



UNIVERSIDAD DE CÓRDOBA



Título

***DESARROLLO Y EVALUACIÓN DE MODELOS PREDICTIVOS
DE CONSUMO DE AGUA BASADOS EN REDES NEURONALES
ARTIFICIALES. APLICACIÓN EN UN SECTOR ESPECÍFICO DE
LA RED DE ABASTECIMIENTO DE EMACSA***

MÁSTER EN TECNOLOGÍA DEL AGUA EN INGENIERÍA CIVIL

Alumno: Diego José Galán López

Director: Javier Estévez Gualda

Co-directora: Amanda P. García Marín

Julio 2017

TÍTULO

**“DESARROLLO Y EVALUACIÓN DE MODELOS PREDICTIVOS
DE CONSUMO DE AGUA BASADOS EN REDES NEURONALES
ARTIFICIALES. APLICACIÓN EN UN SECTOR ESPECÍFICO DE
LA RED DE ABASTECIMIENTO DE EMACSA”**

TITLE

**“DEVELOPMENT AND EVALUATION OF PREDICTIVE MODELS OF
WATER CONSUMPTION BASED ON ARTIFICIAL NEURAL NETWORKS.
APPLICATION ON A “EMACSA’S” SPECIFIC SECTOR OF THE WATER
SUPPLY NETWORK”**

Resumen

Los nuevos sistemas de gestión de recursos llevan integrado el llamado internet de las cosas, orientado hacia un modelo de “Smart cities”. En este punto, donde se genera una gran cantidad de datos, resulta crucial la necesidad de validarlos como requisito previo a su uso en las diferentes gestiones que los requieran. El presente trabajo se centra en la estimación de consumos de agua potable de la red de abastecimiento de la Empresa Municipal de Aguas de Córdoba, a través del desarrollo y evaluación de diferentes modelos predictivos basados en redes neuronales artificiales (RNA). Los modelos empleados han sido el NFIT (ajuste entradas-salida), NARX (Red Neuronal No lineal Autorregresiva con factores externos) y aprendizaje profundo (Red Neuronal Recurrente LSTM). Empleando diversas arquitecturas y evaluando diferentes sets de datos para su validación, se han calculado una serie de estadísticos para evaluar su rendimiento. En base a los resultados obtenidos, la RNA LSTM (aprendizaje profundo) mostró el mejor comportamiento a la hora de pronosticar los consumos de agua con un horizonte de una hora, para todos los contadores analizados. El NARX resultó ser más preciso que el NFIT para todos los casos, a excepción del contador de mayor calibre.

Palabras clave: Redes neuronales artificiales, análisis de series temporales, control de calidad, validación, predicción, consumos, aprendizaje profundo.

Abstract

The new systems of resource management have integrated the so-called Internet of Things, oriented towards a model of "Smart cities". At this point, where a large amount of data is generated, the need to validate them as a prerequisite to their use in the different procedures that require them is crucial. The present work focuses on the estimation of drinking water consumption of the supply network of "EMACSA", through the development and evaluation of different predictive models based on artificial neural networks (ANN). The models used were the NFIT (input-output adjustment), NARX (Neural Line Non-linear Autoregressive with external factors) and deep learning (Recurrent Neural Network LSTM). Using various architectures and evaluating different sets of data for validation, a series of statistics have been calculated to evaluate their performance. Based on the results obtained, the LSTM RNA (deep learning) showed the best behaviour when forecasting the consumption of water with a horizon of one hour, for all the meters analyzed. The NARX proved to be more accurate than the NFIT for all cases, with the exception of the higher caliber counter.

Keywords: Artificial neural networks, analysis of time series, quality control, validation, consumption, prediction, deep learning.

Javier Estévez Gualda y Amanda P. García Marín, ambos Profesores Titulares de Universidad del área de Proyectos de Ingeniería del Departamento de Ingeniería Rural de la Universidad de Córdoba,

INFORMAN:

Que el Trabajo Fin de Máster titulado “DESARROLLO Y EVALUACIÓN DE MODELOS PREDICTIVOS DE CONSUMO DE AGUA BASADOS EN REDES NEURONALES ARTIFICIALES. APLICACIÓN EN UN SECTOR ESPECÍFICO DE LA RED DE ABASTECIMIENTO DE EMACSA” y realizado por el alumno Diego J. Galán López con DNI: 30985310D, reúne los requisitos necesarios para su defensa, dentro del Máster Universitario de Postgrado “Tecnología del Agua en Ingeniería Civil”.

Y para que así conste, lo firman en Córdoba, julio de 2018.

Prof. Javier Estévez Gualda

Profa. Amanda P. García Marín

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1. Introducción y antecedentes.....	9
1.1. Ciclo integral urbano del agua en la ciudad de Córdoba	9
1.2. Sectorización	11
1.3. Descripción actual del sistema de telelectura.	13
2. Objetivos.	17
3. Fuente de Datos.....	18
4. Metodología.....	20
4.1. Control de calidad y pre-procesado de datos brutos.....	20
4.1.1. Test de rangos.....	20
4.1.2. Test de consistencia.	20
4.1.3. Detección de anomalías.	22
4.2. Fundamento teórico de las redes neuronales.....	23
4.2.1. Fundamentos biológicos de las redes neuronales.....	23
4.2.2. Redes neuronales artificiales.....	24
4.2.3. Redes perceptron multicapa.....	27
4.2.4. Redes neuronales de base radial.	39
4.2.5. Redes neuronales modelo NAR y NARX.....	49
4.2.6. Redes neuronales modelo Deep Learning: Red Neuronal Recurrente LSTM.....	51
4.3. Redes neuronales en Matlab.	53
4.3.1. Redes Perceptron Multicapa.	53
4.3.2. Redes neuronales de base radial.	55

4.3.3. Redes neuronales NARX.....	56
4.3.4. Redes neuronales recurrente LSTM.....	58
4.4. Parámetros estadísticos.....	61
5. Resultados y Discusión.	63
5.1. Datos depurados.	63
5.2. Evaluación de los diferentes modelos.....	66
5.2.1. Modelo de ajuste NFIT.	67
5.2.2. Modelo NARX.	69
5.2.3. Función de Base Radial.....	71
5.2.4. Deep Learning: Red Neuronal Recurrente LSTM.....	71
6. Conclusiones.....	79
7. Líneas futuras de mejora.....	81
8. Bibliografía.	83
Anejo I.- Características técnicas del contador.	86

ÍNDICE DE TABLAS

Tabla 1.- Detalle de la instalación de contadores en telelectura con prueba piloto.	14
Tabla 2.- Número de datos brutos de los contadores estudiados.	18
Tabla 3.- Test de rangos establecido para la depuración.....	20
Tabla 4.- Resultados de la clasificación mediante RNA. Distribución según los usos del consumo total en periodo Enero-2008 a Julio 2015.....	21
Tabla 5.- Valores establecidos para detectar anomalías en test de consistencia.	22
Tabla 6.- Evaluación de la depuración de la serie de datos por contador.....	63

Tabla 7.- Tabla muestra de estructuración de datos.	64
Tabla 8.- Resultados de los parámetros estadísticos para el modelo NFIT... ..	68
Tabla 9.- Resultados de los parámetros estadísticos para el modelo NARX.. .	70
Tabla 10.- Valores del error medio cuadrático en el modelo LSTM.....	78

ÍNDICE DE FIGURAS

Figura 1.- Ciclo integral del agua en la ciudad de Córdoba.....	10
Figura 2.- Red de abastecimiento de la ciudad de Córdoba.	11
Figura 3.- Red de abastecimiento sectorizada.	12
Figura 4.- Distinción tipo de red de abastecimiento (Izq.) y materiales de la red de abastecimiento (Dcha.).....	13
Figura 5.- Contadores en batería (Izq.) y contadores individuales (Dcha.)..	15
Figura 6.- Estructura de la red de telelectura WF868.....	15
Figura 7.- Estructura de la red de telelectura VHF169.	16
Figura 8.- Dimensiones de contador Altair V4.....	19
Figura 9.- Modelos de contador Altair V4.....	19
Figura 10.- Esquema de funcionamiento de una neurona.....	25
Figura 11.- Esquema de la arquitectura de una red..	26
Figura 12.- Esquema de red neuronal de base radial..	40
Figura 13.- Esquema de la arquitectura general del modelo neuronal NAR.. ..	49

Figura 14.- Esquema de la arquitectura general del modelo NARX.	51
Figura 15.- Diagrama de la arquitectura de una red LSTM simple.	59
Figura 16.- Consumos del contador C11ALC578265 en los meses de Noviembre- Diciembre de los años 2015 y 2016..	65
Figura 17.- Valores de consumo entre los meses Marzo a Mayo del contador C11ALC578265.	65
Figura 18.- Valores de consumo del contador C12ALC155545 durante la serie de datos.	66
Figura 19.- Esquema del proceso de funcionamiento de la red neuronal NFIT. Fuente: Elaboración propia.	67
Figura 20 (A y B).- Modelo entrenado con NFIT con 4 inputs para contadores de calibre 15mm (A) y 20 mm (B).	68
Figura 21.- Esquema del proceso de funcionamiento de la red neuronal NARX..	69
Figura 22 (A y B).- Modelo entrenado con NARX con 4 inputs para contadores de calibre 15 mm (A) y 20 mm (B).	70
Figura 23.- Aplicación RNN LSTM para el contador C11ALC153667 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D).	72
Figura 24.- Aplicación de RNN LSTM para contador C12ALC156544 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D).	73
Figura 25.- Aplicación de RNN LSTM para contador C12ALC155545 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D).	74



Figura 26.- Aplicación de RNN LSTM para contador C12ALC578265 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D)..

..... 75

Figura 27.- Aplicación de RNN LSTM para contador C12ALC583338 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D)..

..... 76

Figura 28.- Aplicación de RNN LSTM para contador C12ALC071969 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D)..

..... 77

1. Introducción y antecedentes.

Este proyecto se realiza en colaboración con la Empresa Municipal de Aguas de Córdoba (EMACSA). EMACSA fue creada en 1969 y hoy en día ha alcanzado un lugar de vanguardia dentro de las empresas del sector (Martínez, 2017). En la ciudad de Córdoba ha implantado diferentes sistemas de telelectura sobre la que se desarrolla el presente trabajo.

1.1. Ciclo integral urbano del agua en la ciudad de Córdoba

En la ciudad de Córdoba, el ciclo integral urbano del agua es gestionado por la Empresa Municipal de Aguas de Córdoba. Esta empresa creada en 1969 por el pleno del Ayuntamiento de Córdoba, inicia sus actividades en materia de abastecimiento de aguas en este mismo año y más tarde en 1982, tras la cesión de las instalaciones, por parte inicia la gestión de la red de alcantarillado, para así completar el ciclo integral del agua en Córdoba, entendiendo como tal, la captación, aducción, tratamiento, almacenaje y distribución del agua potable, la recogida de las aguas usadas y de lluvia en la red de alcantarillado, su depuración y posterior entrega a los cauces naturales.

En 1969, comienza a dar servicio la Estación de Tratamiento de Agua Potable (ETAP) de Villa Azul, destinada al tratamiento y potabilización del agua, y se aborda el recrecimiento del embalse del Guadalquivir, ya existente, principal fuente de abastecimiento de la ciudad. Se aumenta y mejora la red de abastecimiento, se asume la gestión de la red de alcantarillado, hasta entonces de gestión municipal, se amplía y mejora la ETAP de Villa Azul. Más tarde, en 1982, tras la cesión de las instalaciones por parte del Ayuntamiento de Córdoba, se inicia la gestión de la red de alcantarillado, siendo en 1991, cuando se da un paso importante al entrar en funcionamiento la Estación Depuradora de Aguas Residuales (EDAR) de La Golondrina, esencial para el saneamiento de la ciudad y una de las estaciones depuradoras con tecnología vanguardista hasta el momento en Europa (Romero, 2014).

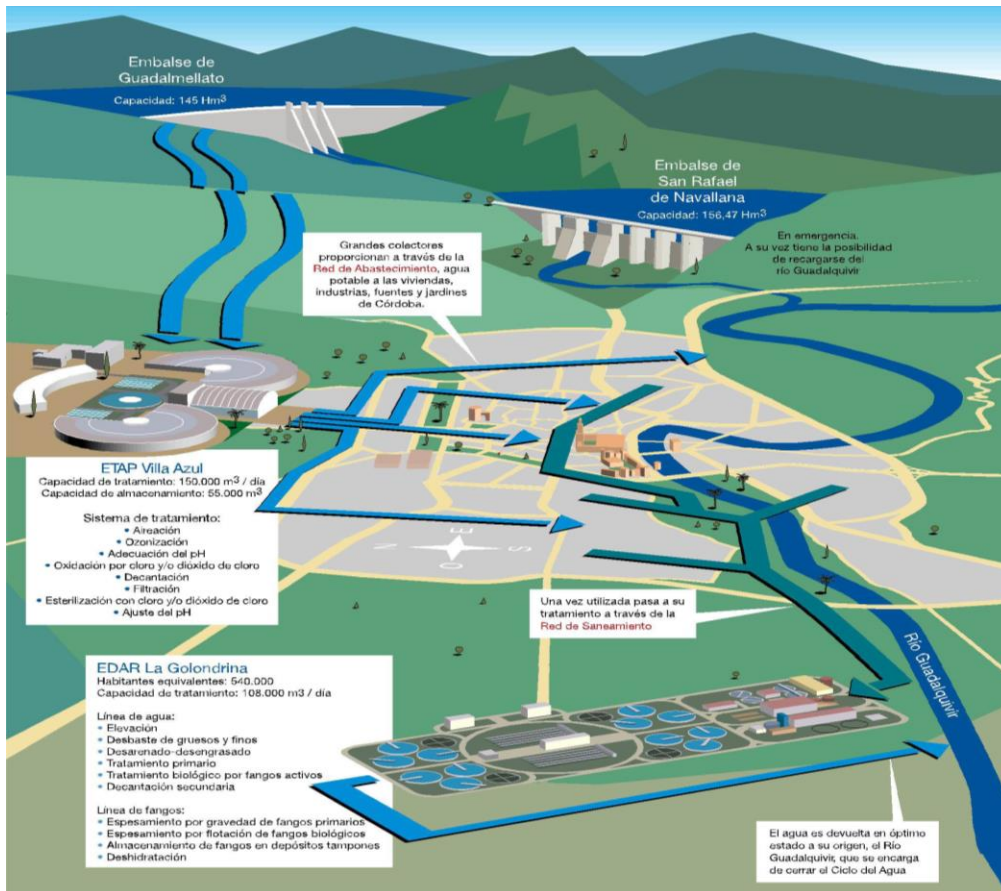


Figura 1.- Ciclo integral del agua en la ciudad de Córdoba. **Fuente:** Presentación “La gestión del servicio de aguas de Córdoba”.

En pocos años, EMACSA ha logrado situarse entre las primeras de España por la calidad de su servicio y continuos desarrollos, adelantándose a las exigencias que establecen las directivas comunitarias. Las transformaciones continúan hasta nuestros días y la ciudad cuenta con un ciclo integral del agua, la informática es el soporte esencial en los procedimientos de trabajo y las inversiones se traducen en modernas instalaciones que potencia el Servicio de Atención al Cliente. Y es que EMACSA ha puesto en práctica un completo y avanzado sistema de gestión por el que “Córdoba no gasta agua, sólo la utiliza”: el agua es tratada para su consumo, distribuida por toda la ciudad y, además, es depurada, con lo que se consigue completar su ciclo integral.

1.2. Sectorización

EMACSA siempre ha utilizado las tecnologías más vanguardistas en la gestión del ciclo urbano del agua y, en concreto, en la gestión de la red de abastecimiento. Prueba de ello, es la aplicación de las tecnologías de la información y la comunicación (Díaz et al., 2016) en la monitorización de la red mediante la instalación de cientos de sensores y el despliegue de telectura en red fija de los contadores domiciliarios o la utilización de sistemas informáticos dotados de los más vanguardistas algoritmos computacionales que implementan redes neuronales o inteligencia artificial.



Figura 2.- Red de abastecimiento de la ciudad de Córdoba. **Fuente:** Elaboración propia.

La sectorización de la red de abastecimiento de Córdoba, se desarrolló a partir de 2007 y se basó en la compartimentación tanto de la red de distribución como la red de transporte en porciones homogéneas que permitieran acotar los eventos que sucedieran en cada uno de ellos de forma aislada. Para ello, a cada porción de la red se le dotó de una media de 2 alimentaciones de agua dotadas cada una de ellas de medidor de caudal, medidor de presión y válvula motorizada

y telecontrolada con transmisión de datos en tiempo cuasireal. La reciente aparición de los medidores de agua inteligentes de alta resolución y análisis de datos avanzados permiten una nueva era de uso de los 'grandes datos' continuos generados por estas flotas de sensores para crear un sistema inteligente de gestión del agua urbana (Nguyen et. al, 2018). Esta configuración, que prioriza la garantía de suministro frente a averías, complicó exponencialmente el entendimiento de los eventos que sucedían en el interior del sector debido a la variabilidad que introducían los 2 puntos de alimentación.

Para salvar esta problemática, se estableció como proyecto estratégico de la corporación el despliegue masivo de telectura en red fija de los contadores domiciliarios y se desarrolló un sistema informático a medida del problema que permitía abordarlo de una forma asumible para el explotador.



Figura 3.- Red de abastecimiento sectorizada. **Fuente:** Plano sectores, EMACSA.

En la actualidad, EMACSA abastece a la población de Córdoba, 328.659 habitantes, de los cuales 87.436 son abonados, a través de una red mallada, de 1.142 km longitud aproximada en la que se pueden diferenciar los tramos de red de transporte y red de distribución y los diferentes materiales que la componen en los siguientes gráficos.

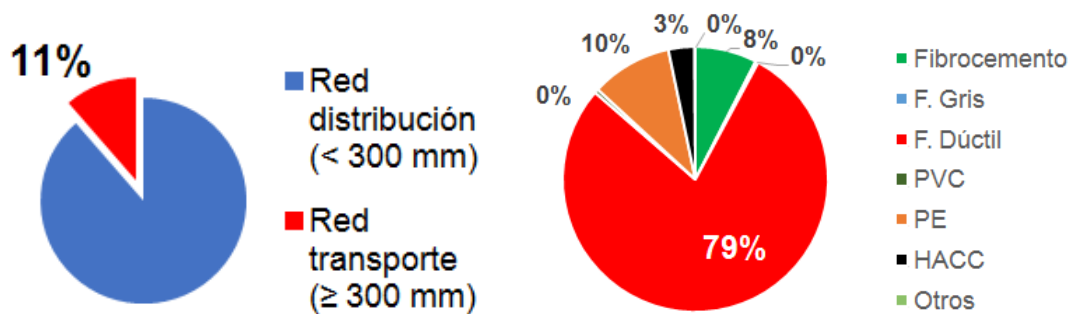


Figura 4.- Distinción tipo de red de abastecimiento (Izq.) y materiales de la red de abastecimiento (Dcha.)

Fuente: Presentación “Sistemas telelectura en EMACSA”

1.3. Descripción actual del sistema de telelectura.

Debido a su conveniencia, los sistemas automáticos de lectura de contadores se están convirtiendo en una solución cada vez más popular para el seguimiento remoto de la recolección de datos de consumo de recursos de servicios públicos (agua, gas, calor, electricidad) (Meskuotiene et al., 2016). En el campo de la dotación de sensores a la red de abastecimiento y el despliegue de telelectura en red fija de los contadores domiciliario, destacan como hitos los años 2003, en el que se instalaron los primeros equipos de telelectura de contadores y 2007 año en el que se comenzó la sectorización de la red de abastecimiento.

Como se ha mencionado anteriormente, en el año 2003 se comenzó la instalación de telelectura a los consumidores más importantes del núcleo urbano, culminando en 2004 la instalación de los 100 primeros equipos de transmisión de datos de lectura a demanda mediante una red GSM. Esta tecnología, aunque permitía obtener información muy valiosa del suministro era prohibitiva para desplegarse de forma masiva al resto de suministros.

En el año 2009, con el avance que se había producido en las redes de comunicaciones, EMACSA licitó una prueba piloto de despliegue de telelectura en red fija para contadores domiciliarios para más de 3.500 contadores, tal y

como puede observarse en la tabla 1, estos contadores se centraron en la barriada, zonas en las que se entendía prioritario establecer la telelectura.

Tabla 1.- Detalle de la instalación de contadores en telelectura con prueba piloto. **Fuente:** Presentación: “Sistemas de telelectura en EMACSA”.

Barriada de Cerro Muriano	1.158 Uds.
Barriada de Sta. M ^a de Trassierra	418 Uds.
Barriada de Sta. Cruz	404 Uds.
Barriada de Alcolea	1.164 Uds.
Barriada del Ángel	381 Uds.
Encinarejo	364 Uds.
TOTAL	3.889 Uds.

Ante el éxito que se estaba cosechando, en 2010 se comenzó otra prueba piloto con otra tecnología de comunicación en otra zona de la ciudad, desplegándose en más de 2.500 contadores (Fernández, 2017). En 2013, tras la validación y consolidación de los sistemas, se decide el despliegue de estas tecnologías en el resto del parque de contadores de la compañía, más de 90.000 equipos, para que se culminase en 2020.

Las tecnologías desplegadas en la ciudad de Córdoba actualmente pueden clasificarse por sus usos principales, siendo uno el destinado en su mayoría a la telelectura de contadores en batería y el otro destinado a leer contadores individuales. Ambos sistemas, basan su estructura en la comunicación por radiofrecuencia, usando las bandas libres 868 y 169. La clasificación anteriormente mencionada se entiende por el tipo de frecuencia en la que trabajan cada tecnología de telelectura.



Figura 5.- Contadores en batería (Izq.) y contadores individuales (Dcha.). **Fuente:** Elaboración propia.

El sistema de telelectura WF868, usado en su mayoría para contadores en batería, tiene una mayor penetración, es un sistema flexible con un alcance prácticamente ilimitado, pero, sin embargo, es una red débil, con problemas de cobertura GPRS. Este sistema, transmite la lectura al ser consultado por el concentrador, en un intento de ahorrar el máximo de batería para llegar con garantía a su vida útil, sin embargo, los problemas de expansión de la red, hacen que esa consulta pueda no realizarse. A pesar de ello, el contador es capaz de almacenar las lecturas, enviándolas en el momento en el que consiga la conexión.

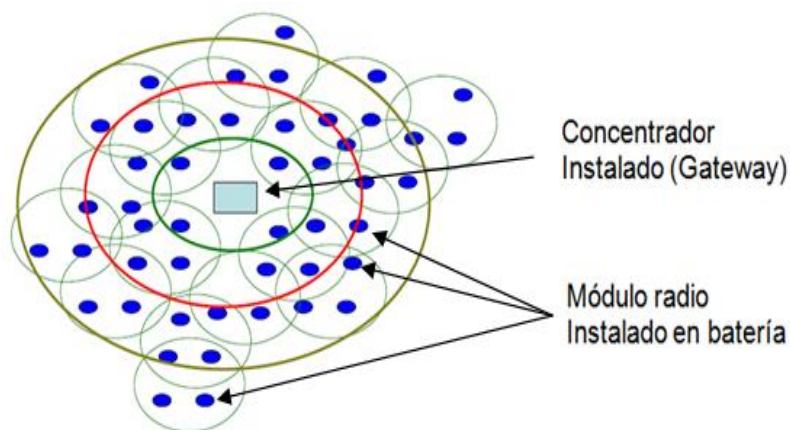


Figura 6.- Estructura de la red de telelectura WF868. **Fuente:** Presentación "Sistemas de telelectura EMACSA"

Por otro lado se encuentra el sistema de telelectura VHF169, el cual se caracteriza por ser un sistema que no requiere apenas infraestructura, con fácil

instalación, pero con poca penetración, alcance limitado y que requiere la instalación de antenas. La estructura de funcionamiento del sistema, se basa en una antena que emite pulsos del contador, 6 veces al día y las antenas concentradoras que recogen estas emisiones, en un radio teórico de hasta 750 metros. La recogida de lecturas, en caso de no producirse, se pierde, dejando un vacío en la serie de datos.

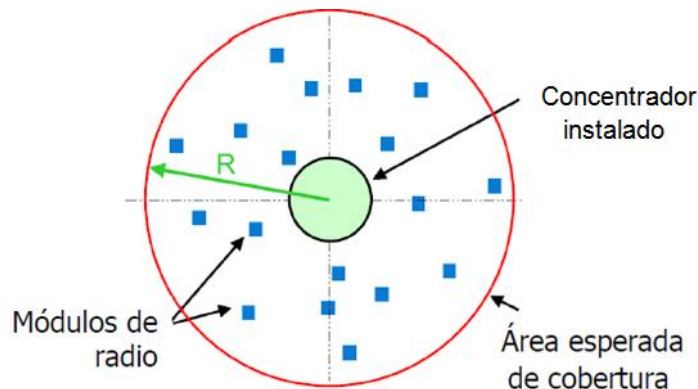


Figura 7.- Estructura de la red de telelectura VHF169. **Fuente:** Presentación "Sistemas de telelectura EMACSA".

Los sistemas de telelectura empleados por EMACSA, ofrecen una gran cantidad de información que permite realizar balances hídricos de los sectores de la red adecuados para la gestión de la misma. Para conseguir realizar esta tarea de una forma eficaz y ágil, también se han realizado desarrollos personalizados de software que permiten manejar de una forma solvente la ingente cantidad de datos con la que es necesario trabajar.

El empleo de estas herramientas, ha puesto de manifiesto la necesidad de implementar sistemas de tratamiento de la información recibida para su adecuación, validación y corrección, de forma que siempre, aunque existan problemas puntuales en la transmisión de la información, se pueda contar con el mejor dato posible para su utilización en el resto de sistemas de la empresa.

2. Objetivos.

El objetivo principal del presente trabajo es el desarrollo y evaluación de varios modelos basados en Redes Neuronales Artificiales (RNA) para la estimación de registros de consumo de agua en un sector específico de la red de la Empresa Municipal de Aguas de Córdoba S.A. (EMACSA). Este es un campo emergente de investigación, que se caracteriza por una amplia variedad de técnicas, una ausencia general de comparaciones intermodelo y un informe inconsistente de la habilidad del modelo (Dawson et al., 2001). Se trata de solventar los problemas relacionados con la ausencia de datos en relación con el envío telemático de los contadores al sistema de almacenamiento y otros de diferente naturaleza, con la intención de disponer de ellos en tiempo real o casi real. En este sentido, el aseguramiento de la calidad de este tipo de registros resulta esencial para el control y gestión de las redes de abastecimiento de agua a la población, así como su distribución y suministro. Experiencias previas en la validación de datos en Hidrología (Estévez et al., 2015; López-Lineros et al., 2014) servirán de base para el desarrollo del trabajo.

Para la consecución de este objetivo general se plantean los siguientes objetivos específicos:

- Detección de anomalías en los sets de datos brutos de consumo de agua y aplicación de procedimientos básicos de control de calidad.
- Calibración y Validación de los modelos empleados.
- Que el estudiante se familiarice con actividades de investigación tales como la revisión bibliográfica y la programación empleando software científico-técnico para la resolución de problemas en el marco del Agua y la Ingeniería Civil.

3. Fuente de Datos.

Los datos brutos que se han utilizado en este estudio fueron suministrados por la Empresa Municipal de Aguas de Córdoba (EMACSA), a través de su Meter Data Management de Aguas de Córdoba (MDMAC), desarrollado a partir de un importante convenio entre la Empresa Municipal de Aguas de Córdoba y la Universidad de Córdoba.

Para este estudio nos centramos en los contadores de un sector en concreto, el 02-ALC, sector que abarca la barriada de Alcolea. Este es uno de los sectores donde comenzó la implantación de la telelectura, en 2009 con la primera prueba piloto que llevó a cabo EMACSA, y en el que la telelectura tiene un amplio bagaje y un correcto funcionamiento. En este sector contamos con más de 1500 contadores, en su totalidad funcionando en el sistema de telelectura VHF 169 y de los cuales, para este estudio se van a tomar los siguientes:

Tabla 2.- Número de datos brutos de los contadores estudiados. **Fuente:** Elaboración propia.

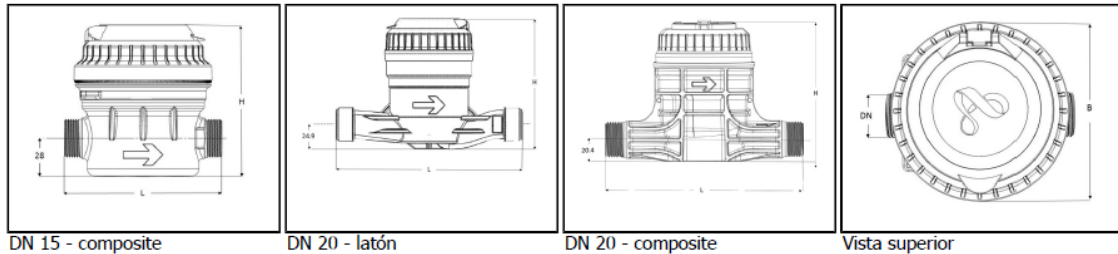
Contadores	Marca	Modelo	Calibre	Nº datos brutos
C11ALC578265	Sappel	Altair V4	15 mm	26243
C11ALC583338	Sappel	Altair V4	15 mm	26324
C12ALC153667	Sappel	Altair V4	15 mm	26353
C12ALC155545	Sappel	Altair V4	15 mm	25715
C12ALC156544	Sappel	Altair V4	15 mm	26440
C12ALC071969	Sappel	Altair V4	20 mm	26266

Estos contadores son contadores instalados entre 2011 y 2012, de la marca Sappel, calibre 15 mm y modelo Altair V4. Este tipo de contador, es un contador volumétrico, pistón rotativo, con capacidad de registrar un caudal de arranque de 0,4 l/h y un caudal nominal de 2,5 m³/h.

La serie de datos escogida, tiene una longitud del 1 de Enero de 2015 hasta el 30 de Noviembre de 2017, contando con lecturas horarias de cada contador, llegan a sumar un total de 131062 lecturas.

En el Anejo I, se pueden ver el documento completo de las características técnicas de los contadores Sappel Altair V4.

DIMENSIONES



Diámetro nominal	DN	mm	15	15	15	15	15	20	20	20
Longitud*	L	mm	110	115	134	165	170	110	165	190
Cuerpo			latón/ composite	latón/ composite	latón/ composite	latón/ composite	latón/ composite	latón	latón	latón/ composite
Anchura	B	mm	92,7	92,7	92,7	92,7	92,7	90	90	90/97
Altura	H	mm	104/108	105/117	105/109	105/108	105/108	132	132	132/135
Rosca		pulgada	3/4"	3/4" & 7/8"	3/4 "	3/4 "	3/4"	1"	1"	1"
Peso		kg	0,90/0,51	0,95/0,51	1/0,55	1,05/0,58	1,07/0,58	1	1,2	1,34/0,72

Figura 8.- Dimensiones de contador Altair V4. **Fuente:** Características técnicas Sappel.

ALTAIR V4

CONTADOR VOLUMETRICO



Figura 9.- Modelos de contador Altair V4. **Fuente:** Características técnicas Sappel.

4. Metodología.

4.1. Control de calidad y pre-procesado de datos brutos.

4.1.1. Test de rangos.

Para comenzar el procesamiento de los datos brutos, establecemos una serie de valores que consideramos no se pueden dar, dadas las características y regímenes de funcionamiento de los contadores. Se establece por tanto el siguiente test de rangos:

Tabla 3.- Test de rangos establecido para la depuración. **Fuente:** Elaboración propia.

	Ø 15mm	Ø 20mm
Caudal arranque	≥ 0,4 l/h	≥ 0,7 l/h
Caudal máximo (Q ₄)	≤ 3125 l/h	≤ 5000 l/h

Con los rangos establecidos, se considera que el contador no puede tener lecturas que den consumos negativos, ya que esto significaría que la presión de la instalación interior sería superior a la existente en la red de abastecimiento de agua, que se ha comprobado estar en torno a los 30-40 m.c.a.

El rango máximo establecido es el dado por el fabricante y que muestra el caudal máximo que puede aportar el contador, entendiendo así que nunca dentro de una hora, puede obtenerse consumos superiores a 3125 litros, en contadores de calibre 15 mm y de 5000 litros en contadores de calibre 20 mm.

4.1.2. Test de consistencia.

El procesamiento de los datos continúa introduciendo un test de consistencia, donde buscamos valores de consumo, que se repitan continuamente, es decir, consumos constantes que nos indiquen posibles periodos de vacaciones, riegos, grifos abiertos, fugas en instalación, etc. En esta línea, son muchos los estudios desarrollan perfiles de demanda basados en medidores inteligentes que permiten obtener conjuntos de datos de uso final del agua (Gurung, 2017).

Para ello tomamos como referencia los datos reflejados en el Cuaderno de I+D+i 25 del Canal de Isabel II “Sistema de reconocimiento de patrones para identificación de usos finales del agua en consumos domésticos” (Canal de Isabel II, 2017).

Tabla 4.- Resultados de la clasificación mediante RNA. Distribución según los usos del consumo total en periodo Enero-2008 a Julio 2015. **Fuente:** Canal Isabel II

Uso	Consumo total (m³)		Consumo medio mensual (m³)	Nº de eventos total		Nº de eventos, media mensual	Consumo medio, por evento (l)
Grifos	54.198	41%	595,58	21.028.397	61%	231.081	2,58
Cisternas	13.662	10%	150,14	2.614.989	8%	28.736	5,22
Duchas	43.323	33%	476,08	2.338.834	7%	25.701	18,52
Lavadora	9.133	7%	100,36	1.291.398	4%	14.191	7,07
Lavavajillas	1.951	1%	21,68	1.883.072	5%	20.923	1,04
Piscina	88	0%	0,99	1.608	0%	18	54,56
Riego	2.751	2%	30,57	48.657	0%	541	56,54
Fugas	5.543	4%	60,91	5.442.828	16%	59.811	1,02
Total	130.649	100%	1.436,30	34.649.783	100%	381.003	3,77

La tabla 4 refleja los resultados de distribución según los usos del consumo total en el periodo de enero de 2008 a Julio de 2015, para el método de clasificación Redes Neuronales Artificiales. Estos resultados, van a servirnos de referencia para establecer los valores de nuestro test de consistencia, además de incluir el valor de consumo = 0, que nos indicaría una vivienda deshabitada, cuando este valor se repita continuamente durante más de 5 días, las 24 horas del día, por ello, para nuestro estudio estableceremos los siguientes valores:

Tabla 5.- Valores establecidos para detectar anomalías en test de consistencia. **Fuente:** Elaboración propia.

Eventos	Consumos (l/h)
Sin consumo	0
Fugas	1,02
Avería	10,40

Cabe destacar que la persistencia de un consumo de forma continuada, nos da a conocer la existencia de un problema. Estos consumos continuados, no forman parte del patrón normal de consumo para los datos que aportan los contadores que hemos tomado en este estudio, por ello podemos conocer la existencia de eventos, tales como los que se muestran en la tabla 4, y que deberán ser sacados de la muestra a la hora de realizar la validación de los consumos.

La persistencia, es el valor que va a determinar la clasificación de los consumos. Para este estudio, estimamos que en caso que un valor de consumo supere los 5 días repitiéndose de forma constante, no se tomará en cuenta para realizar la validación de los datos, suponiendo aproximadamente 0,48% del total de la muestra sobre los datos de un contador.

4.1.3. Detección de anomalías.

El conjunto de datos que tenemos después de realizar el test de rangos y el test de consistencia, necesita una pre-validación para comprobar que todos los datos siguen la estructura correcta antes de su introducción en las RNA. En concreto, esta pre-validación consiste en la detección de líneas duplicadas.

Estas líneas de datos, las detectamos en un primer momento al existir horas del día duplicadas. Este problema en el set de datos, se ha consultado con EMACSA y se relaciona con errores de depurado por parte de la compañía que suministra las lecturas. Se debe a que en el sistema de telelectura de VHF 169, cada contador, debe ser barrido por al menos dos antenas receptoras, un

pequeño desfase de unos segundos, provoca dos líneas de lectura para una misma hora.

4.2. Fundamento teórico de las redes neuronales.

4.2.1. Fundamentos biológicos de las redes neuronales.

Las redes neuronales artificiales se basan en el funcionamiento del sistema neuronal del cuerpo humano. Las propiedades computacionales del uso de organismos biológicos o la construcción de computadoras pueden surgir como propiedades colectivas de sistemas que tienen un gran número de componentes equivalentes simples (o neuronas) (Hopfield, 1982). Los humanos no comienzan su pensamiento desde cero cada segundo, en el cuerpo humano encontramos 3 elementos fundamentales: los órganos receptores que recogen información del exterior; el sistema nervioso que transmite la información, la analiza y en parte almacena, y envía la información elaborada y, los órganos efectores que reciben la información de parte del sistema nervioso y la convierte en una cierta acción. Entiendes cada palabra según tu comprensión de las palabras anteriores, no tiras todo y comienzas a pensar de nuevo, tus pensamientos tienen persistencia (Olah, 2015).

La unidad fundamental del sistema nervioso es la neurona. Las neuronas se unen unas con otras formando redes. Se componen de un cuerpo o núcleo, del axón, que es una ramificación de salida de la neurona, y de un gran número de ramificaciones de entrada llamadas dendritas. Su funcionamiento es el siguiente. Las señales de entrada llegan a la neurona a través de la sinapsis, que es la zona de contacto entre neuronas (u otro tipo de células, como las receptoras). La sinapsis recoge información electro-química procedente de las células adyacentes que están conectadas a la neurona en cuestión. Esta información llega al núcleo de la neurona, a través de las dendritas, que la procesa hasta generar una respuesta, la cual es posteriormente propagada por el axón.

La sinapsis está compuesta de un espacio líquido donde existe una cierta concentración de iones. Este espacio tiene unas determinadas características eléctricas que permiten inhibir o potenciar la señal eléctrica a conveniencia.

Por ello, se puede ver que el sistema neuronal es un conjunto de neuronas conectadas entre sí, que reciben, elaboran y transmiten información a otras neuronas, y que dicha información se ve potenciada o inhibida en la siguiente neurona a conveniencia, gracias a las propiedades del espacio intersináptico.

De hecho, esta propiedad de poder alterar el peso de cada información en la red neuronal nos otorga en cierta medida la capacidad de aprender.

4.2.2. Redes neuronales artificiales.

Las redes neuronales artificiales tratan de emular las características y propiedades de las redes neuronales biológicas. En general, consisten en una serie de unidades denominadas neuronas, conectadas entre sí.

Cada neurona recibe un valor de entrada, el cual se transforma según una función específica denominada función de activación. Dicha señal transformada pasa a ser la salida de la neurona.

Las neuronas se conectan entre sí según una determinada arquitectura. Cada conexión tiene un determinado peso que pondera cada entrada a la neurona. De esta manera la entrada de cada neurona es la suma de las salidas de las neuronas conectadas a ella, multiplicadas por el peso de la respectiva conexión. La figura siguiente ilustra dicho concepto:

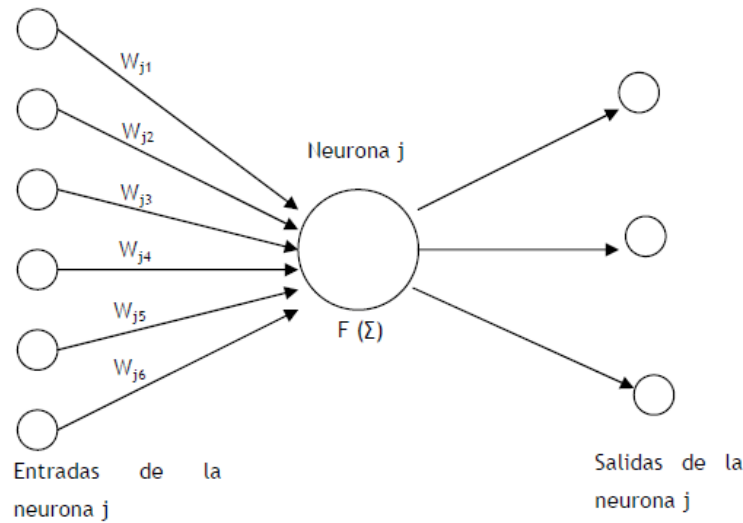


Figura 10.- Esquema de funcionamiento de una neurona. **Fuente:** TFC "Desarrollo de una interfaz gráfica de redes neuronales usando Matlab", (Moreno, 2009).

En este modelo, la neurona j recibe una serie de entradas x_1, x_2, \dots, x_n . Cada señal se multiplica por el peso asociado a su conexión, w_1, w_2, \dots, w_n . Luego, se suman estas entradas ponderadas y se les aplica la función de activación $F(.)$ para generar la señal de salida de la neurona j . Los valores de los pesos son ajustados durante la fase de aprendizaje.

Como se ha comentado anteriormente, estas neuronas están conectadas entre sí de acuerdo a una determinada arquitectura. Es decir, las neuronas se agrupan en distintas capas: una capa de entrada, otra de salida, y en el caso de existir, una o varias capas ocultas. La salida de cada neurona se propaga por igual por estas conexiones hasta las neuronas de destino. Cada conexión tiene un peso asociado que pondera el valor numérico de la señal que viaja por esta. Así pues, una red de neuronas artificial puede verse como un grafo cuyos nodos tienen funcionamiento similar, los cuales propagan la información a través de las distintas conexiones.

Veamos el funcionamiento de una red. Para ello nos referimos a la figura 2.2. Las entradas a la red son introducidas en las neuronas de la capa de entrada, que normalmente genera una salida tal cual o las escala para que las señales se encuentren en un determinado rango. Estas entradas son propagadas a las

neuronas de la siguiente capa. De acuerdo al esquema de la figura 11 cada neurona j de la segunda capa generara una salida de valor:

$$S_{2j} = F_{2j} (X_1 W_{1j}) \quad (2.1)$$

Donde X_1 es el vector de entradas de la capa 1 y W_{1j} el vector de pesos correspondientes a las conexiones que van de todas las neuronas de la primera capa a la neurona j de la segunda capa. La función F_{2j} es la función de activación de la neurona j de la segunda capa. Así con todas las neuronas de la segunda capa. Estas salidas son propagadas a las neuronas de la capa de salida. Estas neuronas generan las salidas de la red. Cada neurona i de la capa de salida generara una salida de valor:

$$S_{2i} = F_{2i} (W_{2i} S_2) \quad (2.2)$$

Donde W_{2i} es el vector de pesos correspondientes a las conexiones que van de las neuronas de la segunda capa a la neurona i de la capa de salida, y S_2 el vector de salidas de las neuronas de la capa dos, que a su vez son entradas de las neuronas de la capa de salida.

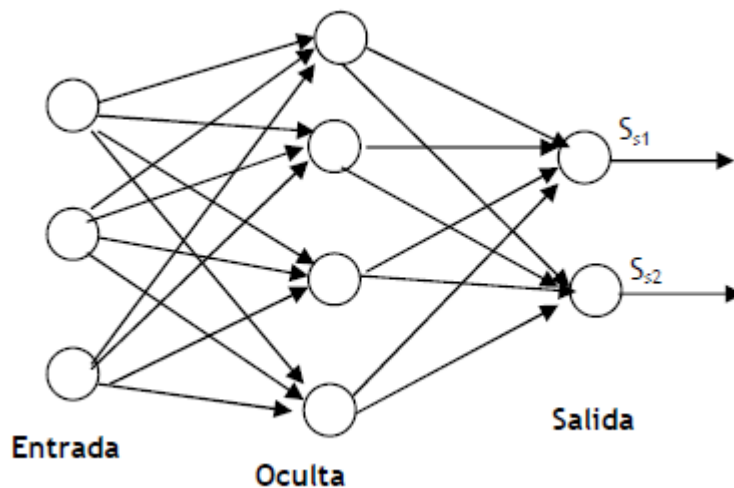


Figura 11.- Esquema de la arquitectura de una red. **Fuente.-** TFC "Desarrollo de una interfaz gráfica de redes neuronales usando Matlab", (Moreno, 2009).

Por último hablar del aspecto más importante y delicado de redes neuronales, el aprendizaje. Las RNA son sistemas de aprendizaje basadas en datos que son utilizados como patrones. Por ello la capacidad de una red de resolver un problema está muy ligada a los patrones utilizados durante su fase de aprendizaje.

El aprendizaje de una red neuronal consiste en hallar los valores precisos de los pesos de sus conexiones para que pueda resolver un determinado problema. El proceso general consiste en ir introduciendo una serie de datos patrón y ajustar los pesos siguiendo un determinado criterio. Los criterios que se van a utilizar en este proyecto se fundamentan en el error cometido por la red, lo que nos obliga a conocer la salida que se debería obtener para cada uno de ellos. Es lo que se conoce como entrenamiento supervisado. De esta manera, primero se introducen los patrones, se reajustan los pesos, posteriormente se comprueba si se ha cumplido un determinado criterio de convergencia, de no ser así se repite todo el proceso.

4.2.3. Redes perceptrón multicapa.

Introducción.

El perceptron multicapa con conexiones hacia adelante es una generalización del perceptron simple. Surge como respuesta a los problemas que tenía dicha red, como por ejemplo, no poder resolver problemas que no fueran linealmente separables. De hecho, algunos autores han demostrado que el perceptron multicapa es un aproximador universal de cualquier función continua en el espacio \mathbb{R}^n .

Arquitectura del perceptron multicapa.

La arquitectura de este tipo de red se caracteriza porque tiene todas sus neuronas agrupadas en distintos niveles llamados capas. El primer nivel corresponde a la capa de entrada, que se encarga únicamente de propagar por el resto de la red las entradas recibidas. El último nivel es el de la capa de salida.

Se encarga de proporcionar los valores de salida de la red. En las capas intermedias denominadas capas ocultas, se realiza un procesamiento no lineal de los patrones recibidos.

Las conexiones del perceptron multicapa son hacia adelante. Generalmente todas las neuronas de un nivel se conectan con todas las neuronas de la capa inmediatamente posterior. A veces, dependiendo de la red, se encuentran conexiones de neuronas que no están en niveles consecutivos, o alguna de las conexiones entre dos neuronas de niveles consecutivos no existe, es decir, el peso asociado a dicha conexión es constante e igual a cero. Además, todas las neuronas de la red tienen un valor umbral asociado. Se suele tratar como una entrada cuyo valor es constante e igual a uno, y lo único que varía es el peso asociado a dicha conexión (que es el umbral realmente).

Por otro lado, las funciones de activación que se suelen utilizar son la función identidad, la función sigmoïdal y la función tangente hiperbólica. A continuación se muestran sus respectivas expresiones.

Función identidad

$$f_1(x) = x \quad (2.3)$$

Función sigmoïdal

$$f_2(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

Función tangente hiperbólica

$$f_3(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (2.5)$$

La principal diferencia entre la función sigmoïdal y la función tangente hiperbólica es el rango de sus valores de salida. Mientras que para la primera su rango es $[0,1]$, para la segunda es $[-1,1]$. De hecho existe una relación entre las dos. Se relacionan mediante la expresión $f_2(x) = 2f_3(x) - 1$, por lo que la elección entre una u otra se elige en función del recorrido que interese.

Algoritmo de retropropagación.

Este algoritmo es el algoritmo básico de aprendizaje que usa el perceptron multicapa. Es el algoritmo mediante el cual se van adaptando todos los parámetros de la red. El tipo de entrenamiento que sigue este tipo de red es supervisado.

El aprendizaje de la red se plantea como un problema de minimización de una determinada función de error. En general se usa como función del error, el error medio cuadrático, es decir:

$$E = \frac{1}{N} \sum_{n=1}^N s(n) \quad (2.6)$$

$$s(n) = \frac{1}{2} \cdot \sum_{n=1}^t (s(n) - y(n))^2 \quad (2.7)$$

Donde $s(n)$ es la salida patrón, $y(n)$ la salida obtenida de la red, t el numero de neuronas de salida y N el numero de patrones.

El problema es no lineal y como tal, el problema de minimización de la función error se resuelve por técnicas de optimización no lineales que se basan en ajustar los parámetros siguiendo una determinada dirección. En este método, la dirección elegida es la negativa del gradiente de la función error. Más adelante veremos que existen otros métodos que no se basan en esta dirección de cambio.

A partir de aquí existen dos opciones. Podemos cambiar los parámetros cada vez que introducimos el patrón, o solamente cambiarlos cuando hayamos introducido todos los parámetros de entrenamiento por cada ciclo. En el primer caso, debemos minimizar $e(n)$ y en el segundo se minimiza la función E . A continuación se presentara el desarrollo para el primer caso, pero la extensión al segundo es inmediata.

De acuerdo a lo que hemos dicho antes, todos los pesos deben variar según la dirección del gradiente del error. Matemáticamente esto se expresa de la siguiente forma:

$$w(n) = W(n - 1) - \alpha \frac{\partial e(n)}{\partial w} \quad (2.8)$$

Por lo tanto el problema consiste en evaluar la derivada. El parámetro α es la tasa de aprendizaje que influye en la magnitud del desplazamiento en la superficie de la función error. A continuación se evalúa el valor del gradiente para los pesos de las conexiones de la última capa oculta a las neuronas de la capa de salida.

Sea $w_{ji}^{c-1}(n)$ el peso de la conexión de la neurona j de la capa $C-1$ a la neurona i de la capa de salida. El error $e(n)$ viene dado por la ecuación (2.7), y además este solo se ve afectado por w_{ji}^{c-1} en el error de la salida de la neurona i . Por tanto:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = -(s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} \quad (2.9)$$

Por otro lado, la salida de la neurona i es igual a la suma de las entradas transformadas según su función de activación. Aplicando la regla de la cadena, y teniendo en cuenta que w_{ji}^{c-1} solo afecta a la entrada de la neurona i porque va multiplicando a_j se tiene que:

$$\frac{\partial y_i(n)}{\partial w_{ji}^{c-1}} = f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) a_j^{c-1}(n) \quad (2.10)$$

Se define δ asociado a la neurona i de la capa C del patrón n , $\delta_i^c(n)$, del siguiente modo:

$$\delta_i^c(n) = -(s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c \right) a_j^{c-1}(n) \quad (2.11)$$

De tal manera que el gradiente se exprese ahora mediante la siguiente expresión:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = \delta_i^c(n) a_j^{c-1}(n) \quad (2.12)$$

Esto es extensible para todos los pesos de las conexiones de las neuronas de la capa C-1 con las neuronas de la capa de salida. Se procede de la misma forma para los valores umbrales obteniéndose una expresión análoga.

A continuación se procede a calcular la magnitud de la variación para el resto de conexiones que se existe entre las capas ocultas. Se va a proceder de la misma forma que hasta ahora y se obtendrá una ley de recurrencia para la modificación de dichos pesos. Consideremos el peso w_{kj}^{c-2} de la conexión de la neurona k de la capa C-2 con la neurona j de la capa C-1. El nuevo valor de dicho peso vendrá dado por una expresión análoga a la dada en (2.8), donde lo único que se desconoce es el valor del gradiente. En este caso w_{kj}^{c-2} influye en todas las salidas de la red, por lo que se tiene que:

$$\frac{\partial e(n)}{\partial w_{ji}^{c-1}} = - \sum_{i=1}^{n_c} (s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{kj}^{c-2}} \quad (2.13)$$

Para calcular la derivada hay que tener en cuenta que la salida w_{kj}^{c-2} influye en la entrada de la neurona j (por lo que influye en su salida), que a su vez influye en la entrada de todas las neuronas de salida. Por tanto, aplicando la regla de la cadena tenemos:

$$\frac{\partial y_i(n)}{\partial w_{kj}^{c-1}} = f' \left(\sum_{j=1}^{n_{c-1}} w_{ij}^{c-1} a_j^{c-1} + u_j^{c-1} \right) \frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} \quad (2.14)$$

Por otro lado la salida de la neurona j, únicamente se ve afectada por w_{kj}^{c-2} que va multiplicando a a_k^{c-2} . Por lo tanto, y volviendo a aplicar la regla de la cadena tenemos:

$$\frac{\partial a_j^{c-1}}{\partial w_{kj}^{c-2}} = f' \left(\sum_{k=1}^{n_{c-2}} w_{kj}^{c-2} a_k^{c-2} + u_j^{c-2} \right) a_k^{c-2}(n) \quad (2.15)$$

Definiendo el valor de δ para las neuronas de la capa C-1 de la siguiente forma:

$$\delta_j^{c-1}(n) = f' \left(\sum_k^{n_{c-2}} w_{kj}^{c-2} a_k^{c-2} + u_j^{c-2} \right) \sum_{i=1}^{n_c} \delta_i^c(n) w_{ji}^{c-1} \quad (2.16)$$

Se tiene que el valor de la derivada del error con respecto al peso w_{kj}^{c-2} es:

$$\frac{\partial e(n)}{\partial w_{kj}^{c-2}} = \delta_j^{c-1}(n) a_k^{c-2}(n) \quad (2.17)$$

Por lo que el nuevo valor de dicho peso viene dado por la expresión:

$$w_{kj}^{c-2} = w_{kj}^{c-2}(n-1) - a \delta_j^{c-1}(n) a_k^{c-2}(n) \quad (2.18)$$

Se observa que la modificación de los pesos de las conexiones entre las neuronas de la capa $C - 2$ y las neuronas de la capa $C - 1$ se ve afectada por la salida de la neurona k de la capa $C - 2$ y el término δ asociado a la neurona a la que llega la conexión. Lo único que varía es la expresión de δ . En este punto es posible generalizar la actualización de cualquiera de los pesos de cualquier capa dentro de la red, acorde a la ecuación (2.18). Por ello, la actualización de uno de los pesos de la conexión que une la neurona k de la capa h con la neurona j de la capa $h + 1$ vendrá dado por la expresión:

$$w_{kj}^h(n) = w_{kj}^h(n-1) - a \delta_j^{h+1}(n) a_k^h(n) \quad (2.19)$$

Donde el término δ viene dado por la siguiente ley de recurrencia:

$$\delta_j^{h+1} = f' \left(\sum_k^{n_h} w_{kj}^h a_k^h + u_j^{h+1} \right) \sum_{i=1}^{n_{h+2}} \delta_j^{h+1}(n) w_{kj}^{h+1} \quad (2.20)$$

El término δ propaga los errores obtenidos a la salida hacia atrás. De esta manera, cada neurona oculta recibe un cierto error o valor δ de todas las neuronas a las cuales se conecta, y la suma de todos estos errores es una medida del error total que comete esta neurona. Para la actualización de los valores umbrales se procede de la misma forma llegando a una expresión para su actualización dado por la siguiente expresión:

$$u_j^{h+1}(n) = u_j^{h+1}(n-1) - a \delta_j^{h+1}(n) \quad (2.21)$$

En este tipo de entrenamiento se define el número de épocas como el número de veces que se han comparado los ejemplos con las salidas de la red para realizar los ajustes en los pesos de las conexiones.

Mejoras del algoritmo de aprendizaje.

En el apartado anterior se ha descrito el algoritmo de retropropagación, que es el algoritmo más conocido para el entrenamiento de redes neuronales, pero que en la práctica su uso se ve muy limitado debido a que su convergencia es muy lenta. Por ello se han introducido mejoras de este algoritmo para acelerar la convergencia del mismo. En esta sección describiremos someramente algunos de ellos.

- Regla delta generalizada con momentos.

La regla delta generalizada se basa en la búsqueda del mínimo de la función error mediante el descenso por la superficie a través del gradiente de la misma. Esto nos puede llevar a un mínimo local de la función error, donde el gradiente vale cero, y por lo tanto los pesos no se ven modificados, pero el error cometido por la red es significativo. Para evitar este fenómeno este método propone que los cambios de los pesos de las conexiones tengan en cuenta el gradiente de la iteración anterior, realizando un promedio de este y el actual. De

esta manera, se desciende por una dirección intermedia entre el gradiente actual y el de la iteración anterior. Matemáticamente esto queda expresado por la expresión:

$$w(n) = w(n - 1) - \alpha \frac{\partial e(n)}{\partial w} + \eta \Delta w(n - 1) \quad (2.22)$$

El término $\Delta w(n - 1)$ es la magnitud del cambio de la iteración anterior. Es donde se tiene en cuenta el gradiente anterior. El parámetro η es el momento del algoritmo. El efecto de la inclusión del momento en este algoritmo es tal que acelera el cambio si las direcciones de las dos iteraciones son similares, y realiza pasos más pequeños si se producen oscilaciones en el gradiente, es decir tiene un efecto estabilizador.

- Tasa de aprendizaje variable.

La tasa de aprendizaje α juega un papel muy importante en el comportamiento de estos algoritmos de aprendizaje. Si es pequeño, la magnitud del cambio de los pesos sinápticos será pequeña y por lo tanto tardara mucho en converger. Si es demasiado grande el algoritmo oscilara y no encontrara nunca un mínimo de la función error. Se demuestra que el valor óptimo de la tasa de aprendizaje para una convergencia rápida es el valor inverso del mayor autovalor de la matriz Hessiana H . Pero computacionalmente este proceso es ineficiente, ya que para obtener la matriz H es necesario evaluar las derivadas segundas de la función error. Por ello se emplea técnicas heurísticas que van variando el valor de la tasa de aprendizaje en cada iteración. Algunas de estas son:

- Incrementar la tasa de aprendizaje cuando $\nabla E(n - 1) \nabla E(n)$ es próximo a 0 y disminuir la en caso contrario.

- Multiplicar la tasa de aprendizaje por $\alpha > 1$ si $\frac{\partial E(n-1)}{\partial w}$ y $\frac{\partial E(n)}{\partial w}$ tienen el mismo signo, o por $b \in (0,1)$ en caso contrario.

- Multiplicar la tasa de aprendizaje por una cantidad mayor que uno cuando haya decrecido la función error con el fin de avanzar más rápidamente, y multiplicarla por una cantidad menor que uno en caso contrario. Este heurístico es el que utiliza Matlab en el entrenamiento *traingda* (ver apartado 31.3).

- Método elástico de retropropagación.

Cuando las entradas a las neuronas de una red son muy grandes, y la función de activación de dichas neuronas es una función sigmoidea, el algoritmo de retropropagación de errores o regla delta generalizada sufre un bloqueo, es decir los pesos sináptico de las conexiones apenas varían debido a la magnitud del gradiente. Se produce dicho fenómeno porque la derivada de la función sigmoidea es cercana a cero cuando el argumento de dicha función es grande. Para evitarlo se utiliza este método que no determina la magnitud de la variación con el valor del gradiente, aunque si utiliza su signo para ver la dirección de cambio. El tamaño del cambio del peso esta determinado exclusivamente por un valor específico conocido como valor de actualización Δ_{ij} . Matemáticamente esto se expresa por:

$$w(n) = \begin{cases} w(n-1) - \Delta_{ij} & \text{si } \frac{\partial E(n)}{\partial w} > 0 \\ w(n-1) + \Delta_{ij} & \text{si } \frac{\partial E(n)}{\partial w} < 0 \\ w(n-1) & \text{si } \frac{\partial E(n)}{\partial w} = 0 \end{cases} \quad (2.23)$$

- Métodos de mayor orden de convergencia.

La rapidez de estos métodos es del orden de 10 a cientos de veces más rápidos que los anteriores. Se basan en la idea de que la dirección del gradiente no tiene porque ser la que más rápido converja hacia el mínimo de la función

error. De esta manera, estos métodos actualizan los pesos de acuerdo con esta expresión:

$$w(n) = w(n - 1) + a(n)d(n) \quad (2.24)$$

Donde $d(n)$ indica la dirección de búsqueda de la iteración n . Esta dirección es la conjugada de las $n - 1$ direcciones anteriores. Se dice que un conjunto de vectores no nulos son conjugados con respecto a una matriz A si son linealmente independientes y $d(i)^t A d(j) = 0, \forall i \neq j$.

Si utilizamos el desarrollo de Taylor generalizado cuadrático para aproximar la función E en el entorno del punto \vec{w}_0 y derivamos para obtener su mínimo obtenemos:

$$E(\vec{w}) = E(\vec{w}_0) + \nabla E(\vec{w}_0)^T (\vec{w} - \vec{w}_0) + \frac{1}{2} (\vec{w} - \vec{w}_0)^T H(w_0) (\vec{w} - \vec{w}_0) \quad (2.25)$$

$$\nabla E(w) = \nabla E(w_0) + H(w_0)(w - w_0) = 0 \quad (2.26)$$

Donde $d(n)$ es la matriz Hessiana. Si existe la inversa de H entonces se puede hallar una fórmula recurrente para la actualización de los pesos despejando de la ecuación 2.26. Es decir:

$$w(n) = w(n - 1) - H(w(n - 1))^{-1} \nabla E(w(n - 1)) \quad (2.27)$$

En general, el método de Newton puede que no converja, si el punto inicial no está lo suficientemente cerca del óptimo. Además requiere el cálculo de la inversa de la matriz Hessiana que requiere un gran esfuerzo computacional, por lo que en la práctica este método no se utiliza. El método de la secante evalúa la matriz Hessiana por medio del cálculo de la secante de la curva para evitar el mayor esfuerzo computacional que requiere el cálculo de las derivadas segundas.

Existen modificaciones del método de Newton para que pueda converger partiendo de cualquier punto. Uno de ellos (algoritmo de Fletcher-Reeves) es generar direcciones conjugadas mediante la regla:

$$d_n = -\nabla E(w_n) + \beta(n)d_{n-1} \quad (2.28)$$

Donde $\beta(n)$ viene dado por la expresión:

$$\beta(n) = \frac{g_n^T g_n}{g_n^T g_{n-1}} \quad (2.29)$$

Donde g son los gradientes de la función error de las respectivas iteraciones. Una vez obtenido la dirección de variación de los pesos sinápticos por la ecuación (2.28) con ayuda de la expresión dada en (2.29) los nuevos pesos vendrán dados por:

$$w(n) = w(n-1) + ad(n) \quad (2.30)$$

- Algoritmo de *Levenberg-Marquardt*

Es una modificación del método de Newton que elimina el cálculo de la matriz Hessiana. Requiere mayor esfuerzo computacional que los métodos de descenso por el gradiente pero es mucho más rápido que estos. De hecho es el algoritmo que Matlab utiliza por defecto para entrenar las redes neuronales.

Como ya se comentó, el problema de actualización de los pesos se resuelve minimizando la función error (se utiliza el error medio cuadrático), es decir:

$$\min(E(w)) = \min\left(\frac{1}{2} \sum_{i=1}^N (s_t - y(w)_t)^2\right) \quad (2.31)$$

La función error se puede expresar de la siguiente forma:

$$E(w) = \frac{1}{2} e(w)^T e(w) \quad (2.32)$$

donde:

$$e(w)^\tau = (e_1(w), e_2(w), \dots, e_N(w)) \quad (2.33)$$

$$e_i(w) = s_i - y(w)_i \quad (2.34)$$

La derivada de la función error $\nabla E(w)$ tiene la siguiente forma:

$$\nabla E(w) = e(w)^\tau \nabla e(w) = J(w)^\tau e(w) \quad (2.35)$$

Asimismo, la matriz Hessiana vendrá dada por:

$$H(w) = \sum_{i=1}^N \left(\nabla e_i(w) (\nabla e_i(w))^\tau + e_i(w) \nabla^2 e_i(w) \right) = J(w)^\tau J(w) + S(w) \quad (2.36)$$

Por otro lado, si tenemos en cuenta que la magnitud del cambio de los pesos sinápticos viene dada por la expresión:

$$\Delta w = -\alpha M(w) \nabla E(w) \quad (2.37)$$

Donde si $M(w) = I$ tenemos el método del descenso en la dirección del gradiente y si $M(w) = H^{-1}(w)$ nos encontramos con el método de Newton. En general, será más conveniente llegar a un compromiso entre estos dos métodos. Para ello se toma $[\mu I + H(w)]^{-1}$. De esta manera la dirección del cambio de los pesos sinápticos vendrá dado por:

$$d = -[\mu I + H(w)]^{-1} \nabla E(w(n)) \quad (2.38)$$

Debido a la dificultad de evaluar la matriz Hessiana, evaluamos solamente su término de primer orden $J(w)^\tau J(w)$, suponiendo así que el término $S(w) \cong 0$. Nos encontramos entonces ante el método de Levenberg-Maquardt. Se suele tomar $\mu = 0.01$. En este método el parámetro se incrementa o disminuye en cada paso:

- Si $E(w(k+1)) \leq E(w(k))$ entonces el parámetro T se divide por un determinado factor β .

- Si ocurre lo contrario entonces se multiplica por el factor β .

4.2.4. Redes neuronales de base radial.

Introducción.

Las redes neuronales de base radial surgen con el objeto de conseguir redes neuronales cuyo proceso de aprendizaje fuera mucho más rápido que en las redes perceptrón multicapa. Para ello se diseña una red en la que las neuronas de su única capa oculta tuvieran un carácter local, de tal manera que a la entrada de nuevos patrones hubiera menos neuronas que procesar. Esta es de hecho su principal diferencia con las redes perceptrón multicapa, el comportamiento local de sus neuronas, de tal manera que solo unas pocas se activan ante la entrada de un determinado patrón. Esto le confiere determinadas ventajas e inconvenientes en comparación con las primeras.

De todas maneras, y al igual que sucedía con el perceptrón multicapa, es un aproximador universal en el espacio \mathbb{R}^n .

Arquitectura.

Las redes neuronales de base radial disponen de una capa de neuronas de entrada, otra capa de neuronas de salida y solo una capa de neuronas ocultas.

Las conexiones en este tipo de red son todas hacia adelante, es decir, la entrada de todas las neuronas de la capa oculta procede de la salida de todas las neuronas de la capa de entrada, y a su vez las neuronas de la capa oculta transmiten su salida a todas las neuronas de la capa de salida. Un ejemplo de arquitectura de este tipo de red se muestra en la figura 12. La capa de neuronas de entrada únicamente propagan las entradas hacia la capa oculta, es decir, el peso de sus conexiones es siempre igual a uno. Por otro lado, las conexiones entre las neuronas de la capa oculta y las de la capa de salida si tienen asociado un peso.

Las neuronas de salida solo realizan una combinación lineal de las entradas que reciben, es decir, su función de activación es la identidad. Por ello el único procesamiento no lineal que se da en este tipo de redes tiene lugar en las neuronas de la capa oculta.

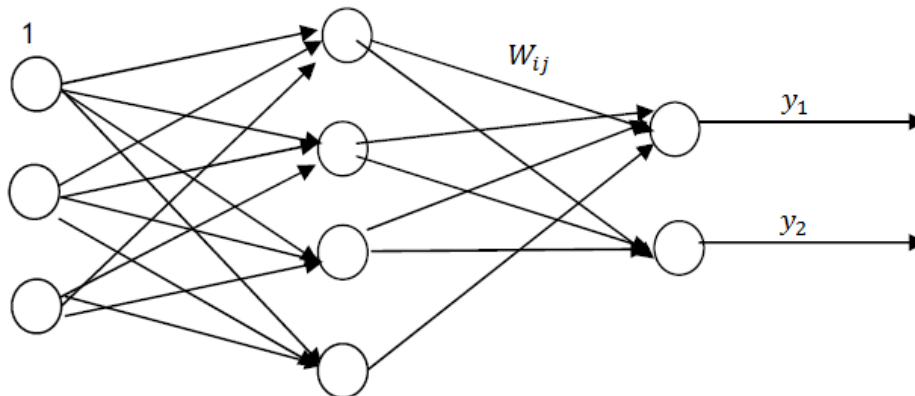


Figura 12.- Esquema de red neuronal de base radial. **Fuente.-** TFC "Desarrollo de una interfaz gráfica de redes neuronales usando Matlab", (Moreno, 2009).

Según lo que hemos dicho de como se propaga la señal a través de la red la salida y_1 bajo un determinado patrón de entrada $X = (x_1, x_2, x_3)$, donde x_i es la entrada a la neurona i de la capa de entrada, será:

$$y_1 = \sum_{i=1}^{n_{co}} w_{i1} \phi_i(X) + u_1 \quad (2.39)$$

La función $\phi_i(X)$ es la función de activación de la neurona i de la capa oculta. Las neuronas de la capa oculta tienen un comportamiento local debido al tipo de función que es. Esta función tiene la forma siguiente:

$$\phi_i(X) = \phi \left(\frac{\|X - C_i\|}{d_i} \right) \quad (2.40)$$

Es decir, el nivel de activación de la neurona i de la capa oculta depende de la distancia entre el patrón y el centro C_i asociado a dicha neurona. El parámetro d_i mide de alguna forma el campo de actuación de la neurona. De esta manera, si la entrada se encuentra cerca del centro de la neurona el nivel

de activación de la misma será grande, y en caso contrario pequeño. Podemos observar como el centro C_i determina la zona de actuación de la neurona y el parámetro d_i la extensión de dicha zona. Por ello se dice que estas neuronas tienen un comportamiento local. Esta función de activación, también conocida como función de base radial puede adoptar diversas formas que se adecuan a este esquema. Algunas de ellas son:

-Función gaussiana

$$\phi(r) = e^{\left(-\frac{r^2}{2}\right)} \quad (2.41)$$

-Función inversa cuadrática

$$\phi(r) = \frac{1}{1 + r^2} \quad (2.42)$$

- Función inversa multicuadrática

$$\phi(r) = \frac{1}{\sqrt{1 + r^2}} \quad (2.42)$$

Donde $r = \frac{\|X - C_i\|}{d_i}$, es decir, la distancia euclídea entre la entrada X y el centro de la neurona dividido por la amplitud. Hay que tener en cuenta que el centro tendrá tantas componentes como entradas tenga la red para situarlo en el espacio \mathbb{R}^n , donde n es el número de neuronas de entrada que existen en la red.

Por todo lo explicado anteriormente llegamos a la conclusión de que únicamente hay que decidir (en relación a la arquitectura de la red) el número de neuronas en la capa de entrada, en la de salida y en la oculta. El número de las dos primeras vendrá determinado por la naturaleza del problema al que nos enfrentamos, es decir, el número de neuronas en la capa de entrada viene determinado por las entradas necesarias para la resolución del problema, y el número en la capa de salida por el número de salidas diferentes que queramos

obtener. Sin embargo la determinación del un mero óptimo de neuronas en la capa oculta es más difícil de determinar. Hay que tener en cuenta que si existen pocas neuronas, se puede representar mal el espacio del problema, y por el contrario, si existen demasiadas neuronas, pueden existir varias neuronas en una determinada región, lo que conlleva consecuencias negativas en la aptitud de la red para resolver el problema. Este problema se suele resolver mediante el procedimiento ensayo-error añadiendo cada vez más neuronas hasta obtener un resultado óptimo.

Entrenamiento.

Como ocurría con la red anterior, en esta fase se adaptaban los parámetros de la red para que esta fuera capaz obtener las salidas apropiadas ante las entradas que se le iban a presentar. En el caso de las redes neuronales de base radial hay que determinar los centros y amplitudes de actuación de cada neurona, así como los pesos de las conexiones y valores umbrales de las neuronas de la capa de salida. Existen dos políticas a la hora de determinar estos parámetros. En la primera de ellas, llamada método híbrido, los centros y amplitudes de cada neurona se determinan mediante un esquema de aprendizaje no supervisado, y los pesos y valores umbrales se determinan mediante un esquema de aprendizaje supervisado minimizando el error que comete la red ante unos determinados patrones de entrenamiento. En el segundo método, denominado método de aprendizaje totalmente supervisado, se determinan todos los parámetros de la red mediante la minimización del error cometido por la red ante unos determinados patrones.

- Método híbrido

Como se comentó anteriormente, en este método primero se determinan los centros y amplitudes de cada neurona mediante un esquema de aprendizaje no supervisado, y posteriormente se determinan los pesos y valores umbrales de las neuronas de salida mediante un esquema de aprendizaje supervisado.

En la determinación de los centros se emplean algoritmos que tienen como objetivo clasificar de manera óptima, en un determinado número de clases, el espacio de entrada definido por los patrones de entrada. Uno de estos algoritmos es el algoritmo de K-medias.

Este algoritmo divide el espacio de entrada en K clases, siendo K el número de neuronas en la capa oculta, de tal manera que se minimice la suma de las distancias de cada entrada a su centro más cercano, es decir, se intenta resolver el problema:

$$\min Z = \sum_{i=1}^K \sum_{n=1}^N \text{Min} \|X(n) - C_i\| \quad (2.44)$$

Para resolver este problema, este algoritmo propone seguir unos determinados pasos.

1. Se inicializan los centros de las K clases de manera aleatoria, o con unos determinados valores obtenidos por otro método.

2. A cada entrada del patrón se le asocia el centro más cercano. Es decir, el patrón $X(n)$ pertenece al centro C_i si se cumple $\|X(n) - C_i\| \leq \|X(n) - C_s\|$ donde C_s es el centro mas cercano encontrado hasta ahora y distinto del centro C_i .

3. Se determinan los nuevos centros de cada clase mediante la media de las entradas que pertenecen a dicha clase.

4. Se repite los pasos 2 y 3 hasta que la posición de los centros no se modifique bajo un determinado margen de tolerancia, es decir que $\|C_i^{\text{nuevo}} - C_i^{\text{anterior}}\|$ para cualquier i .

Este algoritmo converge muy rápidamente, pero tiene el inconveniente de que se pueden obtener diferentes resultados dependiendo de los valores

tomados en la inicialización. Esto ocurre porque el método encuentra mínimos locales de la función Z.

Para la determinación de las amplitudes d_i se calcula de forma que el solapamiento entre zonas de activación de neuronas ocultas se lo menor posible, de manera que se obtengan interpolaciones más suaves. Para ello se emplean diversos heurísticos entre los que destacan los siguientes:

- Media uniforme de la distancia a los Q centros más cercanos. Es decir,

$$d_i = \frac{1}{p} \sum_p \|C_i - C_p\| \quad (2.45)$$

- Media geométrica de la distancia a sus dos centros más cercanos, es decir,

$$d_i = \sqrt{\|C_i - C_s\| \|C_i - C_t\|} \quad (2.46)$$

Por último, hay que determinar los pesos de las conexiones de la red. Para ello se pasa a la fase supervisada del método híbrido. Al igual como ocurría en las redes perceptron consiste en resolver el problema de minimizar una determinada función error. En el caso general esta función error vendrá definida por la expresión dada en la ecuación 2.6. Como ocurría anteriormente, en este método la actualización de los pesos se lleva a cabo mediante la expresión:

$$w_{ik}(n) = w_{ik}(n-1) - \alpha \frac{\partial e(n)}{\partial w_{ik}} \quad (2.47)$$

$$u_k(n) = u_k(n-1) - \alpha \frac{\partial e(n)}{\partial u_k} \quad (2.48)$$

Teniendo en cuenta la expresión del error y que el peso asociado a la conexión entre la neurona oculta i y la neurona de salida k solo afecta a la salida de la neurona k , tenemos,

$$\frac{\partial e(n)}{\partial w_{ik}} = -(s_k(n) - y_k(n)) \frac{\partial y_k(n)}{w_{ik}} \quad (2.49)$$

$$\frac{\partial e(n)}{\partial u_k} = -(s_k(n) - y_k(n)) \frac{\partial y_k(n)}{u_k} \quad (2.50)$$

Además la salida de la neurona k es una combinación lineal de las salidas de las neuronas ocultas ponderada mediante el peso de las respectivas conexiones más el valor umbral, por tanto,

$$\frac{\partial y_k(n)}{\partial w_{ik}} = \phi_i(n) \quad (2.51)$$

$$\frac{\partial y_k(n)}{\partial u_k} = 1 \quad (2.52)$$

De esta manera queda que la actualización de los pesos y valores umbrales dadas por las ecuaciones 2.47 y 2.48 quedan definidos de la forma:

$$w_{ik}(n) = w_{ik}(n-1) + \alpha(s_k(n) - y_k(n))\phi_i(n) \quad (2.53)$$

$$u_k(n) = u_k(n-1) + \alpha(s_k(n) - y_k(n)) \quad (2.54)$$

Debido a que la salida de la red depende linealmente de los pesos y valores umbrales, se puede obtener la solución del problema de minimización del error por medio del empleo del método de los mínimos cuadrados. En este método se obtiene directamente la matriz de pesos y valores umbrales R por medio de la expresión:

$$W = (G^t G)^{-1} G^t \cdot S \quad (2.55)$$

Donde la matriz G viene definida por:

$$G = \begin{pmatrix} \phi_1(1) & \phi_2(1) & \dots & \phi_r(1) \\ \phi_1(2) & \phi_2(2) & \dots & \phi_r(2) \\ \dots & \dots & \dots & \dots \\ \phi_1(N) & \phi_2(N) & \dots & \phi_r(N) \end{pmatrix} \quad (2.56)$$

Donde $s_k(n)$ es el valor de la función de activación de la neurona i ante el patrón $X(n)$. Y la matriz S viene definida por:

$$S = \begin{pmatrix} s_1(1) & s_2(1) & \dots & s_r(1) \\ s_1(2) & s_2(2) & \dots & s_r(2) \\ \dots & \dots & \dots & \dots \\ s_1(N) & s_2(N) & \dots & s_r(N) \end{pmatrix} \quad (2.57)$$

Donde $s_k(n)$ es la salida de la neurona k bajo la entrada a la red del patrón $X(n)$.

- Método de aprendizaje totalmente supervisado.

Como ya se comento, en el aprendizaje totalmente supervisado se determinan los centros, amplitudes y pesos de tal manera que se minimice el error cometido por la red, por lo que no es de esperar que se conserve el comportamiento local de la misma, ya que no se restringe el solapamiento de las regiones de activación de las neuronas.

Para ello hay que tener en cuenta que la dependencia de la salida de la red en relación a los centros y amplitudes de las distintas neuronas de la capa oculta es no lineal, por lo que hay que emplear técnicas de optimización no lineales. A continuación explicaremos el método del descenso por el gradiente. Mediante este método, y como ya se ha repetido en anteriores ocasiones, la actualización de los parámetros (centros, amplitudes y pesos) se lleva a cabo mediante las siguientes expresiones.

$$w_{ik}(n) = w_{ik}(n - 1) - \alpha_1 \frac{\partial e(n)}{\partial w_{ik}} \quad (2.58)$$

$$u_k(n) = u_k(n - 1) - \alpha_2 \frac{\partial e(n)}{\partial u_k} \quad (2.59)$$

$$c_{ik}(n) = c_{ik}(n - 1) - \alpha_3 \frac{\partial e(n)}{\partial c_{ik}} \quad (2.60)$$

$$d_i(n) = d_i(n - 1) - \alpha_4 \frac{\partial e(n)}{\partial d_i} \quad (2.61)$$

Donde $j = 1, 2, \dots, p$, $i = 1, 2, \dots, m$, y $k = 1, 2, \dots, r$, siendo p el numero de neuronas en la capa de entrada, m el numero de neuronas en la capa oculta y r el numero de neuronas de salida.

Pesos y umbrales: Ya se determino en el apartado anterior. La actualización de los pesos y umbrales vienen dadas por las ecuaciones 2.53 y 2.54 respectivamente.

Centros: Teniendo en cuenta que el centro c_{ij} de la neurona i de la capa oculta afecta a todas las salidas de la red, se tiene, aplicando la regla de la cadena:

$$\frac{\partial e(n)}{\partial c_{ij}} = - \sum_{k=1}^r (s_k(n) - y_k(n)) \frac{\partial y_k(n)}{\partial c_{ij}} \quad (2.62)$$

Por otro lado, el centro c_{ij} solo afecta a la neurona i de la capa oculta, es decir, en la salida solo afecta a la entrada de cada neurona de salida procedente de la neurona i de la capa oculta. Por tanto, se tiene:

$$\frac{\partial y_k(n)}{\partial c_{ij}} = w_{ik} \frac{\partial \phi_i(n)}{\partial c_{ij}} \quad (2.63)$$

Además, teniendo en cuenta la ecuación 2.40 para la función ϕ , tenemos:

$$\frac{\partial \phi_i(n)}{\partial c_{ij}} = \phi_i(n) \frac{x_j - c_{ij}}{d_i^2} \quad (2.64)$$

De esta manera, y mediante sustitución en la ecuación 2.60 obtenemos la ley de actualización de los centros c_{ij} .

$$c_{ij}(n) = c_{ij}(n - 1) + \alpha_2 \left(\sum_{k=1}^r (s_k(n) - y_k(n)) w_{ik} \phi_i(n) \frac{x_j - c_{ij}(n - 1)}{d_i^2(n - 1)} \right) \quad (2.65)$$

Amplitudes: Al igual que ocurría con los centros, la amplitud d_i afecta a todas las salidas de la red, por lo tanto:

$$\frac{\partial e(n)}{\partial d_i} = - \sum_{k=1}^r (s_k(n) - y_k(n)) \frac{\partial y_k(n)}{\partial d_i} \quad (2.66)$$

Además, la amplitud d_i solo afecta a la entrada de cada neurona de salida procedente de la neurona i de la capa oculta, al igual que ocurre en el caso de los centros. Por tanto,

$$\frac{\partial y_k(n)}{\partial d_i} = w_{ik} \frac{\partial \phi_i(n)}{\partial d_i} \quad (2.67)$$

Y atendiendo a la expresión dada en la ecuación 2.40 tenemos que:

$$\frac{\partial \phi_i(n)}{\partial d_i} = \phi_i(n) \frac{\|X(n) - C_i(n-1)\|^2}{d_i^3(n-1)} \quad (2.68)$$

De esta manera, y mediante sustitución, se obtiene la ley de actualización para las amplitudes utilizando la ecuación 2.61

$$d_i(n) = d_i(n-1) + \alpha_4 \left(\sum_{k=1}^r (s_k(n) - y_k(n)) w_{ik} \phi_i(n) \frac{\|X(n) - C_i(n-1)\|^2}{d_i^3(n-1)} \right) \quad (2.69)$$

Como en todo proceso iterativo hay que inicializar todos los parámetros. Los centros y amplitudes se deberían inicializar con valores que permitan una buena representación del espacio donde se encuentran los patrones de entrada.

- Combinación de los dos métodos.

El método híbrido tiene la ventaja de que conserva el comportamiento propio de las redes neuronales de base radial, sin embargo su respuesta es menos exacta que la de las redes obtenidas por el método totalmente supervisado. Por ello, se busca una combinación de los dos métodos para obtener las ventajas de los dos tipos de entrenamiento. De esta manera, en este

tipo de entrenamiento, se emplea el algoritmo de K-medias y la determinación de las amplitudes utilizada en el método híbrido, para introducir después estos valores en el entrenamiento totalmente supervisado como valores iniciales. Así se consigue las propiedades locales del primer tipo de entrenamiento, y la exactitud del aprendizaje totalmente supervisado.

4.2.5. Redes neuronales modelo NAR y NARX.

Este es un modelo neuronal autorregresivo no lineal, basado en una única serie de datos que funciona como input y output. Éste modelo predice los valores de la serie utilizando únicamente las observaciones pasadas de dicha serie a pronosticar. Esta propuesta de modelo se considera una de las mejores para analizar y predecir información de series de tiempo y ya se encuentra incluida dentro del módulo de redes neuronales de Matlab, expresamente para este tipo de análisis de series temporales. La arquitectura general se presenta a continuación:

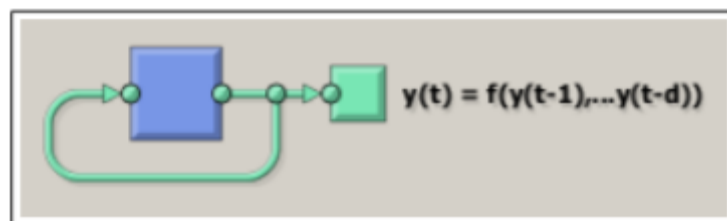


Figura 13.- Esquema de la arquitectura general del modelo neuronal NAR. **Fuente:** Elaboración propia.

Una red NAR, es un tipo de perceptrón multicapa, en donde una vez teniendo la serie a utilizar se definen los rezagos, el número de capas ocultas y la función de entrenamiento. Este tipo de red es de dos capas, una oculta y una de salida. La función de entrenamiento que utiliza es “trainlm”, que es la función de retropropagación de Levenberg-Marquardt (Levenberg, 1994) y (Marquardt, 1963). Esto nos regresa una red neuronal NAR. El término retropropagación se refiere a la forma en que el gradiente de la función de transferencia es calculado para redes neuronales feedforward. El entrenamiento de la red involucra un ajuste de los pesos comparando la salida deseada con la respuesta de la red,

de manera que se minimice el error. La función que usualmente se utiliza para medir el error es la suma de los errores al cuadrado (SSE) o su promedio (MSE). La función de transferencia utilizada en la capa oculta de este tipo de red es la sigmoideal, pues al ser un modelo no lineal, ésta debe ser estrictamente monótona, acotada y diferenciable, requisitos que cubre la función sigmoideal. Para el caso de la capa de salida, la función de transferencia es lineal, la cual produce tanto su entrada como su salida. Como el valor de la salida durante el entrenamiento se encuentra disponible para este tipo de red, se puede utilizar una arquitectura de bucle abierto, en donde el verdadero valor de la salida sea utilizado en lugar de retroalimentar la salida estimada. Las dos ventajas que ofrece esto son que en primer lugar, la entrada a la red de alimentación directa (feedforward) es más precisa, y en segundo lugar que la red resultante tiene una arquitectura completamente feedforward, y por lo tanto un algoritmo más eficaz que puede ser utilizado para su entrenamiento. En Redes Neuronales Artificiales los modelos se obtienen por prueba y error. Esto implica que no hay un procedimiento específico para conseguir la mejor red, se parte de una red inicial, y probando diversas arquitecturas, es que puede obtenerse una red final que cumpla las expectativas del modelador.

El modelo NARX también es un modelo dinámico autorregresivo no lineal pero con variables exógenas, lo cual implica que tiene una variable dependiente de una o varias variables que la explican. Al igual que en el modelo NAR, se considera la mejor propuesta para el análisis con el módulo de series de tiempo en redes neuronales de Matlab. Éste modelo predice los valores de y_t en función de los valores de:

$$y_{t-1} \dots y_{t-n}, x_{1,t-n}, \dots, x_{m,t-1} \dots x_{m,t-n}$$

Su arquitectura general se muestra en la siguiente figura:

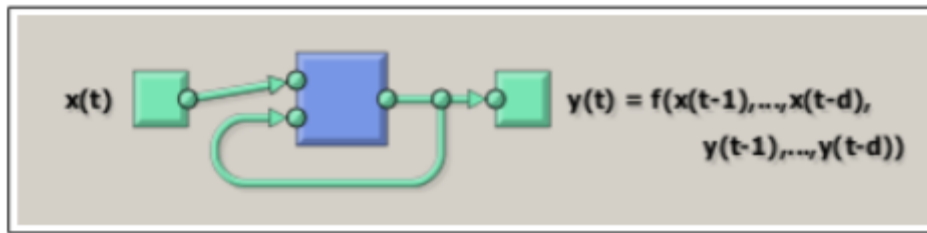


Figura 14.- Esquema de la arquitectura general del modelo NARX. **Fuente:** Elaboración propia.

Similar a una red NAR, la red NARX es un tipo de perceptrón multicapa en donde una vez teniendo las series a utilizar (introducidas de forma independiente la variable dependiente de un grupo de independientes) se definen los rezagos, el número de capas ocultas y la función de entrenamiento. Este tipo de red también se conforma de dos capas, una oculta y una de salida. La función de entrenamiento que utiliza es “trainlm” que es la función de retropropagación de Levenberg-Marquardt. Las funciones de transferencia son las mismas que en el caso de la red NAR: Sigmoidal en capa oculta y lineal en capa de salida.

4.2.6. Redes neuronales modelo Deep Learning: Red Neuronal Recurrente LSTM.

El aprendizaje profundo (Deep Learning) es una técnica de aprendizaje automático que enseña a los ordenadores a hacer lo que resulta natural para las personas: aprender mediante ejemplos.

Con el aprendizaje profundo, un modelo informático aprende a realizar tareas de clasificación directamente a partir de imágenes, texto o sonido. Los modelos de aprendizaje profundo pueden obtener una precisión de vanguardia que, en ocasiones, supera el rendimiento humano. Los modelos se entrenan mediante un amplio conjunto de datos etiquetados y arquitecturas de redes neuronales que contienen muchas capas.

El problema de las dependencias a largo plazo de las RNN es la idea de que podrían ser capaces de conectar información previa a la tarea actual, por ejemplo, el uso de marcos de video previos podría ayudar a comprender el marco

actual. Como algunas veces, los humanos solo necesitamos ver información reciente para realizar la tarea actual. En tales casos, cuando la brecha entre la información relevante y el lugar que se necesita es pequeña, las RNN pueden aprender a usar la información pasada. Pero también hay casos en los que necesitamos más contexto ya que es posible que exista una brecha entre la información relevante y el punto necesario que sea muy grande. Desafortunadamente, a medida que crece esa brecha, las RNN no pueden aprender a conectar la información.

En teoría, las RNN son absolutamente capaces de manejar tales "dependencias a largo plazo". Un humano podría elegir cuidadosamente los parámetros para resolver problemas de esta forma. Tristemente, en la práctica, las RNN no parecen ser capaces de aprenderlas. Este problema fue analizado en profundidad por Hochreiter J. (1991) y Bengio (1994), quien encontró algunas razones bastante fundamentales por las cuales era difícil.

Las redes de memoria a corto plazo, generalmente llamadas "LSTM", son un tipo especial de RNN, capaz de aprender con dependencias a largo plazo. Fueron introducidos por Hochreiter et al. (1997) y funcionan tremendamente bien en una gran variedad de problemas y ahora son utilizados en multitud de campos.

Los LSTM están diseñados explícitamente para evitar el problema de dependencia a largo plazo. Recordar la información durante largos períodos de tiempo es prácticamente su comportamiento predeterminado.

La mayor parte de los métodos de aprendizaje emplean arquitecturas de redes neuronales, por lo que, a menudo, los modelos de aprendizaje profundo se denominan redes neuronales profundas. El término "profundo" suele hacer referencia al número de capas ocultas en la red neuronal. Las redes neuronales tradicionales solo contienen dos o tres capas ocultas, mientras que las redes profundas pueden tener hasta 150. Una popular variante de LSTM, introducida por Gers et al. (2000), agrega "conexiones de mirilla". Esto significa que dejamos que las capas de puerta miren el estado de la celda.

4.3. Redes neuronales en Matlab.

4.3.1. Redes Perceptron Multicapa.

En Matlab se crea una red perceptron multicapa con conexiones hacia delante utilizando la función *newff*. Esta función devuelve una variable que representa a la red. Los argumentos de entrada de esta función (descritos en el orden en el que se colocan) son:

- *R* que es una matriz cuyo número de filas es el número de entradas, y su número de columnas es igual a 2. Esta matriz debe señalar los valores mínimos y máximos que pueden tomar cada una de las entradas a la red.

- [*Slayer1, Slayer2, ...*] es un vector que describe el número de neuronas que tienen todas las capas ocultas de la red y el número de neuronas de salida. Asimismo el tamaño de dicho vector permite conocer al programa el número de capas que debe tener la red que se va a crear.

- {*Funciones*} es un vector de varias cadenas de caracteres en el que se señala las funciones de activación que va a poseer todas las neuronas de una capa. Así, la primera cadena señalará la función de activación de las neuronas de la primera capa oculta, la segunda señalará la función de las neuronas de la segunda capa, y así sucesivamente. Por ello, este vector debe tener tantas cadenas como capas vaya a tener nuestra red. Para que la función de activación sea lineal, la cadena debe ser '*purelin*', para que esta sea una función tangente hiperbólica se debe escribir '*tansig*', y para indicar que queremos una función sigmoideal escribiremos '*logsig*'. Por defecto Matlab utiliza la función tangente hiperbólica.

Simulación de la red.

Para obtener las salidas de una determinada red ante unas ciertas entradas se utiliza la función *sim*. Esta función devuelve un vector cuyas componentes son las salidas que se obtienen de cada neurona de salida de la

red. A dicha función se le introducen como argumentos el nombre del objeto que representa a la red en Matlab, y el vector de entradas a la red.

Se pueden obtener varias simulaciones de un tiempo introduciendo una matriz, cuyos vectores sea cada uno de los vectores de entrada de los que se quiere obtener su salida. De esta manera, si se tiene una red “*net*” de la que se quiere obtener su respuesta ante unas entradas “*a*”, se escribe en Matlab la siguiente línea de código.

p=sim (net,a);

Entrenamiento.

Como se ha comentado anteriormente, el entrenamiento de la red (el que se utiliza en este proyecto) consiste en presentarle unas entradas y sus correspondientes salidas (que son conocidas por nosotros), para que la red vaya reajustando su salida mediante la modificación de sus pesos y valores umbrales, de manera que el error de actuación de la red se minimice. Dicho valor Matlab lo almacena en la variable *net.performFcn*. La medida del error por defecto en Matlab es el error medio cuadrático, que viene dado por la expresión:

Para indicar que se quiere utilizar esta definición del error en Matlab, la variable *net.performFcn* debe igualarse a *mse* (en cualquier caso, este valor lo toma por defecto).

$$E(n) = \frac{1}{N} \sum_{k=1}^N \sum_{n=1}^r (s_k(n) - y_k(n))^2 \quad (3.1)$$

Para entrenar una red en Matlab se utiliza la función *train*. Dicha función utiliza como argumentos el nombre de la red que se quiere entrenar (y que anteriormente ha debido ser creada) y los patrones, compuestos por unos vectores de entradas (unidos en una matriz “*a*”) y sus correspondientes salidas (todas ellas en una matriz “*p*”). De esta manera la línea de código que se debería escribir es:

$[net,pr] = train (net,a,p);$

La variable *pr* contiene información sobre el proceso de entrenamiento y la variable *net* contiene a la red ya entrenada, es decir con sus pesos y valores umbrales ajustados. Por otro lado, destacar que esta función utiliza una serie de variables para definir el entrenamiento y que pueden definirse con anterioridad. Estas variables varían según el tipo de entrenamiento utilizado por lo que se hablara de ellas cuando se describan como se definen los diferentes algoritmos de aprendizaje en Matlab. Se pueden utilizar diferentes tipos de entrenamiento para las redes del tipo multipercpetrón.

4.3.2. Redes neuronales de base radial.

Las redes neuronales de base radial podrían requerir más neuronas que las redes neuronales estándar de retropropagación. Pero su entrenamiento es similar a estas. Trabajan mejor cuando hay muchos datos de entrenamiento disponibles.

Las redes neuronales de base radial pueden ser implementadas con cualquiera de estas dos funciones, *newrbe* y *newrb*.

- Función *newrbe*.

Esta función coge una matriz de vectores de entrada y sus correspondientes salidas patrón y una constante *SPREAD*, que señala el campo de actuación de cada neurona, y devuelve una red cuyos pesos y valores umbrales son tal, que devuelve exactamente las salidas esperadas *P* para las entradas *A*. Su sintaxis en Matlab es la siguiente:

$net = newrbe(A,P,SPREAD)$

Esta función crea una red cuya capa de neuronas de base radial tiene un número de neuronas igual al número de entradas diferentes que se le proporciona a la red durante su entrenamiento. De esta manera cada neurona

de esta capa actúa como un detector de un tipo de entrada en concreto. Como se ha dicho antes la red tiene un error cero para los valores de entrada.

La constante *SPREAD* es muy importante porque definirá el campo de actuación de cada neurona. Por ejemplo si dicha constante tiene un valor cuatro, la neurona responderá con un valor 0.5 a cualquier entrada que se encuentre a una distancia euclídea de cuatro con respecto a su centro de actuación. Debe ser lo suficientemente grande para que las neuronas actúen correctamente en regiones de solapamiento. Ello provoca que la respuesta de la red sea muy suave y que tenga buena capacidad de generalización. El problema estriba, en que si se necesitan demasiados vectores de entrada para caracterizar correctamente la red, esta tendrá en su capa oculta demasiadas neuronas.

- Función *newrb*

Esta función se puede emplear de otra forma para generar una red más eficiente. En este método, la función crea la red de forma iterativa, creando una neurona más en cada iteración. Las neuronas son añadidas hasta que el error medio cuadrático caiga por debajo del parámetro *GOAL*, o se haya alcanzado un determinado número de neuronas máximo. La sintaxis de dicha función en Matlab es:

```
net = newrb(P,T,GOAL,SPREAD);
```

4.3.3. Redes neuronales NARX.

El modelo de estimación NARX proporciona mejores predicciones que el modelo de entrada-salida, porque usa la información adicional contenida en los valores previos de $y(t)$. Sin embargo, puede haber algunas aplicaciones en las que los valores previos de $y(t)$ no estarían disponibles y esos son los únicos casos en los que se debería usar el modelo de entrada-salida en lugar del modelo NARX (Demuth et al., 2016).

La red NARX estándar es una red de alimentación directa de dos capas, con un sigmoide en la función de transferencia en la capa oculta y una función de transferencia lineal en la capa de salida. Esta red también usa líneas de retraso intervenidas para almacenar secuencias de valores previos de $x(t)$ y $y(t)$. Hay que tener en cuenta que la salida de la red NARX, $y(t)$, se retroalimenta a la entrada de la red (a través de retrasos), ya que $y(t)$ es una función de $y(t-1)$, $y(t-2)$, ..., $y(t-n)$. Sin embargo, para un entrenamiento eficiente, este circuito de retroalimentación se puede abrir (Beale et al, 2015).

Debido a que la salida real está disponible durante el entrenamiento de la red, puede usarse la arquitectura de bucle abierto, en la que se usa la salida verdadera en lugar de retroalimentación de la producción estimada. Esto tiene dos ventajas, la primera es que la entrada a la red *feedforward* es más precisa y la segunda es que el resultado la red tiene una arquitectura puramente *feedforward* y por lo tanto se puede usar un algoritmo más eficiente para el entrenamiento.

Para crea una red NARX, *narxnet*, se trata de una red de *feedforward* con función de transferencia tan-sigmoide predeterminada en la capa oculta y función de transferencia lineal en la capa de salida. Esta red tiene dos entradas, una es una entrada externa, y la otra es una conexión de retroalimentación desde la salida de la red (después de que la red haya sido entrenada, esta conexión de retroalimentación puede cerrarse). En cada una de estas entradas, hay una línea de retardo para almacenar valores previos.

El número predeterminado de neuronas ocultas se establece en 10. El número predeterminado de retrasos o *delays* es 2. Con estas configuraciones, los vectores de entrada y los vectores de destino se dividirán al azar en tres conjuntos de la siguiente manera:

- 70% se usará para entrenamiento.

- 15% se usará para validar que la red se está generalizando y para detener el entrenamiento antes de sobreajustar.
- El último 15% se usará como una prueba de generalización de la red completamente independiente.

4.3.4. Redes neuronales recurrente LSTM.

Una red LSTM es un tipo de red neuronal recurrente (RNN) que puede aprender las dependencias a largo plazo entre los pasos de tiempo de la secuencia de datos.

Para pronosticar los valores de los espacios de tiempo futuros de una secuencia, se entrena una regresión de secuencia a secuencia de red LSTM, donde las respuestas son las secuencias de entrenamiento con valores desplazados por un espacio de tiempo. Es decir, en cada espacio de tiempo de la secuencia de entrada, la red LSTM aprende a predecir el valor del siguiente espacio de tiempo. Para pronosticar los valores de varios espacios de tiempo en el futuro, en Matlab se usa la función `predicAndUpdateState` para predecir los espacios de tiempo de uno en uno y actualizar el estado de la red en cada predicción.

Arquitectura de red LSTM

Los componentes principales de una red LSTM son una capa de entrada de datos y una capa LSTM. Una capa de entrada de datos ingresa la secuencia o datos de series de tiempo en la red. Una capa LSTM aprende las dependencias a largo plazo entre los espacios de tiempo de los datos de la secuencia.

Este diagrama ilustra la arquitectura de una red simple LSTM. La red comienza con una capa de entrada de datos seguida de una capa LSTM. Para predecir las etiquetas de clase, la red termina con una capa completamente conectada, una capa de softmax y una capa de salida de clasificación.

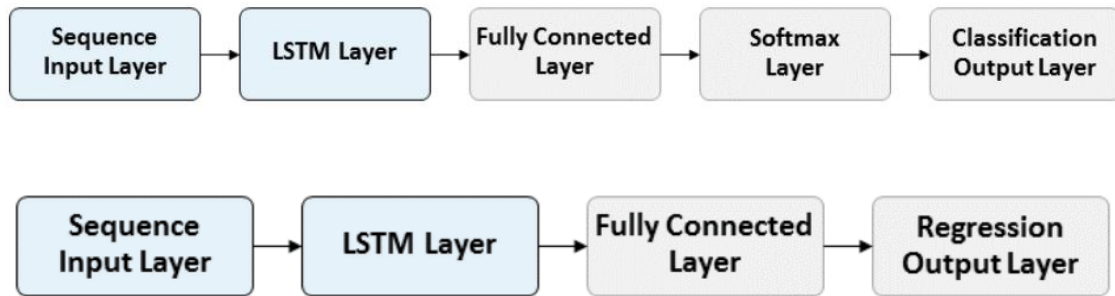


Figura 15.- Diagrama de la arquitectura de una red LSTM simple. **Fuente:** es.mathworks.com

Este diagrama ilustra la arquitectura de una red LSTM simple para una regresión. La red comienza con una capa de entrada de datos seguida de una capa LSTM. La red termina con una capa completamente conectada y una capa de salida de regresión.

Clasificación LSTM Networks

Para crear una red LSTM para la clasificación de datos a etiqueta, será necesario crear una matriz de capas que contenga una capa de entrada de datos, una capa LSTM, una capa totalmente conectada, una capa de softmax y una capa de salida de clasificación.

En primer lugar, hay que especificar el tamaño de la capa de entrada de datos para que sea la dimensión de la entrada de datos. Se deberá añadir el tamaño de la capa totalmente conectada para que sea el número de clases. Para la capa LSTM, especificar un tamaño de salida y el modo de salida 'último'.

Capas

- Capa de entrada de datos.

Una capa de entrada de datos introduce la secuencia a una red. Este tipo de capas se pueden crear usando `sequenceInputLayer`.

- Capa LSTM.

Una capa LSTM es una capa de red neuronal recurrente (RNN) que permite la compatibilidad con series de tiempo y datos de secuencia en una red. La capa realiza interacciones aditivas, que pueden ayudar a mejorar el flujo de gradiente en secuencias largas durante el entrenamiento. Las capas LSTM son las más adecuadas para aprender dependencias a largo plazo (dependencias de pasos de tiempo distantes). Esta capa se puede crear usando `lstmLayer`.

- Capa Bidireccional LSTM.

Una capa bidireccional LSTM (BiLSTM) es una capa RNN que aprende las dependencias bidireccionales a largo plazo entre los pasos de tiempo. Estas dependencias pueden ser útiles cuando desea que la red aprenda de la serie temporal completa en cada paso de tiempo. Para crear esta capa será necesario usar la función `bilstmLayer`.

Clasificación y predicción.

Para clasificar o hacer predicciones sobre nuevos datos, use en Matlab la función `classify` y `predict`.

Las redes LSTM pueden recordar el estado de la red entre predicciones. El estado de la red es útil cuando no tiene la serie de tiempo completa por adelantado, o si desea hacer predicciones múltiples en una serie larga. Para predecir y clasificar partes de una serie temporal y actualizar el estado de la red, se pueden usar las funciones `predictAndUpdateState` y `classifyAndUpdateState`, y para restablecer el estado de la red entre predicciones, `resetState`.

Acolchado de secuencia, truncamiento y división.

Al pasar datos a través de la red, el software rellena, trunca o divide secuencias en cada mini lote para tener la longitud especificada. En Matlab se puede especificar las longitudes de secuencia y el valor, utilizando para rellenar los espacios los argumentos del par nombre-valor `SequenceLength` y `SequencePaddingValue` en `trainingOptions`.

- Ordenar secuencias por longitud.

Si se ordenan los datos por longitud de secuencia, puede reducir la cantidad de relleno o los datos descartados al rellenar o truncar las secuencias.

Para ordenar los datos por longitud de secuencia, primero será necesario obtener el número de columnas de cada secuencia aplicando el tamaño $(X, 2)$ a cada secuencia usando *cellfun*. Posteriormente clasificar las longitudes de secuencia usando *sort*, y use la segunda salida para reordenar las secuencias originales.

- Secuencias de Pad

Si se especifica la longitud de la secuencia 'más larga', el software rellena las secuencias en cada mini lote para tener la misma longitud que la secuencia más larga en ese mini lote. Esta opción es la predeterminada.

- Secuencias Truncadas

Por otro lado, si se especifica la longitud de la secuencia que será 'más corta', entonces el software trunca las secuencias en cada mini-lote para tener la misma longitud que la secuencia más corta en ese mini-lote. Los datos restantes en las secuencias se descartan.

- Secuencias divididas

Por último, si se especifica la longitud de la secuencia para que sea un valor entero, entonces el software rellena las secuencias en cada mini lote para tener la misma longitud, de forma que sea como la secuencia más larga, y luego divide las secuencias en secuencias más pequeñas de la longitud especificada. Si se produce una división, el software crea mini lotes adicionales.

4.4. Parámetros estadísticos.

Para la evaluación de los diferentes modelos empleados, se compararon los datos estimados con los datos de consumo real. Para cada contador se han

calculado los siguientes parámetros estadísticos (Willmott, 1982): error medio absoluto, raíz del error cuadrático medio, error relativo y el ratio.

$$MAE = \frac{\sum_{i=1}^n \text{abs}(y_i - x_i)}{n} \quad (4.1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (4.2)$$

$$RE = \frac{MAE}{x_{ave}} \times 100 \quad (4.3)$$

$$R = \frac{y_{ave}}{x_{ave}} \quad (4.4)$$

5. Resultados y Discusión.

Tras aplicar los procedimientos de control de calidad para la validación descritos y de uso en el ámbito hidrogeológico (Sciuto et al., 2009), se han descartado todos los registros anómalos. El modelado de series de tiempo es necesario para la planificación y gestión de los recursos hidrológicos (Lohani et al., 2012). Por tanto, se ha construido un set de datos verosímil y con garantías para cada uno de los 6 contadores, con el que hemos trabajado en el desarrollo de los modelos basados en RNA que se han evaluado.

5.1. Datos depurados.

Los sets de datos depurados de cada uno de los contadores, arrojan huecos importantes debido a las características propias de la serie de datos y de la validación realizada.

Tabla 6.- Evaluación de la depuración de la serie de datos por contador. **Fuente:** Elaboración propia.

Contadores	Nº datos brutos	Test rangos	Test consistencia	Nº datos depurados
C11ALC578265	26243	484	1333	24426
C11ALC583338	26324	658	842	24730
C12ALC153667	26353	587	1548	24218
C12ALC155545	25715	851	2137	22727
C12ALC156544	26440	1116	340	24984
C12ALC071969	26266	1081	1595	23590

Al ver el resultado de la depuración en la tabla 6, podemos comprobar que en la longitud de la serie de datos, que va de 1 de Enero de 2015 hasta 30 de Noviembre de 2017, contamos con 1049 días, que aportan un máximo de 25176 datos. Como se puede comprobar, el set de datos bruto contenía valores fuera de rango y tras la evaluación de la consistencia verificamos la existencia de valores permanentes que deben ser sacados, para usarlos con garantías para la validación de los consumos con RNA.

Por lo tanto, hemos organizado los datos, con el fin de crear un procedimiento que detecte las anomalías. En la tabla 7, se puede comprobar

para el día 1 de Enero, un ejemplo de las lecturas diarias. Esta será la estructura que tendrán los datos antes de ser introducidos en Matlab para la validación de los consumos con RNA.

Tabla 7.- Tabla muestra de estructuración de datos. **Fuente:** Elaboración propia.

CONTADOR	AÑO	Nº HORA	Día del año	NºMES	Nº Día Semana	Día Semana	Lectura	Consumo (l)
C11ALC583338	2015	1	1	1	4	Jueves	320357	0
C11ALC583338	2015	2	1	1	4	Jueves	320370	13
C11ALC583338	2015	3	1	1	4	Jueves	320374	4
C11ALC583338	2015	4	1	1	4	Jueves	320375	1
C11ALC583338	2015	5	1	1	4	Jueves	320376	1
C11ALC583338	2015	6	1	1	4	Jueves	320377	1
C11ALC583338	2015	7	1	1	4	Jueves	320377	0
C11ALC583338	2015	8	1	1	4	Jueves	320378	1
C11ALC583338	2015	9	1	1	4	Jueves	320382	4
C11ALC583338	2015	10	1	1	4	Jueves	320430	48
C11ALC583338	2015	11	1	1	4	Jueves	320440	10
C11ALC583338	2015	12	1	1	4	Jueves	320456	16
C11ALC583338	2015	13	1	1	4	Jueves	320504	48
C11ALC583338	2015	14	1	1	4	Jueves	320544	40
C11ALC583338	2015	15	1	1	4	Jueves	320544	0
C11ALC583338	2015	16	1	1	4	Jueves	320548	4
C11ALC583338	2015	17	1	1	4	Jueves	320548	0
C11ALC583338	2015	18	1	1	4	Jueves	320568	20
C11ALC583338	2015	19	1	1	4	Jueves	320595	27
C11ALC583338	2015	20	1	1	4	Jueves	320599	4
C11ALC583338	2015	21	1	1	4	Jueves	320602	3
C11ALC583338	2015	22	1	1	4	Jueves	320602	0
C11ALC583338	2015	23	1	1	4	Jueves	320618	16
C11ALC583338	2015	24	1	1	4	Jueves	320622	4

A continuación, en la figura 16, se representan los consumos del contador C11ALC578265 durante los meses de Noviembre y Diciembre de los años 2015 y 2016, se aprecia la importancia de la introducción del test de rangos. Se destacan algunos picos que evidencian valores de consumo por encima de los valores de capacidad de caudal máximo del contador en estudio.

Con la depuración de los datos, nos resalta la circunstancia que se muestra en la figura 18, en la que el consumo va aumentando a lo largo de la serie de datos. En este caso, no somos capaces de identificar una avería, pero a pesar de ello, vemos que los consumos aumentan a lo largo del tiempo pudiendo esto deberse a múltiples circunstancias y también se aprecia que la curva de consumos se acentúa en los meses de verano.

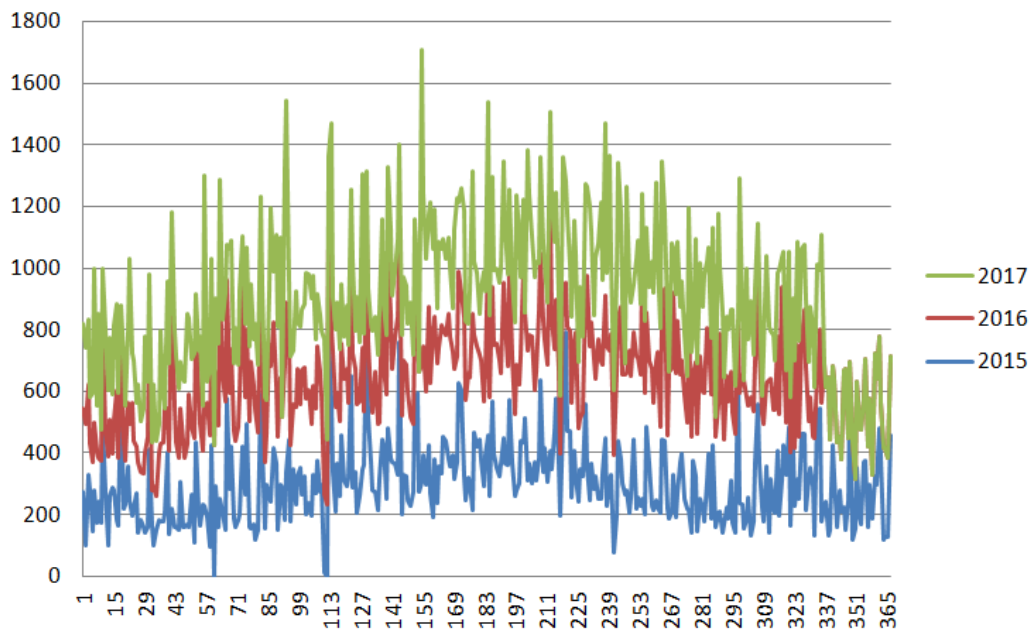


Figura 18.- Valores de consumo del contador C12ALC155545 durante la serie de datos. **Fuente:** Elaboración propia.

5.2. Evaluación de los diferentes modelos.

Una vez tenemos el set de datos depurados de cada contador se ha procedido a entrenar y evaluar los diferentes modelos basados en redes neuronales artificiales mediante su desarrollo en Matlab. Hemos utilizado un 15% del set de datos para validar los datos estimados por los modelos de ajuste RNA y NARX. Es decir, de forma aleatoria hemos eliminado ese 15% (más de 3000 datos) y los hemos estimado utilizando cada uno de los dos modelos. En el caso del modelo Deep Learning (Red Neuronal Recurrente LSTM), hemos tomado series temporales previas de 15, 30, 45 y 60 días, debido a que sets de datos

excesivamente grandes en este tipo de aprendizaje profundo pueden introducir ruido, arrojando en estos casos resultados menos precisos.

En estos modelos de ajuste hemos utilizado el algoritmo de entrenamiento Levenberg-Mardkuard, el cual ha dado buenos resultados en Hidrología (López-Lineros et al., 2014; Sciuto et al., 2009). Este procedimiento requiere mayor esfuerzo computacional que los métodos de descenso por el gradiente pero es mucho más rápido que estos tal y como se describe en el apartado 4.2.6 de este trabajo.

5.2.1. Modelo de ajuste NFIT.

En este apartado se detallan los resultados obtenidos a partir del modelo de ajuste NFIT. Se trata de un modelo de ajuste que ha utilizado 4 variables de entrada (inputs) que han sido las siguientes: Día del año, día de la semana, mes y hora del día. La arquitectura de este modelo puede verse en la figura 19, a partir de estos cuatro inputs y con 100 neuronas en la capa oculta, el modelo es capaz de aportarnos el valor de consumo estimado. Se realizaron pruebas con 10, 20, 40 y 100 neuronas, siendo esta última configuración ligeramente más adecuada.

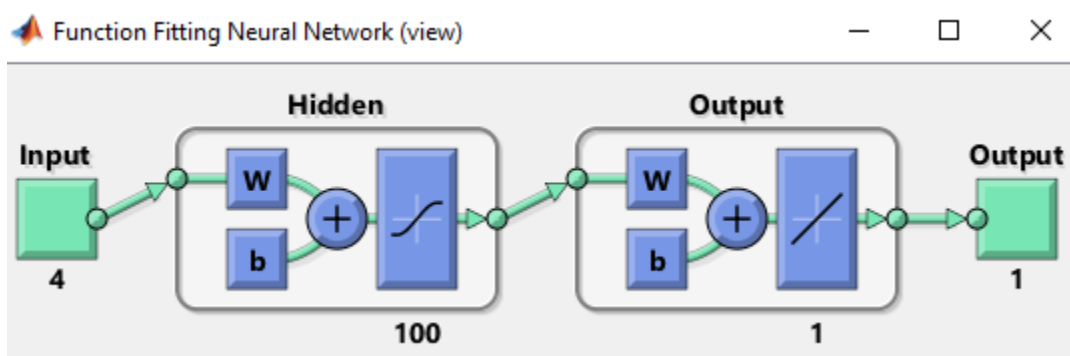


Figura 19.- Esquema del proceso de funcionamiento de la red neuronal NFIT. **Fuente:** Elaboración propia.

En la Figura 20 se muestran los resultados de la validación del modelo NFIT para los cinco contadores de calibre 15mm (A) y para el contador de 20 mm (B). Se puede apreciar una clara diferencia de rango, donde los consumos

horarios son bastante más elevados en este último (Figura 20B). Las diferentes nubes de puntos resultan de enfrentar los consumos reales frente a los consumos horarios estimados por el modelo para esas horas concretas. Téngase en cuenta que el objetivo de graficar estas parejas de valores es la de dar una idea orientativa sobre la tendencia en relación con la línea 1:1 y visualizar de forma aproximada la dispersión, ya que la cuantificación de los diversos errores y otros coeficientes se detallan en la Tabla 8.

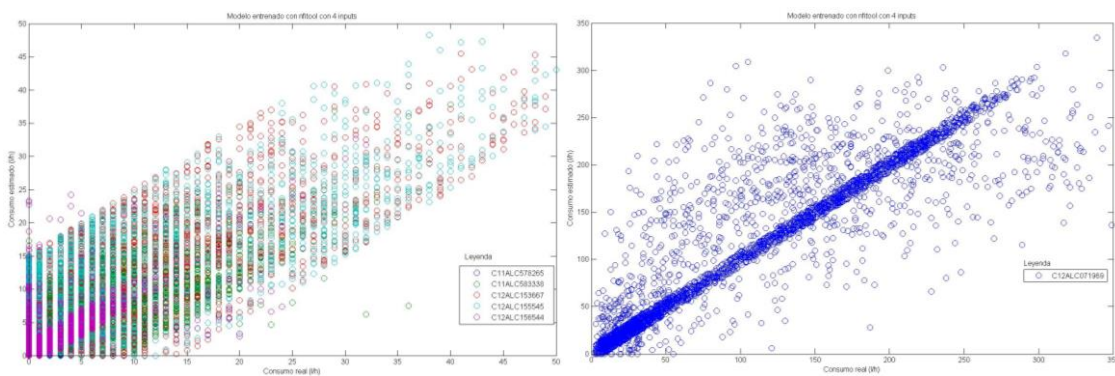


Figura 20 (A y B).- Modelo entrenado con NFIT con 4 inputs para contadores de calibre 15mm (A) y 20 mm (B). **Fuente:** Elaboración propia.

Tabla 8.- Resultados de los parámetros estadísticos para el modelo NFIT. **Fuente:** Elaboración propia.

	Nº de datos empleados	Neural fit				
		R ²	RMSD (l/h)	MAE (l/h)	RE	Ratio
C11ALC578265	3364	0,458	1,093	0,839	60,6%	1,645
C11ALC583338	3708	0,360	4,084	3,368	35,6%	1,177
C12ALC153667	3631	0,495	6,415	4,837	35,2%	1,447
C12ALC155545	3411	0,579	5,553	4,658	32,2%	1,529
C12ALC156544	3725	0,580	2,170	1,195	23,7%	1,503
C12ALC071969	3713	0,705	45,053	31,395	21,5%	1,004

Estos resultados muestran a través del Ratio, que, en promedio, este modelo sobreestima en todos los contadores los consumos horarios pronosticados, si bien el contador de calibre 20 mm es el que compensa los errores de predicción (ratio casi igual a 1), siendo C11ALC578265 el que muestra mayores sobreestimaciones. En términos de dispersión, sin embargo, el que mayor coeficiente de determinación tiene (0,705) es el contador de mayor calibre,

siendo C11ALC583338 el que peor se comporta, oscilando el resto entre 0,45 y 0,58. En términos de RMSD, los errores oscilan entre 1 y 6,4 l/h para los contadores de menor calibre, obteniendo el valor más alto en C12ALC153667. Los valores del error medio absoluto son en líneas generales aceptables para todos los contadores (entre 21,5% y 35,6%), con la excepción de C11ALC578265 donde se obtuvo un error demasiado elevado. El contador de mayor calibre, aunque con errores RMSD y MAE aparentemente altos fue el que menor error relativo presentó.

5.2.2. Modelo NARX.

Este modelo está basado en una red neuronal no lineal autorregresiva con entradas externas. Esta RNA pronostica consumos horarios dados d valores pasados (delay) y utiliza además los registros de las series temporales de otras variables de entrada. Se han empleado los mismos 4 inputs que en el modelo anterior y un delay=2. En la Figura 21 se representa la arquitectura y funcionamiento del modelo NARX evaluado.

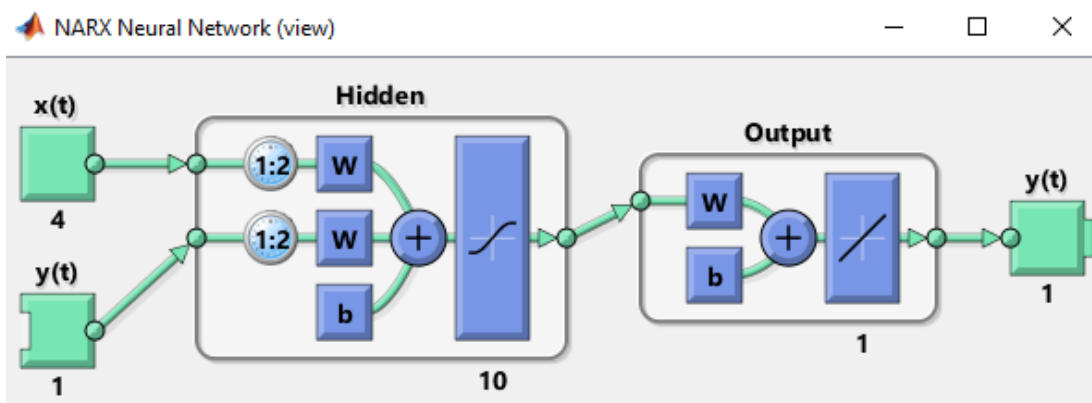


Figura 21.- Esquema del proceso de funcionamiento de la red neuronal NARX. **Fuente:** Elaboración propia.

En la Figura 22 se representan los resultados de graficar consumos reales vs. consumos estimados correspondiente al set de datos utilizado para la validación de este modelo, donde la Figura 20A corresponde a los 5 contadores de calibre 15 mm y la Figura 20B al contador de 20 mm. Los estadísticos que se

han evaluado están recogidos en la Tabla 9, por lo que la representación de estas nubes de puntos es meramente orientativa de cara a dar una idea cualitativa de la tendencia de los consumos estimados por el modelo NARX.

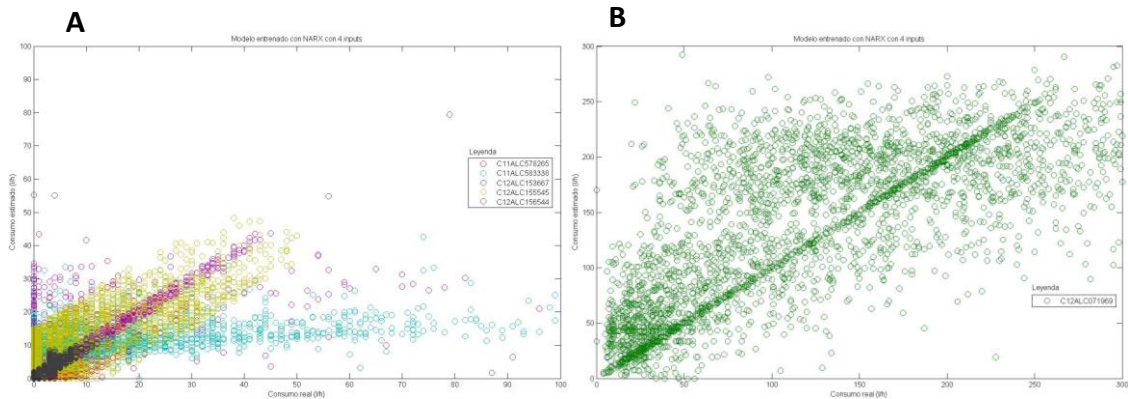


Figura 22 (A y B).- Modelo entrenado con NARX con 4 inputs para contadores de calibre 15 mm (A) y 20 mm (B). **Fuente:** Elaboración propia.

Tabla 9.- Resultados de los parámetros estadísticos para el modelo NARX. **Fuente:** Elaboración propia.

	Nº de datos empleados	NARX				
		R ²	RMSD (l/h)	MAE (l/h)	RE	Ratio
C11ALC578265	3364	0,740	0,911	0,635	45,9%	1,327
C11ALC583338	3708	0,770	1,720	2,480	26,2%	0,946
C12ALC153667	3631	0,730	5,600	3,070	22,3%	1,141
C12ALC155545	3411	0,760	4,550	4,620	32,0%	1,530
C12ALC156544	3725	0,700	1,950	0,910	18,0%	1,499
C12ALC071969	3713	0,640	55,870	46,230	31,7%	0,993

En primer lugar, se puede observar como los coeficientes de determinación han mejorado notablemente en todos los contadores de calibre 15 mm en relación con el modelo NFIT. Sin embargo, para el contador de 20 mm los resultados mostraron un comportamiento algo peor que el NFIT en relación con el grado de dispersión y también con el RMSD, MAE y RE que registraron valores más elevados. En promedio, el modelo NARX sobreestimó los consumos pronosticados en los contadores C11ALC578265, C12ALC153667, C12ALC155545 y C12ALC156544, mientras que subestimó ligeramente en los contadores C11ALC583338 y C12ALC071969. Los valores de RMSD y MAE

para todos los contadores de 15 mm fueron más bajos que en el modelo anterior, mostrando por tanto un mejor comportamiento en las estimaciones, oscilando entre 0,911 l/h (C11ALC578265) y 5,6 l/h (C12ALC153667). El error relativo más bajo se obtuvo en el contador C12ALC156544, con un valor del 18%, registrándose también en el resto de contadores de 15 mm valores inferiores a los obtenidos en la evaluación del modelo NFIT.

5.2.3. Función de Base Radial.

El modelo de RNA basado en funciones de base radial evaluado en el presente trabajo, debido a que las series de datos son limitadas (apenas 3 años) no ha dado resultados satisfactorios, obteniendo un mal comportamiento en las estimaciones de consumo.

5.2.4. Deep Learning: Red Neuronal Recurrente LSTM

Para evaluar el modelo LSTM en cada uno de los contadores y debido al alto coste computacional de este tipo de aprendizaje profundo se ha utilizado diferentes longitudes de cada serie temporal para estimar consumos de dos días consecutivos (48 registros). Los sets de datos han sido de 2 meses, 1 mes y medio, 1 mes y 15 días. Siguiendo un proceso aleatorio para seleccionar la fracción de la serie inicial a utilizar, se ha repetido el proceso 100 veces en cada uno de los contadores. Aunque el objetivo primordial es evaluar cómo se comporta el modelo estimando el consumo futuro una hora después, en las siguientes figuras se muestra una comparación de valores reales y pronosticados con un horizonte de dos días, en una de esas cien iteraciones realizadas.

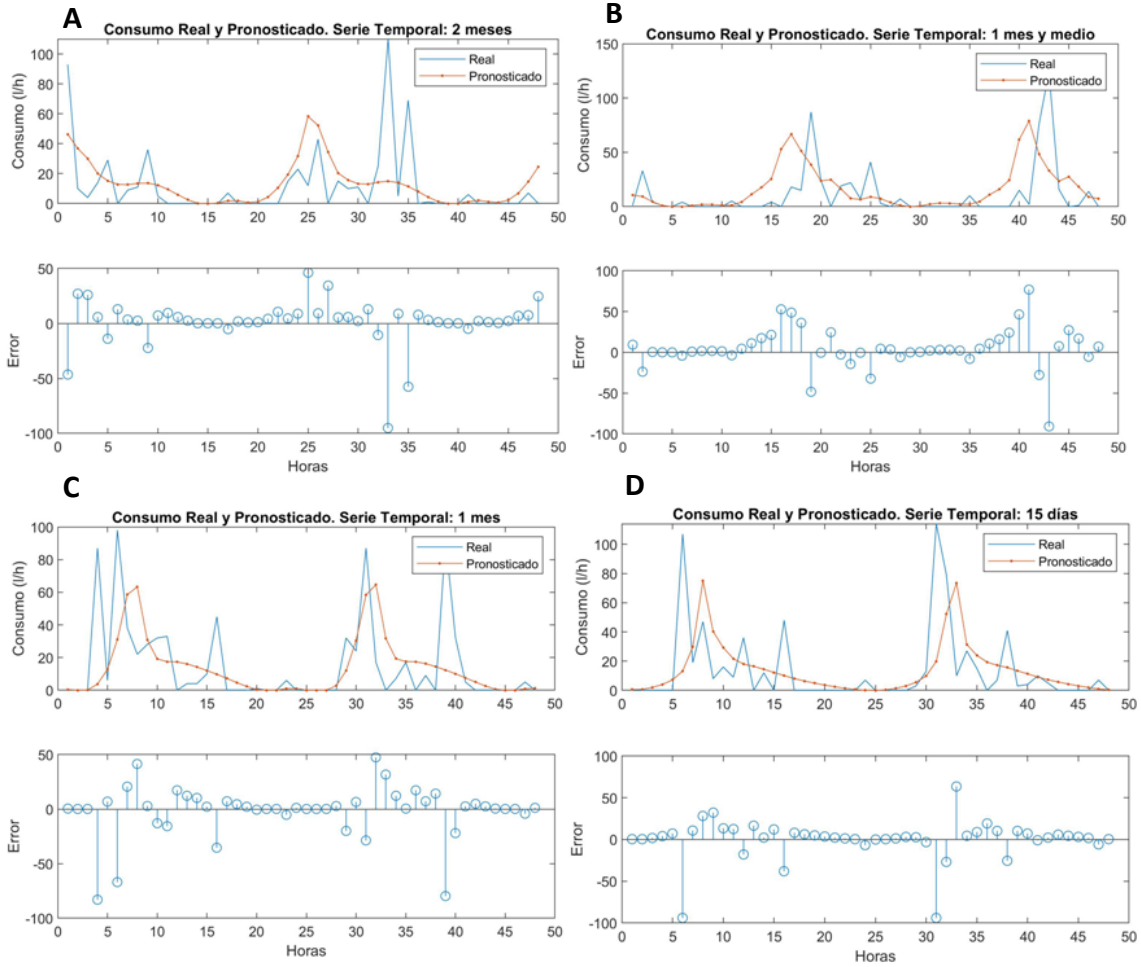


Figura 23.- Aplicación RNN LSTM para el contador C11ALC153667 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D). **Fuente:** Elaboración propia.

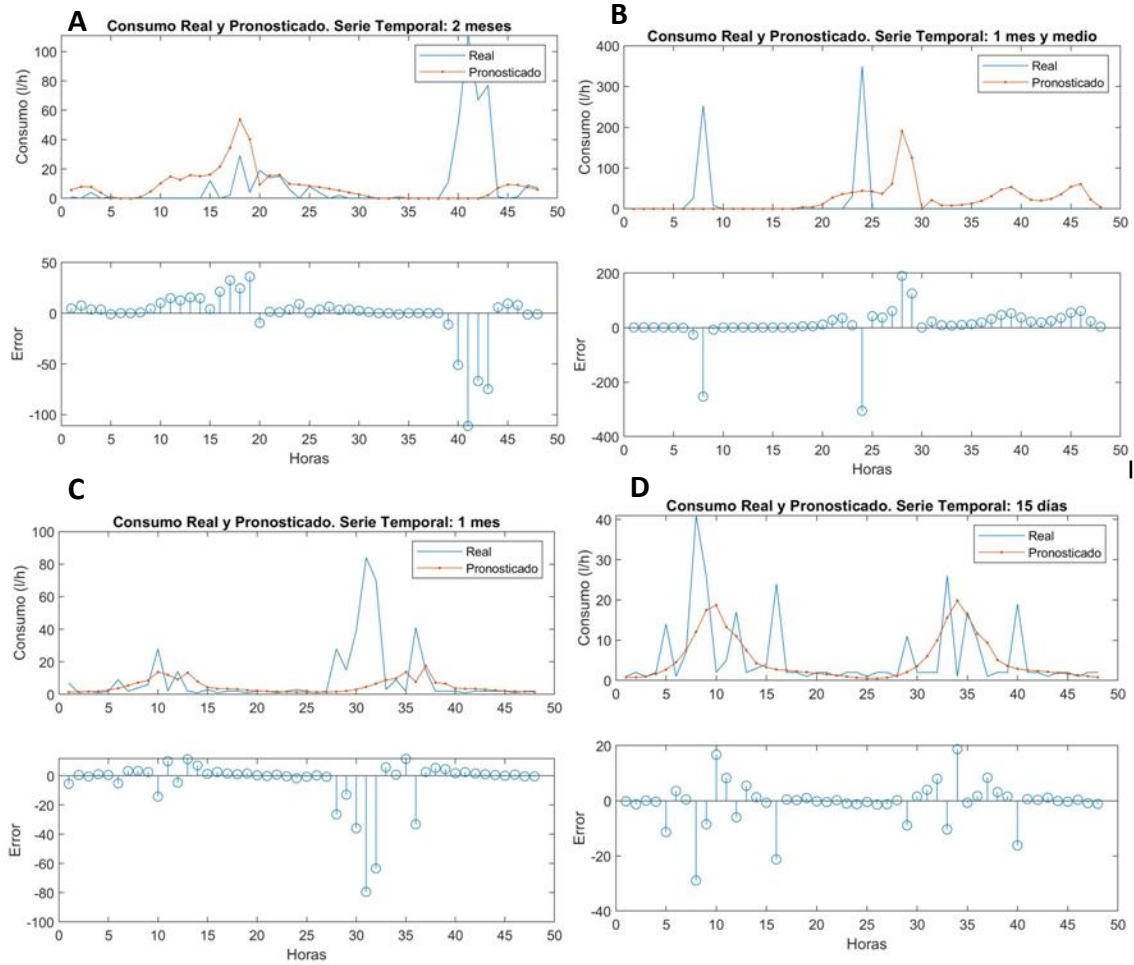


Figura 24.- Aplicación de RNN LSTM para contador C12ALC156544 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D). **Fuente:** Elaboración propia.

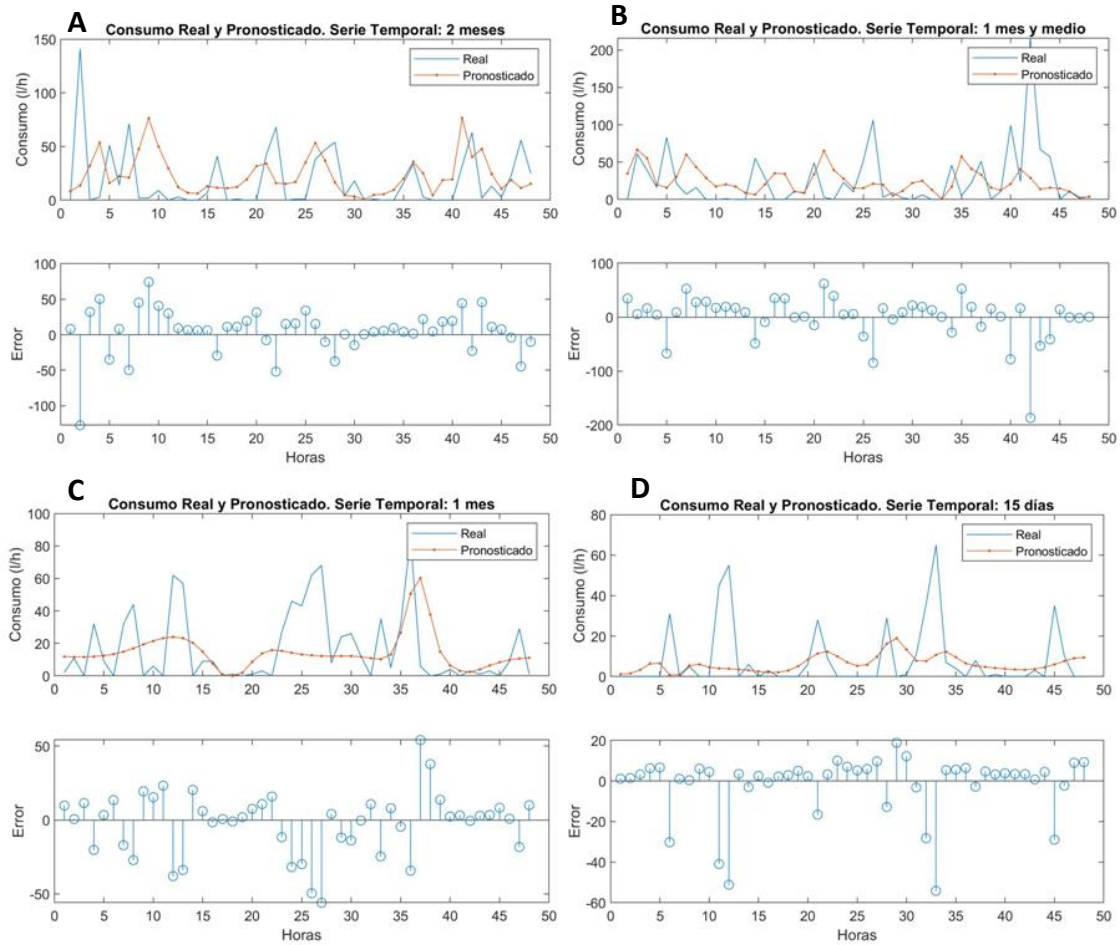


Figura 25.- Aplicación de RNN LSTM para contador C12ALC155545 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D). **Fuente:** Elaboración propia.

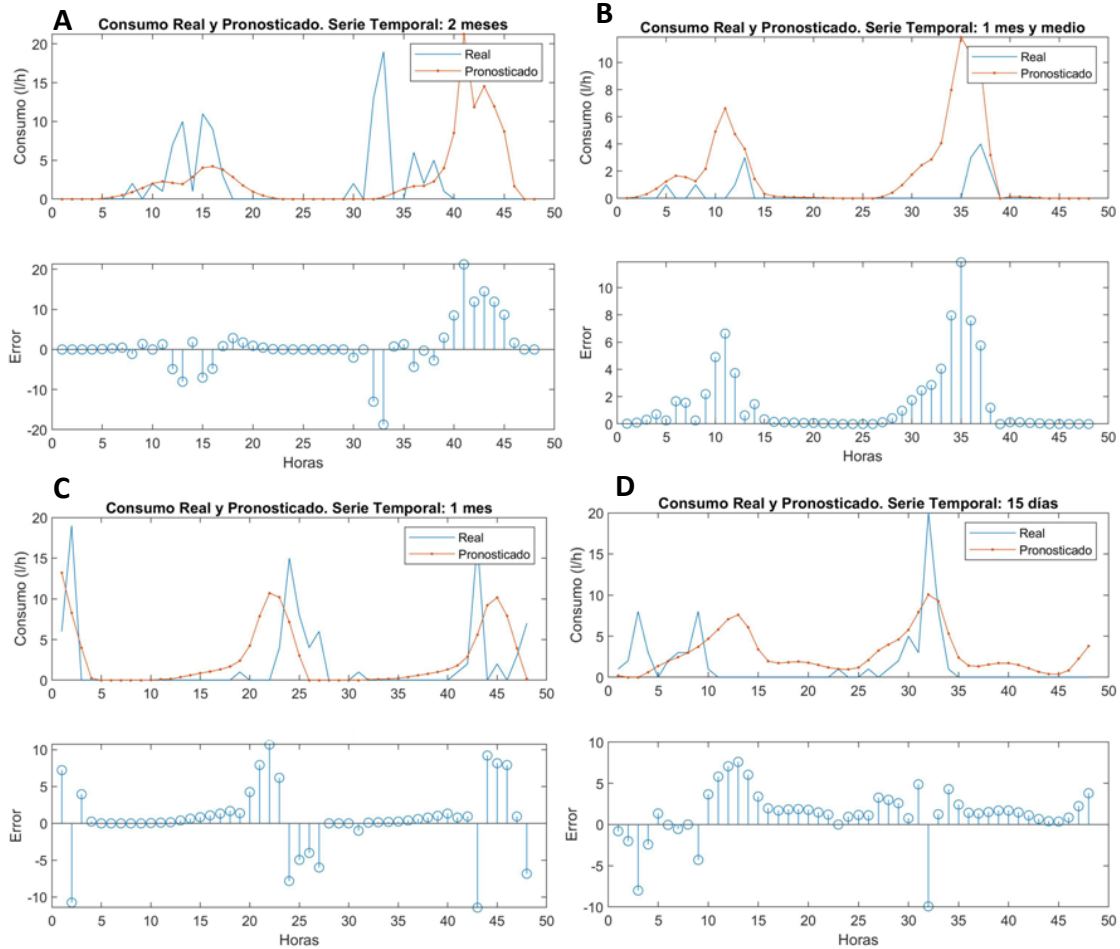


Figura 26.- Aplicación de RNN LSTM para contador C12ALC578265 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D). **Fuente:** Elaboración propia.

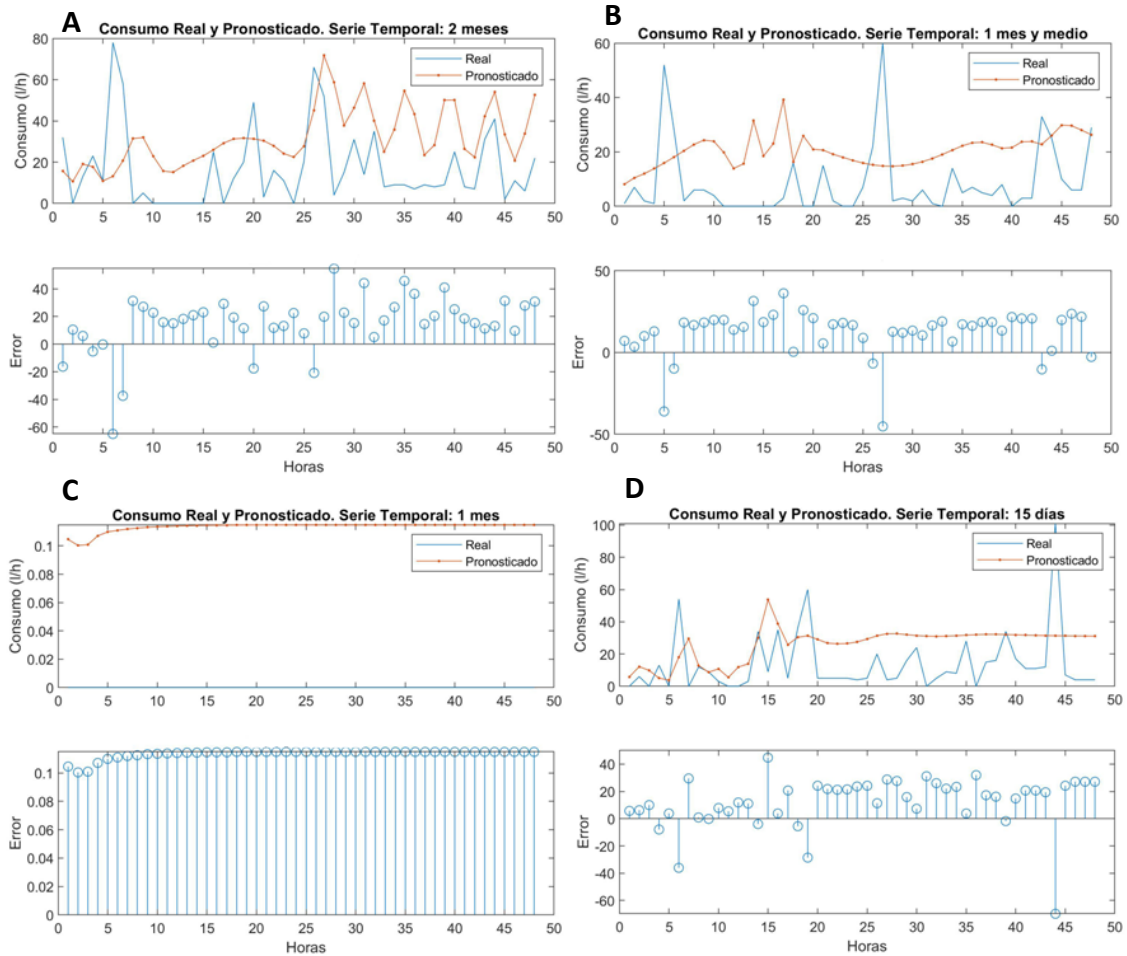


Figura 27.- Aplicación de RNN LSTM para contador C12ALC583338 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D). **Fuente:** Elaboración propia.

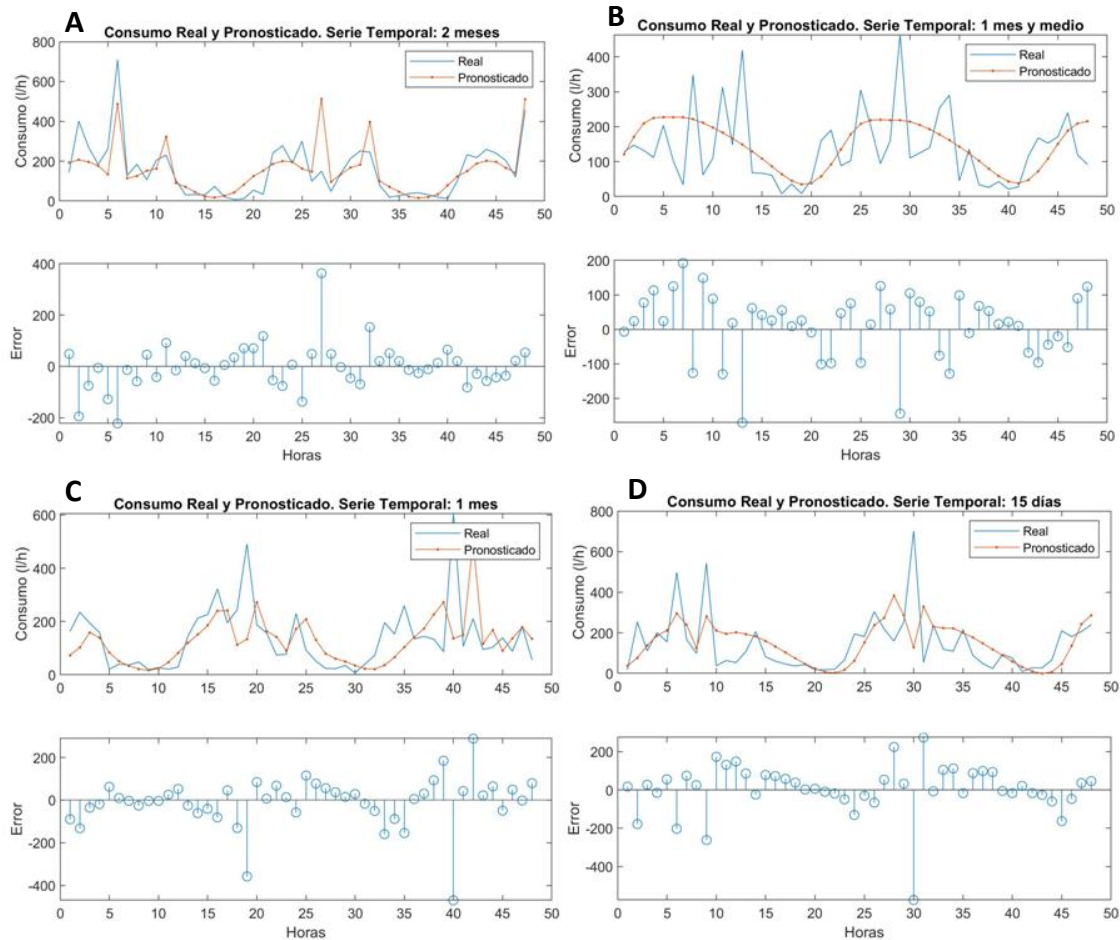


Figura 28.- Aplicación de RNN LSTM para contador C12ALC071969 empleando series temporales de 2 meses (A), 1 mes y medio (B), 1 mes (C) y 15 días (D). **Fuente:** Elaboración propia.

En primer lugar, hay que destacar la gran variabilidad en los errores obtenidos en las estimaciones de consumo en los diferentes contadores (Figuras 23-28) con un horizonte de 48h. Desde casos donde el modelo estima con acierto, en líneas generales, las 48 estimaciones futuras (Figura 28A) hasta casos donde el modelo va perdiendo eficacia (Figura 25B). En la Figura 25C, por ejemplo, el modelo muestra un buen comportamiento, teniendo en cuenta con un posterior filtro de precisión decimal, los errores serían nulos. La Figura 26B en cambio, parece indicarnos que el modelo es capaz de pronosticar mejor, valores bajos de consumo.

Nuestro objetivo como se comentó anteriormente es el de evaluar las predicciones de una hora después de la serie temporal considerada por lo que

en la Tabla 10 se recogen los valores de RMSD obtenidos para las cien iteraciones aleatorias evaluadas en esa primera hora pronóstico.

Tabla 10.- Valores del RMSD tras la aplicación del modelo LSTM. **Fuente:** Elaboración propia.

	RMSD (l/h) (Predicción 1h)			
	2 meses	1,5 meses	1 mes	15 días
C11ALC578265	2,478	0,865	1,247	0,411
C11ALC583338	3,444	1,699	0,216	0,942
C12ALC153667	5,211	3,689	0,123	0,358
C12ALC155545	4,398	3,218	2,433	0,249
C12ALC156544	2,547	1,116	1,024	0,112
C12ALC071969	3,187	8,145	9,741	3,251

En general, los resultados obtenidos en términos de la raíz cuadrada del error cuadrático medio son bastante mejores que los obtenidos en los dos modelos anteriores para todos los contadores. Se puede apreciar también, que en los contadores C11ALC578265, C12ALC155545 y C12ALC156544 el modelo fue capaz de estimar con bastante precisión, obteniendo valores de RMSD de 0,411, 0,249 y 0,112 l/h respectivamente. En estos casos, los resultados parecen indicar que las series más cortas (15 días) son más determinantes para el aprendizaje profundo que las series más largas. En cambio, fueron las series de 1 mes las que resultaron más idóneas de cara a la obtención de valores más bajos de RMSD, en los contadores C11ALC583338 y C12ALC153667. Finalmente, y mostrando una dinámica -al igual que en la evaluación de los dos anteriores modelos- totalmente diferente, el contador de calibre 20 mm arrojó los mejores resultados (RMSD=3,187 l/h) utilizando la serie temporal de mayor tamaño, 2 meses.

6. Conclusiones.

En el presente trabajo se han evaluado los diferentes modelos que se han desarrollado para la estimación de consumos de agua horarios, basados en RNA. Previamente se ha aplicado una pre-validación de los datos, necesaria para tener un set de datos verosímil y con garantías, con el objetivo de entrenar adecuadamente estos modelos de aprendizaje. Se han analizado los modelos NFIT, NARX, función de base radial y aprendizaje profundo (LSTM).

A la vista de los resultados obtenidos, sin duda el modelo LSTM es el que mejor comportamiento ha mostrado a la hora de estimar un consumo futuro con un horizonte de una hora, para todos los contadores evaluados, obteniendo unos errores realmente bajos. El presente trabajo también concluye que la dinámica de cada contador es única y que eso requiere un análisis pormenorizado a la hora de seleccionar la longitud de la serie utilizada para estimar el valor futuro.

Los modelos de base radial y empleando las series temporales dadas, no han resultado adecuados para este tipo de variable.

Se aprecia también que existen grandes diferencias entre los contadores de calibre 15 mm y el contador de calibre 20 mm, que, por su constante funcionamiento, no tiene en su serie de datos de consumo valores de 0. Esta circunstancia hace que para el contador de calibre 20 mm el modelo LSTM sea el único que precise la serie temporal de 2 meses para el aprendizaje, bastando 15 o 30 días para el resto de contadores. En relación con este modelo de RNA hay que precisar que el coste computacional es bastante elevado en relación con los otros dos modelos evaluados.

En líneas generales, hay que destacar que el modelo NARX funciona mejor que el modelo NFIT en los contadores de calibre 15 mm, no siendo así para el contador de 20 mm. Es importante matizar que el modelo NFIT, pese a no mostrar en términos globales un comportamiento excelente en los contadores evaluados, si que nos permite estimar con cierta fiabilidad consumos horarios



**Modelos predictivos de consumo de agua
basados en Redes Neuronales Artificiales**

Máster en Tecnología del Agua en Ingeniería Civil



utilizando solo y exclusivamente los 4 inputs disponibles: día del año, día del mes, día de la semana y hora. Ambos modelos podrían aumentar su rendimiento si se entrenan con series temporales más largas y se clasifican previamente las dinámicas de consumo.

7. Líneas futuras de mejora.

El presente trabajo abre una línea de investigación en el campo de la telelectura, debido a que el estudio muestra las diversas posibilidades que aportan los modelos de RNA en la estimación de consumos de agua y por tanto en la validación de series continuas de registros de calidad.

Para futuros estudios, sería interesante separar antes de entrenar cualquiera de los modelos evaluados (en especial NFIT y NARX), las diferentes dinámicas observadas en los consumos, como por ejemplo, programas de riego, dinámicas de consumo de los electrodomésticos, fugas y averías contadores domiciliarios o de empresas, etc., utilizando también métodos de clasificación mediante inteligencia artificial como paso previo a su entrenamiento y aprendizaje. Para el modelo LSTM se recomienda modificar la arquitectura, pese a aumentar el coste computacional, y evaluar en detalle los resultados.

Por motivos de tiempo, en el presente TFM no se ha llevado a cabo un estudio pormenorizado de la distribución de errores de cada uno de los modelos, así como los factores que influyen en ellos, lo cuál sería muy interesante de cara a su optimización. Asimismo, la posibilidad de implementar otro tipo de técnicas como la lógica difusa o algoritmos genéticos sería una vía a explorar de gran interés, pudiendo desarrollar modelos híbridos más eficientes.

Además, sería recomendable la utilización de series más largas de datos, permitiendo caracterizar con mayor precisión las dinámicas de consumo de cada contador. También sería crucial la evaluación de series de datos de calibre de contadores superiores y/o de otros sectores, teniendo así una muestra más representativa de los consumos reales dentro de la red de abastecimiento de agua potable de la Empresa Municipal de Aguas de Córdoba S.A.

Por último, destacar la posibilidad que ofrecen las tecnologías actuales, de poder instalar un dispositivo junto al contador que nos permita chequear en tiempo real o cuasireal los consumos leídos, y realizar mediante este tipo de



**Modelos predictivos de consumo de agua
basados en Redes Neuronales Artificiales**

Máster en Tecnología del Agua en Ingeniería Civil



modelos estimaciones in situ. Esta posibilidad podría ser factible utilizando dispositivos que incorporen un microprocesador, ejecutándose directamente en los momentos en los que se producen huecos en la serie de datos, debido a los diferentes fallos de comunicación en la red de telelectura. En este sentido, sería muy interesante la evaluación de los tiempos consumidos por cada uno de los modelos para realizar esos pronósticos en este tipo de hardware.

8. Bibliografía.

Beale, M. H., Hagan M.T., Demuth, H.B. (2015). Neural Network Toolbox "Getting Started Guide" R2015b.

Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw, 5(2), 157-166.

Canal de Isabel II. (2017). Cuaderno I+D+i 25. "Sistemas de reconocimiento de patrones para identificación de usos finales del agua en consumos domésticos"

Dawson, C., Wilby, R. (2001). Hydrological modelling using artificial neural networks. Progress in Physical Geography. Vol.25, 80-108.

Demuth, H., Beale, M., Hagan M. (2016). Neural Network Toolbox 6 "User's Guide". Matlab.

Díaz, S., Mínguez, R., González, J. (2016). Aproximación estocástica al análisis de observabilidad en redes de abastecimiento de agua. Ingeniería del agua. Vol. 20, Número 3.

Estévez J., García-Marín, A.P., Zardi, D., Gavilán P. (2015). Detection of spurious precipitation signals from automatic weather stations in irrigated areas. International Journal of Climatology, 35, págs. 1556-1568.

Fernández, J. M. (2017). Presentación "Sistemas de telelectura de EMACSA". Documento interno EMACSA.

Gurung, T., Stewart., R.A., Beal, C.D., Sharma, A.K. (2017). Reprint of: Smart meter enabled informatics for economically efficient diversified water supply infrastructure planning. Journal of Cleaner Production, vol 163, suplemento 1, 138-147.

Hochreiter, J. (1991). Diplomarbeit im fach informatik "Untersuchungen zu dynamischen neuronalen Netzen". Technische Universität, München.

Hochreiter, S. y Schmidhuber, J. (1997). Long short-term memory. Neural computation 9 (8), 1735-1780. Technische Universität, München.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences of the United States of America-Biological Sciences, vol 79, nº8, 2554-2558.

Marquardt, D. (1963). An algorithm for least-square estimation of nonlinear parameters. SIAM Journal on Applied Mathematics 11 (2). 431-441. doi: 10.1137/0111030.

Levenberg, K. (1944). A method for the solutions of certain non-linear problems in least squares. Quarterly of Applied Mathematics 2. 164-168

Li, Y., Zhu, P., Xie, X., He, G., Aihara, K. (2012). Learning-induced pattern classification in a chaotic neural network. Physics Letters A, vol 376, issue 4, 412-417.

Lohani, A., Kumar, R., Singh, R.D. (2012). Hydrological time series modeling: A comparison between adaptive neuro-fuzzy, neural network and autoregressive techniques. Journal of Hydrology, vol 442-443, 23-35.

López-Lineros, M., Estévez, J., Giráldez, J.V., Madueño, A. (2014). A new quality control procedure based on non-linear autoregressive neural network for validating raw river stage data. Journal of Hydrology. 510, págs. 103-109.

Martínez, 2017. Web "Red de agua pública". Presentación " Experiencias y retos para una gestión del ciclo urbano del agua". Obtenido de: <https://redaguapublica.wordpress.com/page/7/>

Meskuotiene, A., Kaskonas, P., Joneliunas, S., Urbonavicius, B.G. (2016). Conformity Assessment and Validation of Automatic Meter Reading Systems. MAPAN-Journal of Metrology Society of India, 257-264.

Moreno, A. (2009). Proyecto fin de carrera Ingeniería Industrial "Desarrollo de una interfaz gráfica de redes neuronales usando Matlab". Universidad Carlos III, Madrid.

Nguyen, K., Stewart, R.A., Zhang, H., Sahin, O., Siriwardene, N. (2018). Re-engineering traditional urban water management practices with smart metering and informatics. Environmental modelling & software, vol 101, 256-267.

Olah, C. (27 de Agosto de 2015). Colah's blog. Obtenido de <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Romero, M. Á. (2014). Documento "Descripción de instalaciones-2014". Documento interno EMACSA.

Schmidhuber, G. &. (2000). "Recurrent Nets that time and count". Idisia, Lugano.

Sciuto, G., Bonaccorso, B., Cancelliere, A., Rossi, G. (2009). Quality control of daily rainfall data with neural networks. Journal of Hydrology, vol 364, issues 1-2, 13-22.

Willmott, C. J. (1982). Some comments on the evaluation of model performance. Bull. Am. Meteorol. Soc., 63 (11), 1309-1369.



**Modelos predictivos de consumo de agua
basados en Redes Neuronales Artificiales**
Máster en Tecnología del Agua en Ingeniería Civil



Anejo I.- Características técnicas del contador.

ALTAIR V4

CONTADOR VOLUMETRICO

DIEHL
Metering



DESCRIPCIÓN

ALTAIR V4 representa la nueva generación de contadores volumétricos desarrollada de acuerdo con las últimas normas MID (Directiva de Instrumentos de Medición) OIML R49, EN 14154 e ISO 4064 y con las exigencias del terreno. ALTAIR V4 presenta una metrología muy precisa y un diseño muy compacto, lo que le permite adaptarse a los entornos más difíciles. ALTAIR V4 es un contador resistente, compatible con las distintas calidades de agua.

Al ser de tipo modular, ALTAIR V4 puede equiparse en cualquier momento con el sistema de radio IZAR, el dispositivo IZAR DOSING o el emisor de impulsos IZAR PULSE, lo que le abre la puerta a la integración de otros sistemas.

PRINCIPALES CARACTERÍSTICAS

- ▶ Amplio rango de dinámica
- ▶ Caudal de arranque de 0,4 l/h
- ▶ Instalación en cualquier posición
- ▶ Compacto
- ▶ Homologado según MID hasta R=800
- ▶ Disponible con cuerpo de latón y composite
- ▶ Totalizador modular "Ti" de serie
- ▶ Totalizador disponible en cobre/vidrio en opción
- ▶ DN 15 & 20 mm

ALTAIR V4

CONTADOR VOLUMETRICO

CARACTERÍSTICAS METROLÓGICAS

Diámetro nominal	DN	mm	15	15	15	15	15	20	20	20
Longitud*	L	mm	110	115	134	165	170	110	165	190
Cuerpo			latón/ composite	latón/ composite	latón/ composite	latón/ composite	latón/ composite	latón	latón	latón/ composite
Caudal nominal	Q ₃	m ³ /h	2,5	2,5	2,5	2,5	2,5	4	4	4
R*	Q ₃ /Q ₁		160	160	160	160	160	160	160	160
Caudal de arranque		l/h	0,4	0,4	0,4	0,4	0,4	0,7	0,7	0,7
Caudal mínimo*	Q ₁	l/h	15,62	15,62	15,62	15,62	15,62	25	25	25
Caudal de transición	Q ₂	l/h	25	25	25	25	25	40	40	40
Caudal máximo	Q ₄	m ³ /h	3,125	3,125	3,125	3,125	3,125	5	5	5

* Otros valores bajo pedido

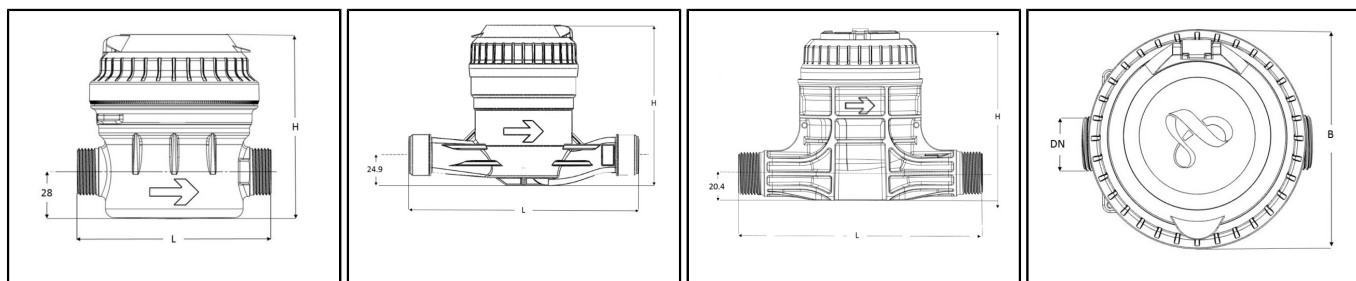
HOMOLOGACIÓN

ALTAIR V4	
Homologación MID	LNE - 6250 (DN 15 & 20)
Alimentaridad	ACS - WRAS - KTW - BELGAQUA

TEMPERATURAS Y PRESIÓN

ALTAIR V4	
Intervalos de temperatura del agua	°C 0 ... +50
Presión nominal	bar 16

DIMENSIONES



DN 15 - composite

DN 20 - latón

DN 20 - composite

Vista superior

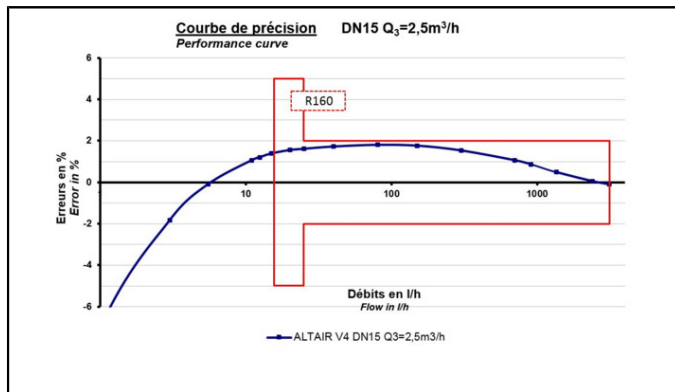
Diámetro nominal	DN	mm	15	15	15	15	15	20	20	20
Longitud*	L	mm	110	115	134	165	170	110	165	190
Cuerpo			latón/ composite	latón/ composite	latón/ composite	latón/ composite	latón/ composite	latón	latón	latón/ composite
Anchura	B	mm	92,7	92,7	92,7	92,7	92,7	90	90	90/97
Altura	H	mm	104/108	105/117	105/109	105/108	105/108	132	132	132/135
Rosca		pulgada	3/4"	3/4" & 7/8"	3/4 "	3/4 "	3/4"	1"	1"	1"
Peso		kg	0,90/0,51	0,95/0,51	1/0,55	1,05/0,58	1,07/0,58	1	1,2	1,34/0,72

*otros valores disponibles previa solicitud

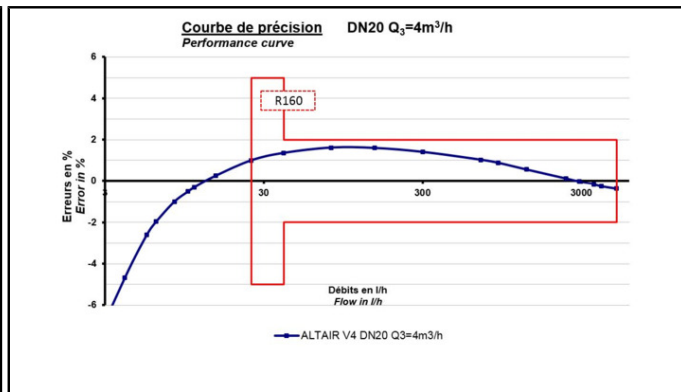
ALTAIR V4

CONTADOR VOLUMETRICO

CURVA DE PRECISI3N



DN 15



DN 20

PERDIDA DE CARGA

ALTAIR V4

kvs ($\Delta P = Q^2 / Kvs^2$)

3,25 (DN15) / 5,41 (DN20 lat3n) / 5,30 (DN20 composite)

Con esta tecnolog3a no se necesitan que instalar tramos rectos.