

A mis primos más cercanos que siendo ellos ya profesionales me dieron el ejemplo y los consejos necesarios para no desmayar en todo este camino, en especial a mi prima Johana que nunca dejó de confiar en mí.

A todos los docentes que con su conocimiento y consejos supieron guiarme para alcanzar este anhelado objetivo.

Stalin Fabián Toaquiza Chango.

DEDICATORIA

Dedicado para toda mi familia que siempre me alentó y brindó toda la ayuda posible para que pueda cumplir este anhelado sueño, de manera especial a mis padres que me apoyaron bajo toda circunstancia.

Alvaro Leonel Mora Silva.

DEDICATORIA

Este logro se lo dedico a mis padres y en especial a mi hermana, esperando que le sirva como fuente de motivación y así logre alcanzar todos sus objetivos académicos y personales en el transcurso de su vida.

Stalin Fabian Toaquiza Chango.

RESUMEN.

Este trabajo de investigación tiene como finalidad realizar la proyección de la demanda de energía eléctrica mediante un modelo de Redes Neuronales Artificiales (RNA). Para desarrollar esta proyección se utilizan datos de consumos históricos (demanda máxima mensual) entregados por la empresa eléctrica CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO en el periodo de tiempo 2003 - 2008. La proyección de la demanda de energía eléctrica es fundamental para poder garantizar principalmente la continuidad del suministro eléctrico generando confiabilidad en los consumidores. En este proyecto se pone a prueba un modelo de RNA, las cuales han venido desarrollándose con mayor frecuencia a lo largo del tiempo. El procesamiento de los datos se realiza a través de 3 capas que forman parte de la RNA, estas capas consisten en: capa de entrada, oculta y salida, las cuales mediante procesos matemáticos entregan valores continuos que representan el resultado. El desarrollo de los resultados se divide en etapas, las cuales consisten en: entrenamiento de la RNA con el 85% de los datos histórico, prueba de la RNA bajo el criterio de evaluación Error Porcentual Absoluto Medio (MAPE) entre el 15% de los valores no utilizados en el entrenamiento frente a la proyección para dicha cantidad de históricos, esta etapa permite demostrar la confiabilidad del modelo en base al porcentaje de error, y así realizar una proyección con mayor horizonte de tiempo.

Palabras Claves: RNA, retropropagación, demanda eléctrica, descenso de gradiente, MAPE

ABSTRACT

The purpose of this research is to perform the projection of the demand for electrical energy by using an Artificial Neural Networks (ANN) model. To develop this projection, historical consumption data (maximum monthly demand) provided by the electricity company CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO are used from the period 2003 - 2008. The projection of electricity demand is essential to guarantee mainly the continuity of the electricity supply generating reliability in consumers. In this project, a model of ANNs is tested, which has been developed with greater frequency over time. The data processing is done through 3 layers that are part of the ANN, these layers consist of: input layer, hidden layer and output layer, which through mathematical procedures delivers continuous values that represent the result. The development of the results is divided into stages, which consist of: ANN training with 85% of the historical data, ANN test under the evaluation standard Mean Absolute Percentage Error (MAPE) between 15% of the Values not used in the training versus the projection for historical data, this stage allows demonstrating the reliability of the model based on the percentage error, and thus making a projection with a longer time horizon.

Keywords: ANN, backpropagation, electrical demand, gradient decrease, MAPE

TABLA DE CONTENIDO

PORTADA	i
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS	ii
CERTIFICACIÓN DE CULMINACIÓN DEL PROYECTO DE INVESTIGACIÓN.....	iv
CERTIFICADO DEL REPORTE DE LA HERRAMIENTA DE PREVENCIÓN DE COINCIDENCIA Y/O PLAGIO ACADÉMICO.....	vi
CERTIFICADO DE APROBACIÓN POR TRIBUNAL DE SUSTENTACIÓN	vii
AGRADECIMIENTO	viii
DEDICATORIA.....	x
RESUMEN.	xii
ABSTRACT	xiii
CÓDIGO DUBLIN.....	xviii
INTRODUCCIÓN.....	1
1. CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN.....	2
1.1. Problema de investigación.	3
1.1.1. Planteamiento del problema.	3
1.1.2. Formulación del problema.....	4
1.1.3. Sistematización del problema.....	4
1.2. Objetivos.....	5
1.2.1. Objetivo general.	5
1.2.2. Objetivos específicos.....	5
1.3. Justificación.	6
2. FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN	7
2.1. Marco conceptual.....	8
2.1.1. Análisis de series temporales.....	8
2.1.2. Neurona biológica.....	9
2.1.3. Neurona artificial.....	10
2.1.4. Red Neuronal Artificial.	11
2.1.5. Función de activación.....	13
2.1.6. Capas de una Red Neuronal Artificial.....	15
2.1.7. Clasificación de las Redes Neuronales Artificiales.....	16
2.1.8. Perceptrón multicapa.	17
2.1.9. Propagación hacia adelante (feedforward).	21

2.1.10.	Algoritmo backpropagation (propagación hacia atrás).	21
2.1.11.	Evaluación del modelo.	32
2.1.12.	Determinación de arquitectura y aplicación.	34
2.1.13.	Proyección en un paso de tiempo.	34
2.1.14.	Proyección en múltiples pasos de tiempo.	36
2.2.	Marco Referencial.	38
3.	METODOLOGÍA DE LA INVESTIGACIÓN	40
3.1.	Localización.	41
3.2.	Tipo de investigación.	41
3.2.1.	Investigación bibliográfica.	41
3.2.2.	Investigación explicativa.	41
3.3.	Método de investigación.	41
3.3.1.	Método analítico.	41
3.3.2.	Método inductivo.	42
3.4.	Fuentes de recopilación de información.	42
3.5.	Diseño de investigación.	42
3.6.	Instrumentos de investigación.	42
3.7.	Tratamiento de los datos.	43
3.8.	Recursos humanos y materiales.	43
4.	RESULTADOS Y DISCUSIÓN	45
4.1.	Resultados.	46
4.1.1.	Base de datos.	46
4.1.2.	Preparación de datos.	48
4.1.2.1.	Normalización de datos.	48
4.1.2.2.	Conjuntos de entrenamiento y de prueba.	49
4.1.3.	Proceso de entrenamiento.	50
4.1.4.	Proceso de prueba.	53
4.1.5.	Proyección.	56
4.1.6.	Implementación de la Red Neuronal Artificial.	58
4.1.6.1.	Diagramas de flujo de la Red Neuronal Artificial.	58
4.1.6.2.	Desarrollo del programa en MATLAB.	62
4.1.7.	Diseño de la Red Neuronal Artificial.	65
4.1.8.	Validación de los modelos.	73
4.1.8.1.	Error de proyección.	75

4.2. Discusión.....	76
5. CONCLUSIONES Y RECOMENDACIONES.....	78
5.1. Conclusiones.....	79
5.2. Recomendaciones.	80
6. BIBLIOGRAFÍA.....	81
7. ANEXOS.....	85

ÍNDICE DE FIGURAS

Figura 1: Neurona biológica.....	9
Figura 2: Neurona biológica y neurona artificial.....	10
Figura 3: Modelo estándar de neurona artificial.....	11
Figura 4: Partes de una Red Neuronal Artificial.	12
Figura 5: Función lineal.....	13
Figura 6: Función tangente sigmoide.	14
Figura 7: Capas de una Red Neuronal Artificial.	16
Figura 8: Arquitectura perceptrón simple.....	18
Figura 9: Arquitectura perceptrón multicapa.....	19
Figura 10: Modelo de proyección en un paso de tiempo.....	36
Figura 11: Modelo de proyección en múltiples pasos de tiempo.	37
Figura 12: Curva de consumo de potencia eléctrica de 72 meses en escala original.	48
Figura 13: Curva de consumo de potencia eléctrica de 72 meses normalizados.....	49
Figura 14: Resultado de entrenamiento de una red sobreentrenada.	52
Figura 15: Resultado de entrenamiento con 3 neuronas de entrada.	53
Figura 16: Proyección para la prueba de la RNA sobreentrenada.....	55
Figura 17: Proyección para la prueba de la RNA con 3 neuronas de entrada.	56
Figura 18: Prueba y proyección de la RNA sobreentrenada.	57
Figura 19: Prueba y proyección de la RNA con 3 neuronas de entrada.	58
Figura 20: Diagrama de flujo 1 del proceso de entrenamiento.....	60
Figura 21: Diagrama de flujo 2 del proceso de proyección.....	61
Figura 22: Arquitectura del modelo 1.....	66

Figura 23: Arquitectura del modelo 2.....	67
Figura 24: Proceso de entrenamiento de los modelos 1 y 2.	69
Figura 25: Decrecimiento del error de entrenamiento de los modelos 1 y 2.....	70
Figura 26: Prueba y proyección del modelo 1 y 2 para N = 100.....	72
Figura 27: Proceso de entrenamiento, prueba y proyección para los modelos 1 y 2 con N = 36.	73

ÍNDICE DE TABLAS

Tabla 1: Recursos humanos.....	43
Tabla 2: Recursos de hardware.....	43
Tabla 3: Recursos de software.....	44
Tabla 4: Recursos de materiales.	44
Tabla 5: Valores del consumo de potencia eléctrica mensual del periodo 2003 – 2008.....	46
Tabla 6: División de datos para entrenamiento y prueba.	50
Tabla 7: Matriz de entrada y salida para el entrenamiento.....	51
Tabla 8: Datos históricos usados en el proceso de prueba.	54
Tabla 9: Características de los modelos 1 y 2.	66
Tabla 10: Parámetros de proyección de los modelos 1 y 2.	68
Tabla 11: Error cuadrático medio obtenido en MW del entrenamiento de los modelos 1 y 2.	70
Tabla 12: Proyección normalizada y desnormalizada de los modelos 1 y 2.	74
Tabla 13: Resultado del cálculo del MAPE para los modelos 1 y 2.	75

CÓDIGO DUBLIN.

Título:	PREDICCIÓN DE DEMANDA ELÉCTRICA UTILIZANDO REDES NEURONALES ARTIFICIALES PARA CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO				
Autor:	Mora Silva Alvaro Leonel, Toaquiza Chango Stalin Fabian				
Palabras clave:	RNA	backpropagation	demanda eléctrica	descenso de gradiente	MAPE
Fecha de publicación:					
Editorial:	Quevedo, Universidad Técnica Estatal de Quevedo, 2021				
Resumen:	<p>Este trabajo de investigación tiene como finalidad realizar la proyección de la demanda de energía eléctrica mediante un modelo de Redes Neuronales Artificiales (RNA). Para realizar esta proyección se utilizan datos de consumos históricos (demanda máxima mensual) entregados por la empresa eléctrica CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO en el periodo de tiempo 2003 - 2008. (...)</p> <p>The purpose of this research is to perform the projection of the demand for electrical energy by using an Artificial Neural Networks (ANN) model. To develop this projection, historical consumption data (maximum monthly demand) provided by the electricity company CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO are used from the period 2003 - 2008. (...)</p>				
Descripción:	108 hojas. Dimensiones, 29 x 21 cm + CD-ROM				
URI:	--				

INTRODUCCIÓN

La demanda eléctrica es la suma de los consumos individuales de los diferentes clientes pertenecientes al sistema de distribución, los cuales, varían según diversos factores que caracterizan dicho consumo. Los clientes residenciales mantienen un consumo muy variado de acuerdo a las horas del día; los clientes industriales y comerciales varían el consumo de energía según las jornadas de trabajo. Éstos y otros factores como la época del año, festividades, feriados, clima, entre otros, provocan que el consumo de energía eléctrica varíe en el tiempo.

El consumo de energía eléctrica a lo largo de los años indica el aumento de la demanda y con esto se debe repotenciar el sistema de distribución para cumplir con los parámetros que deben garantizar la confiabilidad, continuidad, seguridad, niveles de tensión y frecuencia a todos los clientes que están conectados al sistema de distribución. Realizar la proyección de la demanda eléctrica es sumamente importante para el sistema de distribución, permitiendo cumplir con el objetivo de entregar la energía eléctrica necesaria a todos los consumidores.

La empresa eléctrica CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO mantiene control de las proyecciones de demanda. Se realizan por medio de métodos estadísticos con el objetivo de llevar a cabo una adecuada distribución de energía aplicando planes de operación y expansión con antelación. En este trabajo de investigación se utiliza una RNA como un método alternativo para realizar la proyección de demanda, haciendo uso del software MATLAB para llevar a cabo su funcionamiento.

El proyecto se desarrolla en cinco capítulos en los que se mencionan los temas más relevantes dirigidos a la proyección con RNA. Estos son: funciones de activación, topología de RNA, procesamiento de datos históricos, entrenamiento, prueba y proyección.

CAPÍTULO I
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1.1. Problema de investigación.

1.1.1. Planteamiento del problema.

El correcto funcionamiento del Sistema Nacional Interconectado (SNI) implica que todos los agentes que participan en él, tales como la generación, transmisión y distribución, estén siempre interconectados y comunicados. Es así que ante la presencia de una falla el sistema pueda responder adecuadamente direccionando recursos para aislarla y corregirla.

El constante crecimiento poblacional y desarrollo de la ciudad provoca el incremento de la demanda de energía eléctrica, siendo la empresa de distribución la responsable de abastecer este recurso de forma continua y estable al menor costo. Por ello, está en la necesidad de desarrollar planes de expansión que reduzcan los costos de inversión, cubriendo el total de la demanda requerida, considerando que una inadecuada proyección genera problemas operativos y mal dimensionamiento de elementos ocasionando pérdidas para la empresa.

Dada la importancia de conocer el comportamiento futuro de la demanda de energía eléctrica en la empresa de distribución, la cual tiene entre sus objetivos mantener un adecuado plan de expansión y correcta operación del sistema, en el presente proyecto se realiza la proyección de demanda eléctrica mediante el uso de RNA, ampliamente usado en problemas de proyección de demanda, por los resultados confiables que ofrece.

Diagnóstico.

Ningún método logra realizar la proyección de la demanda de manera precisa debido a que el consumo de energía eléctrica de la ciudad varia en el tiempo y aumenta con el desarrollo de la población. Sin embargo, es necesario el uso de un método que se aproxime lo más posible a la realidad. El uso de un método u otro varía dependiendo de la complejidad de los datos para realizar la proyección de demanda.

Pronóstico.

No contar con una correcta proyección de la demanda de energía eléctrica que consume la ciudad de Quevedo, provincia de Los Ríos, en Ecuador, afecta directamente al creciente desarrollo de la ciudad, así también, perjudica a la empresa distribuidora disminuyendo la confiabilidad del sistema. Esto puede llegar a comprometer a los equipos e instalaciones necesarias para entregar la energía a los hogares.

1.1.2. Formulación del problema.

¿Cómo se podrá proyectar la demanda de energía eléctrica en la ciudad Quevedo?

1.1.3. Sistematización del problema.

¿Cuál es el comportamiento que presenta el consumo de electricidad en la ciudad de Quevedo?

¿De qué manera se va a proyectar la demanda de energía eléctrica en el periodo de tiempo estimado?

¿De qué forma se comprueba que el método brinda una proyección aceptable considerando el porcentaje de error obtenido en la etapa de prueba?

¿Cuál sería el comportamiento a futuro de la demanda de energía eléctrica en la ciudad de Quevedo?

1.2. Objetivos.

1.2.1. Objetivo general.

Implementar un método de proyección de demanda eléctrica mensual usando redes neuronales artificiales para la empresa de distribución eléctrica CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO.

1.2.2. Objetivos específicos.

- Analizar los datos de consumo de energía eléctrica en la ciudad de Quevedo.
- Diseñar una red neuronal artificial para la proyección de demanda eléctrica mensual utilizando el software MATLAB.
- Comprobar los resultados de proyección de demanda eléctrica de la red neuronal con los valores reales entregados por CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO.
- Realizar la proyección de demanda para un horizonte de tiempo fuera de los datos históricos.

1.3. Justificación.

La importancia de una correcta proyección de demanda de energía eléctrica radica en que ésta se ajuste a los valores reales. La aplicación de cualquier método no garantiza esta finalidad; sin embargo, es importante contar con uno que demuestre ser efectivo. Por ello se plantea el uso de RNA para llevar a cabo la tarea de proyección de demanda. En los sistemas de distribución la correcta proyección de la demanda resulta útil en la toma de decisiones y operación del sistema.

El presente trabajo consiste en realizar la proyección de demanda de energía eléctrica considerando el valor del porcentaje de error obtenido en la etapa de prueba, que a su vez depende del proceso de entrenamiento. Las etapas de entrenamiento y prueba se realizan con los datos entregados por CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO. La efectividad del método depende del porcentaje de error resultante en la etapa de prueba.

Las RNA se asemejan al funcionamiento de una red neuronal biológica, es decir, se encadenan múltiples neuronas dentro de capas, una a continuación de otra. Cada neurona genera una respuesta de salida que se entrega a las neuronas de la siguiente capa, las que procesan los datos entregándolos a las neuronas de la última capa dando el resultado final. El entrenamiento es un proceso en el que una parte del total de datos son ingresados para que la red identifique y aprenda los patrones presentes en la serie de tiempo. Este proceso busca ajustar el valor de pesos y bias que sirven para realizar la proyección después del proceso de entrenamiento.

Con el uso de RNA se espera obtener resultados aceptables dado que su algoritmo de aprendizaje supervisado permite modificar los parámetros mediante ensayos de prueba y error hasta lograr un modelo que dé como resultado el menor porcentaje de error en la etapa de prueba.

CAPÍTULO II

FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN

2.1. Marco conceptual.

2.1.1. Análisis de series temporales.

Es necesario conocer el comportamiento de las series temporales, debido a que podrían tener un comportamiento continuo o discreto. Una serie temporal continua es aquella que puede tomar un valor cualquiera de múltiples observaciones posibles en un intervalo de tiempo y que se registran de manera constante [1]. Una serie temporal discreta es aquella que toma valores específicos en un intervalo regular pudiendo ser en días, meses, años, etc. [2].

Los principales objetivos del análisis de una serie temporal son [1]:

- a) Descripción: Se puede presentar un resumen estadístico o también describir el conjunto mediante una gráfica, la cual resulta de gran ayuda al momento de entender su comportamiento.
- b) Modelado: Es una tarea que permite describir el modelo generado por el proceso.
- c) Pronóstico: Consiste básicamente en proyectar valores futuros de la serie temporal.
- d) Control: Haber realizado un correcto pronóstico permite tomar decisiones de control sobre el proceso determinado.

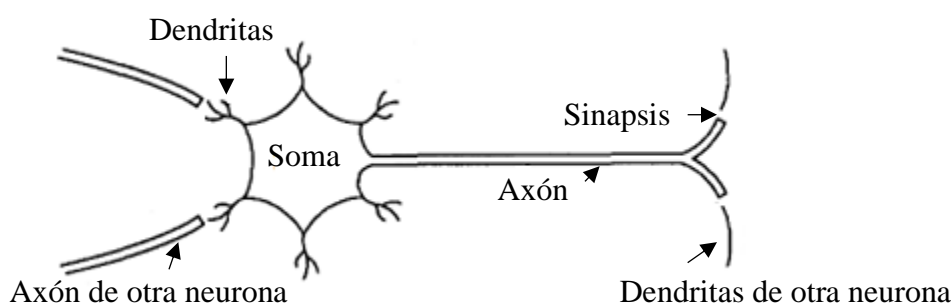
Los componentes de una serie temporal son [2]:

- a) Tendencia: Describe el comportamiento de la serie a través del tiempo y si presenta un crecimiento o decrecimiento.
- b) Estacionalidad: Hace referencia a fluctuaciones cíclicas que se repiten cada cierto tiempo, es decir, patrones que se repiten.
- c) Error: Representa las fluctuaciones a corto plazo que normalmente son impredecibles, también es conocido como ruido o componente irregular.

2.1.2. Neurona biológica.

Debido a que las RNA están basadas en la estructura del cerebro e imitan el comportamiento de las redes neuronales biológicas es conveniente exponer la estructura de las neuronas biológicas. Existen diferentes tipos de neuronas biológicas, pero se va a considerar la estructura más común, como se muestra en la figura 1 [5].

Figura 1: Neurona biológica.



Fuente: [5].

Elaboración: Autores.

Esta neurona biológica está formada por un cuerpo central llamado Soma que contiene el núcleo celular; una prolongación del Soma, llamada Axón; una ramificación terminal, llamada Dendritas; y una zona de conexión entre neuronas, llamada Sinapsis.

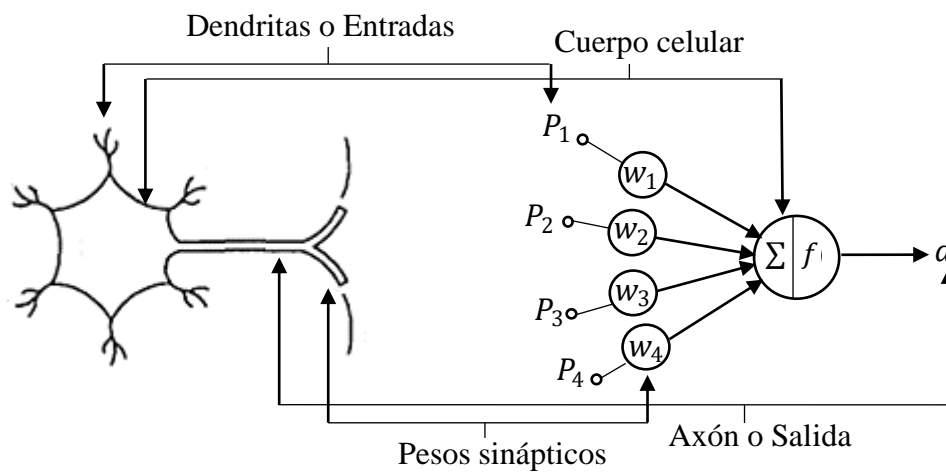
La función de las neuronas biológicas es la de transmitir impulsos nerviosos que recorren por todo el sistema neuronal. Este recorrido empieza por las Dendritas, pasa por el axón y finalmente es transferida a otra neurona por medio de la sinapsis. En términos generales, se puede decir que las neuronas biológicas reciben señales de entrada, el cuerpo celular las combina e integra y emite una señal de salida. Las RNA intentan imitar este proceso [7].

2.1.3. Neurona artificial.

Una neurona artificial es un elemento de cálculo simple que, a partir de un conjunto o vector de datos de entrada y/o a partir de estímulos de neuronas adyacentes, emite una respuesta. [3],[4].

Una neurona artificial tiene mucha similitud con una neurona biológica como se muestra en la figura 2.

Figura 2: Neurona biológica y neurona artificial.



Fuente: [10].

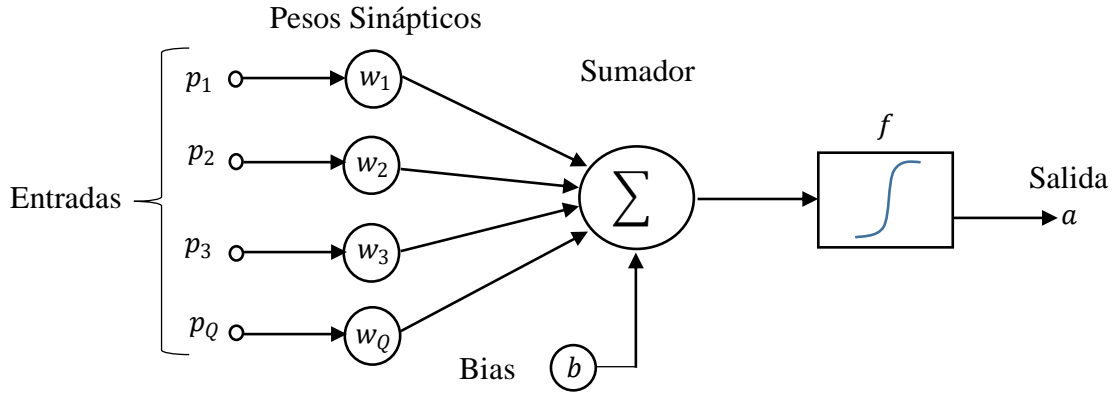
Elaboración: Autores.

En toda neurona se tiene [6]:

- Señales de entrada provenientes del exterior receptoras por las dendritas.
- Conjunto de conexiones o pesos sinápticos.
- Un cuerpo celular o sumador donde se procesa las señales de entrada.
- Función de activación que limita la amplitud de salida de la neurona.
- Salida que es entregada a las siguientes neuronas o al entorno.

Para comprender de mejor manera la estructura de la neurona artificial, en la figura 3 se indica su modelo estándar. A la neurona artificial se le añade un bias que actúa como otro peso.

Figura 3: Modelo estándar de neurona artificial.



Fuente: [7].
Elaboración: Autores.

De esta estructura podemos destacar los siguientes elementos:

- Entrada: p_1, p_2, \dots, p_Q .
- Pesos sinápticos: w_1, w_2, \dots, w_Q , a cada entrada le corresponde un peso.
- Bias: b , determina el valor por encima del cual la neurona se activa.
- Proceso de sumatoria de las señales de entrada.
- Función de activación: f .
- Salida de la neurona artificial: a .

La señal de salida de la neurona artificial viene dada por la ecuación (1):

$$a = f \left(\sum_{i=1}^Q (w_i \times p_i) + b \right) \quad (1)$$

2.1.4. Red Neuronal Artificial.

Una RNA está inspirada en las redes neuronales biológicas e intentan imitar su comportamiento [3], [5]. Las RNA han venido tomando importancia en las últimas décadas ya que permiten

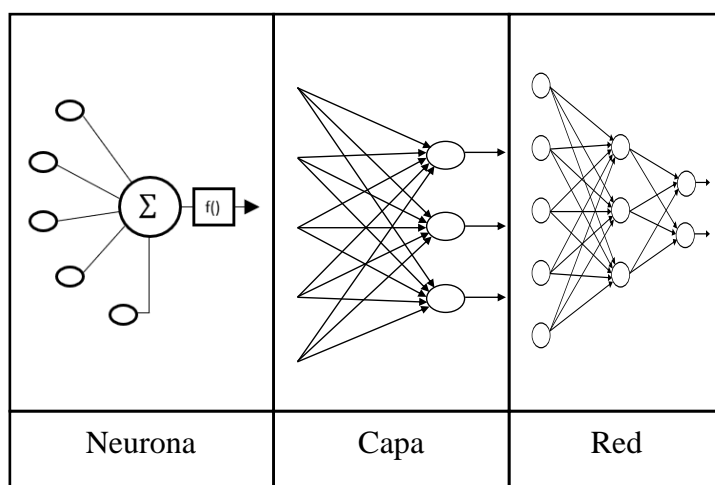
resolver problemas con gran cantidad de datos. La propiedad más importante es su capacidad de aprender a partir de un conjunto de patrones de entrenamiento [2].

Las aplicaciones más exitosas de las RNA son: procesamiento de imágenes y voz, reconocimiento de patrones, interfaces adaptativos para sistemas hombre-maquina, proyecciones, control, optimización y filtrado de señales [10].

Para que una RNA imite el comportamiento de una red neuronal biológica, debe tener una estructura jerarquizada como se muestra en la figura 4, de tal manera que permite generar un procesador de señales, formando conexiones entre ellas de tal forma que estas puedan recibir señales del entorno o bien de otra neurona, generando una salida que pueda seguir propagándose en la red [7].

La RNA está compuesta por capas dirigidas a realizar una tarea específica, es así que, la capa de entrada recibe las señales del entorno, la capa oculta procesa la información y la capa de salida entrega el resultado al entorno. Estas a su vez se componen de múltiples neuronas interconectadas entre ellas. Sin embargo, la RNA aún no está completa, debido a que hace falta realizar el entrenamiento utilizando algún algoritmo de aprendizaje para este fin.

Figura 4: Partes de una Red Neuronal Artificial.



Fuente: [7].
Elaboración: Autores.

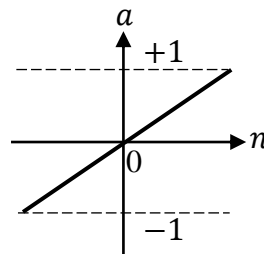
2.1.5. Función de activación.

La función de activación combina la entrada y pesos recibidos con el estado actual de la neurona para producir un nuevo estado de activación o desactivación acorde a la información recibida [3]. Esta función limita la amplitud de salida de la neurona. Normalmente el estado de las neuronas consiste en valores de -1 a 0 en estado no activo y 0 a 1 para estado activo [4]. Las funciones de activación utilizadas son la función tangente sigmoide y la función lineal.

a) Función Lineal.

La función de activación lineal permite obtener una única salida en base a varias entradas, siendo muy útil en los casos de regresión al generar un único valor en la salida. La RNA configurada con una única neurona con función lineal en la última capa, permite realizar aproximaciones numéricas de funciones.

Figura 5: Función lineal.



Fuente: [12].

Elaboración: Autores.

Esta función está dada por la siguiente ecuación:

$$a = n \quad (2)$$

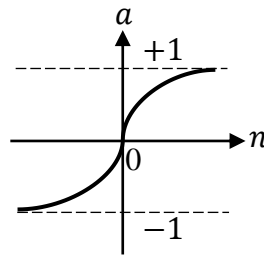
Donde:

- n : valores de entrada.
- a : resultado de la función.

b) Función Tangente Sigmoide.

Esta es una de las funciones más usadas dentro de las redes neuronales artificiales, debido a su flexibilidad de amplitud, siendo del intervalo de -1 a 1 como se observa en la figura 6, esta función adapta valores próximos a cero cuando n es pequeña con la ventaja de ampliar los resultados a la parte negativa. A medida que n se aleja del cero, la función adapta los valores hasta 1 o menos 1.

Figura 6: Función tangente sigmoide.



Fuente: [12].

Elaboración: Autores.

Esta función se expresa por la siguiente ecuación:

$$a = \frac{2}{1 + e^{-2n}} - 1 \quad (3)$$

Donde:

- n : valores de entrada.
- a : resultado de la función.

2.1.6. Capas de una Red Neuronal Artificial.

a) Capa de entrada.

Es la capa que contiene las neuronas que reciben los datos de entrada. Se establece que los datos que ingresen deben tener algún método estadístico que cambie la escala original evitando que sean difíciles de procesar por la RNA [9]. Algunos autores no consideran a esta sección como una capa, debido a que solo se ingresa datos de entrada y no se lleva a cabo ningún proceso.

b) Capa oculta.

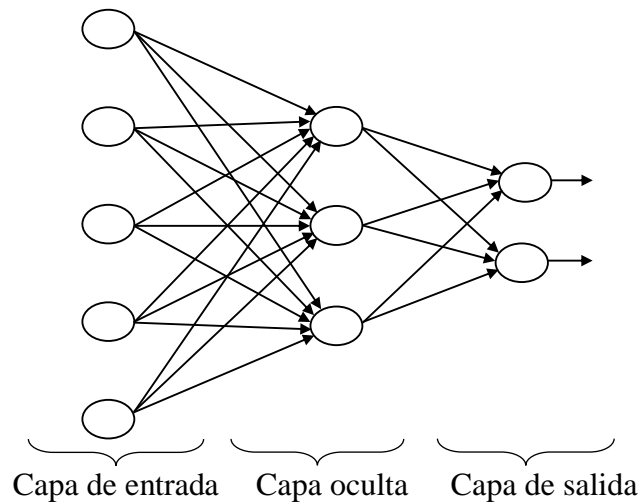
La o las capas ocultas no tienen contacto con el exterior y son las que determinan la topología de la red. La determinación del número de neuronas de la capa oculta es muy importante debido a que afecta directamente al desempeño de la red. El número de neuronas se determinan mediante ensayos de prueba y error [10].

c) Capa de salida.

Corresponde a la o las neuronas que entregan el resultado al exterior o hacia otras neuronas [9]. El número de neuronas de la última capa depende de la finalidad de la red, si la respuesta está dirigida al exterior es necesario convertir los valores a su escala original y así poderlos presentar de manera correcta y entendible.

La figura 7 presenta un esquema de RNA con tres capas diferentes, en la capa de entrada se presentan 5 neuronas receptoras de los elementos del exterior, la capa oculta presenta 3 neuronas para procesamiento de los valores y la capa de salida presenta 2 neuronas que entregan los resultados al exterior.

Figura 7: Capas de una Red Neuronal Artificial.



Fuente: [4].
Elaboración: Autores.

2.1.7. Clasificación de las Redes Neuronales Artificiales.

a) Por su arquitectura.

Este tipo de clasificación considera los niveles de capas, clasificando a una red como [7]:

- Monocapa: cuando consta de una sola capa de entrada conectada a una única capa de salida donde se procesan los datos y entregan resultados.
- Multicapa: muy similar a la anterior, pero con la inserción de una o varias capas ocultas que se encuentran entre la capa de entrada y salida.
- Recurrente: Este tipo de arquitectura se caracteriza porque existen lazos de realimentación, estos lazos pueden ser entre las neuronas de diferentes capas o dentro de una misma capa.

b) Tipo de aprendizaje.

La RNA utiliza un vector de datos del exterior del cual aprende, en la mayoría de casos la forma en la que se hace uso de este vector de datos caracteriza el tipo de aprendizaje [6]:

- Aprendizaje supervisado: consiste en mostrar a la salida de la RNA un valor deseado con el cual comparar, con la finalidad de calcular un error y modificar los bias y pesos sinápticos.
- Aprendizaje no supervisado: este tipo de aprendizaje no tiene un valor de comparación a la salida, este tipo RNA tiene la capacidad de autoajustar sus pesos sin necesidad de un agente externo que controle el desarrollo de la misma [6].

c) Según su aplicación.

La aplicación de las RNA puede variar según el caso de estudio. Existen configuraciones con las cuales se podría realizar [6]:

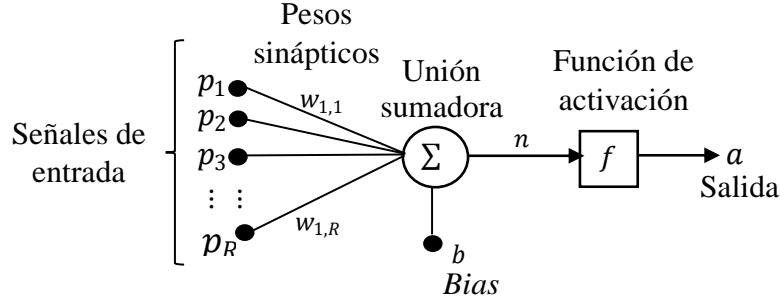
- Clasificación: hacen uso de cualquiera de los dos aprendizajes, es decir, supervisado o no supervisado, con el fin de modificar los pesos sinápticos, en base a las entradas del problema.
- Proyección: el objetivo es determinar la salida de una función en base a los datos históricos.

2.1.8. Perceptrón multicapa.

Es posible tratar problemas no lineales gracias a la unión de varios perceptrones simples, es decir, agregar capas ocultas a la red, esto da como resultado la arquitectura perceptrón multicapa [13]. Actualmente el uso de la arquitectura perceptrón multicapa es la más usada en la solución de problemas, debido a que, se lo considera un aproximador universal de funciones [4].

El tipo de arquitectura perceptrón simple intenta imitar el comportamiento de la neurona biológica, y a diferencia del perceptrón multicapa consta únicamente de señales de entrada, un punto de sumatoria y la función de salida para generar una respuesta al estímulo [7]. Las ecuaciones de la figura 8 serán representadas con una matriz W y un vector P , con la finalidad de representar de manera abreviada las operaciones.

Figura 8: Arquitectura perceptrón simple.



Fuente: [9].

Elaboración: Autores.

Se denota a W como una matriz de pesos que corresponde a $w_{1,1}, w_{1,2}, \dots, w_{1,R}$ donde los subíndices señalan la primera neurona y su interconexión con las entradas p_1, p_2, \dots, p_R respectivamente, P en este caso representa el vector de entrada y n es el resultado del producto de la matriz W por el vector P mas el bias b como se expresa en la ecuación (4).

$$n = \sum_{i=1}^R (p_i \times w_{1,i}) + b \quad (4)$$

Empleando matrices, la entrada neta n se expresa como:

$$n = P \times W + b \quad (5)$$

La señal de salida a de la neurona es el resultado de aplicar la función de activación f a la entrada neta n como se muestra en la ecuación (6).

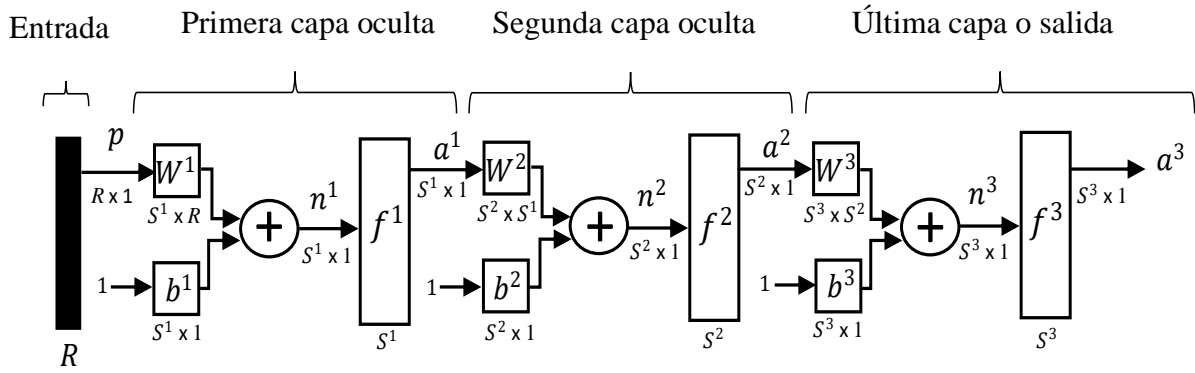
$$a = f(n) \quad (6)$$

También expresado como:

$$a = f(W \times P + b) \quad (7)$$

La arquitectura del perceptrón multicapa se caracteriza por tener neuronas agrupadas en capas de diferentes niveles, en la figura 9 se distinguen tres capas diferentes en notación abreviada [13]: capa de entrada, capas ocultas y capa de salida. Las neuronas de la capa de entrada no actúan como una capa propiamente dicha, sino que, reciben señales del exterior y las propagan hacia las demás neuronas de la capa oculta, mientras que la neurona de salida entrega la respuesta de la RNA. Los superíndices de W , b , S , n , f y a indican el número de la capa a la cual pertenecen, de igual manera, se define el tamaño de los vectores: bias, entradas netas y salidas como: número de neuronas de la capa (filas) por el número de entradas (columnas).

Figura 9: Arquitectura perceptrón multicapa.



Fuente: [11].

Elaboración: Autores.

A continuación, se describe las matrices y vectores que conforman la figura 9.

El vector de entrada p presenta un tamaño de $R \times 1$, donde R es el número de entradas y 1 es la cantidad de columnas del vector. La matriz de pesos W^1 presenta un tamaño de $S^1 \times R$, donde S^1 es el número de neuronas de la primera capa oculta y R representa el número de entradas. Los vectores b^1 , n^1 y a^1 presentan un tamaño de $S^1 \times 1$ donde S^1 es el número de neuronas de la primera capa oculta y 1 representa un único valor de entrada. La función f^1 presenta un tamaño de S^1 , es decir, el número de neuronas en la primera capa oculta.

La matriz de pesos W^2 presenta un tamaño de $S^2 \times S^1$, donde S^2 es el número de neuronas de la segunda capa oculta y S^1 es el número de neuronas de la primera capa oculta siendo

considerado como el número de entradas. Los vectores b^2 , n^2 y a^2 presentan un tamaño de $S^2 \times 1$ donde S^2 es el número de neuronas de la segunda capa oculta y 1 representa un único valor de entrada. La función f^2 presenta un tamaño de S^2 , es decir, el número de neuronas en la segunda capa oculta.

La matriz de pesos W^3 presenta un tamaño de $S^3 \times S^2$, donde S^3 es el número de neuronas de la capa de salida y S^2 es el número de neuronas de la segunda capa oculta siendo considerado como el número de entradas. Los vectores b^3 , n^3 y a^3 presentan un tamaño de $S^3 \times 1$ donde S^3 es el número de neuronas de la capa de salida y 1 representa un único valor de entrada. La función f^3 presenta un tamaño de S^3 , es decir, el número de neuronas en la capa de salida.

Matriz de entrada y vector de salidas esperadas.

Cuando el objetivo de una RNA es aprender el comportamiento de los datos secuenciales de un vector de entrada P , la matriz se adecua de la siguiente manera:

$$P = \begin{bmatrix} p_k & p_{k+13} & \cdots & p_{k+47} \\ p_{k-1} & p_k & \cdots & p_{k+46} \\ p_{k-2} & p_{k-1} & \cdots & p_{k+45} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k-11} & p_{k-10} & \cdots & p_{k+36} \end{bmatrix} \quad (8)$$

$$T = [p_{k+1} \quad p_{k+2} \quad \cdots \quad p_{k+48}] \quad (9)$$

El vector de salidas T está conformado por los datos de la misma secuencia, pero son representados como t debido a que se tratan de las salidas esperadas, por lo tanto, se denota como:

$$T = [t_{k+1} \quad t_{k+2} \quad \cdots \quad t_{k+48}] \quad (10)$$

Donde k representa un instante de tiempo seleccionado como actual, para esta representación establecido como 12.

El vector de salidas T está conformado por los datos de la misma secuencia, representados como t debido a que se tratan de las salidas esperadas u objetivo.

2.1.9. Propagación hacia adelante (feedforward).

La propagación de la información desde la capa de entrada hacia la capa de salida se considera como propagación hacia adelante, para ello se debe encontrar la salida de cada una de las capas y en el caso de una red multicapa la salida de una capa se convierte en la entrada de la siguiente.

La salida de cada una de las capas se obtiene a partir de las siguientes ecuaciones:

$$a^1 = f^1 (W^1 \times p + b^1) \quad (11)$$

$$a^2 = f^2 (W^2 \times a^1 + b^2) \quad (12)$$

$$a^3 = f^3 (W^3 \times a^2 + b^3) \quad (13)$$

Donde:

- p : vector de entrada.
- a^1, a^2, a^3 : vector de salidas de cada una de las capas.
- f^1, f^2, f^3 : funciones de activación.
- W^1, W^2, W^3 : matrices de pesos.
- b^1, b^2, b^3 : vectores de bias.

2.1.10. Algoritmo backpropagation (propagación hacia atrás).

Es un tipo de entrenamiento supervisado que hace uso del método del descenso del gradiente para realizar la propagación del error entre las salidas deseadas y las proyectadas a partir de un conjunto de datos de entrada, actualizando los pesos de las neuronas en las diferentes capas, este proceso se lo realiza en dos fases [11]:

- 1) Se ingresa el vector de entrada a la red y se propaga por las capas de la red emitiendo una señal de salida. Esta salida se compara con la salida deseada y se calcula el error cometido por la neurona de salida, este error se propaga hacia atrás hasta las neuronas de la capa oculta.
- 2) A cada neurona se le atribuye un error proporcional a su contribución en el error total de la red, y basados en este error, se ajustan los pesos sinápticos y bias de cada neurona.

El método de entrenamiento backpropagation es usado para obtener una combinación correcta de pesos de una RNA de tal manera que se minimice el error. La base de esta minimización está en la retropropagación del error entre la capa de salida y la capa oculta de tal manera que se encuentre las derivadas parciales necesarias para minimizar este error [16].

Índice de rendimiento.

El algoritmo de retropropagación utiliza el índice de rendimiento: error cuadrático medio. El algoritmo se alimenta con un conjunto de ejemplos adecuados:

$$(p_1, t_1), (p_2, t_2), \dots, (p_R, t_R)$$

Donde:

- p_R : es la entrada a la RNA.
- t_R : es la salida esperada correspondiente al objetivo.

A medida que se aplica cada entrada, su respuesta se compara con su objetivo correspondiente, dando como resultado el error medio cuadrático (MSE) para un número total de valores M .

$$MSE = \frac{\sum_{i=1}^M (t - a)^2}{M} \quad (14)$$

Donde:

- a : respuesta estimada por cada patrón de entrada.
- t : salida esperada u objetivo.

El algoritmo debe ajustar los parámetros de la red para minimizar el error cuadrático medio:

$$F(x) = E[e^2] = E[(t - a)^2] \quad (15)$$

Donde:

- $F(x)$: representación del índice de rendimiento.
- x : representa el vector de pesos y bias.
- E : denota el valor esperado del error cuadrático e^2 .

Si la red tiene múltiples salidas, esto se generaliza de forma matricial a:

$$F(x) = E[e^T \times e] = E[(t - a)^T \times (t - a)] \quad (16)$$

El valor esperado del error cuadrático $E[e^2]$ se representa como una matriz transpuesta del error e^T que multiplica al vector de errores e , y de igual manera para $E[(t - a)^2]$.

Se aproxima el error cuadrático medio mediante:

$$\hat{F}(x) = e^T(k) \times e(k) \quad (17)$$

Donde:

- \hat{F} : es la aproximación del error cuadrático medio.
- k : número de iteración.

El valor esperado del error cuadrático $E[e^2]$ se ha sustituido por el producto de los vectores $e^T(k) \times e(k)$ para el numero de iteraciones k .

El algoritmo de descenso para el error cuadrático medio aproximado (descenso de gradiente estocástico) es:

$$w_{i,j}^m(k + 1) = w_{i,j}^m(k) - \alpha \frac{\partial \hat{F}}{\partial w_{i,j}^m} \quad (18)$$

$$b_i^m(k+1) = b_i^m(k) - \alpha \frac{\partial \hat{F}}{\partial b_i^m} \quad (19)$$

Donde:

- α : es la tasa de aprendizaje.
- b : son los bias.
- w : son los pesos.
- i : es la neurona de partida.
- j : es la neurona de llegada.
- m : es el número de la capa.

A continuación, se realiza el cálculo de las derivadas parciales.

Regla de la cadena.

Para la red multicapa el error no es una función explícita de los pesos en las capas ocultas, por lo que estas derivadas no se calculan tan fácilmente. Como el error es una función indirecta de los pesos en las capas ocultas, se utiliza la regla de la cadena para calcular las derivadas. Para repasar la regla de la cadena, se supone una función f que es una función explícita sólo de la variable n . Se toma la derivada de f con respecto a una tercera variable w . La regla de la cadena es entonces:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \times \frac{dn(w)}{dw} \quad (20)$$

Por ejemplo, si:

$$f(n) = e^n \quad (21)$$

Donde:

$$n = 2w \quad (22)$$

Para que:

$$f(n(w)) = e^{2w} \quad (23)$$

Entonces:

$$\frac{df(n(w))}{dw} = \frac{df(n)}{dn} \times \frac{dn(w)}{dw} = e^n \times 2 \quad (24)$$

Utilizaremos este concepto para encontrar las derivadas de la función de aproximación con respecto al peso y bias de las ecuaciones (18) y (19) respectivamente:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \quad (25)$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m} \quad (26)$$

El segundo término de cada una de estas ecuaciones puede calcularse fácilmente, ya que la entrada neta n a la capa m es una función explícita de los pesos y bias de esa capa:

$$n_i^m = \sum_{j=1}^{s^{m-1}} w_{i,j}^m \times a_j^{m-1} + b_i^m \quad (27)$$

Por lo tanto:

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1} \quad (28)$$

$$\frac{\partial n_i^m}{\partial b_i^m} = 1 \quad (29)$$

Si ahora se define la sensibilidad como:

$$s_i^m \equiv \frac{\partial \hat{F}}{\partial n_i^m} \quad (30)$$

Donde s es la sensibilidad de \hat{F} a los cambios en el i -ésimo elemento de entrada neta en la capa m , y representa la cantidad de error correspondiente a cada neurona i de la capa m .

Considerando la ecuación (28) y (30), la ecuación (25) puede simplificarse como:

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = s_i^m \times a_j^{m-1} \quad (31)$$

Así mismo, considerando la ecuación (29) y (30), la ecuación (26) puede simplificarse como:

$$\frac{\partial \hat{F}}{\partial b_i^m} = s_i^m \quad (32)$$

Las ecuaciones (31) y (32) serán consideradas para realizar la actualización de pesos y bias en las ecuaciones (38) y (39).

Se necesita calcular como punto de partida la sensibilidad s^m de la capa de salida:

$$s_i^m = \frac{\partial \hat{F}}{\partial n_i^m} = \frac{\partial (t - a)^T \times (t - a)}{\partial n_i^m} = \frac{\partial \sum_{j=1}^{s^m} (t_j - a_j)^2}{\partial n_i^m} = -2 \times (t_i - a_i) \times \frac{\partial a_i}{\partial n_i^m} \quad (33)$$

Ahora, como:

$$\frac{\partial a_i}{\partial n_i^m} = \frac{\partial n_i^m}{\partial n_i^m} = \frac{\partial f^m(n_i^m)}{\partial n_i^m} = f^m(n_i^m) \quad (34)$$

Donde $\dot{f}^m(n_i^m)$ representa la función derivada de la entrada neta en la capa de salida.

La sensibilidad se puede escribir como:

$$s_i^m = -2 \times (t_i - a_i) \times \dot{f}^m(n_i^m) \quad (35)$$

La ecuación (35) permite calcular las sensibilidades en la capa de salida.

Si en la última capa se tiene una neurona de salida, \dot{f}^m representará la derivada de la función de activación de esta neurona. En el caso de que exista más de una neurona en la capa de salida la sensibilidad s_i^m se expresará en forma de matriz debido a que \dot{f}^m será una matriz diagonal \dot{F}^m que representará el número de neuronas de salida con su respectiva función de activación derivada.

La sensibilidad expresada en forma matricial de la siguiente manera:

$$s_i^m = -2 \times \dot{F}^m(n^m) \times (t - a) \quad (36)$$

Donde $\dot{F}^m(n^m)$ representa la matriz diagonal de funciones de activación derivadas de las entradas netas para la capa m , y se expresa como:

$$\dot{F}^m(n^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \dots & 0 \\ 0 & \dot{f}^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \dot{f}^m(n_{s^m}^m) \end{bmatrix} \quad (37)$$

Actualización de pesos y bias en la capa de salida

Reemplazando la ecuación (31) en la ecuación (18) se obtiene la actualización de pesos de la capa de salida:

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha \times s_i^m \times a_j^{m-1} \quad (38)$$

Reemplazando la ecuación (32) en la ecuación (19) se obtiene la actualización de bias de la última neurona:

$$b_i^m(k+1) = b_i^m(k) - \alpha \times s_i^m \quad (39)$$

En forma de matriz esto se convierte en:

$$W^m(k+1) = W^m(k) - \alpha \times s^m \times (a^{m-1})^T \quad (40)$$

$$b^m(k+1) = b^m(k) - \alpha \times s^m \quad (41)$$

Donde s expresado de forma matricial equivale a:

$$s^m \equiv \frac{\partial \hat{F}}{\partial n^m} = \begin{bmatrix} \frac{\partial \hat{F}}{\partial n_1^m} \\ \frac{\partial \hat{F}}{\partial n_2^m} \\ \vdots \\ \frac{\partial \hat{F}}{\partial n_{s^m}^m} \end{bmatrix} \quad (42)$$

Retroalimentación de las sensibilidades.

Posteriormente se calcula las sensibilidades s^m , que requiere otra aplicación de la regla de la cadena, esto entrega el término de retropropagación, debido a que describe una relación de recurrencia en la que la sensibilidad en la capa m se calcula a partir de la sensibilidad en la capa $m+1$.

Para derivar la relación de recurrencia de las sensibilidades, se utiliza la siguiente matriz jacobiana:

$$\frac{\partial n^{m+1}}{\partial n^m} \equiv \begin{bmatrix} \frac{\partial n_1^{m+1}}{\partial n_1^m} & \frac{\partial n_1^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_1^{m+1}}{\partial n_{s^m}^m} \\ \frac{\partial n_2^{m+1}}{\partial n_1^m} & \frac{\partial n_2^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_2^{m+1}}{\partial n_{s^m}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_1^m} & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_2^m} & \dots & \frac{\partial n_{s^{m+1}}^{m+1}}{\partial n_{s^m}^m} \end{bmatrix} \quad (43)$$

Donde:

- n^{m+1} : entrada neta de la capa siguiente.
- n^m : entrada neta de la capa anterior.
- m : número de la capa.

A continuación, se requiere encontrar una expresión para esta matriz. Se considera i, j como elementos de la matriz:

$$\begin{aligned} \frac{\partial n_i^{m+1}}{\partial n_j^m} &= \frac{\partial (\sum_{l=1}^{s^m} w_{i,l}^{m+1} \times a_l^m + b_i^{m+1})}{\partial n_j^m} = w_{i,j}^{m+1} \times \frac{\partial a_j^m}{\partial n_j^m} = \\ &= w_{i,j}^{m+1} \times \frac{\partial f^m(n_j^m)}{\partial n_j^m} = w_{i,j}^{m+1} \times \dot{f}^m(n_j^m) \end{aligned} \quad (44)$$

Donde:

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m} \quad (45)$$

\dot{f}^m representa la derivada de la función de activación de la entrada neta con respecto a la entrada neta n^m de la capa m .

Por lo tanto, la matriz jacobiana se puede escribir:

$$\frac{\partial n^{m+1}}{\partial n^m} = W^{m+1} \times \dot{F}^m(n^m) \quad (46)$$

Donde:

\dot{F}^m se presenta como una matriz diagonal con la finalidad de multiplicar los pesos W^{m+1} con su respectiva función de activación derivada $\dot{f}^m(n_1^m), \dot{f}^m(n_2^m), \dots, \dot{f}^m(n_{s^m}^m)$.

Se escribe la relación de recurrencia para la sensibilidad utilizando la regla de la cadena en forma de matriz:

$$\begin{aligned} s^m &= \frac{\partial \hat{F}}{\partial n^m} = \left(\frac{\partial n^{m+1}}{\partial n^m} \right)^T \times \frac{\partial \hat{F}}{\partial n^{m+1}} = \dot{F}^m(n^m) \times (W^{m+1})^T \times \frac{\partial \hat{F}}{\partial n^{m+1}} = \\ &= \dot{F}^m(n^m) \times (W^{m+1})^T \times s^{m+1} \end{aligned} \quad (47)$$

La ecuación (47) realiza el cálculo de las sensibilidades en las capas ocultas, utilizando el cálculo de las sensibilidades de la capa $m + 1$. Las sensibilidades se propagan hacia atrás a través de la red desde la última capa hasta la primera:

$$s^M \rightarrow s^{M-1} \rightarrow \dots \rightarrow s^2 \rightarrow s^1$$

M indica el número total de capas, que es igual al número de la última capa, por ejemplo, si M es igual a 2, las sensibilidades se propagan desde la última capa s^2 hasta la primera capa s^1 .

Resumen

A continuación, se presentan las ecuaciones más importantes utilizadas para la implementación de la RNA en el software MATLAB.

Ecuación (7), propagación hacia adelante:

$$a = f(W \times P + b)$$

Ecuación (14), evaluación del error:

$$MSE = \frac{\sum_{i=1}^M (t - a)^2}{M}$$

Ecuación (36), cálculo de sensibilidades en la última capa:

$$s_i^m = -2 \times \dot{F}^m(n^m) \times (t - a)$$

Ecuación (47), cálculo de sensibilidades en la o las capas ocultas:

$$s^m = \dot{F}^m(n^m) \times (W^{m+1})^T \times s^{m+1}$$

Ecuación (40), actualización de pesos:

$$W^m(k + 1) = W^m(k) - \alpha \times s^m \times (a^{m-1})^T$$

Ecuación (41), actualización de bias:

$$b^m(k + 1) = b^m(k) - \alpha \times s^m$$

Derivada de la función de activación logarítmica sigmoide:

$$\dot{f}(n) = a(1 - a) \quad (48)$$

Derivada de la función de activación tangente sigmoide:

$$\dot{f}(n) = 1 - a^2 \quad (49)$$

Derivada de la función de activación lineal:

$$\dot{f}(n) = 1 \quad (50)$$

2.1.11. Evaluación del modelo.

Para el proceso de prueba se requiere de alguna técnica que permiten evaluar la proyección y conocer el error cometido. Entre las técnicas para llevar a cabo esta evaluación están las siguientes [18]:

a) Error Cuadrático Medio (MSE: Mean Squared Error).

Es la medida de error más común y viene expresada por la ecuación (48). El error resultante entre el valor real y el valor esperado se eleva al cuadrado para resaltar el error cometido y evitar valores negativos. Se denomina como M al número total de valores que serán evaluados y el resultado se obtiene como la suma de los errores divididos para M [18]:

$$MSE = \frac{\sum_{i=1}^M (y_i - \hat{y}_i)^2}{M} \quad (51)$$

Donde:

- y_i : valor esperado (real).
- \hat{y}_i : valor estimado (proyectado).

b) Raíz del error Cuadrático Medio (RMSE: Root Mean Squared Error).

El proceso es al igual que la técnica anterior de MSE, con la diferencia de añadir una raíz cuadrada al resultado. La raíz se introduce con el objetivo de obtener la escala de los errores en la misma escala que los objetivos [17]. La ecuación de esta técnica es la siguiente:

$$RMSE = \sqrt{\frac{\sum_{i=1}^M (y_i - \hat{y}_i)^2}{M}} \quad (52)$$

Donde:

- y_i : valor esperado.
- \hat{y}_i : valor estimado.
- M : número total de valores.

c) Error Absoluto Medio (MAE: Mean Absolute Error).

Esta técnica es muy útil cuando se desea medir el error cometido en las mismas unidades que la serie original [7]. La medida del error entre los valores esperados y los valores obtenidos se plantea como valor absoluto para evitar valores negativos, el resultado se obtiene como la suma de los errores cometidos dividida para el total de valores [18]: Esta técnica viene expresada en la siguiente ecuación:

$$MAE = \frac{\sum_{i=1}^M |y_i - \hat{y}_i|}{M} \quad (53)$$

Donde:

- y_i : valor esperado.
- \hat{y}_i : valor estimado.
- M : número total de valores.

d) Porcentaje de Error Medio Absoluto (MAPE: Mean Absolute Porcentaje Error).

Esta técnica permite expresar el resultado en forma porcentual, lo que resulta más útil que en términos de valor absoluto [2]. Se calcula el error relativo porcentual para cada proyección y luego se obtiene el promedio de estos errores relativos, esta técnica viene expresada en la siguiente ecuación [7]:

$$MAPE = \frac{\sum_{i=1}^M 100 \times \frac{|y_i - \hat{y}_i|}{y_i}}{M} \quad (54)$$

Donde:

- y_i : resultado esperado.
- \hat{y}_i : resultado estimado.
- M : número total de valores.

De las técnicas presentadas se selecciona el criterio de evaluación MAPE, debido a que al estar expresado en porcentaje facilita la interpretación de los resultados. Se considera que el resultado del criterio de evaluación MAPE es “bueno” cuando se encuentra entre 10% y 20%, “moderado” cuando se encuentra entre 20% y 30% y “pobre” al resultar por encima del 30% [16].

2.1.12. Determinación de arquitectura y aplicación.

No existe en la literatura un conjunto de pasos que indiquen cual es la forma correcta de definir la estructura de la RNA, sin embargo, la mayoría de autores recomienda ensayos de prueba y error, para ello se plantea un procedimiento que recomienda el cumplimiento de tres etapas [19]:

- 1) Determinación de los parámetros del modelo.
- 2) Diseño y entrenamiento.
- 3) Prueba del modelo.

En la primera etapa se elige la arquitectura de RNA a utilizar, el número de neuronas en las diferentes capas, función de activación y algoritmo de entrenamiento. En la segunda etapa se presenta el conjunto de datos de entrada a partir de los cuales la red va a aprender. Finalmente, la tercera etapa consiste en comprobar que la RNA ha aprendido, para ello se presenta un conjunto de datos desconocidos, que no han sido usados en el entrenamiento.

2.1.13. Proyección en un paso de tiempo.

El problema de la proyección de series de tiempo resulta que, a partir de un conjunto de muestras conocidas de la serie, determinar la evolución de la misma para obtener sus valores futuros. Esta proyección se la realiza para el instante inmediatamente siguiente del actual t , usando las muestras disponibles hasta ese instante de tiempo; es decir, proyectar el valor $x(t + 1)$ utilizando un cierto número de muestras anteriores $x(t)$, $x(t - 1)$, $x(t - 2)$, ..., $x(t - n)$ [21].

Cuando no se dispone de una ecuación que describe el comportamiento de la serie temporal, se recurre a métodos aproximativos para construir el modelo que resuelva el problema de

proyección. Estos modelos recogen el comportamiento temporal de la serie expresando el valor de la serie en el instante $t + 1$ como una función no lineal de $r + 1$ valores de la serie en instantes anteriores de tiempo, es decir:

$$x(t + 1) = F(x(t), x(t - 1), \dots, x(t - r)) \quad (55)$$

Donde:

- t : variable discreta tiempo.
- F : función no lineal desconocida.
- r : valores pasados que describen el patrón de la serie temporal.
- x : muestras de la serie temporal.

Si se considera un vector de entrada como el descrito en la ecuación (46), la red neuronal puede aproximar la función F obteniendo un modelo de proyección definido por:

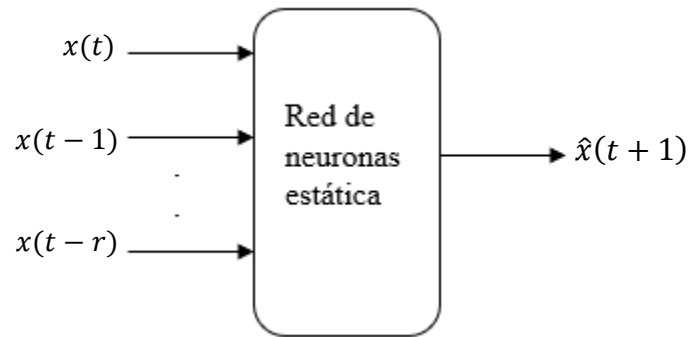
$$\hat{x}(t + 1) = \hat{F}(x(t), x(t - 1), \dots, x(t - r)) \quad (56)$$

Donde:

- \hat{F} : aproximación neuronal de la función F .
- $\hat{x}(t + 1)$: proyección para el instante de tiempo $t + 1$.

La figura 10 muestra el modelo de proyección para un instante de tiempo con $r + 1$ neuronas de entrada que reciben de los valores de la serie temporal en los $r + 1$ instantes anteriores, y una neurona de salida que representa la proyección en un paso de tiempo.

Figura 10: Modelo de proyección en un paso de tiempo.



Fuente: [19].

Elaboración: Autores.

Dado el conjunto de muestras y el valor de r , cada patrón se construye utilizando los $r + 1$ valores anteriores de la serie, de este modo cada patrón de entrada recoge la información temporal necesaria para proyectar el valor de la serie en el instante de tiempo $t + 1$. Al realizar el entrenamiento se pretende que la red neuronal capture la relación entre el valor de la serie en el instante de tiempo y los r valores pasados. Uno de los problemas existentes es la longitud que se debe presentar en el conjunto de patrones para los instantes anteriores puesto que esto depende de cada serie [21].

2.1.14. Proyección en múltiples pasos de tiempo.

Esta proyección consiste en conocer el comportamiento de la serie no solo en un instante siguiente al actual, sino en un futuro más lejano, específicamente en un intervalo de proyección $[t + 1, t + N + 1]$ siendo N el horizonte de proyección [21].

Si una serie no es lo suficientemente larga y se desean proyectar los valores futuros de la misma, se realiza una realimentación de proyecciones, este método se denomina como proyección por evolución de red [22]. Este método consiste en utilizar de manera recurrente el modelo construido para la proyección en un paso de tiempo, ahora el propósito es proyectar el valor de la serie de tiempo en el instante $t + N + 1$ que viene dado por la ecuación (47):

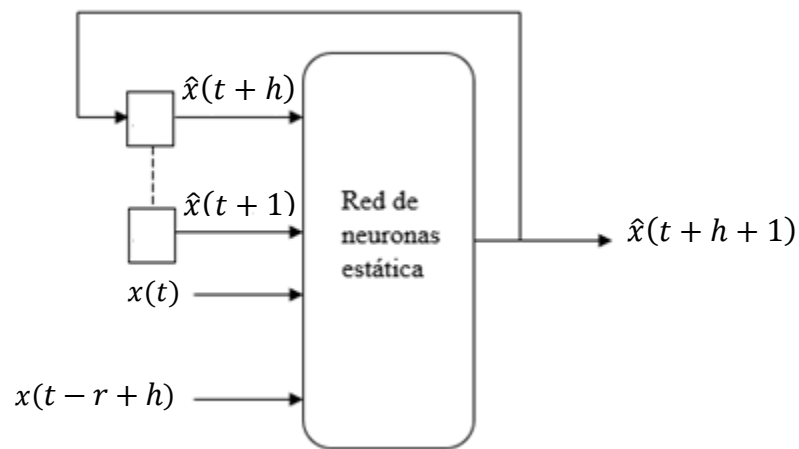
$$\hat{x}(t + 1 + N) = \hat{F}(x(t + N), x(t + N - 1), \dots, x(t + N - r)) \quad (57)$$

Donde:

- t : variable discreta tiempo.
- \hat{F} : función no lineal proyectada.
- r : valores pasados que describen el patrón de la serie temporal.
- \hat{x} : valor proyectado de la serie temporal.
- x : muestra de la serie temporal.
- N : horizonte de tiempo.

Sin embargo, no toda la información de entrada está disponible, debido a que los valores $x(t + N), x(t + N - 1), \dots, x(t + 1)$ para $N > 1$ no se conocen. Para afrontar este problema se utilizan como entradas al modelo los valores proyectados en instantes anteriores de tiempo $-\hat{x}(t + N), x(t + N - 1), \dots, x(t + 1)$, de este modo el modelo dado en la ecuación (46) puede usarse con el propósito de proyección en múltiples pasos de tiempo retroalimentando la salida de la red neuronal hacia la entrada, como se muestra en la figura 11.

Figura 11: Modelo de proyección en múltiples pasos de tiempo.



Fuente: [21]

Elaborado: Autores

El inconveniente que se presenta en este modelo es que durante el entrenamiento, la red neuronal ha capturado las relaciones entre la muestra actual y las muestras observadas en instantes anteriores, sin embargo, cuando se utiliza el modelo para proyectar en múltiples pasos de tiempo, las neuronas de entrada reciben la información de los valores proyectados en instantes anteriores de tiempo, de manera que los errores cometidos en la proyección de dichos valores son propagados hacia las proyecciones futuras siendo afectadas de forma negativa.

2.2. Marco Referencial.

La proyección de demanda eléctrica radica en su importancia para la operación y expansión de los sistemas de distribución, por lo cual, es considerado un factor determinante para la toma de decisiones [23]. La industria eléctrica requiere proyecciones cada vez más ajustadas a la necesidad de corto, mediano y largo plazo, debido a que esta proyección es fundamental para los procesos que realizan los agentes de mercado, pero resulta ser una tarea difícil debido a la complejidad de los factores que influyen en su comportamiento. La proyección a corto plazo es usada para planificar la operación del sistema, mientras que la predicción de mediano y largo plazo es usada para la toma de decisiones de expansión en capacidad de generación y distribución del sistema [24].

Para realizar la proyección es necesario contar con información cuantitativa de la evolución de la demanda a través del tiempo, es decir, una serie de tiempo, siendo el análisis de series de tiempo la técnica más usada para estimar su comportamiento, pero el uso de este método está limitado por la existencia de relaciones no lineales entre los datos [25], tal como sucede con la serie de tiempo en este caso de estudio [25].

Las RNA para la proyección de series de tiempo fueron usadas por primera vez para el pronóstico del clima, pero debido a la ausencia de un algoritmo de entrenamiento la investigación quedó limitada [26]. No fue sino hasta que se introdujo el algoritmo backpropagation (propagación hacia atrás) que las RNA empezaran a tomar mayor importancia. Las aplicaciones de las RNA han sido implementadas para el pronóstico de series de tiempo en áreas de finanzas, generación de energía, medicina, recursos de agua, entre otras [27].

Muchas investigaciones se han realizado usando RNA en el campo de la proyección de series temporales, algunas de ellas las comparan con los métodos tradicionales, este es el caso de [23], en donde se presenta una investigación comparando el método ARIMA con una RNA para la proyección de la demanda de energía eléctrica en Colombia, concluyendo que el método de RNA proyecta los datos de la demanda de energía eléctrica de mejor manera.

En otro caso de estudio se realiza una investigación para conocer el error entre el método RNA y el método ARIMA, haciendo énfasis en la importancia del pre procesamiento de los datos, es decir, evaluar las muestras antes de ser ingresadas a procesar, concluyendo que las RNA proporcionan menor error al comparar los dos métodos con los valores reales [27]. Se presenta un trabajo con la recopilación de algunas investigaciones (artículos científicos) sobre la proyección de demanda de energía eléctrica analizando el uso de diferentes métodos, entre ellos están: series de tiempo, RNA, ARIMA, regresión lineal, modelo espacio tiempo, entre otros, concluyendo que no existen razones empíricas, metodológicas o teóricas para preferir un modelo específico entre las alternativas [26].

CAPÍTULO III
METODOLOGÍA DE LA INVESTIGACIÓN

3.1. Localización.

Este trabajo de investigación se realiza en la ciudad de Quevedo, provincia de Los Ríos, en Ecuador, con ayuda de la empresa eléctrica CNEL EP -UNIDAD DE NEGOCIOS AGENCIA QUEVEDO quien fue la que aportó los datos de demanda de la ciudad.

3.2. Tipo de investigación.

3.2.1. Investigación bibliográfica.

Al estar tratando con un tema desconocido para los autores del proyecto se realiza una ardua investigación recurriendo a artículos científicos, trabajos de investigación, libros, tesis de grado y posgrado que abordan el tema desde diferentes metodologías y puntos de vista, permitiendo conocer el tema a mayor profundidad.

3.2.2. Investigación explicativa.

Las metodologías aplicadas en clase a lo largo de la carrera permiten complementar las destrezas necesarias para el entendimiento y desarrollo de este proyecto, sentando como bases conocimientos sobre temas relacionados con demanda de energía eléctrica, consumo de energía, probabilidades, entre otros.

3.3. Método de investigación.

3.3.1. Método analítico.

La recopilación de datos de diferentes fuentes de información consultadas permite conocer las características principales que se debe tener en cuenta para el tratamiento de los datos iniciales, el proceso de entrenamiento y prueba de la RNA.

3.3.2. Método inductivo.

En este método se canaliza la teoría antes analizada, infiriendo ciertas conclusiones al tomar en cuenta las condiciones que se podrían generar al desarrollar el proceso de proyección mediante RNA.

3.4. Fuentes de recopilación de información.

La información para este proyecto de investigación fue obtenida de diversas fuentes, tales como: CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO, artículos científicos, libros y tesis relacionadas con el caso de estudio que permitieron el entendimiento y conceptualización del tema.

3.5. Diseño de investigación.

Los datos que se usan para el desarrollo de este proyecto son valores de consumo de potencia eléctrica correspondiente a 6 años, estos datos fueron entregados por la empresa CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO. Los datos corresponden al periodo 2003 – 2008, los cuales no son tan actuales debido a que en la empresa eléctrica se manifestó que el seguimiento del consumo energético de la ciudad de Quevedo fue trasladado a oficinas centrales ubicadas en el cantón Durán, a lo cual no se tuvo acceso por motivos de pandemia. Cabe indicar que se realizó la debida solicitud para la obtención de los datos históricos de consumo de energía eléctrica.

3.6. Instrumentos de investigación.

Los instrumentos empleados para este proyecto de investigación son: tabla de consumo de potencia eléctrica (MW) del año 2003 al 2008 de la ciudad de Quevedo, artículos científicos, libros y tesis referentes al tema de investigación, Microsoft Word para realizar diagramas de flujo y el software MATLAB, empleado para la implementación de la RNA.

3.7. Tratamiento de los datos.

Los datos facilitados por CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO son agrupados en dos categorías que corresponden a datos de entrenamiento y datos de prueba, posteriormente son tabulados e ingresados en el software MATLAB donde se realiza el cambio de escala a unidades más pequeñas (normalización). Mediante ensayos de prueba y error se obtiene los resultados de entrenamiento, posteriormente se realiza la prueba de la RNA con los valores en su escala original (desnormalización), obteniendo el porcentaje de error al comparar los valores históricos de demanda y la proyección para ese lapso de tiempo, finalmente se realiza la proyección para un tiempo posterior y fuera de los datos históricos, limitado por un horizonte de tiempo.

3.8. Recursos humanos y materiales.

Tabla 1: Recursos humanos.

Ítem	Descripción
1	Alvaro Leonel Mora Silva
2	Stalin Fabian Toaquiza Chango
3	Ing. Diego Patricio Peña Banegas

Elaboración: Autores.

Tabla 2: Recursos de hardware.

Ítem	Cantidad	Descripción
1	1	Computadora HP, Intel Core i5, 525GB
2	1	Computadora ASUS, Intel Core i7, 1TB
3	1	Impresora EPSON L3110
4	1	USB capacidad de 4GB

Elaboración: Autores.

Tabla 3: Recursos de software.

Ítem	Cantidad	Descripción
1	1	MATLAB r2017b
2	1	Paquete Office 2020

Elaboración: Autores.

Tabla 4: Recursos de materiales.

Ítem	Cantidad	Descripción	Costo Unitario	Total
1	1	Resma de hojas A4	3.50 \$	3.50 \$
2	1	Cuaderno de 100 hojas académico	1.50 \$	1.50 \$
3	2	Esferos de color negro y azul	0.35 \$	0.70 \$
			Total \$	5.70 \$

Elaboración: Autores.

CAPÍTULO IV
RESULTADOS Y DISCUSIÓN

4.1. Resultados.

4.1.1. Base de datos.

Los datos para realizar el proyecto son valores de consumo de potencia eléctrica tabulados de forma mensual pertenecientes a la ciudad de Quevedo en el periodo de tiempo 2003 - 2008. Los datos son entregados por la empresa eléctrica CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO y se presentan en la siguiente tabla:

Tabla 5: Valores del consumo de potencia eléctrica mensual del periodo 2003 – 2008.

Número de muestras	Tiempo (Año)	Tiempo (Mes)	Demanda máxima (MW)	Número de muestras	Tiempo (Año)	Tiempo (Mes)	Demanda máxima (MW)
1	2003	1	62.90	37	2006	1	71.77
2	2003	2	67.48	38	2006	2	76.48
3	2003	3	59.42	39	2006	3	73.58
4	2003	4	67.38	40	2006	4	75.36
5	2003	5	60.42	41	2006	5	65.17
6	2003	6	56.51	42	2006	6	64.45
7	2003	7	48.45	43	2006	7	61.91
8	2003	8	57.39	44	2006	8	65.18
9	2003	9	56.92	45	2006	9	62.26
10	2003	10	56.92	46	2006	10	62.45
11	2003	11	58.35	47	2006	11	64.72
12	2003	12	70.87	48	2006	12	81.24
13	2004	1	69.25	49	2007	1	77.26
14	2004	2	72.41	50	2007	2	83.45
15	2004	3	67.57	51	2007	3	78.48
16	2004	4	70.50	52	2007	4	78.27
17	2004	5	61.20	53	2007	5	73.45
18	2004	6	57.68	54	2007	6	69.45
19	2004	7	56.25	55	2007	7	68.45
20	2004	8	61.45	56	2007	8	68.16
21	2004	9	56.57	57	2007	9	64.88

22	2004	10	62.54	58	2007	10	68.35
23	2004	11	54.90	59	2007	11	64.88
24	2004	12	64.25	60	2007	12	80.75
25	2005	1	68.42	61	2008	1	79.05
26	2005	2	76.21	62	2008	2	86.29
27	2005	3	76.67	63	2008	3	82.56
28	2005	4	71.25	64	2008	4	71.58
29	2005	5	63.24	65	2008	5	77.45
30	2005	6	63.28	66	2008	6	71.28
31	2005	7	61.84	67	2008	7	72.58
32	2005	8	66.35	68	2008	8	71.52
33	2005	9	62.15	69	2008	9	69.53
34	2005	10	61.48	70	2008	10	70.25
35	2005	11	62.25	71	2008	11	71.65
36	2005	12	75.45	72	2008	12	81.54

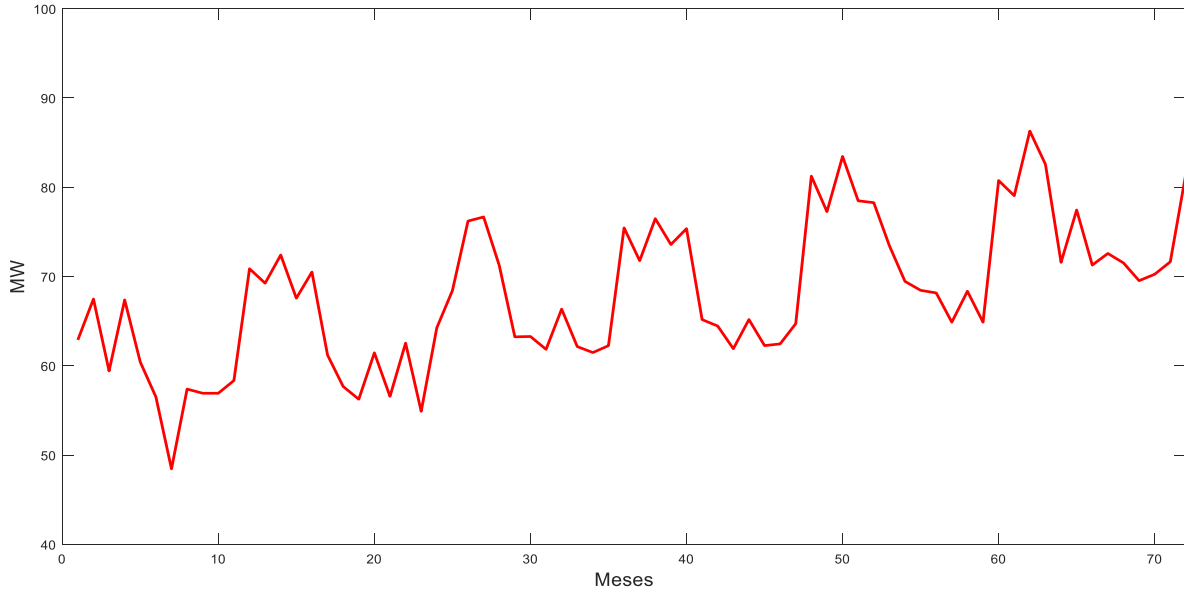
Fuente: CNEL EP – UNIDAD DE NEGOCIOS AGENCIA QUEVEDO.
Elaboración: Autores.

De la base de datos se analiza que los valores de mínimo y máximo consumo fueron en el mes 7 del año 2003 y el mes 2 del año 2008 que corresponden a 48.45 MW y 86.29 MW respectivamente. Estos valores mínimo y máximo de la serie de tiempo permiten realizar la normalización de los datos lo cual es muy importante para el procesamiento en la RNA.

Del conjunto de datos históricos se estima que inicialmente en el año 2003 el valor medio de consumo es de 60.25 MW creciendo anualmente de manera progresiva con una media de 3.04 MW, hasta alcanzar en el año 2008 un valor medio de 75.44 MW.

Los valores de consumo de potencia eléctrica de los 72 meses se presentan en la figura 12 como una curva que describe una serie en el tiempo, las fluctuaciones que presentan corresponden a las variaciones típicas de consumo.

Figura 12: Curva de consumo de potencia eléctrica de 72 meses en escala original.



Fuente: Investigación.
Elaboración: Autores.

4.1.2. Preparación de datos.

4.1.2.1. Normalización de datos.

Los datos que se presentan en la serie de tiempo no pueden ser procesados directamente por la RNA, debido a que al tratar de procesar valores muy grandes tardaría mucho tiempo realizando los cálculos, retrasando el proceso y en el peor de los casos colapsando el sistema. Para evitar esto se realiza la normalización de los datos de entrada a valores en el intervalo 0 a 1, la fórmula utilizada para normalizar los valores es la siguiente:

$$P_{norm} = \frac{P_{desn} - P_{min}}{P_{max} - P_{min}} \quad (58)$$

Donde:

P_{norm} : valor normalizado.

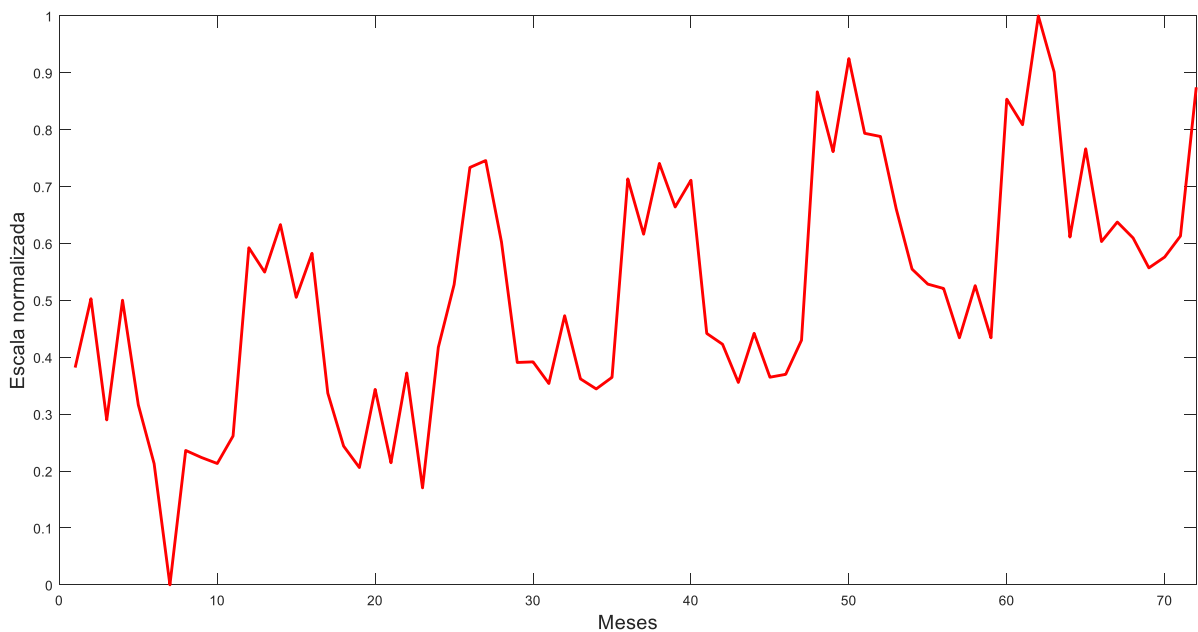
P_{desn} : valor desnormalizado o en escala inicial.

P_{min} : valor mínimo de la serie.

P_{max} : valor máximo de la serie.

Los datos de consumo de potencia eléctrica para los 72 meses normalizados se presentan en la figura 13.

Figura 13: Curva de consumo de potencia eléctrica de 72 meses normalizados.



Fuente: Investigación.

Elaboración: Autores.

4.1.2.2. Conjuntos de entrenamiento y de prueba.

Del total de datos mostrados en la serie de tiempo se dividen en dos grupos, de la siguiente manera: 85% del total de datos para el proceso de entrenamiento, que corresponde a los datos en el intervalo del mes 1 al 60; y, el 15% restante para el proceso de prueba que corresponde a los datos de la serie de tiempo pertenecientes al intervalo de los meses 61 al 72. La división de

los datos se realiza de forma arbitraria, pero siempre considerando que el mayor porcentaje de datos es necesario para el entrenamiento y el porcentaje de datos restantes es muy importante para validar el desempeño de la RNA mediante la comparación de la proyección en el proceso de prueba, si se seleccionase el 100% de datos para el entrenamiento, no se podría validar si la proyección de la RNA es correcta, debido a que no se tendría valores nuevos para probar la RNA.

Tabla 6: División de datos para entrenamiento y prueba.

Proceso	# datos	% datos
Entrenamiento	60	85
Prueba	12	15
Total	72	100

Fuente: Investigación.

Elaboración: Autores.

4.1.3. Proceso de entrenamiento.

Para realizar el proceso de entrenamiento se debe seleccionar de forma adecuada el tamaño del vector de entrada, el cual contiene los datos seleccionados previamente para este proceso. El tamaño del vector de entrada debe contener un patrón que marque una tendencia en el tiempo; asimismo, se debe contar con un vector de salida u objetivo que corresponde a la salida deseada de entrenamiento. Durante el entrenamiento se generan errores al evaluar la salida de la RNA con la salida esperada, esto provoca que el algoritmo de entrenamiento modifique los parámetros relacionados al proceso, es decir, los pesos y bias de las neuronas de la o las capas ocultas y salida.

La matriz de entrada P y el vector de salida T están definidos en la Tabla 7, el tamaño de la matriz de entrada es de 12×48 , y el vector de salida es de 1×48 , es importante que las dos variables tengan la misma cantidad de columnas, debido a que cada columna de la matriz de entrada corresponde al patrón que describe cada valor del vector de salida.

Tabla 7: Matriz de entrada y salida para el entrenamiento.

P	70.87	69.25	72.41	67.57	70.50	61.20	57.68	56.25	61.45	56.57	...
	58.35	70.87	69.25	72.41	67.57	70.50	61.20	57.68	56.25	61.45	...
	56.92	58.35	70.87	69.25	72.41	67.57	70.50	61.20	57.68	56.25	...
	56.92	56.92	58.35	70.87	69.25	72.41	67.57	70.50	61.20	57.68	...
	57.39	56.92	56.92	58.35	70.87	69.25	72.41	67.57	70.50	61.20	...
	48.45	57.39	56.92	56.92	58.35	70.87	69.25	72.41	67.57	70.50	...
	56.51	48.45	57.39	56.92	56.92	58.35	70.87	69.25	72.41	67.57	...
	60.42	56.51	48.45	57.39	56.92	56.92	58.35	70.87	69.25	72.41	...
	67.38	60.42	56.51	48.45	57.39	56.92	56.92	58.35	70.87	69.25	...
	59.42	67.38	60.42	56.51	48.45	57.39	56.92	56.92	58.35	70.87	...
	67.48	59.42	67.38	60.42	56.51	48.45	57.39	56.92	56.92	58.35	...
	62.90	67.48	59.42	67.38	60.42	56.51	48.45	57.39	56.92	56.92	...
T	69.25	72.41	67.57	70.50	61.20	57.68	56.25	61.45	56.57	62.54	...

Elaboración: Autores.

El tamaño de las filas de la matriz de entrada corresponde al número de neuronas de la capa de entrada y es el desplazamiento que tendrá el resultado de entrenamiento con respecto a los datos históricos, es decir, si el número de neuronas de entrada es 12, el resultado de entrenamiento inicia en el mes número 13, como se observa en la figura 14.

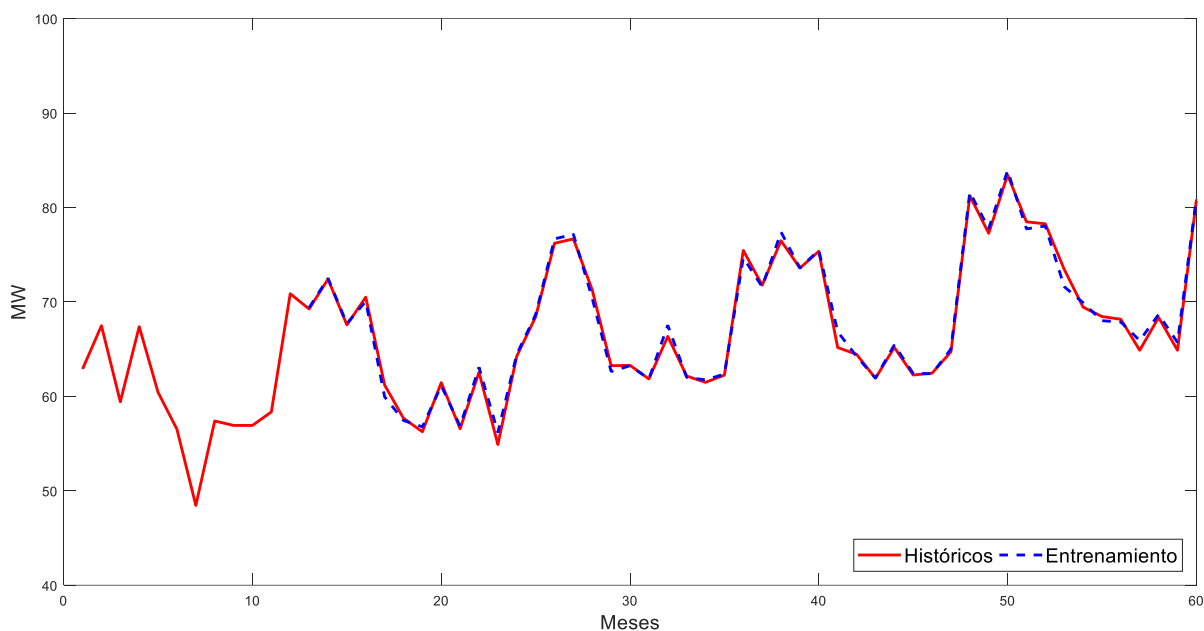
En este proyecto se consideran dos inconvenientes durante el desarrollo de ensayos con la RNA, el sobreentrenamiento y la poca cantidad de neuronas de entrada, no permiten lograr el objetivo de una adecuada proyección, a continuación, se presentan los resultados de entrenamiento realizados con este tipo de problemas.

a) Sobreentrenamiento.

El sobreentrenamiento es producido cuando la RNA no ha aprendido y no puede generalizar el modelo. Como resultado de este proceso se establece que la RNA memoriza, más no ha aprendido. En la figura 14 se muestra el resultado de un proceso de entrenamiento en el cual se evidencia que la salida de la RNA (línea entrecortada azul) es la misma que los valores históricos

(línea continua roja), estando sobrepuesta una encima de otra. El eje horizontal se extiende hasta el mes 60 debido a que los 12 meses restantes han sido omitidos para luego ser utilizados en el proceso de prueba, mientras que en el eje vertical los valores se presentan en el intervalo de 40 a 100 en su unidad de medición original (MW) y este será el intervalo para todas las demás figuras.

Figura 14: Resultado de entrenamiento de una red sobreentrenada.



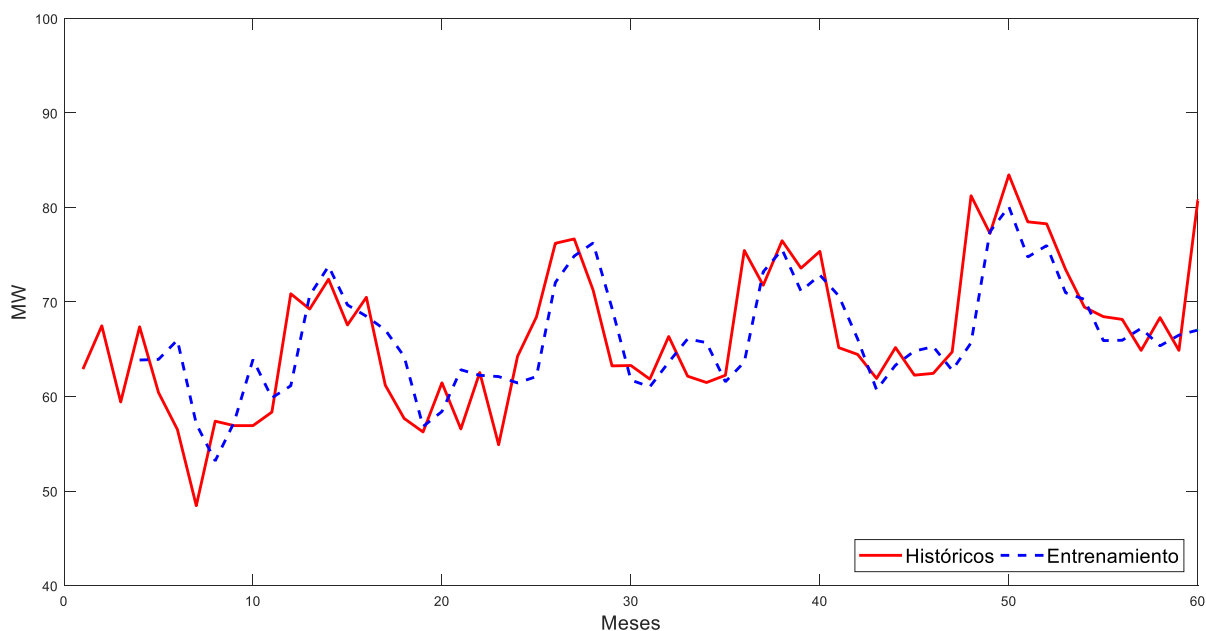
Fuente: Investigación.
Elaboración: Autores.

b) Número de neuronas de entrada inadecuado.

Un número muy bajo de neuronas de entrada no es suficiente para que la RNA aprenda la tendencia de la serie temporal, mientras que un número excesivo de neuronas de entrada provoca la saturación y de igual forma no entrega resultados adecuados. Durante los ensayos se determina que el número adecuado de neuronas de entrada tiene que ser un valor que abarque un patrón que se repita cierto tiempo.

En la figura 15, se presenta el resultado del proceso de entrenamiento para la RNA con 3 neuronas de entrada. La salida estimada por la RNA es muy distinta a la salida deseada, considerando a esta respuesta como un entrenamiento pobre. El resultando de entrenamiento para este caso inicia en el mes número 4, debido a que se la RNA tiene 3 neuronas de entradas lo que provoca el desplazamiento en el resultado.

Figura 15: Resultado de entrenamiento con 3 neuronas de entrada.



Fuente: Investigación.
Elaboración: Autores.

En los siguientes procesos se muestra los resultados de la RNA con los dos problemas mencionados.

4.1.4. Proceso de prueba.

En el proceso de prueba se obtiene el MAPE entre los valores proyectados y valores históricos que se destinaron para este proceso y no fueron usados durante el entrenamiento. La validación

del modelo se realiza utilizando la ecuación (51), el uso de este criterio de evaluación facilita la interpretación del error debido a que entrega el resultado en porcentaje.

En la tabla 8 se presenta el conjunto de datos históricos de consumo de potencia eléctrica seleccionados para realizar la prueba, los valores corresponden a los 12 meses del año 2008 que han sido omitidos del proceso de entrenamiento.

Tabla 8: Datos históricos usados en el proceso de prueba.

Año	Mes	MW
2008	1	79.05
	2	86.29
	3	82.56
	4	71.58
	5	77.45
	6	71.28
	7	72.58
	8	71.52
	9	69.53
	10	70.25
	11	71.65
	12	81.54

Fuente: Investigación.
Elaboración: Autores.

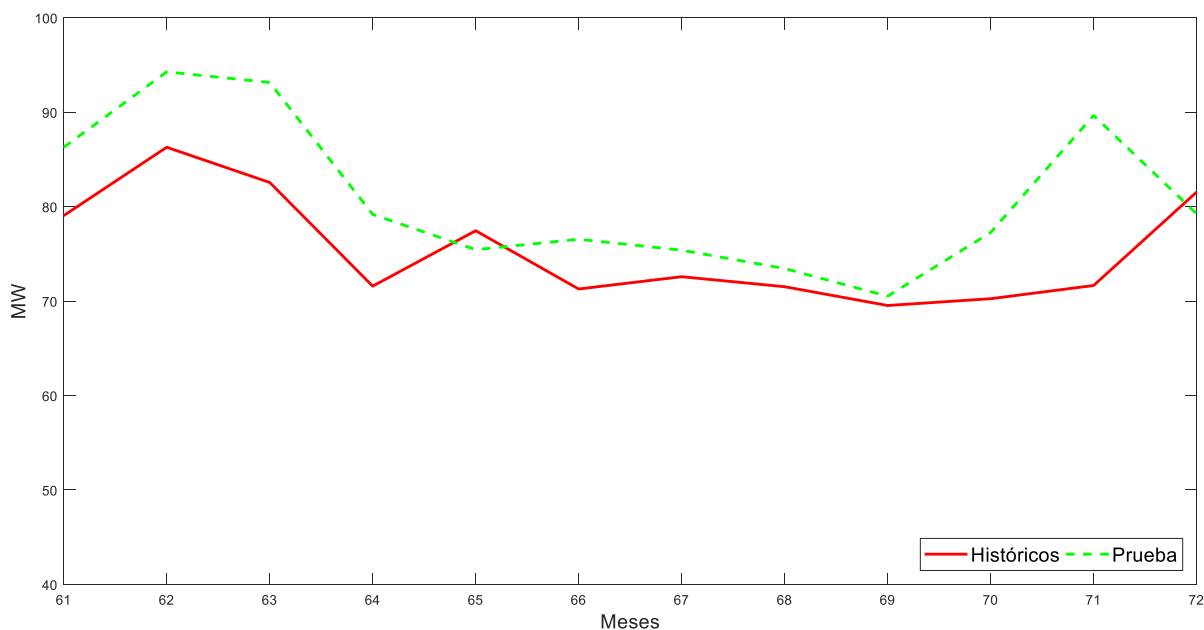
Una vez finalizado el proceso de entrenamiento la RNA está en capacidad de realizar la proyección de los valores para el proceso de prueba, sin embargo, la proyección se encuentra en la escala normalizada por lo que se procede a desnormalizar y obtener los valores en la unidad inicial, es decir, en MW, para ello se utiliza la ecuación (56):

$$P_{desn} = (P_{norm} \times (P_{max} - P_{min})) + P_{min} \quad (59)$$

En la figura 16 se muestra el resultado de la proyección (línea entrecortada verde) para el proceso de prueba de la RNA sobreentrenada y en la figura 17 el resultado de proyección de la RNA con 3 neuronas de entrada. El eje horizontal se presenta en el intervalo del mes 61 al 72

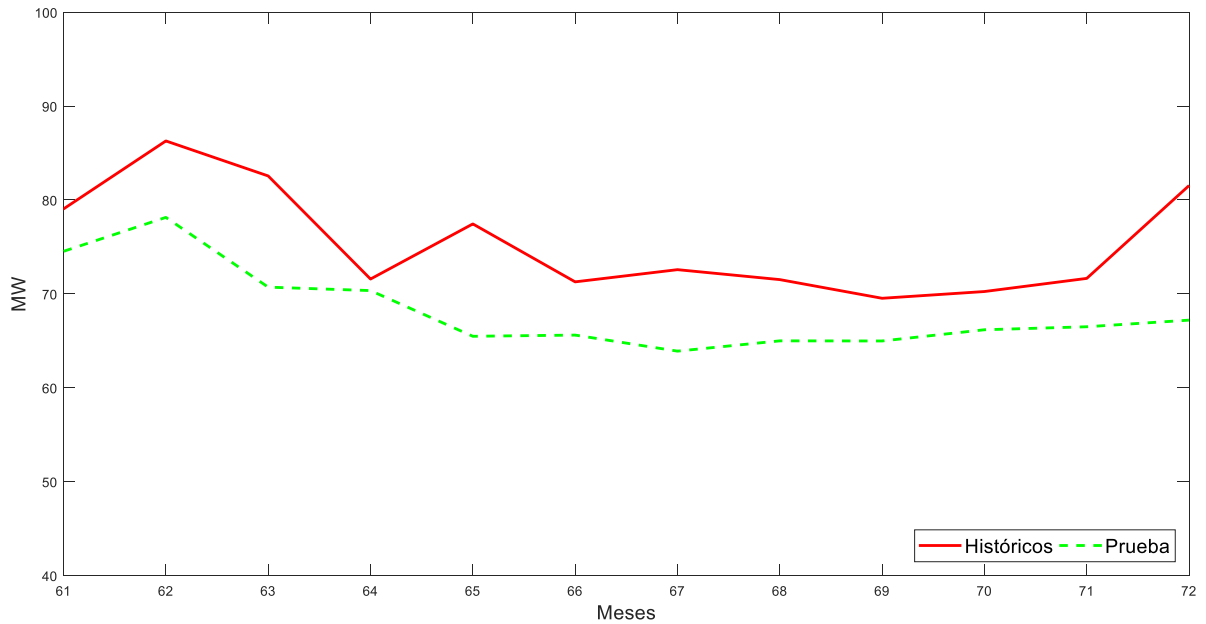
que corresponde a los 12 meses históricos seleccionados para este proceso, mientras que en el eje vertical se mantiene el mismo intervalo anterior en MW.

Figura 16: Proyección para la prueba de la RNA sobreentrenada.



Fuente: Investigación.
Elaboración: Autores.

Figura 17: Proyección para la prueba de la RNA con 3 neuronas de entrada.



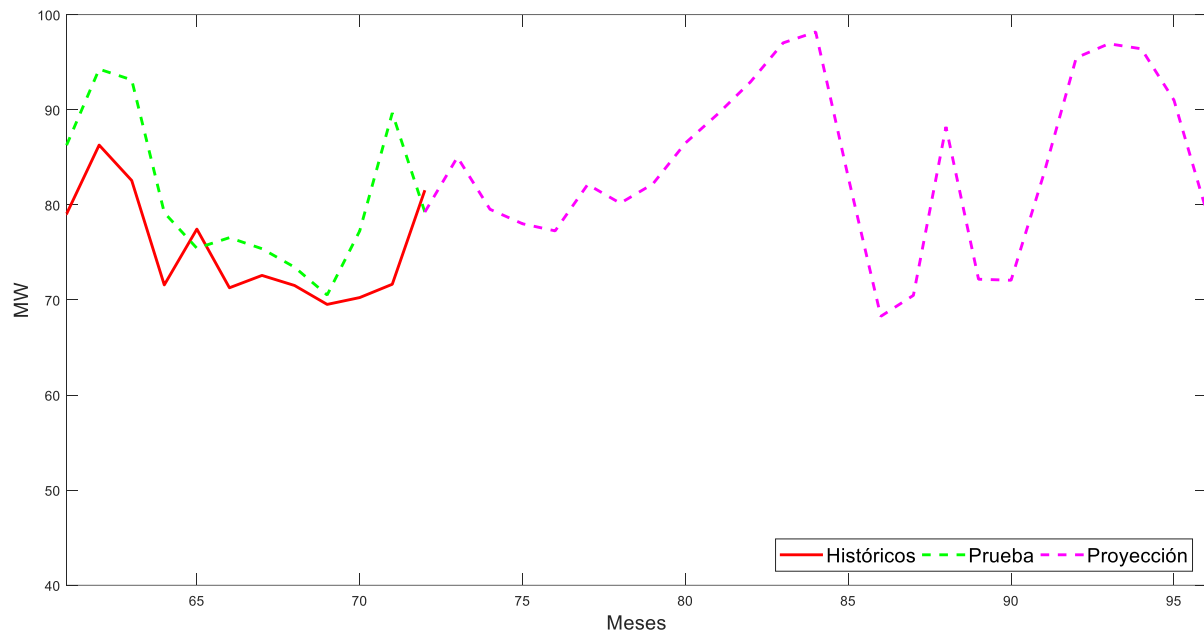
Fuente: Investigación.
Elaboración: Autores.

4.1.5. Proyección.

El proceso de proyección consiste en realizar una extensión de los valores proyectados en la prueba de la RNA, se realiza considerando el método de proyección en múltiples pasos de tiempo, en donde los nuevos datos proyectados serán ingresados a la matriz de entrada formando el patrón de tendencia a la largo del tiempo. La variable N indica el horizonte de tiempo hasta el cual se va a realizar la proyección, para el proceso de prueba el valor de esta variable está definido en 12, mientras que, para la proyección fuera de la prueba se puede definir en cualquier valor.

En la figura 18 se muestra el resultado de la prueba y proyección de la RNA sobreentrenada, se evidencia que, al obtener el entrenamiento demasiado ajustado entre la salida estimada y la salida esperada, se generan proyecciones inadecuadas sin ningún patrón de comportamiento.

Figura 18: Prueba y proyección de la RNA sobreentrenada.

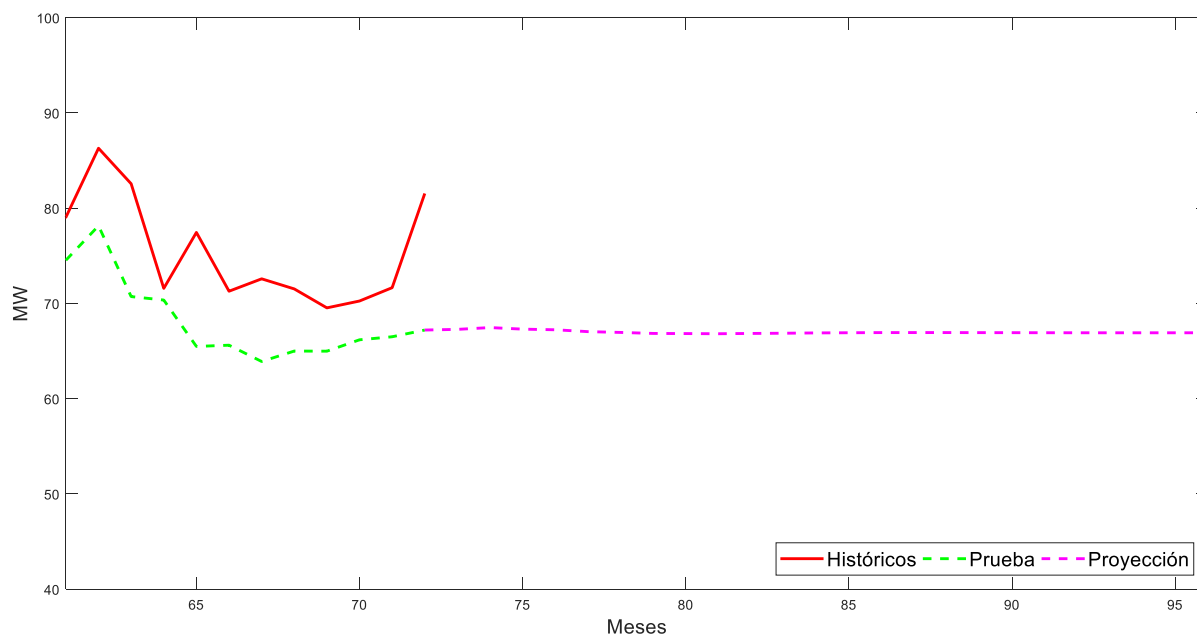


Fuente: Investigación.

Elaborado: Autores.

En la figura 19 se muestra el resultado de la prueba y proyección de la RNA con 3 neuronas de entrada, de igual manera se evidencia que en el entrenamiento no se ajusta adecuadamente la salida estimada y la salida esperada, generando proyecciones que resultan en una línea recta.

Figura 19: Prueba y proyección de la RNA con 3 neuronas de entrada.



Fuente: Investigación.

Elaboración: Autores.

En la figura 18 y figura 19 se presentan los procesos de prueba (líneas entrecortadas verdes), los valores históricos (línea continua roja) y la proyección (líneas entrecortadas magenta) con horizonte de tiempo = 36, que corresponden a la proyección de 12 meses para la prueba y extensión de la proyección para 24 meses más.

4.1.6. Implementación de la Red Neuronal Artificial.

4.1.6.1. Diagramas de flujo de la Red Neuronal Artificial.

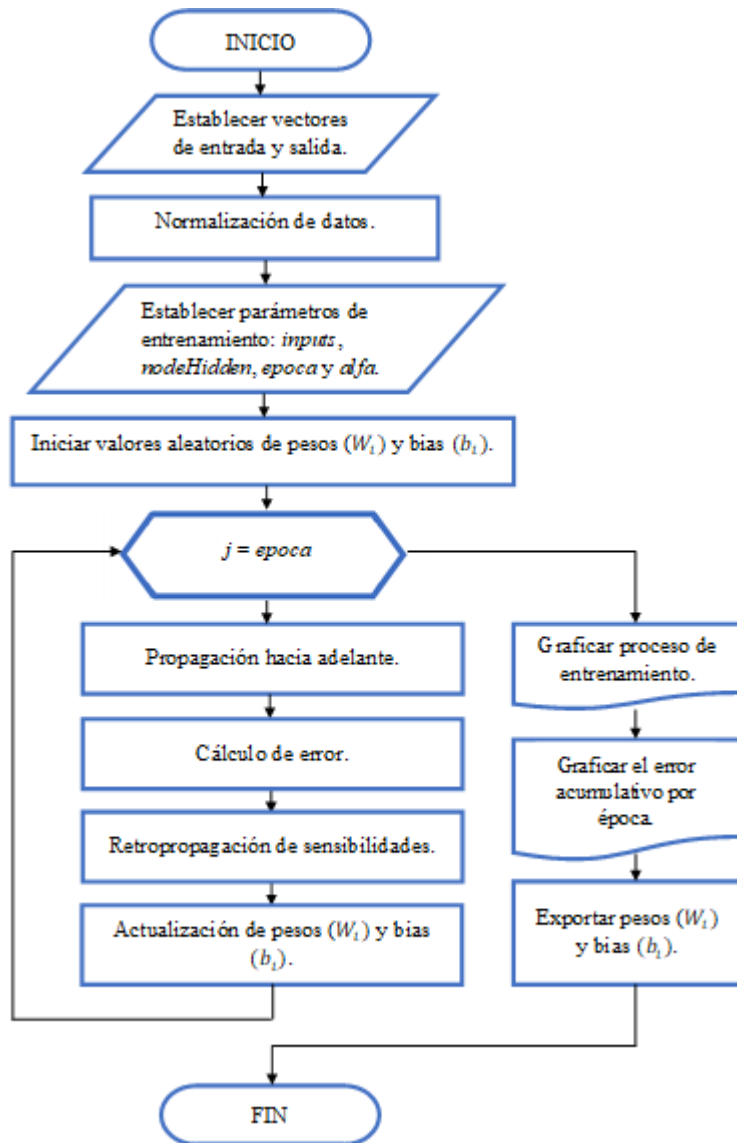
Antes de empezar a implementar la RNA en el software MATLAB se debe contar con los diagramas de flujos que expliquen la conexión existente entre los diferentes procesos. Los diagramas de flujos que explican estas conexiones se presentan a continuación.

La figura 20 presenta el diagrama de flujo 1 correspondiente al proceso de entrenamiento, en la cual se establecen la matriz de entrada y vector de salida que serán normalizados, además, se

definen la cantidad de las neuronas de entrada en la variable *inputs*, en la variable *nodeHidden* se indica el número de neuronas de la o las capas ocultas, en la variable *epoca* se define el número de iteraciones y en la variable *alfa* se define el valor de la tasa de aprendizaje.

La matriz de pesos W_i y vector bias b_i inicialmente se genera de manera aleatoria y con ello se realiza la propagación hacia adelante, a continuación, se realiza la comparación de la salida estimada con la salida deseada y se calcula el error permitiendo obtener un valor de sensibilidad para ser propagado hacia atrás, esto permite ajustar el valor de pesos y bias con cada iteración, este proceso se repite hasta que la variable j es del mismo valor que la variable *época*, posteriormente se generan las gráficas de entrenamiento y del error por cada iteración, finalmente se exportan la matriz de pesos W_i y el vector bias b_i a un archivo denominado **Pesos.m** para ser usados en el siguiente diagrama de flujo.

Figura 20: Diagrama de flujo 1 del proceso de entrenamiento.



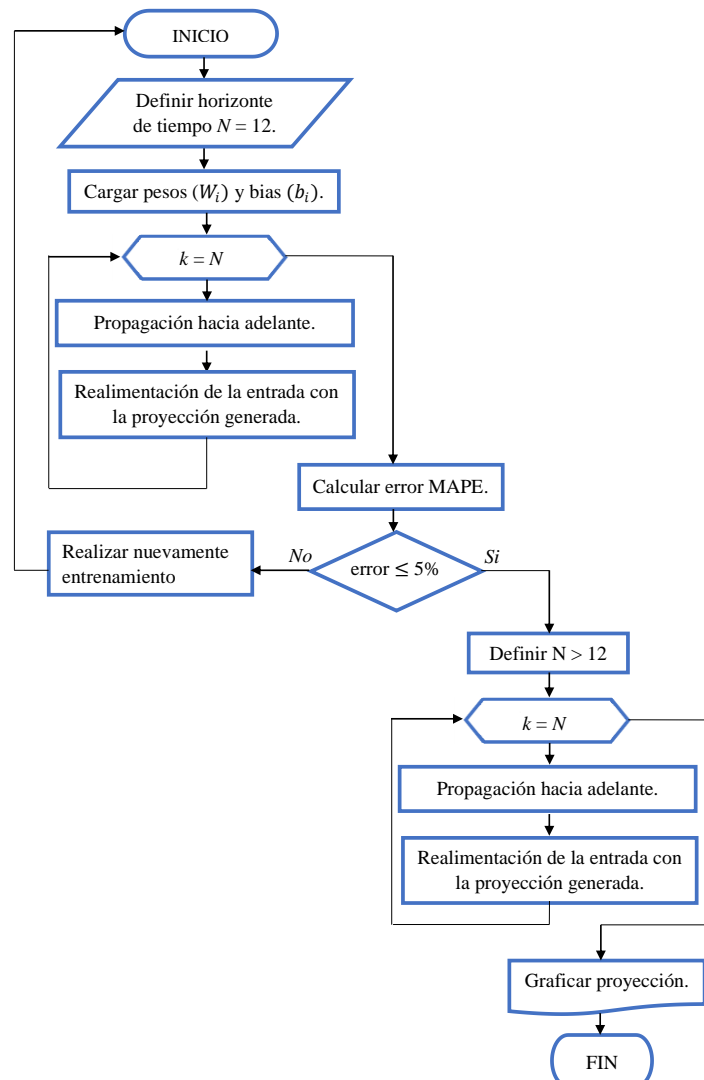
Fuente: Investigación.

Elaboración: Autores.

La figura 21 presenta el diagrama de flujo 2 correspondiente al proceso de proyección. El requerimiento inicial en este diagrama es que para realizar la primera proyección el valor de N sea igual a 12, las variables almacenadas en el archivo **Pesos.m** son cargadas en este nuevo proceso para utilizar el valor de la matriz de pesos W_i y vector bias b_i en la propagación hacia adelante, el vector de entrada es realimentado con los nuevos datos proyectados y genera nuevas

proyecciones, este proceso se repite hasta que el valor de la variable k sea igual al número de la variable N . El objetivo de establecer el horizonte de tiempo N igual a 12 para obtener la primera proyección es realizar la prueba en ese intervalo de tiempo con el criterio de evaluación MAPE, y si es porcentaje de error obtenido cumple con la condición de ser inferior al establecido como aceptable se procede a extender la proyección cambiando el valor de N a un número mayor a 12, y se vuelve a repetir el proceso, es decir, se carga el archivo **Pesos.m**, se realiza la propagación hacia adelante, el vector de entrada es realimentado con los nuevos datos proyectados generando nuevas proyecciones hasta que el valor de la variable k sea igual al de la variable N .

Figura 21: Diagrama de flujo 2 del proceso de proyección.



Fuente: Investigación.
Elaboración: Autores.

4.1.6.2. Desarrollo del programa en MATLAB.

La implementación de la RNA se realiza en el software MATLAB debido que brinda la facilidad para trabajar con vectores y matrices, lo cual permite rapidez en el procesamiento de datos. A continuación, se presentan algunas líneas de código con las que se realiza los procesos más sobresalientes.

La variable *Dataset* almacena la base de datos históricos correspondiente a los 72 meses en unidades de MW.

```
Dataset = [62.90 67.48 59.42 ...
```

Las siguientes líneas de código, *inputs* y *train* representan el número de neuronas de entrada y el intervalo de valores a seleccionar correspondientes al *Dataset* para el proceso de entrenamiento respectivamente.

```
inputs = 12;  
train = Dataset(1:60);
```

Para el ordenamiento de los datos se utilizan dos ciclos de repetición, el ciclo interno genera un arreglo con la cantidad de datos de entrada y son almacenados en la variable *inputTrain* que representa el arreglo de entrada *P* y el ciclo externo genera un vector de salida *T* almacenado en la variable *outputTrain*.

```
for i=1:length(train)-inputs  
    for j=1:inputs  
        inputTrain(j,i) = train(i+inputs-j);  
    end  
    outputTrain(i) = train(i+inputs);  
end
```

Esto es igual a tener los datos representados como en la Tabla 7.

La variable *nodeHidden* para el caso de una red neuronal con dos capas ocultas contiene un vector con la cantidad de neuronas de la primera y segunda capa, la variable *epoca* establece el número de iteraciones que se realizará para el entrenamiento y *alfa* corresponde al valor de la tasa de aprendizaje. Los valores para estas variables se obtienen mediante ensayos de prueba y error, buscando obtener un modelo adecuado.

```
nodeHidden = [2 2];  
epoca = 20000;  
alfa = 0.000192;
```

En el caso de la red neuronal con una sola capa el valor de la variable *nodeHidden* contendrá un escalar y no un vector, así como se muestra a continuación:

```
nodeHidden = 9;
```

La siguiente línea de código llama a la función de entrenamiento **neuralTrain** que requiere de las variables *inputTrain*, *outputTrain*, *nodeHidden*, *epoca* y *alfa* para realiza la propagación hacia adelante, cálculo de error cuadrático medio, cálculo de sensibilidades, retropropagación y actualización de pesos y bias. Las variables W_1 , b_1 , W_2 y b_2 contendrán los valores resultantes correspondientes a pesos y bias actualizados, mientras que el error cuadrático medio se guarda en la variable *error*.

```
[W1,b1,W2,b2,error]=neuralTrain(inputTrain,outputTrain,...  
                                nodeHidden,epoca,alfa);
```

Se realiza la propagación de la información hacia adelante generando las salidas por cada capa.

```
a1 = tansig(W1*inputTrain + b1);  
a2 = W2*a1 + b2;
```

La variable $a1$, guarda los datos de salida de la primera capa oculta, $a2$ guarda las salidas de la última capa con la que se realiza la gráfica de entrenamiento.

Finalmente, el comando “save” guarda las variables W_1 , b_1 , W_2 y b_2 en el archivo denominado **Pesos.m**, que será utilizado posteriormente en el proceso de proyección.

```
save('Pesos.mat','W1','b1','W2','b2')
```

El comando “load” permite cargar las variables previamente guardadas en el entrenamiento, pero ahora con pesos y bias ya definidos para realizar la proyección.

```
load Pesos.mat
```

La variable N corresponde al horizonte de tiempo hasta el cual se va a realizar la proyección. Inicialmente N igual a 12, se considera como la etapa de prueba. Considerando que el resultado que se obtenga del criterio de evaluación es aceptable, se cambia el valor de N por uno mayor a 12.

```
N = 12;
```

A continuación, se presenta las líneas del código en donde se realiza la realimentación de la entrada con cada proyección de manera recurrente.

```
for k=1:N
    a1 = tansig(W1*P + b1);
    a3(k) = W2*a1 + b2;

    for i=inputs:-1:2
        P(i) = P(i-1);
    end
    P(1) = a3(k);
end
```

El ciclo externo se repite hasta se igual al valor de horizonte de proyección establecido. Las variables *a1* y *a3* corresponden a las salidas de cada una de las capas, la propagación hacia adelante se realiza con los nuevos pesos y bias cargados anteriormente. Internamente se encuentra un ciclo que va a realizar la realimentación de la entrada tomando a la última salida como nueva entrada.

Las siguientes líneas de código implementan el criterio de evaluación MAPE con la cual se obtiene el valor porcentual del error medio absoluto para el proceso de prueba con los parámetros obtenidos en el entrenamiento.

```
real = [79.05 86.29 82.56 71.58 77.45 71.28  
a3desno = (a3*(maxData-minData)+minData);  
div = (abs(real-a3desno)./real);  
promedio = mean(div);  
mape = promedio*100;  
disp (mape + "%")
```

La variable *real* contiene un vector con los datos históricos seleccionados para la prueba en unidades de MW, la variable *a3desno* almacena los valores de proyección de la RNA cambiándolos a su unidad original (MW), el cálculo del MAPE se realiza en las variables *div*, *promedio* y *mape*, mediante el comando “disp” se visualiza en pantalla el valor calculado. Las variables *maxData* y *minData* contienen el valor máximo y mínimo de la serie temporal usados para desnormalizar los valores de proyección.

4.1.7. Diseño de la Red Neuronal Artificial.

Se presentan dos modelos de RNA con arquitecturas diferente, la figura 22 presenta el primer modelo de dos capas ocultas con dos neuronas en cada capa y una neurona en la capa de salida, mientras que, la figura 23 presenta el segundo modelo de una sola capa oculta con nueve neuronas y una neurona en la capa de salida, los dos modelos se plantean con doce neuronas de entrada.

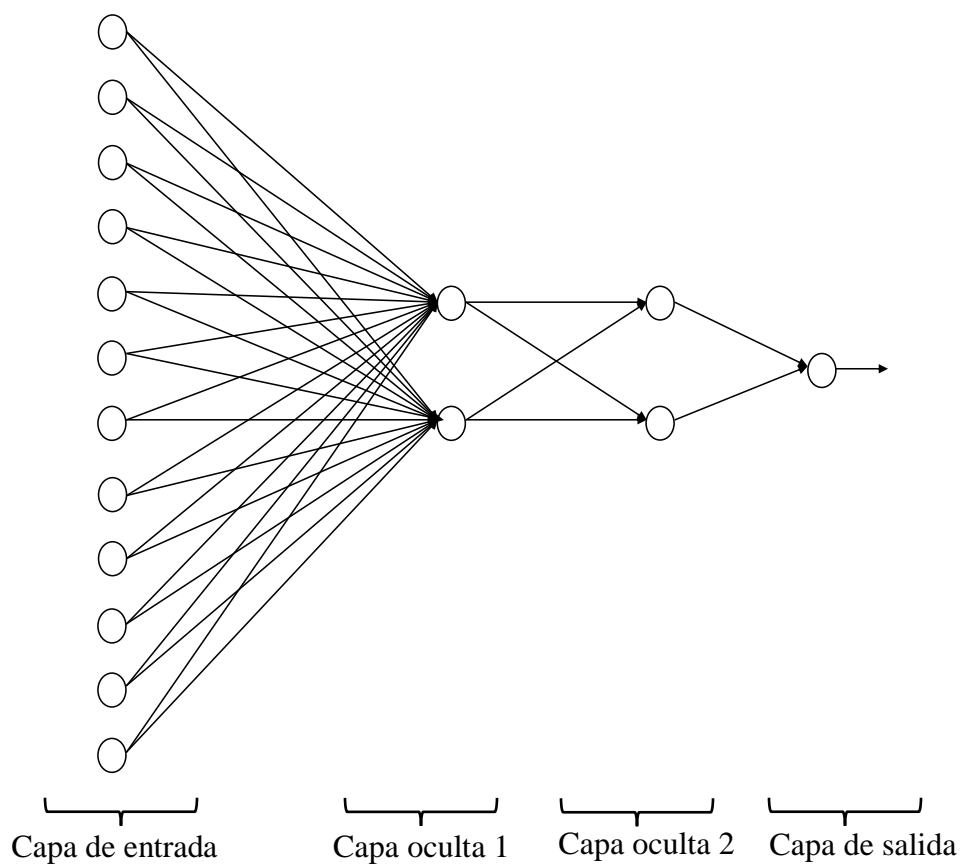
Tabla 9: Características de los modelos 1 y 2.

Parámetro	Modelo 1	Modelo 2
Número de neuronas de entrada	12	12
Número de neuronas de salida	1	1
Número de capas ocultas	2	1
Número de neuronas en primera capa oculta	2	9
Número de neuronas en segunda capa oculta	2	-
Función de activación capa oculta	Tangente hiperbólica	Tangente hiperbólica
Función de activación capa de salida	Lineal	Lineal
Algoritmo de aprendizaje	Backpropagation	Backpropagation
Aprendizaje	Supervisado	Supervisado

Fuente: Investigación.

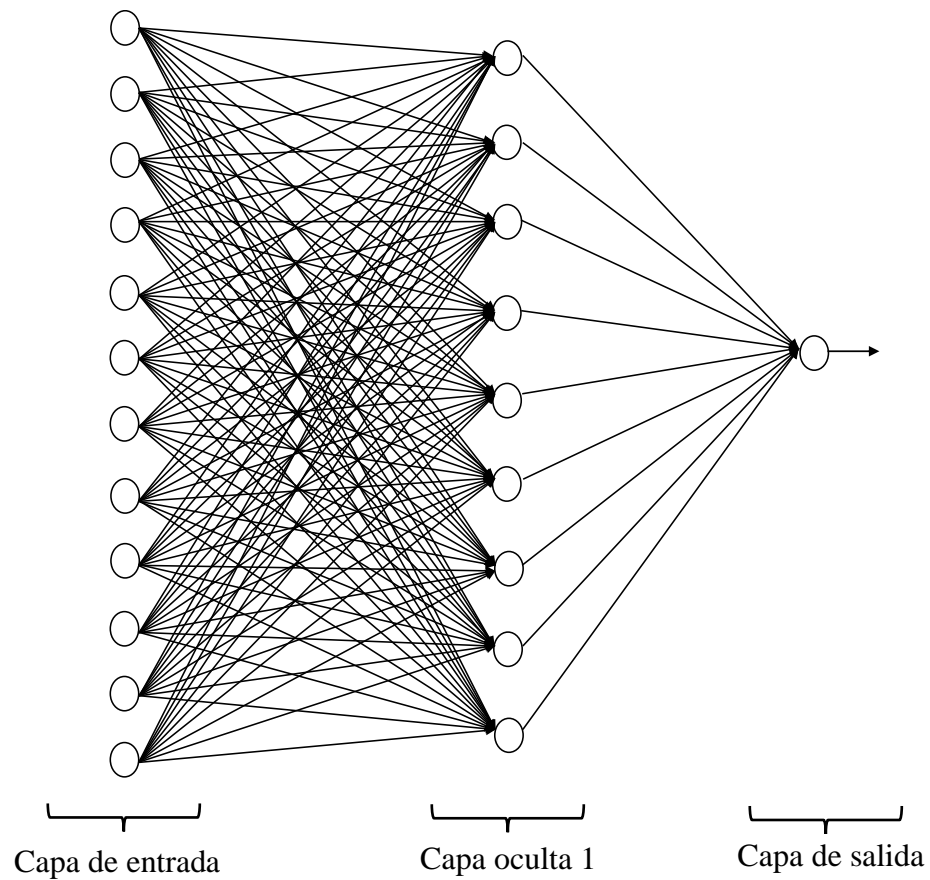
Elaboración: Autores.

Figura 22: Arquitectura del modelo 1.



Fuente: Investigación.
Elaboración: Autores.

Figura 23: Arquitectura del modelo 2.



Fuente: Investigación.
Elaboración: Autores.

Los parámetros con los cuales los modelos 1 y 2 de RNA han tenido resultados adecuados se presentan en la tabla 10:

Tabla 10: Parámetros de proyección de los modelos 1 y 2.

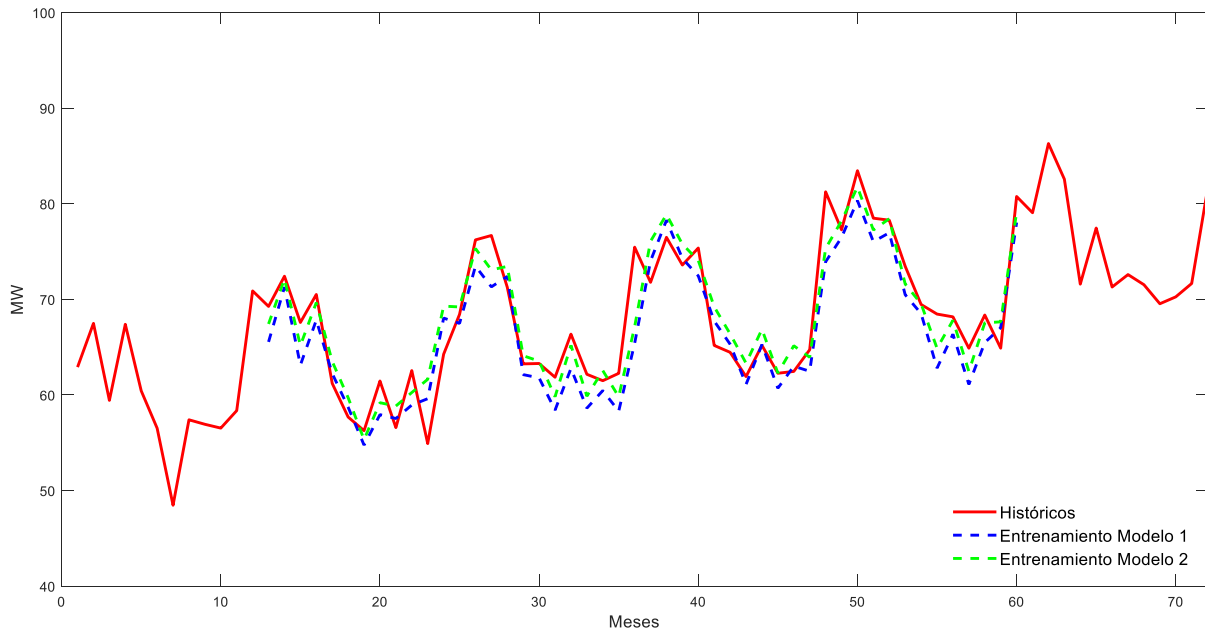
	Modelo 1	Modelo 2
<i>inputs</i>	12	12
<i>train</i>	1:60	1:60
<i>nodeHidden</i>	2;2	9
<i>epoca</i>	1000	1000
<i>alfa</i>	0.032	0.018
<i>prueba</i>	61:72	61:72
<i>N</i>	36	36

Fuente: Investigación.
Elaboración: Autores.

En la figura 24 se presentan los resultados de entrenamiento del modelo 1 (línea entrecortada azul) y modelo 2 (línea entrecortada verde) que inician en el mes número 13 a causa del desplazamiento que surge al presentar 12 neuronas de entrada, el resultado finaliza en el mes número 60 cubriendo el total de valores seleccionados para el entrenamiento, además se incluye el total de valores históricos (línea continua roja). Inicialmente las curvas de los modelos 1 y 2 no se asemejan a los valores históricos, pero a medida que más valores de entrenamiento recibe la RNA, se ajusta de mejor manera.

El eje horizontal presenta la cantidad de meses del total de valores históricos, es decir, 72 meses, mientras que, en el eje vertical se mantiene el intervalo de 40 a 100 en unidad de MW, se presenta en esta unidad debido a que es de mejor comprensión que al tratar con valores normalizados.

Figura 24: Proceso de entrenamiento de los modelos 1 y 2.



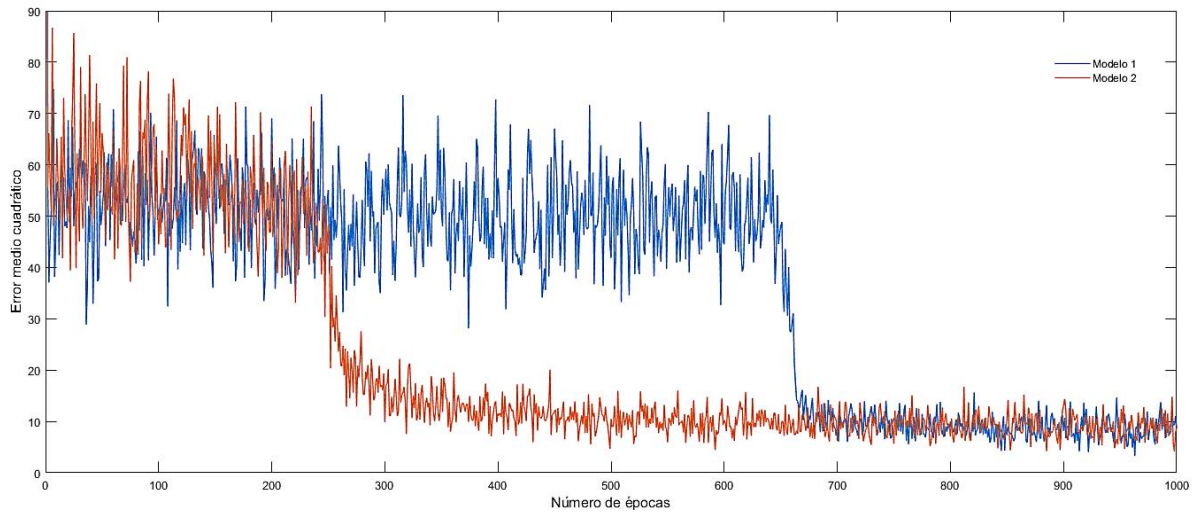
Fuente: Investigación.

Elaboración: Autores.

La figura 25 presenta el descenso del error durante el proceso de entrenamiento para los modelos 1 y 2. El error se calcula como la diferencia entre el valor esperado y el valor estimado, en base a este error se calculan las sensibilidades y se actualización los pesos y bias después de cada iteración, cabe indicar que cuando se han ingresado el total de valores de entrada se completa una iteración.

El eje horizontal de la figura 25 señala la cantidad de iteraciones que se han realizado, mientras que el eje vertical muestra la diferencia existente entre cada punto de la serie temporal de datos históricos y la proyección de los modelos 1 y 2 al finalizar cada iteración.

Figura 25: Decrecimiento del error de entrenamiento de los modelos 1 y 2.



Fuente: Investigación.

Elaborado: Autores.

Con la finalidad de interpretar de mejor manera los resultados, a continuación, se presentan los datos extraídos de la etapa de entrenamiento de los dos modelos y los datos históricos utilizados durante este proceso.

Tabla 11: Error cuadrático medio obtenido en MW del entrenamiento de los modelos 1 y 2.

		MODELO 1	MODELO 2	MODELO 1	MODELO 2
HISTÓRICOS		ENTRENAMIENTO	ENTRENAMIENTO	DIFERENCIA AL CUADRADO	
MES	MW	MW	MW	MW	MW
13	69.25	67.22	67.51	4.13	3.02
14	72.41	72.08	72.32	0.11	0.01
15	67.57	64.56	65.29	9.08	5.19
16	70.50	69.03	71.05	2.15	0.31
17	61.20	62.93	64.04	3.01	8.08
18	57.68	59.68	60.42	3.98	7.49
19	56.25	55.96	55.22	0.08	1.06
20	61.45	59.18	59.81	5.13	2.69
21	56.57	58.70	59.04	4.55	6.08

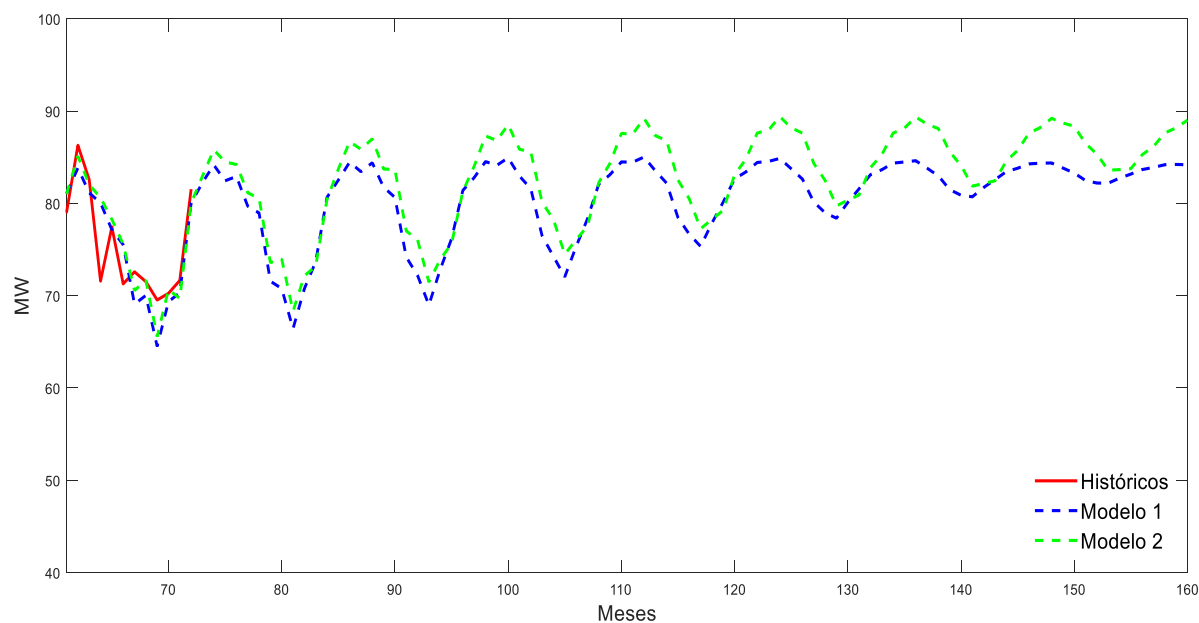
22	62.54	60.34	60.52	4.82	4.06
23	54.90	61.47	61.80	43.19	47.57
24	64.25	69.19	69.70	24.42	29.68
25	68.42	68.60	68.52	0.03	0.01
26	76.21	74.98	75.63	1.51	0.33
27	76.67	72.22	73.15	19.77	12.40
28	71.25	72.90	73.47	2.73	4.95
29	63.24	63.20	64.98	0.00	3.02
30	63.28	63.12	64.15	0.02	0.75
31	61.84	59.66	60.29	4.75	2.40
32	66.35	64.01	66.06	5.48	0.08
33	62.15	59.52	59.87	6.91	5.19
34	61.48	61.65	63.29	0.03	3.27
35	62.25	59.90	60.03	5.54	4.93
36	75.45	67.44	67.67	64.15	60.48
37	71.77	76.13	76.38	19.01	21.26
38	76.48	78.47	78.24	3.97	3.11
39	73.58	74.60	75.27	1.03	2.85
40	75.36	73.63	74.84	2.98	0.27
41	65.17	68.72	69.63	12.62	19.88
42	64.45	66.06	66.56	2.58	4.43
43	61.91	62.34	64.07	0.19	4.68
44	65.18	66.12	67.50	0.89	5.36
45	62.26	61.55	62.40	0.50	0.02
46	62.45	64.73	65.68	5.20	10.44
47	64.72	64.02	64.43	0.49	0.09
48	81.24	75.16	75.33	36.97	34.97
49	77.26	78.12	78.19	0.74	0.87
50	83.45	80.79	81.20	7.07	5.05
51	78.48	76.63	76.71	3.43	3.13
52	78.27	77.82	79.00	0.20	0.53
53	73.45	71.12	71.90	5.45	2.41
54	69.45	69.33	69.71	0.02	0.07
55	68.45	63.82	65.89	21.43	6.57
56	68.16	67.05	68.15	1.22	0.00
57	64.88	62.05	62.93	8.03	3.78
58	68.35	67.33	68.20	1.05	0.02
59	64.88	68.68	68.09	14.45	10.31
60	80.75	78.73	79.03	4.08	2.97
Error cuadrático medio obtenido en MW:				7.69	7.42

Fuente: Investigación.
Elaborado: Autores.

Los resultados de la tabla 11 reflejan que el error cuadrático medio obtenido en la época número 1000 (última época) es de 7.69 MW para el Modelo 1 y 7.42 MW para el Modelo 2, que corresponden a las iteraciones de los 48 patrones de entrenamiento presentados a la RNA.

El resultado de los procesos de prueba y proyección se muestra en la figura 26 para el modelo 1 (línea entrecortada azul) y modelo 2 (línea entrecortada verde). El tramo de curva comprendido entre el mes número 61 hasta el mes número 72 corresponde al proceso de prueba de los dos modelos donde también se incluye los valores históricos (línea continua roja) para este proceso, a partir del mes número 73 se presenta la proyección de los dos modelos para un horizonte de tiempo N igual a 100, con la finalidad de analizar la tendencia de cada uno de los modelos en el tiempo.

Figura 26: Prueba y proyección del modelo 1 y 2 para $N = 100$.

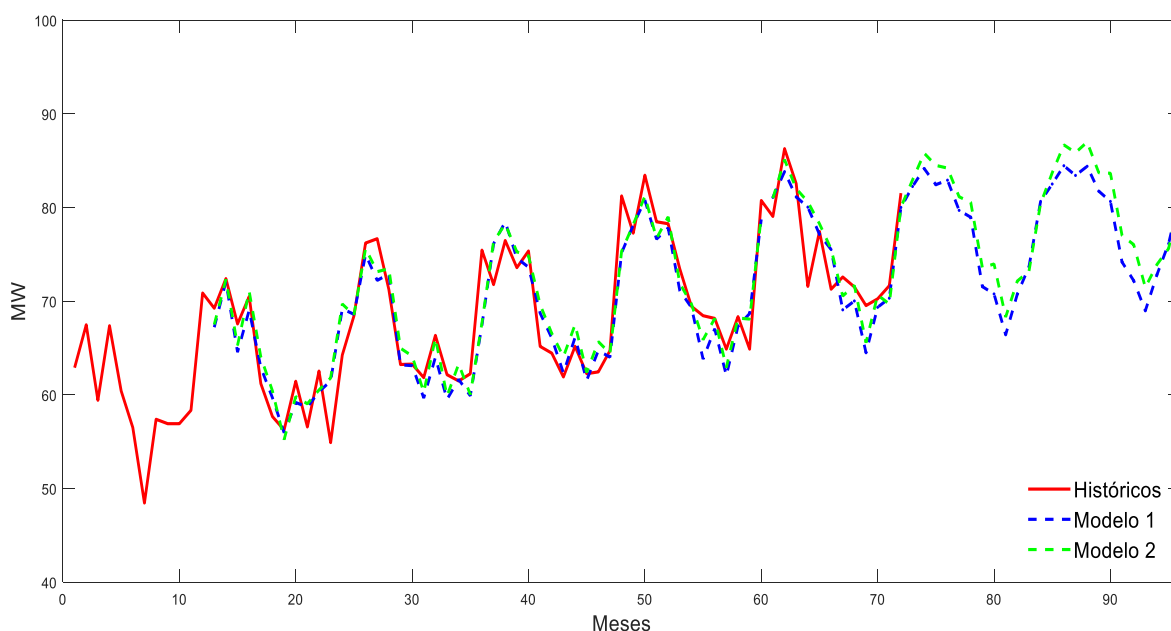


Fuente: Investigación.
Elaboración: Autores.

En la proyección de los dos modelos se aprecia que el modelo 2 presenta mayor tendencia en el tiempo en comparación con el modelo 1 que tiende a perderse más rápido.

Considerando que tanto el modelo 1 como el modelo 2 presentan una proyección adecuada hasta 24 meses después de la prueba, en la figura 27 se presenta el resultado de entrenamiento que inicia en el mes número 13 hasta el mes número 60, el resultado de la proyección para el proceso de prueba que va desde el mes número 61 hasta el mes número 72 y la proyección con horizonte de tiempo $N = 36$ que inicia en el mes número 73 hasta el mes número 96.

Figura 27: Proceso de entrenamiento, prueba y proyección para los modelos 1 y 2 con $N = 36$.



Fuente: Investigación.
Elaborado: Autores.

4.1.8. Validación de los modelos.

La tabla 12 presenta los datos de proyección normalizados y desnormalizados (MW) del modelo 1 y modelo 2. Los meses comprendidos en el intervalo del mes número 61 al mes número 72 corresponden al año 2008, y son usados en el proceso de prueba, mientras que, los meses que se encuentran en el intervalo del mes número 73 al mes número 96 corresponden a los años 2009 y 2010, es decir, los 36 meses proyectados.

Tabla 12: Proyección normalizada y desnormalizada de los modelos 1 y 2.

Año	Mes	Proyección normalizada		Proyección desnormalizada	
		Modelo 1	Modelo 2	Modelo 1	Modelo 2
2008	61	0.96	0.96	81.09	81.00
	62	1.05	1.08	83.83	85.11
	63	0.97	0.99	81.14	82.02
	64	0.93	0.95	80.04	80.62
	65	0.85	0.88	77.14	78.27
	66	0.80	0.80	75.52	75.57
	67	0.61	0.65	69.08	70.55
	68	0.64	0.69	70.02	71.68
	69	0.47	0.51	64.50	65.57
	70	0.62	0.66	69.35	70.66
	71	0.65	0.63	70.32	69.70
	72	0.93	0.93	80.01	80.04
2009	73	1.00	1.02	82.25	82.95
	74	1.06	1.10	84.16	85.80
	75	1.00	1.07	82.44	84.50
	76	1.02	1.06	82.92	84.21
	77	0.92	0.97	79.70	81.15
	78	0.90	0.95	78.96	80.54
	79	0.68	0.74	71.56	73.58
	80	0.66	0.75	70.77	73.97
	81	0.53	0.59	66.42	68.31
	82	0.66	0.70	70.75	72.15
	83	0.75	0.73	73.67	73.19
	84	0.95	0.94	80.65	80.42
2010	85	1.01	1.04	82.51	83.71
	86	1.06	1.13	84.46	86.69
	87	1.03	1.11	83.43	85.86
	88	1.06	1.14	84.40	86.99
	89	0.98	1.04	81.75	83.72
	90	0.95	1.04	80.65	83.66
	91	0.76	0.84	74.18	77.02
	92	0.70	0.82	72.19	76.04
	93	0.61	0.68	68.98	71.47
	94	0.72	0.75	72.83	73.92
	95	0.82	0.80	76.16	75.62
	96	0.97	0.97	81.41	81.30

Fuente: Investigación.

Elaboración: Autores.

4.1.8.1. Error de proyección.

En la tabla 13 se presenta el resultado del proceso de prueba de los modelos 1 y 2, consiste en aplicar el criterio de evaluación MAPE, para lo cual se dispone de los valores esperados que corresponde a los 12 meses del año 2008 en unidades originales (MW), las proyecciones de los modelos 1 y 2 se presenta en la misma unidad de medida y el resultado de aplicar el criterio MAPE se expresa en porcentaje.

Tabla 13: Resultado del cálculo del MAPE para los modelos 1 y 2.

		Históricos	Proyección		MAPE	
Año	Mes	MW	Modelo 1	Modelo 2	MAPE 1	MAPE 2
2008	1	79.05	81.09	81.00	2.58	2.47
	2	86.29	83.83	85.11	2.86	1.36
	3	82.56	81.14	82.02	1.72	0.65
	4	71.58	80.04	80.62	11.82	12.63
	5	77.45	77.14	78.27	0.40	1.05
	6	71.28	75.52	75.57	5.95	6.02
	7	72.58	69.08	70.55	4.82	2.79
	8	71.52	70.02	71.68	2.10	0.23
	9	69.53	64.50	65.57	7.23	5.70
	10	70.25	69.35	70.66	1.29	0.59
	11	71.65	70.32	69.70	1.85	2.73
	12	81.54	80.01	80.04	1.88	1.84
Error:					3.71%	3.17%

Fuente: Investigación.

Elaboración: Autores.

Se evidencia que el modelo 2 entrega menor porcentaje de error, siendo considerado como el modelo más adecuado en base al criterio MAPE.

El porcentaje de error aceptable en proyecciones de demanda eléctrica a mediano y corto plazo se encuentra establecido en 5%, reflejando un 95% de confianza en las proyecciones del modelo

a evaluar [28]. La importancia de evaluar los modelos recae en la correlación que presente el modelo y sus datos reales, y así realizar proyecciones ajustándose a estos datos.

Se considera que las proyecciones que se encuentren dentro del margen de 95% de confiabilidad se pueden considerar como “buenas”, la selección de un modelo de proyección y otro, depende del criterio de evaluación que se aplique, pudiendo ser el criterio de evaluación MAPE, MSE, RMSE, MAE, etc. [29].

4.2. Discusión.

Entre las consideraciones a tomar en cuenta para el desarrollo de la RNA es la forma en la que los datos van a ser ingresados, para ello, en el diagrama de flujo se indica la forma en que se consideran los datos, los datos de entrada forman un conjunto de patrones para la comparación de un valor futuro que se encuentra dentro de los datos históricos, así se va construyendo los vectores con los conjuntos de patrones que sirven para entrenar la red.

La normalización de los datos de entrada es de mucha importancia para evitar saturar el proceso de entrenamiento debido a que al trabajar con datos muy grandes requiere mucho tiempo procesar estos datos. Después de realizar los procesos de entrenamiento, prueba y proyección se realiza el proceso de desnormalización para devolverlos a las unidades de origen.

En este caso la división de los datos usados para el entrenamiento corresponde al 85% del total y el 15% restantes, para el proceso de prueba. Siendo este proceso fundamental para el desempeño y evaluación del modelo de red neuronal.

El procesamiento de la RNA se basa en ensayos de prueba y error hasta encontrar la mejor solución con la cual se obtenga el menor porcentaje de error en el proceso de prueba de la red. Los parámetros que predominan en el proceso de entrenamiento son los que se van modificando hasta encontrar el mejor modelo, para ello se puede cambiar el número de neuronas de entradas, número de capas ocultas, número de neuronas por cada capa oculta, funciones de activación, número de épocas para entrenamiento y valor de la tasa de aprendizaje.

Durante los ensayos de prueba y error se pueden observar que existen diferentes comportamientos que entregan como resultado la RNA. Los comportamientos que destacan son el sobreentrenamiento que entregan resultados de predicción totalmente erróneos, también se destaca que para un horizonte de proyección muy lejano la RNA pierde la tendencia de los datos y se convierte en una recta, también se conoce que, con muy pocas neuronas de entrada, la red no tiene los suficientes patrones de entrenamiento, por lo cual la misma resulta ineficiente para realizar las predicciones.

La prueba se realiza comparando los datos no usados para el proceso de entrenamiento que consisten en el intervalo de meses del 61 al 72 para los modelos 1 y 2, de los cuales se obtiene un porcentaje de error de 3.71% y 3.17% respectivamente a partir del criterio de evaluación MAPE.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones.

- Se concluye que el modelo del perceptrón multicapa con función de activación logística en la capa oculta y función de activación lineal en la capa de salida sirve como un aproximador universal de funciones.
- Para este caso de estudio, el cual presenta una sola variable de entrada, el número de neuronas de la capa de entrada es seleccionado mediante el análisis de la serie temporal identificando patrones de comportamiento que se repiten cada cierto tiempo.
- El modelo 1 es más sensible a cambios en la respuesta de la RNA por cambios en los parámetros de pesos y bias en comparación al modelo 2.
- El resultado del criterio de evaluación MAPE aplicado durante la etapa de prueba, no asegura que la proyección para los meses siguientes tendrá el mismo margen de error, debido a que la proyección resulta adecuada para los primeros meses de evaluación, pero pierde tendencia muy pronto.
- Aplicando el criterio de evaluación MAPE al modelo 1 y modelo 2, se concluye que el modelo 2 resulta ser el más adecuado debido a que entrega un porcentaje de error igual a 3.17%, siendo menor al obtenido del modelo 1, que es de 3.71%.
- La cantidad de neuronas en la capa oculta intervienen de forma significativa en los resultados de la RNA, debido a que esta cantidad de neuronas depende del problema a resolver, por lo que se debe realizar ensayos de prueba y error hasta encontrar el número adecuado de neuronas ocultas que lo resuelvan.

5.2. Recomendaciones.

- Se recomienda aplicar otro tipo de RNA con otros tipos de entrenamiento, con el objetivo de encontrar la mejor configuración para realizar proyecciones.
- Se recomienda el uso de lenguaje de programación Python, el cual posee librerías de RNA, donde el diseño y entrenamiento se realiza de manera sencilla y rápida.
- Se recomienda obtener la mayor cantidad de datos históricos para realizar el entrenamiento de la RNA, dejando un porcentaje mínimo del 15% para el proceso de validación.
- Se recomienda presentar las respuestas de la RNA en las unidades de medición correspondientes a su escala original, facilitando la interpretación de los resultados.

CAPÍTULO VI

BIBLIOGRAFÍA

- [1] D. Cabrera, “*Diseño de una red neuronal artificial para la predicción de demanda eléctrica*”, trabajo de fin de grado, Universidad Nacional de Loja, Loja, 2014.
- [2] L. Quisimalin, “*Predicción de demanda eléctrica en una barra de 69 kV del sistema eléctrico de subtransmisión de la Empresa Eléctrica Riobamba S.A*”, trabajo de fin de grado, Escuela Politécnica Nacional, Quito, 2017.
- [3] A. Ariza. “*Métodos utilizados para el pronóstico de energía eléctrica en sistemas de distribución*”, trabajo de fin de grado, Universidad Tecnológica de Pereira, Pereira, 2013.
- [4] M. Hagan, H. Demuth, M. Beale, “*Neural Network Desing*”, 2nd Edition, Madrid, 2014.
- [5] C. Tablada y G. Torres, “*Redes Neuronales Artificiales*”, Revista de Educación Matemática, vol. 24, octubre. 2009.
- [6] R. Flores y J. Fernández, “*Las redes neuronales artificiales fundamentos teóricos y aplicaciones prácticas*”, Netbiblo, Madrid, 2008.
- [7] P. Cruz, “*Inteligencia artificial con aplicaciones a la ingeniería*”, Primera edición, Alfaomega Grupo Editor, S.A., México, 2010.
- [8] W. Rivas y B. Mazón, “*Redes neuronales artificiales aplicadas al reconocimiento de patrones*”, Machala, UTMACH, 2010.
- [9] J. Freeman y D. Skapura, “*Redes Neuronales algoritmos, aplicaciones y técnicas de programación*”, Addison – Wesley / Diaz de Santos, 2010.
- [10] Y. López, “*Predicción de demanda eléctrica empleando redes neuronales artificiales*”, trabajo de grado, Universidad Central Martha Abreu de Las Villas, Santa Clara, 2009.

- [11] I. Flores, “*Modelo predictivo de la demanda de energía eléctrica en el campus central de la Universidad de El Salvador*”, tesis de grado, Universidad de El Salvador, San Salvador, 2018.
- [12] L. Llanos, A. Hoyos, F. Arias y J. Velázquez, “*Comparación del desempeño de funciones de activación en redes feedforward para aproximar funciones*”, Revista Avances en Sistemas e Informática, vol. 4, septiembre 2007.
- [13] A. Morera, “*Introducción a los modelos de redes neuronales artificiales El perceptrón simple y multicapa*”, tesis de grado, Universidad zaragoza, Zaragoza, 2018.
- [14] A. Palmer, J. Montaña y R. Jiménez, “*Tutorial sobre redes neuronales artificiales: El perceptrón multicapa*”, Universidad de las Islas Baleares, 2001.
- [15] J. Marín, “*Introducción a las redes neuronales aplicadas*”, Universidad Carlos III de Madrid, 2012.
- [16] M. Norgaard, O. Ravn, N. Poulsen y L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*, 1st Edition, 2003.
- [17] B. Caicedo y J. López, "Perceptrón multicapa y algoritmo backpropagation" en *Una Aproximación Práctica a Las Redes Neuronales Artificiales*, 2009.
- [18] A. Bajaan, “*Pronóstico de producción hidroeléctrica del Ecuador usando redes neuronales*”, tesis de grado, Universidad de Guayaquil, 2019.
- [19] R. López y J. Fernández, “*Las Redes Neuronales Artificiales Fundamentos Teóricos y Aplicaciones Prácticas*”, Netbiblo, 2008.
- [20] C. Gonzales, “*Predicción de la demanda eléctrica horaria mediante redes neuronales*”, Universidad de Oviedo, 2004.

- [21] D. Ortiz, “*Aplicación de redes neuronales artificiales en el pronóstico de la demanda eléctrica a corto plazo en el SNP*”, tesis de grado, Universidad Politécnica Salesiana, 2014.
- [22] P. Viñuela y I. Galván, “*Redes neuronales artificiales Un enfoque práctico*”, Madrid, Pearson: 2004.
- [23] D. Verdugo, M. Lorenzo, C. López y J. Hernández, “*Estudio de metodologías empleadas para el pronóstico de demanda eléctrica*”, Culiacán, 2008.
- [24] J. Velázquez, C. Franco y H. García, “*Un modelo no lineal para la predicción de la demanda mensual de electricidad en Colombia*”, Estudios Gerenciales vol. 25 pp. 37–54, septiembre 2009.
- [25] J. Oscullo y L. Haro, “*Pronóstico de la demanda diaria del sistema nacional interconectado utilizando redes neuronales artificiales*”, Revista Politécnica, vol. 38, pp. 77 septiembre 2016.
- [26] J. Mendoza, “*Pronóstico de carga a corto plazo utilizando redes neuronales Artificiales*”, Scientia et Technica, vol. 3, enero 2006.
- [27] M. Salazar y M. Cabrera, “*Pronóstico de demanda por medio de redes neuronales artificiales*”, Nuevo León, 2007.
- [28] J. Barrezueta y A. Encalada,” *Análisis de factores que afectan a la demanda eléctrica y su estimación. Sector residencial del área de concesión de la empresa CENTRO SUR. C. A.*”, tesis de grado, Cuenca, 2014.
- [29] E. Mejía y S. Gonzales, “*Predicción del consumo de energía eléctrica residencial de la Región Cajamarca mediante modelos Holt -Winters*”, Cajamarca, 2019.

CAPÍTULO VII

ANEXOS

Anexo 1: Código del archivo entrenamiento.m

```
clc
clear
close all

%Base de datos (serie temporal)
Dataset = [62.9 67.48 59.42 67.38 60.42 56.51 48.45 57.39 56.92 56.52 58.35
70.87 69.25 72.41 67.57 70.5 61.2 57.68 56.25 61.45 56.57 62.54 54.9 64.25
68.42 76.21 76.67 71.25 63.24 63.28 61.84 66.35 62.15 61.48 62.25 75.45
71.77 76.48 73.58 75.36 65.17 64.45 61.91 65.18 62.26 62.45 64.72 81.24
77.26 83.45 78.48 78.27 73.45 69.45 68.45 68.16 64.88 68.35 64.88 80.75
79.05 86.29 82.56 71.58 77.45 71.28 72.58 71.52 69.53 70.25 71.65 81.54];

%Valor máximo y mínimo de la serie temporal
maxData = max(Dataset);
minData = min(Dataset);

%Normalizacion de la serie temporal
Datasetnor = (Dataset-minData)/(maxData-minData);

%Cantidad de entradas
inputs = 12;

%Cantidad de datos para entrenamiento
train = Datasetnor(1:60);

%Bucle para acomodar la matriz de entrada y vector de salida u objetivos
for i=1:length(train)-inputs

    for j=1:inputs
        inputTrain(j,i) = train(i+inputs-j);
    end
    outputTrain(i) = train(i+inputs);
end

%Parámetros de entrenamiento
nodeHidden = 9;
epoca = 1000;
alfa = 0.018;

%Llama a la función de entrenamiento neuralTrain
[W1,b1,W2,b2,error,errorAcumulado,Tdesnor,a2desnor,a2,e2] =
neuralTrain(inputTrain,outputTrain,nodeHidden,epoca,alfa);

%Plotea error
figure
plot(error)
title ('Resultado del descenso del MSE')
xlabel ('Número de épocas','FontName','Arial','FontSize',12)
ylabel ('Error medio cuadrático','FontName','Arial','FontSize', 12)
```

```

%Propagación hacia adelante
a3 = tansig(W1*inputTrain + b1);
a4 = W2*a3 + b2;

%Plotea históricos y resultado de entrenamiento normalizados
figure
plot(Datasetnor,'r','LineWidth',2),
hold on,plot(13:length(a4)+12,a4,'b--','LineWidth',2)
legend('Históricos','Entrenamiento')

%Guardar pesos y bias
save('Pesos.mat','W1','b1','W2','b2')

```

Anexo 2: Código de la función neuralTrain.m

```

function
[W1,b1,W2,b2,error,errorAcumulado,Tdesnor,a2desnor,a2,e2]=neuralTrain(P,T,n
odeHidden,epoca,alfa)

Q=size(P,2); % Cantidad de ejemplos

inputs=size(P,1); % Cantidad de entradas
outputs=size(T,1); % Cantidad salidas

W1=rand(nodeHidden(1),inputs); %Pesos capa de salida aleatoriamente
b1=rand(nodeHidden(1),1); % Bias de capa de salida inician aleatoriamente

W2=rand(outputs,nodeHidden(1)); %Pesos capa oculta inician aleatoriamente
b2=rand(outputs,1); % Bias de capa oculta inician aleatoriamente

maxData=86.29; %Valor máximo de la serie temporal
minData=48.45; %Valor mínimo de la serie temporal

for Epocas = 1:epoca % Cantidad de repeticiones total del bucle

    errorAcumulado = zeros(outputs,1); % Suma de los errores

    for i = 1:Q % Algoritmo de optimizacion

        q=randi(Q); % Selección aleatoria de entradas

        % Propagación hacia adelante
        a1 = tansig(W1*P(:,q) + b1);
        a2 = W2*a1 + b2;

        %Desnormalización para evaluar el MSE, PARA GRAFICAR EL ERROR
        Tdesnor=(T(:,q)*(maxData-minData)+minData);
        a2desnor=(a2*(maxData-minData)+minData);

        %Cálculo del error desnormalizado
        e2=Tdesnor-a2desnor;
    end
end

```

```

    %Retropropagación de las sensibilidades
    e = T(:,q)-a2; % error de proyección
    s2 = -2*e; % sensibilidad de la última capa
    s1 = diag((1-a1.^2))*W2'*s2; % sensibilidad de la capa oculta

    % Actualización de pesos sinápticos y bias
    W2 = W2 - alfa*s2*a1';
    b2 = b2 - alfa*s2;
    W1 = W1 - alfa*s1*P(:,q)';
    b1 = b1 - alfa*s1;

    %actualización del error Acumulado
    errorAcumulado = e2^2 + errorAcumulado;

end
% MSE de la última epoca, ESTE VALOR SERÁ GRÁFICADO
error(Epocas) = errorAcumulado/Q;
end
end

```

Anexo 3: Código del archivo prediccion.m

```

%Base de datos (serie temporal)
Dataset = [62.90 67.48 59.42 67.38 60.42 56.51 48.45 57.39 56.92 56.52
58.35 70.87 69.25 72.41 67.57 70.5 61.2 57.68 56.25 61.45 56.57 62.54 54.9
64.25 68.42 76.21 76.67 71.25 63.24 63.28 61.84 66.35 62.15 61.48 62.25
75.45 71.77 76.48 73.58 75.36 65.17 64.45 61.91 65.18 62.26 62.45 64.72
81.24 77.26 83.45 78.48 78.27 73.45 69.45 68.45 68.16 64.88 68.35 64.88
80.75 79.05 86.29 82.56 71.58 77.45 71.28 72.58 71.52 69.53 70.25 71.65
81.54];

%Valor máximo y mínimo de la serie temporal
maxData = max(Dataset);
minData = min(Dataset);

%Normalizacion de la serie temporal
Datasetnor = (Dataset-minData)/(maxData-minData);

%Carga el archivo que contiene los pesos y bias del entrenamiento
load Pesos.mat

%Cantidad de entradas (iguales a las del entrenamiento)
inputs = 12;

%Define el vector de entrada P para proyección
i = 1;
for k=60:-1:61-inputs
    P(i,1) = Dataset(k);
    i=i+1;
end

%Horizonte de tiempo a proyectar
N = 12;

```

```

%Bucle para ingresar la salida como nueva entrada
for k=1:N
    a1 = tansig(W1*P + b1);
    a3(k) = W2*a1 + b2;
    for i=inputs:-1:2
        P(i) = P(i-1);
    end
    P(1) = a3(k);
end

%Plotea históricos (mes 61 al 72) y proyección según N
figure
plot(Datasetnor,'r','LineWidth',2),
hold on,plot(61:length(a3)+60,a3,'b--','LineWidth',2)
legend('Históricos','Proyección')

%Cálculo del error MAPE para evaluar proyección
real=Dataset(61:end); %Selección de los últimos 12 valores de la ST
a3sel=a3(1:12); %Selección de los 12 primeros valores de la proyección
a3desn=(a3sel*(maxData-minData)+minData); %Desnormalización de la salida
div=(abs(real-a3desn)./real); % abs(x-xi)/x
promedio=mean(div); % Promedio
mape=promedio*100; % Convertir en %
disp(mape+" %") % Presentación en pantalla

```