



UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO
RENOVABLES

MAESTRÍA EN ELECTROMECÁNICA

TÍTULO

“MODELO DE PREDICCIÓN DE DEMANDA DEL SECTOR
RESIDENCIAL DE LA CIUDAD DE LOJA”

TESIS DE GRADO PREVIO A OPTAR POR EL
TÍTULO DE MAGISTER EN
ELECTROMECÁNICA

Autor: Ing. Juan Pablo Cabrera Samaniego.

Director: Ing. Jorge Enrique Carrión González, Mg. Sc.

LOJA - ECUADOR

2013

1859

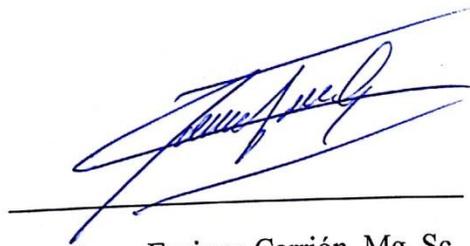
Ing. Jorge Enrique Carrión. Mg. Sc

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en “**Modelo de Predicción de Demanda del Sector Residencial de la Ciudad de Loja**”, previa a la Obtención del título de Magister en Electromecánica, realizado por el señor Ingeniero: **Juan Pablo Cabrera Samaniego**, la misma que cumple con la reglamentación y políticas de Investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 2 de Octubre de 2013

A handwritten signature in blue ink, appearing to read 'Jorge Enrique Carrión', is written over a horizontal line.

Ing. Jorge Enrique Carrión. Mg. Sc.

DIRECTOR DE TESIS

AUTORÍA

Yo, Juan Pablo Cabrera Samaniego, declaro ser autor del presente trabajo de tesis, y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional – Biblioteca Virtual.

Autor: Ing. Juan Pablo Cabrera Samaniego.

Firma: _____

Cédula: 1104179559

Fecha: 2 de octubre de 2013

CARTA DE AUTORIZACIÓN

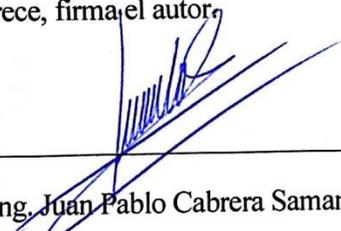
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, Juan Pablo Cabrera Samaniego, declaro ser autor de la tesis titulada: “**MODELO DE PREDICCIÓN DE DEMANDA DEL SECTOR RESIDENCIAL DE LA CIUDAD DE LOJA**”, como requisito para optar por el grado de: **Magister en Electromecánica**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de las tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los 2 días del mes de octubre del dos mil trece, firma el autor,

Firma: 

Autor: Ing. Juan Pablo Cabrera Samaniego.

Cédula: 1104179559

Dirección: Lourdes 11-36 entre Olmedo y Juan José Peña. Correo: juan.cabrera@unl.edu.ec

Teléfono: 07-2577345

Celular: 0999362767

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Jorge Enrique Carrión González, Mg. Sc.

Tribunal de Grado: Ing. Darwin Giovanni Tapia Peralta, Mg. Sc. (Presidente)

Ing. Julio César Cuenca Tinitana, Mg. Sc.

Ing. Edwin Bladimir Paccha Herrera, Mg. Sc.

DEDICATORIA

Dedicado a la memoria de mi querida abuelita

Luz María Castro.

AGRADECIMIENTO

A la Universidad Nacional de Loja.

A mi director de tesis Ing. Jorge Enrique
Carrión González, Mg. Sc.

A mi madre, Dra. Zoila Samaniego
Castro, Mg.Sc

A la Lic. Guadalupe Roa por su
desinteresada colaboración

ÍNDICE

ÍNDICE	1
a. TÍTULO	3
b. RESUMEN	4
b.1. ABSTRACT	5
c. INTRODUCCIÓN	6
d. REVISIÓN DE LITERATURA	8
d.1. CRITERIOS SOBRE LA DEMANDA DE POTENCIA	8
d.1.1. Introducción	8
d.1.2. Factores que afectan en la curva de carga	11
d.1.2.1. Condiciones meteorológicas	11
d.1.2.2. Calendario	12
d.1.2.3. Factores no predecibles	13
d.1.3. Modelos de predicción de demanda de potencia	14
d.1.4. Necesidad de pronosticar la demanda	15
d.2. REDES NEURONALES ARTIFICIALES	16
d.2.1. Redes neuronales biológicas y redes neuronales artificiales	17
d.2.2. Características principales de una rna	20
d.2.3. Funciones de activación, umbral o de transferencia	21
d.2.3.1. Limitador fuerte (HARDLIM)	21
d.2.3.2. Función de transferencia lineal (PURELIN)	22
d.2.3.3. Función de transferencia sigmoideal (LOGSIG)	22
d.2.4. Topología de una RNA	23
d.2.5. Tipos de conexiones entre capas	24
d.2.6. Clasificación de las redes neuronales	26
d.2.6.1. Clasificación según el tipo de conexión	26
d.2.6.1.1. RNA con conexión hacia adelante	26
d.2.6.1.2. RNA recurrentes	26
d.2.6.2. Clasificación según el tipo de información	27
d.2.6.2.1. RNA estáticas	27
d.2.6.2.2. Redes dinámicas	27
d.2.7. Entrenamiento de rna	27
d.2.7.1. Aprendizaje supervisado	30
d.2.7.2. Aprendizaje no supervisado	31
e. MATERIALES Y MÉTODOS	31
e.1. ANÁLISIS DE LA SERIE TEMPORAL Y CONSISTENCIA DE LOS DATOS	31
e.1.1. Descripción de los datos	31
e.1.2. Análisis de la serie temporal	32

e.1.2.1 Determinación de valores máximos de la serie	33
e.1.2.2. Análisis de inconsistencias en los datos.....	35
e.1.3. Diseño y programación de la rna	40
e.1.4. Entrenamiento de la RNA.....	45
e.1.4.1. RNA de predicción de potencia activa	46
e.1.4.1.1. Entrenamiento de RNA potencia activa 1	46
e.1.4.1.2. Entrenamiento de RNA potencia activa 2	49
e.1.4.1.3. Entrenamiento de RNA potencia activa 3	51
e.1.4.1.4. Entrenamiento de RNA potencia activa 4	54
e.1.4.1.5. Entrenamiento de RNA potencia activa 5	56
e.1.4.2. RNA de predicción de potencia reactiva	60
e.1.4.2.1. Entrenamiento de RNA potencia reactiva 1.....	60
e.1.4.2.2. Entrenamiento de RNA potencia reactiva 2.....	63
e.1.4.2.3. Entrenamiento de RNA potencia reactiva 3.....	66
e.1.4.2.4. Entrenamiento de RNA potencia reactiva 4.....	69
e.1.4.2.5. Entrenamiento de RNA potencia reactiva 5.....	72
e.1.4.3. Selección de la rna de predicción	76
f. RESULTADOS.....	78
f.1. Evaluación y predicción de potencia activa mediante RNA.....	78
f.1.1. Evaluación de RNA de predicción de potencia activa	78
f.1.2. Predicción de potencia activa mediante RNA	82
f.1.3. Evaluación y predicción de potencia reactiva	85
f.1.4. Predicción de potencia reactiva mediante RNA	89
g. DISCUSIÓN.....	94
h. CONCLUSIONES.....	96
i. RECOMENDACIONES	97
j. BIBLIOGRAFÍA.....	99
k. ANEXOS.....	101

a. TÍTULO

**MODELO DE PREDICCIÓN DE DEMANDA DEL SECTOR
RESIDENCIAL DE LA CIUDAD DE LOJA.**

b. RESUMEN

En esta memoria se exponen los resultados del diseño de un modelo de predicción de demanda eléctrica del sector residencial de la ciudad de Loja, se tomó como caso estudio el alimentador “Calvario”, perteneciente a la subestación “San Cayetano”, por las características de los usuarios donde se encuentra el alimentador la demanda se la consideró como residencial.

Para la creación del modelo de predicción se consideraron métodos estrictamente no lineales, basados en Redes Neuronales Artificiales (RNA), seleccionando la arquitectura de RNA No-lineal Auto-regresiva, dicha arquitectura posibilita la predicción de series temporales, contando únicamente con la serie de datos histórica, la cual fue proporcionada por la Empresa Eléctrica Regional del Sur (EERSSA).

Para llegar a determinar la base datos que alimentan las entradas de la RNA se realizó el procesamiento integral de la base de datos proporcionada, llegando a eliminar datos y en algunos casos se efectuó el suavizado de la curva de datos obtenida.

En el proceso de validación y prueba del modelo de predicción de demanda eléctrica, se comprobó que los resultados obtenidos por la RNA seleccionada, para cada predicción, se ajusta a las métricas plantadas previo al proceso de diseño.

Para el desarrollo de la investigación se hizo uso del software de ingeniería MatLab (Matrix Laboratory) usado para el desarrollo del proceso de creación del modelo predictivo a través de RNA.

b.1. ABSTRACT

This report presents the results of designing a model predicting electricity demand in the residential sector of the Loja city, was taken as a case study the feeder "Calvario", belonging to the substation "San Cayetano", the nature of where users demand feeder was considered as residential.

To create the prediction model were considered strictly nonlinear methods based on Artificial Neural Networks (ANN), selecting ANN architecture Nonlinear Auto-regressive, this architecture enables the prediction of time series, relying solely with the number of historical data, which was provided by the Empresa Eléctica Regional de Sur (EERSSA).

To get to determine the database that feed entries RNA processing was performed comprehensive database provided, reaching delete data and in some cases made the curve smoothing the data obtained.

In the process of testing and validation of the prediction model of electrical demand, it was found that the results obtained by the selected ANN for each prediction fits planted metrics prior to the design process.

For the development of the research was done using MatLab software engineering (Matrix Laboratory) used to develop the process of creating predictive model through ANN.

c. INTRODUCCIÓN

La dependencia cada vez mayor del consumo de energía eléctrica para las diferentes actividades humanas, ha traído como consecuencia que los usuarios exijan mayor calidad del servicio, por tal motivo las empresas suministradoras en todo el mundo buscan mejorar los esquemas de operación y predicción de demanda eléctrica de las redes de distribución, para mejorar su confiabilidad y economía.

La E.E.R.S.S.A, es la encargada de suministrar energía eléctrica a la región sur del país (Ecuador), con calidad y eficiencia para promover el desarrollo sostenible de la sociedad, mejorando sus condiciones de vida, con la visión de suministrar energía de calidad a toda la región sur y en especial a la ciudad de Loja.

El uso de modelos de predicción de demanda va tomando un interés relevante, tanto desde el punto de vista de ingeniería como financiero.

Desde el punto de vista de ingeniería, los modelos de predicción influyen en el hecho del reordenamiento de los picos máximos en la curva de demanda, mejorar la planificación y asignación de recursos, así como colaborar en la expansión, el diseño y remodelación de los sistemas de suministro eléctrico.

Desde el punto de vista financiero, permiten valorar la influencia del medio, la afectación de las características de explotación al sistema de suministro eléctrico, sin la necesidad de hacer una inversión previa. En el presente trabajo, se desarrolló un modelo de predicción a corto plazo (Horario) mediante el cual se podrá determinar la demanda de consumo eléctrico aproximada, se trabajó con la base de datos del alimentador Calvario, perteneciente a la Subestación San Cayetano. En el desarrollo de todo el documento se especifica el procedimiento a manera de metodología, la cual, podría ser replicada en diferentes sectores, además de poder aplicarse no solo a los sectores residenciales sino a todos los alimentadores que conforman las cuatro subestaciones de la ciudad de Loja, e inclusive, con los datos suficientes, predecir la demanda horaria de consumo de toda la ciudad.

La técnica escogida para la realización del modelo de predicción, es la de Redes Neuronales Artificiales (RNA), las cuales por sus características de; absorción de no linealidades, procesamiento de datos con tiempos de respuesta relativamente rápidos, análisis de una gran cantidad de datos, posibilidad de crear modelos multi-variable de baja complejidad, flexibilidad y robustez, permiten desarrollar modelos de predicción con altos índices de confiabilidad.

Por todo lo anterior se declara como problema de investigación:

No se cuenta con una herramienta que permita pronosticar el consumo de energía eléctrica en el sector residencial de la ciudad de Loja en base al régimen de operación del sistema.

Teniendo como objetivos de la presente investigación:

- Examinar las particularidades de la demanda de energía eléctrica para determinar las variables de mayor afectación en el comportamiento del sistema.
- Determinar la mejor estructura de la Red Neuronal Artificial que permita predecir la demanda de energía eléctrica.

d. REVISIÓN DE LITERATURA

d.1. CRITERIOS SOBRE LA DEMANDA DE POTENCIA

d.1.1. INTRODUCCIÓN

Sin duda la electricidad constituye una de las principales fuentes energéticas con las que cuenta la humanidad. Su empleo se extiende desde los usos industriales hasta el consumo doméstico. Más aún, en un contexto mundial de creciente competitividad, no se concibe una economía en crecimiento y competencia en el ámbito mundial sin un suministro eléctrico confiable y con precios competitivos. Esta marcada dependencia ha convertido a la energía eléctrica en un eje estratégico con marcados efectos de arrastre, no solo de carácter económico sino también social. De modo que podemos afirmar que vivimos en un mundo que funciona con electricidad.

Desde un punto de vista técnico la energía eléctrica es tomada como una “fuente de energía secundaria”, dado que se obtiene a partir fuentes denominadas primarias: carbón, gas, petróleo (combustibles fósiles), hidráulica y nuclear. Al margen de estas y otras consideraciones técnicas, desde un punto de vista económico debemos de referir dos importantes características que nos ayudarán a comprender el funcionamiento del mercado eléctrico, (Murto, 1998), (García Delgado, Myro, & Martínez Serrano, 1997):

- Con independencia de la fuente primaria utilizada, es decir, la energía eléctrica no puede ser almacenada.
- En la mayor parte de usos la electricidad juega el papel de factor productivo que no puede ser reemplazado a corto plazo. Si bien la mayor parte de equipos eléctricos podrían ser adaptados para funcionar con otras fuentes de energía (diésel, gas, gasolina, etc.), a corto plazo estas modificaciones no serían factibles.

Estas características dotan al mercado eléctrico de una serie de peculiaridades interesantes.

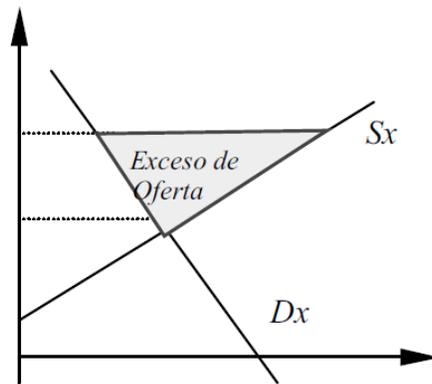


Figura 1.- Exceso de oferta.

En la figura 1 se muestra lo que en un mercado eléctrico sería un exceso de oferta. El área sombreada correspondería con el valor acumulado. Sin embargo, al ser la electricidad un bien no acumulable, el exceso de oferta, es decir la electricidad producida y no consumida inmediatamente, se perdería sin más. En esta situación el área sombreada representaría las pérdidas asociadas a una asignación deficiente de las fuentes primarias, que constituyen recursos escasos y limitados.

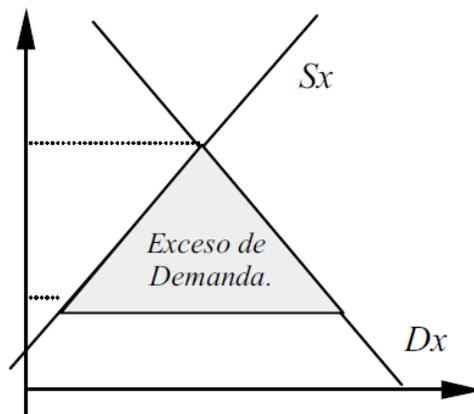


Figura 2.- Exceso de demanda.

Por otra parte, la figura 2 muestra un exceso de demanda. El área sombreada representa el valor de la demanda no atendida. Dada esta situación los demandantes tendrían la posibilidad de esperar a ser atendidos, lo que provocaría un retraso en el suministro eléctrico, generando consecuencias económicas y sociales. En la nueva situación el área

sombreada representaría un coste de eficiencia asociado a la situación de exceso de demanda.

Estas características en el comportamiento de la energía eléctrica motivaron a la creación de organismos dedicados a operador del mercado eléctrico que actúa como regulador, tratando de coordinar en todo momento la oferta y la demanda. En Ecuador las entidades que tiene responsabilidad sobre el mercado eléctrico son:

El Consejo Nacional de Electricidad (CONELEC). Se encarga de desarrollar planes para el desarrollo de la energía eléctrica y ejercer además todas las actividades de regulación y control definidas en la Ley de Régimen del Sector Eléctrico.

El Centro Nacional de Control de Energía (CENACE). Es una corporación civil de derecho privado, de carácter eminentemente técnico, sin fines de lucro, conformado por miembros de todas las empresas de generación, transmisión, distribución y los grandes consumidores. Se encarga del manejo técnico y económico de la energía en bloque, garantizando en todo momento una operación adecuada que redunde en beneficio del usuario final.

El correcto funcionamiento de los entes de control del mercado garantizaría el funcionamiento y aportaría liquidez a generadores y consumidores. El esquema de funcionamiento por el operador debe ser el siguiente:

Conocer el valor que toma la demanda eléctrica en cada momento (curva de carga).

El operador se dirigiría a los centros de producción, indicándoles la carga eléctrica que deben de introducir en la red en cada momento para garantizar la cobertura de la demanda y minimizar la sobreproducción.

Si la demanda eléctrica tuviese un comportamiento puramente lineal el proceso anterior no tendría ningún inconveniente. Sin embargo la demanda eléctrica es muy variable a lo largo de los años, los meses, los días e incluso las horas. Estas características aportan una incertidumbre al sistema eléctrico provocando costes por pérdidas o falta de suministro. Para reducir estos costes se necesita conocer los elementos que provocan la variación de la

demanda, mediante de la curva de carga, con esta información se logran dos objetivos, (Bao, 2000):

- Prever y adaptarse a la demanda en los distintos plazos. A corto, en la operación del sistema, y a largo en la creación de nuevas centrales y redes para el suministro.
- Gestionar la demanda. Es decir, inducir a los consumidores a modificar sus pautas de consumo eléctrico para lograr una mayor eficiencia conjunta del proceso de suministro y uso.

d.1.2. FACTORES QUE AFECTAN LA CURVA DE CARGA

La parte más compleja en desarrollo de un buen modelo de RNA es la relacionada con selección y recogida de la información que alimentará al mismo en la etapa de entrenamiento. Una serie de investigaciones realizadas sobre la demanda eléctrica y sus determinantes en diferentes países, han permitido identificar un conjunto de variables básicas que explican la evolución de la demanda eléctrica a lo largo del tiempo, siendo este conjunto común para la mayor parte de los países. De este modo para modelar la curva de carga. A continuación se detallan las variables básicas que inciden sobre el comportamiento de la curva de carga (Murto, 1998), (Bao, 2000).

d.1.2.1. CONDICIONES METEOROLÓGICAS

En este punto se pueden englobar algunas variables:

- Temperatura
- Velocidad del viento
- Humedad relativa
- Nubosidad
- Pluviosidad, etc.

No todos estos factores inciden sobre la carga, la mayoría de ellos son aleatorios y otros son interrelacionados. Por ejemplo, la temperatura viene explicada parcialmente por la nubosidad, la pluviometría, la humedad relativa, etc.

Entre todos estos factores la temperatura es el más importante, dado que influye de forma directa sobre múltiples fuentes de consumo eléctrico como, aire acondicionado, refrigeradores, etc.

d.1.2.2. CALENDARIO

Existen diferentes efectos relacionados con el calendario que inciden sobre la curva de demanda eléctrica:

Hora del Día: Resulta evidente que la demanda eléctrica realizada a las tres de la madrugada no será igual a la realizada a las tres de la tarde.

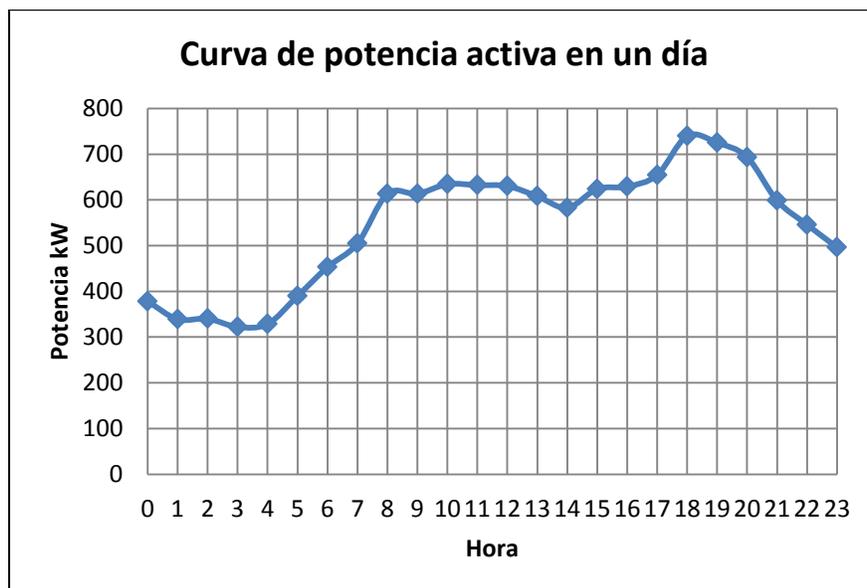


Figura 3.- Curva de potencia activa de un día normal.

La figura 3 muestra como la demanda de potencia tiene dos máximos una de 8AM a 12PM y otro de 3PM a 6PM que es en donde alcanza su máximo para este día, este comportamiento resulta algo lógico si tenemos en cuenta que dichos periodos se corresponde con la jornada laboral estándar.

Fin de semana: En la figura 4 podemos observar la presencia de un factor que incide sobre la demanda eléctrica y que está asociado a los fines de semana y a los días no laborables.

En estos días, se produce una reducción generalizada en el consumo eléctrico como se puede ver en la figura 4.

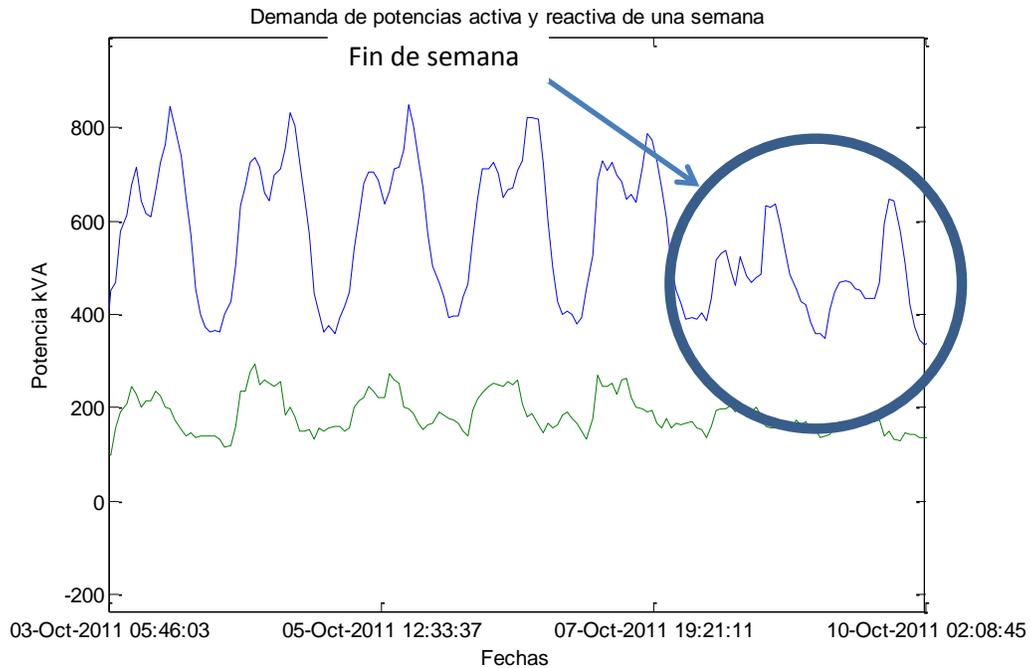


Figura 4.- Curva de demanda de fin de semana.

d.1.2.3. FACTORES NO PREDECIBLES

Debemos de considerar la existencia de una serie de factores que, siendo puramente aleatorios, pueden afectar sensiblemente a la demanda de energía eléctrica. Entre dichos factores destacamos, entre otros: Paros y huelgas generales, eventos deportivos, cierre de instalaciones industriales, desconexiones, mantenimientos, etc. En función del factor del que se trate podemos tratar el mismo como un “Outlier” o dato atípico (figura 5), siendo por tanto necesario realizar un filtrado de los datos de entrada.

También existen factores cuyo efecto sobre la demanda eléctrica puede ser considerado como un “Ruido blanco”, en cuyo caso no será necesario realizar filtrado alguno.

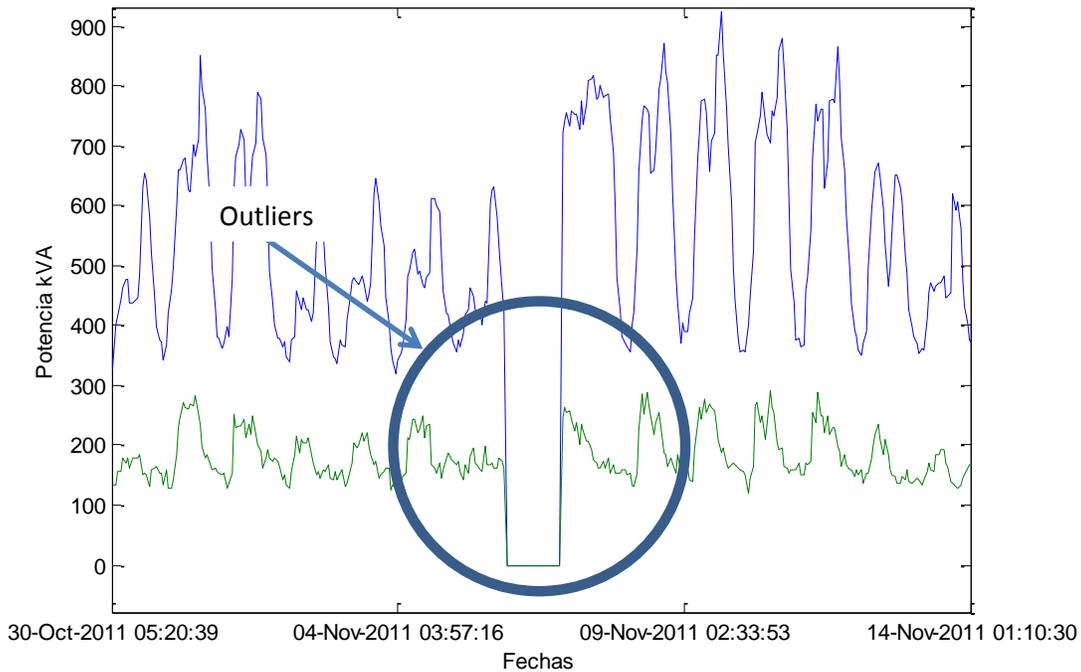


Figura 5.- Outliers.

Este tipo de anomalías causa además una inconsistencia y disminución en la calidad de los datos, lo cual afecta de manera directa a la calidad de la RNA que se vaya a diseñar.

d.1.3. MODELOS DE PREDICCIÓN DE DEMANDA ELÉCTRICA

La predicción de la demanda de electricidad es un insumo fundamental para la toma de decisiones operativas y estratégicas, cuya falta de precisión puede traer altos costos económicos (Bunn & Farmer, 1985), (Pai & Hong, 2005).

La gran cantidad de publicaciones realizadas en los últimos 10 años son una evidencia de los esfuerzos realizados por la comunidad académica en el afán de desarrollar nuevos y mejores métodos para el pronóstico de la demanda de electricidad.

El desarrollo de modelos de predicción de demanda de energía eléctrica acertados, implica un conjunto de dificultades inherentes; entre ellas se cuentan: la complejidad de los factores que determinan la demanda, las características de los datos, el objetivo del pronóstico y las limitantes de las metodologías empleadas.

Existe una tendencia marcada en las series de demanda hacia un comportamiento no lineal, bajo este precedente, las metodologías implementadas son de ese mismo carácter, existen muchos tipos de metodologías que comparten esta característica, las que mas se destacan dentro de los procesos de investigación de predicción de demanda eléctrica son las redes neuronales artificiales (RNA).

d.1.4. NECESIDAD DE PRONOSTICAR LA DEMANDA

La predicción de la demanda de electricidad se realiza para horizontes de corto, mediano y largo plazo (Medina & García, 2005).

El corto plazo corresponde a la predicción en tiempo real para la hora siguiente y con un horizonte de hasta una semana hacia delante; el pronóstico se basa en los datos históricos de demanda diaria y factores climáticos. El mediano plazo corresponde al pronóstico mensual con un horizonte de hasta un año a partir de la demanda histórica de electricidad, y la predicción de algunas variables explicativas como el crecimiento de la economía o el clima. Finalmente, el largo plazo abarca un horizonte de uno hasta diez años; la predicción de la demanda en el largo plazo es usada para la planeación y expansión de la generación y de la transmisión; los datos usados para este pronóstico son la demanda anual, el precio de la electricidad, el precio de productos sustitutos, la evolución demográfica e indicadores económicos, entre otros (Bastidas, Montoya, & Velásquez, 2008).

Dentro de los modelos de predicción de demanda de energía eléctrica, se puede tomar en cuenta el tipo de agente involucrado o interesado en el pronóstico, en función del agente, la predicción de la demanda se justifica por las siguientes razones:

Para los generadores: El pronóstico de demanda en el corto plazo permite definir la cantidad de energía a generar (ofertar) para el día siguiente, programar la operación de unidades y fijar el precio de oferta de la electricidad en el mercado. En el mediano plazo es un insumo fundamental para la definición de planes de mercadeo y ventas en contratos, y análisis de la evolución del mercado. En el largo plazo, permite detectar los periodos para los cuales existen excesos y faltantes de capacidad de generación con el fin de planificar las

inversiones en la expansión del sistema de generación y adoptar medidas de conservación de la energía; en el largo plazo, este es un problema fundamental debido a que la construcción de una nueva planta generadores puede tomar varios años, afectando la confiabilidad del suministro de energía eléctrica (Valanecia, Lozano, & Moreno, 2007).

Para los distribuidores y transmisores: En el corto plazo, permite realizar el monitoreo del comportamiento de la red con el fin de identificar puntos críticos que en el futuro pueden llevar a grandes fallas. Se utiliza en el mediano plazo para el planeamiento de los mantenimientos del sistema. Es un insumo fundamental para definir el programa de expansión del sistema de transmisión en el largo plazo (Valanecia, Lozano, & Moreno, 2007).

Para los comercializadores: En el mediano y largo plazo, es un insumo fundamental para el análisis del comportamiento del mercado y la definición de los planes estratégicos y operativos para la comercialización de la electricidad (Valanecia, Lozano, & Moreno, 2007).

Para el operador del mercado: En el corto plazo es un insumo fundamental para la programación y operación de las unidades de generación, minimizando los ajustes del programa de operación en tiempo real, con el fin de evitar los sobrecostos en la producción de la electricidad (Kher, Sioshanshi, & Soroosshian, 1987).

d.2. REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales son sistemas de procesamiento de información, desarrolladas por científicos cognitivos con el propósito de entender el sistema nervioso biológico e imitar los métodos computacionales del cerebro y su impresionante habilidad para reconocer patrones. (Tkacz & Hu, 1999), (Shachmurove, 2002).

d.2.1. REDES NEURONALES BIOLÓGICAS Y REDES NEURONALES ARTIFICIALES

El elemento funcional básico del cerebro es la neurona (ver figura 6). La neurona, a su vez, está conformada por un cuerpo o soma, unas dendritas y un axón. Cada neurona recibe estímulos eléctricos de otras neuronas a través de las dendritas. En el soma se lleva a cabo la integración de toda la información obtenida en las dendritas. Estos estímulos son amplificados o disminuidos durante la sinapsis y luego sumados. Finalmente, si la suma de todos los estímulos es mayor que el umbral de resistencia máximo de la neurona, entonces el axón transmite a otras células el mensaje resultante de la integración. Estas conexiones sinápticas, cuya intensidad es variable, se usan para enviar mensajes entre neuronas. Las neuronas coleccionan la información y aprenden patrones al reforzar sus conexiones (Hernández López, 2006).

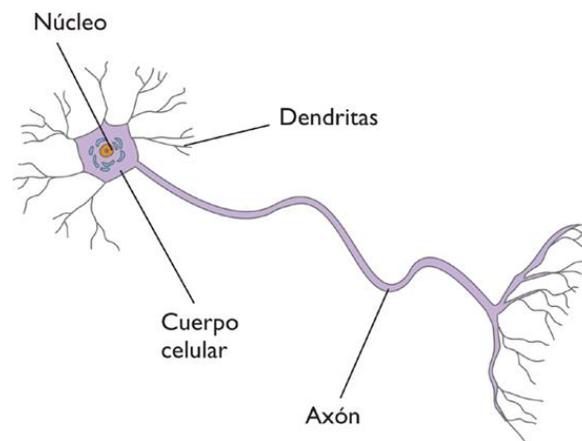


Figura 6.- Neurona biológica.

Las Redes Neuronales Artificiales se inspiran en la estructura y funciones de las neuronas biológicas. Una red neuronal artificial es esencialmente una colección de neuronas interconectadas, agrupadas en capas. Haciendo un paralelo con el esquema recién descrito de procesamiento del cerebro, la neurona artificial recibe distintos valores de entrada “inputs” que son multiplicados por una ponderación. En el escenario más simple, estos productos son sumados para obtener un valor de salida “output”. La forma más básica de red neuronal se encuentra estrechamente vinculada con las técnicas econométricas de

regresión estándar. Este tipo de red simplificada posee dos capas, una de inputs (entradas) y otra de outputs (salidas). La figura 7 ilustra la representación gráfica estándar de una red neuronal feedforward (alimentada hacia delante, es decir que la información fluye desde los inputs hacia el output).

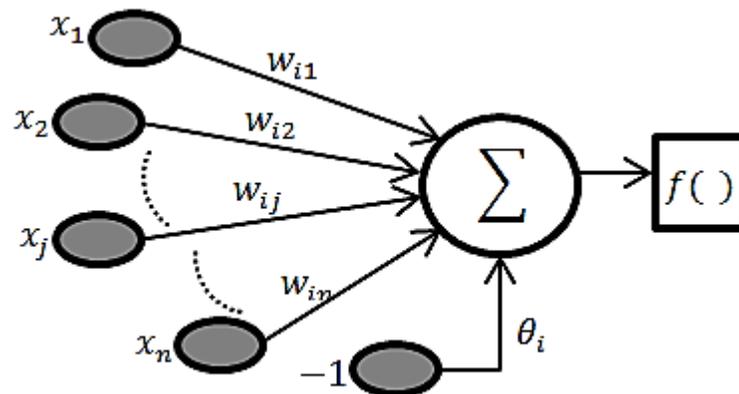


Figura 6.- Modelo de neurona artificial feedforward.

Cada conexión entre un input y un output está caracterizada por un peso ω_i que expresa la importancia relativa de un input particular en el cálculo del output. Para calcular el valor del output en el momento t , la neurona output colecciona los valores de cada neurona input en la observación t y multiplica cada uno de ellos por un peso asociado con la conexión relevante. A continuación se suman estos productos y se obtiene (Yegnanarayana, 2004):

$$y_t = \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_i x_i = \sum_{i=1}^j \omega x_i \quad \text{Ecuación (1)}$$

La ecuación 1 indica que Y es una suma ponderada de X_i , donde cada X_i (las neuronas input) se vincula con Y (la neurona output) por los parámetros ω_i (las ponderaciones). En este modelo lineal, cuando X_i cambia en una unidad, y cambia ω_i unidades.

La neurona output luego procesa este valor usando una función de activación. En la forma más simple de red neuronal, la función de activación es la identidad. En sus cálculos, la red tratará de reproducir el valor del output, dados los valores de los inputs.

Una idea más clara de este proceso se muestra en la figura 7, en donde se puede observar el recorrido de un conjunto de señales que entran en la neurona artificial.

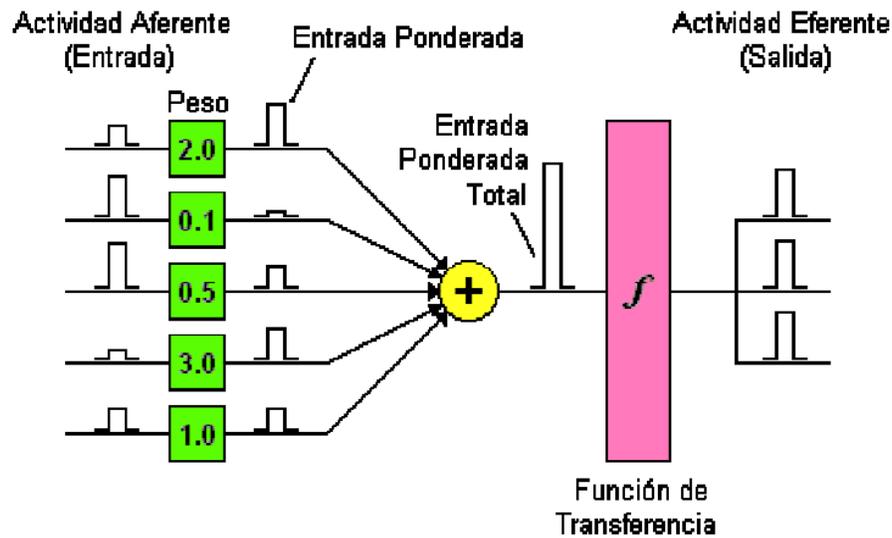


Figura 7.- Recorrido de señales por la neurona artificial.

Ahora, si se cree que existen asimetrías entre los inputs y los outputs, es decir que la relación entre estas variables depende de la magnitud y la dirección de los inputs, entonces la ecuación (1) debe generalizarse con la introducción de no linealidades en la relación. Esto puede lograrse incorporando una función de umbral, que permita que una suma ponderada de los inputs suficientemente grande active un cambio de régimen discreto. Sin embargo, el cambio de régimen no tiene que ser abrupto y para ello se emplean funciones de activación suaves, tales como la función logística.

$$G(z) = \frac{1}{1+e^{-z}} \quad \text{Ecuación (2)}$$

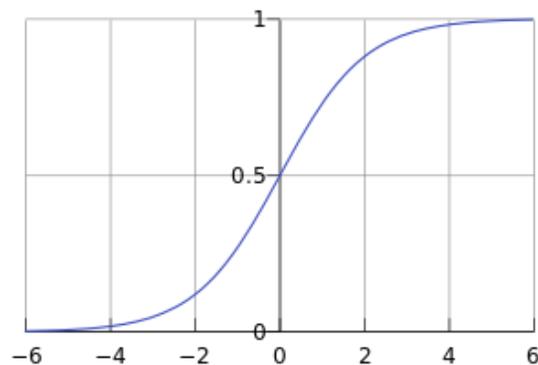


Figura 8.- Función logística.

La ecuación (2) puede aplicarse al modelo lineal en la ecuación (1) para permitir una relación no lineal entre los inputs y el output. Si además se cree que el efecto de los inputs sobre el output no es directo, sino que existen variables intermedias que operan entre ellas; entonces el uso de unidades escondidas o “Hidden Layers” como etapas intermedias donde los inputs X_i y sus pesos son sometidos a una nueva ponderación antes de afectar al output, esto permite que la red capture la relación no lineal entre las variables input y el output (Haykin, 2009).

d.2.2. CARACTERÍSTICAS PRINCIPALES DE UNA RNA

Las RNA deben su capacidad de procesar información a su estructura distribuida y paralela y a su capacidad de entrenamiento y por lo tanto de generalización. Estas dos propiedades hacen que las RNA sean capaces de resolver cierto tipo de problemas muy complejos.

Entre las principales ventajas de las RNA tenemos (Hernández López, 2006) (Arbib, 2003):

- **No linealidad:** las neuronas son elementos de proceso generalmente no lineales. La interconexión de estos elementos genera estructuras de transformación de datos donde este carácter no lineal queda distribuido por toda la red. Esta característica permite modelar procesos intrínsecamente no lineales, pero complica también los métodos de análisis de las estructuras resultantes, impidiendo la aplicación de técnicas de análisis bien establecidas como son las de los sistemas no lineales.
- **Modelado de relaciones entrada salida:** La RNA es capaz de aprender a partir de un conjunto de parejas de datos entrada-salida sin necesidad de conocer las características del problema. Estos pares permiten optimizar los pesos de la red, de tal forma que sea capaz de generar un mapa de entrada-salida para después poder reproducir casos no representados en el conjunto de datos original.
- **Aprendizaje adaptativo:** las RNA tienen la cualidad de aprender a realizar tareas a partir de un entrenamiento, debido a la capacidad de auto-ajuste de los elementos de procesado (neuronas) que componen el sistema. En el proceso de entrenamiento, los pesos que conectan las neuronas se van ajustando de manera que la red reproduce los resultados buscados.

d.2.3. FUNCIONES DE ACTIVACIÓN, UMBRAL O DE TRASFERENCIA

Como se mencionó anteriormente para incluir las no linealidades al modelo, se utiliza las funciones de umbral, las principales funciones son las siguientes:

d.2.3.1. LIMITADOR FUERTE (HARDLIM)

Como se muestra en la figura 9, esta función de transferencia acerca la salida de la red a cero, si el argumento de la función es menor que cero y la lleva a uno si este argumento es mayor que uno. Esta función crea neuronas que clasifican las entradas en dos categorías diferentes, característica que le permite ser empleada en la red tipo Perceptrón (Flórez López & Fernández Fernández, 2008).

$$a = \begin{cases} 1 & \text{si } n \geq 0 \\ 0 & \text{si } n < 0 \end{cases} \quad \text{Ecuación (3)}$$

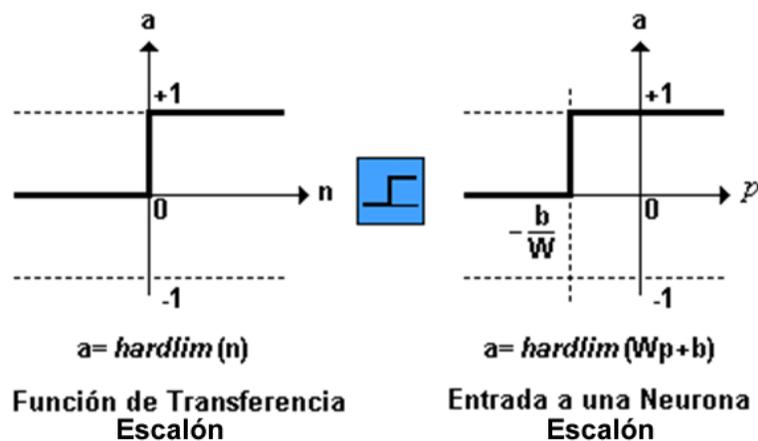


Figura 9.- Función de activación Harlim.

d.2.3.2. FUNCIÓN DE TRASFERENCIA LINEAL (PURELIN)

La salida de una función de transferencia lineal es igual a su entrada.

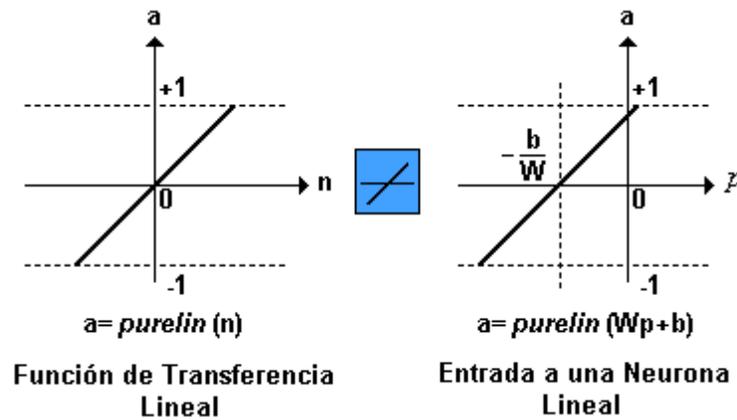


Figura 10.- Función de activación Purelin.

En la gráfica derecha de la figura 10, puede verse la relación entre la salida a de la red, y la entrada p , teniendo en cuenta el valor de ganancia b ; neuronas que emplean esta función de transferencia son utilizadas en la red tipo Adeline (Flórez López & Fernández Fernández, 2008).

d.2.3.3. FUNCIÓN DE TRASFERENCIA SIGMOIDAL (LOGSIG)

También denominada como función logística, figura 11, toma los valores de entrada, los cuales pueden oscilar entre más y menos infinito, y restringe la salida a valores entre cero y uno, de acuerdo a la expresión:

$$a = \frac{1}{1+e^{-n}} \quad \text{Ecuación (4)}$$

Esta función es comúnmente usada en redes multicapa, como la Backpropagation, en parte porque la función Logsig es diferente (Flórez López & Fernández Fernández, 2008).

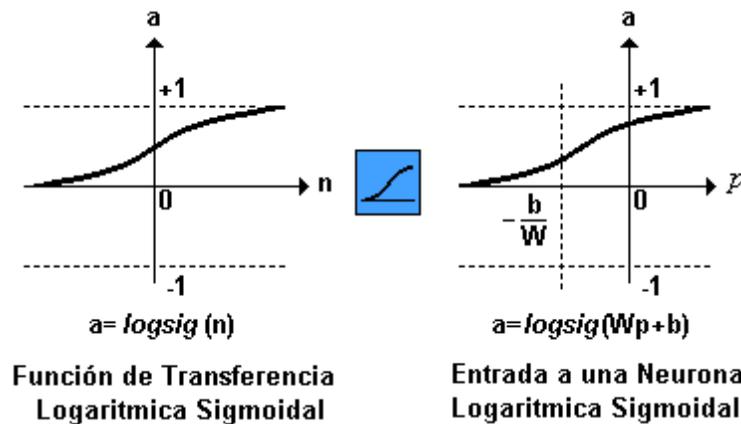


Figura 11.- Función de activación Logsig.

d.2.4. TOPOLOGÍA DE UNA RNA

Normalmente las neuronas se organizan en distintos niveles dentro de la red neuronal, que se suelen denominar capas. Estas capas se clasifican según su situación dentro de la red en base al siguiente criterio (Priddy & Keller, 2005):

De entrada: Es siempre la primera capa de una RNA, y tiene tantas neuronas como entradas a la red. Cada neurona tiene una sola entrada que proviene del exterior y varias salidas a neuronas de capas posteriores. A menudo, en el recuento de capas de una RNA no se incluye la entrada, ya que sus neuronas se limitan a distribuir las entradas externas a las capas posteriores, pero no realizan ninguna operación “típica” de estos elementos de procesado.

Ocultas: Estas neuronas son las que realizan el trabajo y no tienen contacto directo con las señales externas a la red. Las neuronas de estas capas son las que mejor se ajustan a la definición de neurona. Puede haber varias capas de neuronas ocultas, desde cero hasta un número elevado. La primera de las capas ocultas estará conectada con la capa de entrada y la última de estas capas comunicará generalmente con la capa de salida. La forma de interconexión entre las distintas capas ocultas junto con el número de neuronas en cada capa determinará la topología de la red.

De salida: Es la capa de neuronas que trasmite la información generada por la red al exterior. El número de neuronas de esta capa será igual al número de salidas de la RNA.

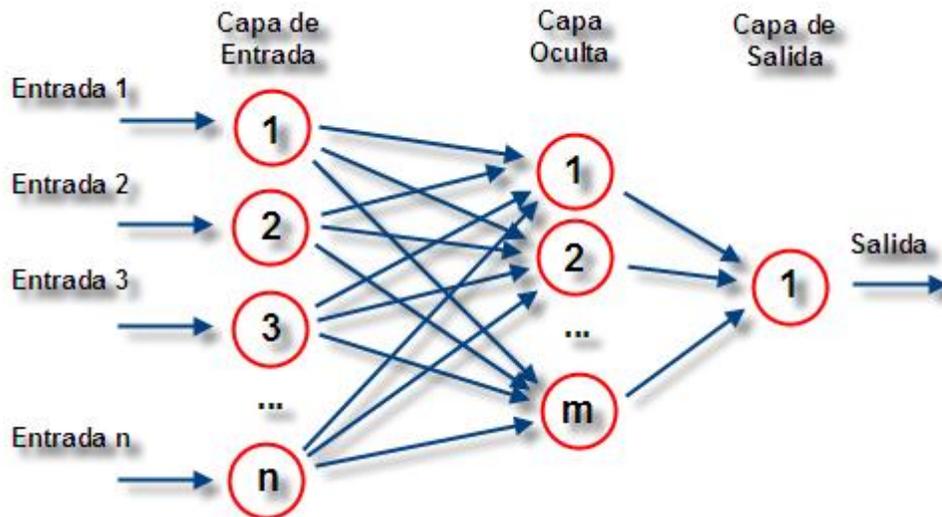


Figura 12.- Red neuronal multicapa.

En la figura 12 se muestra una RNA compuesta por 3 capas: la capa de entrada “Input layer” posee n neuronas de entrada, la capa oculta “Hidden layer” posee m neuronas y la capa de salida “Output layer” posee una neurona de salida, cada capa se encuentra totalmente interconectada con la siguiente, es decir, la salida de una neurona de la capa anterior llega a la entrada de todas las neuronas de la capa siguiente.

d.2.5. TIPOS DE CONEXIONES ENTRE CAPAS

Conexiones hacia atrás: estas conexiones llevan los datos de las neuronas de una capa superior a otras de una capa inferior. Son también llamadas feedbackward (Graupe, 2007).

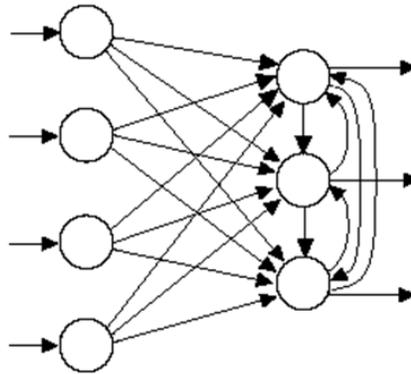


Figura 13.- Red con conexiones hacia atrás.

En la figura 13 se puede apreciar el tipo de conexión hacia atrás, en donde la RNA consta de dos capas con 4 neuronas en la capa de entrada y 3 neuronas en la capa de salida, las cuales se encuentran interconectadas hacia atrás entre sí.

Conexiones laterales: son las conexiones que se hacen entre neuronas de una misma capa. Estas conexiones se diseñan como excitadoras, permitiendo la cooperación (con peso positivo) o la inhibición (con peso negativo) entre neuronas, lo que establece una competición entre neuronas de la misma capa, la conexión se puede apreciar en la figura 14 (Graupe, 2007).

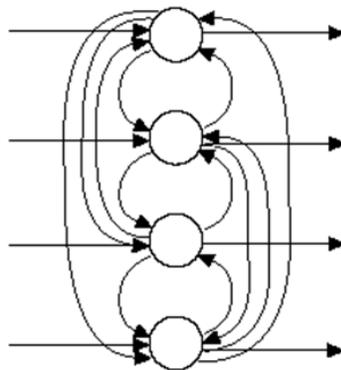


Figura 14.- Red con conexiones laterales.

Conexiones con retardo: los elementos de retardo se incorporan en las conexiones para implementar modelos dinámicos temporales (redes dinámicas). Se almacenan los datos antiguos para después procesarlos junto con la información actual (Graupe, 2007).

Conexiones autorecurrentes: la salida de una neurona se conecta con su propia entrada (Graupe, 2007).

d.2.6. CLASIFICACIÓN DE LAS REDES NEURONALES

Existen varios tipos de criterios para clasificar las RNA, en el presente trabajo nos interesa la clasificación en base al tipo de conexiones y el tipo de datos de entrada, ya que posteriormente en el desarrollo de la red se modificarán las conexiones y los instantes temporales de los datos para poder optimizar los entrenamientos de la misma y poder obtener los mejores resultados posibles.

d.2.6.1. CLASIFICACIÓN SEGÚN EL TIPO DE CONEXIÓN

Dependiendo del tipo de conexión que se establece entre las neuronas de una RNA, estas se pueden clasificar en (Kumar, 2004):

d.2.6.1.1. RNA CON CONEXIÓN HACIA ADELANTE

También conocidas como redes *feedforward*. En este tipo de redes las señales neuronales se propagan siempre hacia adelante a través de las capas de la red. No existen conexiones hacia atrás y normalmente tampoco auto-recurrentes ni laterales. Ejemplos de redes *feedforward* son: Perceptrón multicapa.

Este tipo de redes tienen la ventaja de que la respuesta de la red es rápida y que además admiten algoritmos de entrenamiento del tipo *Backpropagation*.

d.2.6.1.2. RNA RECURRENTE

En este tipo de redes, la información circula tanto hacia adelante (*feedforward*) como hacia atrás (*feedback*), por medio de conexiones que comunican con neuronas de capas anteriores. Las redes recurrentes están especialmente indicadas para los casos en los que la dinámica del proceso es importante, es decir, cuando los patrones de entrada-salida son secuencias temporales de datos. A diferencia de las anteriores, la obtención de las respuestas es más lenta y su entrenamiento es más complicado y largo.

Algunos ejemplos de estas redes neuronales recurrentes son: Perceptrón multicapa recurrente, red de datos temporales, red Hopfield, etc.

d.2.6.2. CLASIFICACIÓN SEGÚN EL TIPO DE INFORMACIÓN

En base a la información que recibe la RNA, estas se pueden clasificar en (Sivanandam & Deepa, 2006):

d.2.6.2.1. RNA ESTÁTICAS

El tipo de información que manejan las redes estáticas tiene carácter estacionario, es decir, el valor de la salida solo depende del valor de las entradas en el mismo momento en el que ésta se produce.

Estas redes se pueden caracterizar estructuralmente por la inexistencia de bucles de realimentación y de elementos de retardo entre los distintos elementos de proceso que las forman. Normalmente van asociadas a las redes con conexiones hacia adelante (feedforward).

d.2.6.2.2. REDES DINÁMICAS

La información que se maneja en este tipo de redes incluye la variable tiempo, es decir, la salida depende de las variables de entrada en el momento actual y de los valores de entrada y/o salida en momentos anteriores.

Estas redes suelen asociarse al tipo de redes recurrentes explicadas con anterioridad. Para incluir la variable tiempo, se puede incorporar en la red retardos temporales o conexiones de tipo feedbackward entre las capas de neuronas. El entrenamiento para las redes dinámicas puede plantear problemas de convergencia y estabilidad, y los resultados que generan son en general de difícil análisis.

d.2.7. ENTRENAMIENTO DE RNA

Construir un sistema que aprenda ha sido tradicionalmente uno de los objetivos más escurridizos de la inteligencia artificial. El aprendizaje, es un proceso de adaptación al

entorno; durante él se crean y manipulan representaciones que sean capaces de explicar dicho entorno.

En las redes neuronales la esencia del aprendizaje se encuentra en la modificación colectiva de los pesos de los elementos de procesado. Una definición de aprendizaje en redes neuronales podría ser la siguiente: Proceso por el cual los parámetros libres de una red neuronal son ajustados a través de un proceso continuo de estimulación por parte del entorno en donde se sitúa el sistema. El tipo de aprendizaje viene determinado por la forma en la que tienen lugar dichos cambios. Esta definición, implica tres hechos fundamentales:

- La red neuronal se encuentra estimulada por el entorno, cambia como consecuencia de esos estímulos y responde de manera diferente al entorno a causa de los cambios que se han producido en sus estructura interna.
- Existen tres paradigmas de aprendizaje: Supervisado, No supervisado (o auto supervisado) y Reforzamiento (se puede considerar como un caso especial del aprendizaje supervisado).
- Existen una gran cantidad variedad de algoritmos dentro de cada una de estas categorías.
- El MatLab (Matrix Laboratory), cuenta con un número interesante de algoritmos de entrenamiento para la RNA, en la tabla 1 se muestra dichos algoritmos y una pequeña definición de los mismos (Sivanandam & Deepa, 2006).

Tabla 1. Algoritmos de entrenamiento en MatLab.

Función Matlab	Descripción
TRAINGD	Los pesos y sesgos de la RNA se actualizan en la dirección del gradiente negativo de la función de ejecución. Respuesta lenta.
TRAINGDM	Los pesos y sesgos de la RNA se actualizan en la dirección del gradiente negativo de la función de ejecución. Se emplea un nuevo parámetro (momento)

	que permite esquivar mínimos locales. De esta forma la actualización es función del último cambio realizado y del gradiente.
TRAINGDA	Se emplea un factor de entrenamiento variable. El factor de entrenamiento se establece en base a las magnitudes de error obtenidas en la iteración actual y anterior. De igual forma los cambios en pesos y sesgos son función de sus valores actuales y anteriores.
TRAINGDX	Es similar al entrenamiento TRAINGDA con la diferencia que en este método se emplea además un parámetro de momento.
TRAINR	Actualización incremental de orden aleatorio.
TRAINRP	Retro propagación con Resiliencia. Emplea el signo de la derivada de la función de ejecución para determinar el cambio en los pesos y sesgos, la magnitud de la derivada no se considera.
TRAINCGF	En los algoritmos de gradiente descendente conjugado se realiza una búsqueda a lo largo de direcciones conjugadas de decremento del gradiente para determinar la magnitud del cambio en los pesos y sesgos. En este caso se emplea la actualización mediante el método Fletcher-Reeves, en el cual se determina la dirección del cambio en base a la dirección de búsqueda de descenso de gradiente actual con la anterior.
TRAINCGP	Método de gradiente descendente conjugado con el algoritmo Polak-Ribière
TRAINCGB	Método de gradiente descendente conjugado empleando el método de restablecimiento de dirección de búsqueda Powell-Beale.
TRAINSCG	Emplea un gradiente conjugado escalado. Es el único algoritmo de gradiente conjugado que no requiere una línea de búsqueda.
TRAINBFG	Emplea métodos de Newton para el cálculo de las actualizaciones. Requiere

	almacenar una aproximación de la matriz Hessiana
TRAINOSS	Método de la secante de un paso. Es una combinación de los métodos de gradiente conjugado y Newton.
TRAINLM	Algoritmo de Levenberg-Marquardt. Emplea modificaciones para disminuir el uso de memoria.
TRAINBR	Regulación Bayesiana. Modificación al algoritmo Levenberg-Marquardt.

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

d.2.7.1 APRENDIZAJE SUPERVISADO

En el aprendizaje supervisado (aprender con un maestro), la adaptación sucede cuando el sistema compara directamente la salida que proporciona la red con la salida que se desearía obtener de dicha red. Existen tres tipos básicos: por corrección de error, por refuerzo y estocástico.

En el aprendizaje por corrección de error el entrenamiento consiste en presentar al sistema un conjunto de pares de datos, representando la entrada y la salida deseada para dicha entrada (este conjunto recibe el nombre de conjunto de entrenamiento). El objetivo, es minimizar el error entre la salida deseada y la salida que se obtiene.

El aprendizaje por refuerzo, es más lento que el anterior. No se dispone de un ejemplo completo del comportamiento deseado pues no se conoce la salida deseada exacta para cada entrada sino que se conoce el comportamiento de manera general para diferentes entradas. La relación entrada-salida, se realiza a través de un proceso de éxito o fracaso, produciendo este una señal de refuerzo que mide el buen funcionamiento del sistema. La función del supervisor es más la de un crítico que la de un maestro.

El aprendizaje estocástico consiste básicamente en realizar cambios aleatorios de los valores de los pesos y evaluar su efecto a partir del objetivo deseado.

d.2.7.2. APRENDIZAJE NO SUPERVISADO

En el aprendizaje no supervisado (o auto supervisado), la red se sintoniza a las regularidades estadísticas de datos de entrada de forma tal que se formen categorías que optimizan, respecto a los parámetros libres de la red, una medida de calidad de la tarea que se quiere realizar con la red. El funcionamiento de estas redes se basa en la búsqueda de características, regularidades, correlaciones y categorías del conjunto de datos de entrada.

Existen diferentes interpretaciones que se le pueden dar a las salidas generadas por una red que utilice este tipo de aprendizaje: similaridad (semejanzas entre la información actual y la información pasada), clusterización (establecimiento de clases), codificación (versión codificada de las entradas) o mapeo (representación topográfica de los datos de entrada).

e. MATERIALES Y MÉTODOS

Previo al desarrollo de las diversas etapas que implica la creación y validación del modelo, se estructura una metodología (figura 15), dicha metodología servirá como base y secuencia organizada para obtener el producto terminado, en nuestro caso el modelo de predicción.



Figura 15.- Metodología para la creación de un modelo de predicción basado en RNA.

e.1. ANÁLISIS DE LA SERIE TEMPORAL Y CONSISTENCIA DE LOS DATOS

e.1.1. DESCRIPCIÓN DE LOS DATOS

Los datos obtenidos para el desarrollo del modelo de predicción usando RNA fueron obtenidos del sistema SCADA de la Empresa Eléctrica Regional del Sur S.A., los datos proporcionados corresponden a los últimos diez meses (Septiembre 2011, Agosto 2012), medidos con un intervalo de tiempo de 15 minutos, durante las 24 horas de cada día, en la tabla 2 de muestra el resumen de descripción de los datos.

Tabla 1. Descripción de datos de entrenamiento.

DESCRIPCIÓN DE LOS DATOS	
Empresa	EERSSA
Subestación	San Cayetano
Alimentador	Calvario
Fecha de inicio	18/09/2011
Hora de inicio	0:15
Fecha de termino	12/08/2012
Hora de termino	0:00
Periodo de muestreo	15 minutos
Muestras	36387

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

e.1.2. ANÁLISIS DE LA SERIE TEMPORAL

Para iniciar con el proceso de desarrollo de la RNA, se hace necesario realizar un paso previo, el cual consiste en analizar la consistencia de los datos que van a ser ingresados en la RNA para la predicción.

El primer paso consiste en llevar los datos históricos de consumo de un archivo Excel a un archivo que reconozca Matlab, en este caso se separó los datos en dos vectores, un vector de tipo “Cell Array” (arreglo de datos de diversos tipos) que contiene los “String” (arreglo de datos de tipo alfanumérico) con las fechas definidas en dos tipos de formatos distintos:

- 21/06/2012 4:15:00 que está en formato dd/mm/aa hh/mm/ss.
- 11-AUG-2012 21:14:59.967 que está en formato dd/mm/aa hh/mm/ss.

El código fuente que permite la lectura y clasificación de los datos en los vectores descritos se detalla en el Anexo 1.

La variedad en los formatos de representación del tiempo, causan a nivel de programación una incertidumbre, por lo cual se llevaron a un solo formato codificado mediante Matlab a través de la función *FechNumVec.m* (ver Anexo 2), como ejemplo de la codificación aplicada tenemos:

- String con la fecha (datos originales): '24-Oct-2003 12:45:07'
- Vector con la fecha (procesada por función): [2003 10 24 12 45 07]
- Fecha en formato serial (codificado en Matlab): 7.3188e+005

La opción de la codificación permite tener a mano y por separado datos de año, mes, día, hora, minuto, segundo, debido a que se toma en promedio 5 muestras por cada hora de los datos como se muestra en la tabla 2:

Tabla 2. Muestra de datos temporales.

18/09/2011 0:15	400,2661214
18/09/2011 0:30	406,5944395
18/09/2011 0:44	380,2264473
18/09/2011 0:45	380,2264473
18/09/2011 1:00	390,7736442

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

e.1.2.1 DETERMINACIÓN DE VALORES MÁXIMOS DE LA SERIE

Para el pronóstico de la demanda de consumo, resulta de mayor utilidad conocer los valores máximos por lo cual se determinará el máximo por cada hora dentro de los datos obtenidos, para llevar a cabo esta tarea se utilizó la función *DataMax.m* (ver Anexo 3), la cual devuelve 7920 muestras, pertenecientes a los valores máximos de potencia activa y reactiva por cada hora, mediante este proceso se aplicó un primer filtro a los datos reduciendo su cantidad y complejidad en el análisis de los mismos.

Los datos se clasificaron por horas, se re-mapeo los intervalos de tiempo, despreciando los valores de los minutos y los segundos lo cual añade un error inherente a la precisión de la ubicación del máximo valor en el tiempo.

Para crear la nueva serie de datos máximos de cada hora re-mapeados, se utilizó la función *tstoollog.m*, (ver Anexo 4) la cual crea una “Time Series” con una fecha de inicio y final similares a las de la serie original con los valores de minutos y segundos en cero, en la figura 15 se muestra la gráfica de la serie temporal de los valores máximos de potencia activa y reactiva por hora.

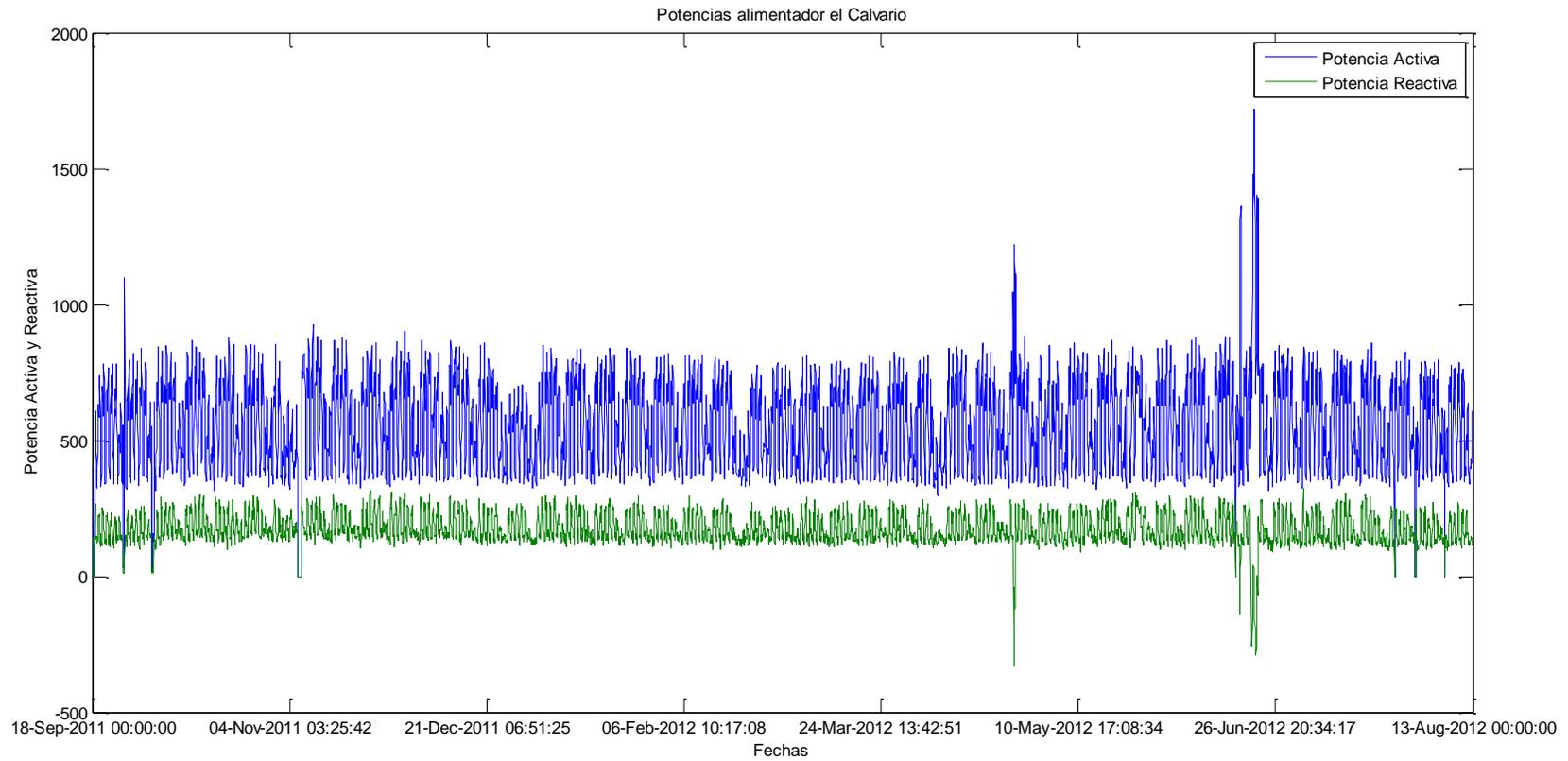


Figura 16.- Serie temporal de datos de potencia activa y reactiva.

e.1.2.2. ANÁLISIS DE INCONSISTENCIAS EN LOS DATOS

La figura 15 muestra una inconsistencia (Outliers) dentro de los datos la cual causa que la calidad de los datos disminuya, dentro de todo proceso de medición se toma en cuenta los errores inherentes al proceso en sí mismo, sumado a fallas en calibración de sensores procesos aleatorios o señales exógenas externas, se generan datos atípicos, los cuales necesitan ser removidos para garantizar la calidad de la RNA y reducir la complejidad en el entrenamiento de la misma.

Para poder eliminar las inconsistencias o los “Outliers” se hace uso de un proceso estadístico, mediante el cual se establece los límites de confianza para la estabilidad de los datos, el cual se ha fijado en 95% o dos desviaciones estándar de la media, límite que permite concentrar la mayor cantidad de datos que siguen una tendencia o un patrón, para el proceso de eliminación se utilizó la función *MoveOutLi.m* (ver Anexo 5), la delimitación de los límites de confianza se muestra en la figura 16 y la figura 17.

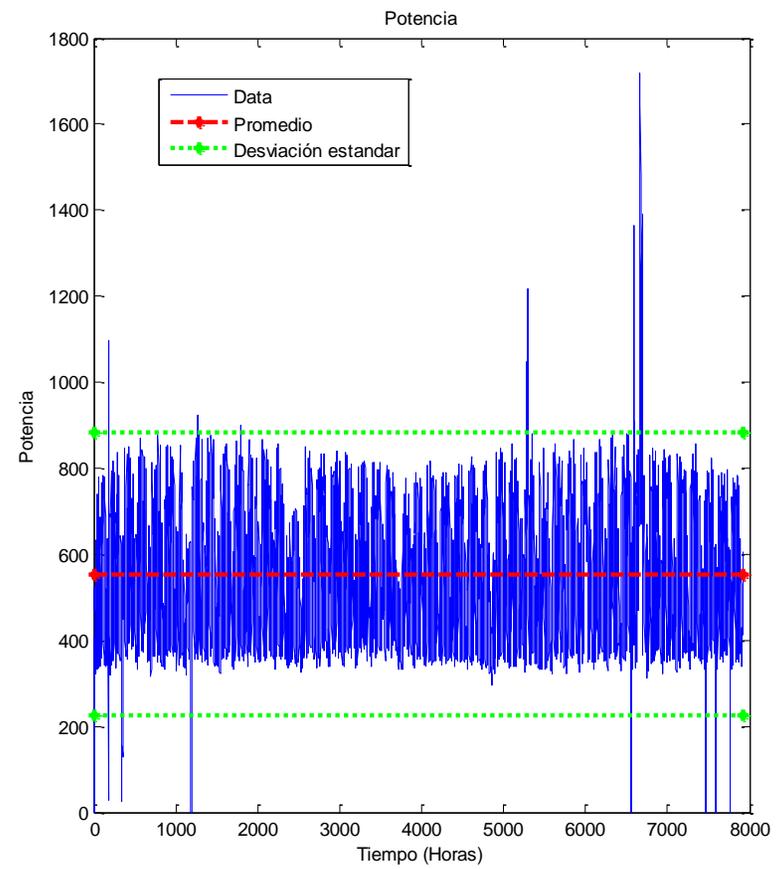
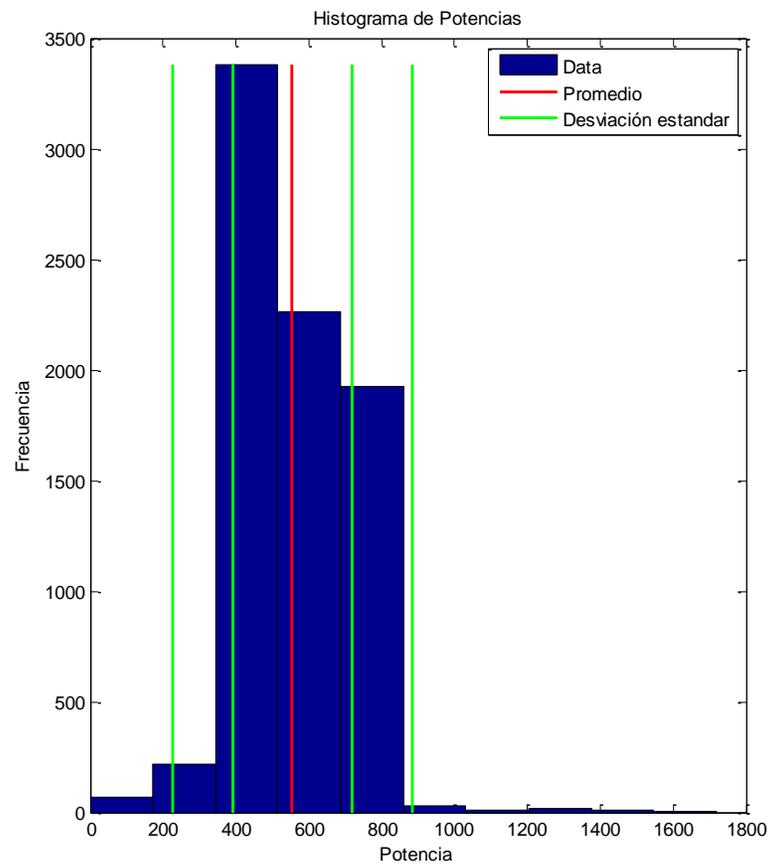


Figura 17.- Límites de confianza para filtrado de potencia activa.

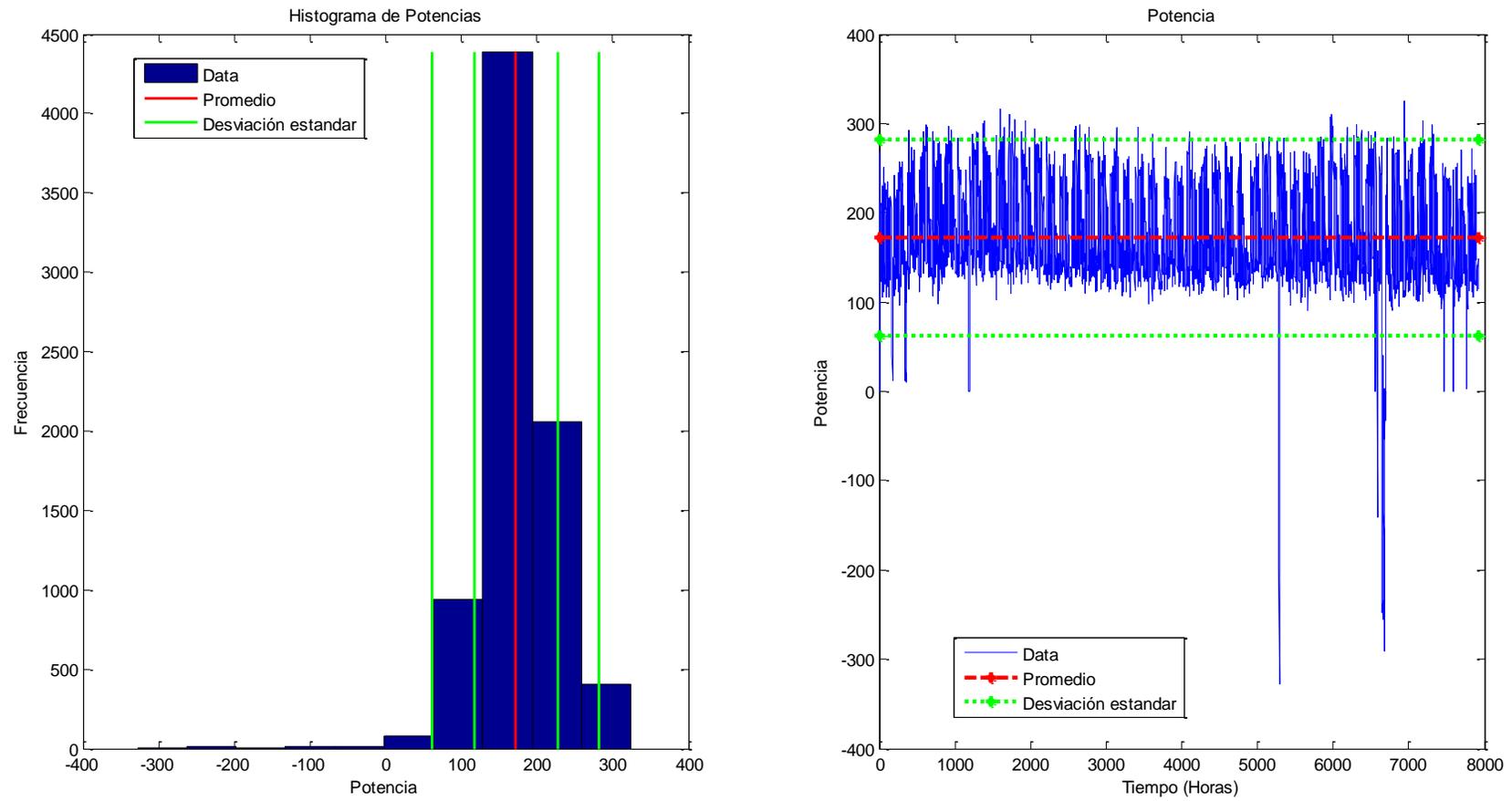


Figura 18.- Límites de confianza para filtrado de potencia reactiva.

Como se observa en los histogramas de frecuencia, debido a que los datos tienen una consistencia o una propensión central delimitado por la media, estos tienden a concentrarse alrededor de un valor, el cual se repite con mayor frecuencia, los datos a sus alrededor forman la tendencia o patrón central de los datos, fuera de dichos intervalos de confianza, los datos son eliminados, puesto que se toman como anomalías lejanas de la tendencia central.

Una vez eliminadas las inconsistencias de los datos quedan huecos que deben ser rellenados para darle continuidad al proceso, la función *MoveOutLi.m* reemplaza los datos atípicos por cero y la función *AjusteData.m* (ver Anexo 6), se encarga de rellenar los ceros con valores coherentes (alisado de la curva) a los datos analizados a través del proceso de interpolación denominado “Nearestinterp”.

La función *AjusteData.m*, devuelve los datos completos e interpolados sin inconsistencias y listos para ser ingresados a la RNA, la figura 18 muestra los datos de salida.

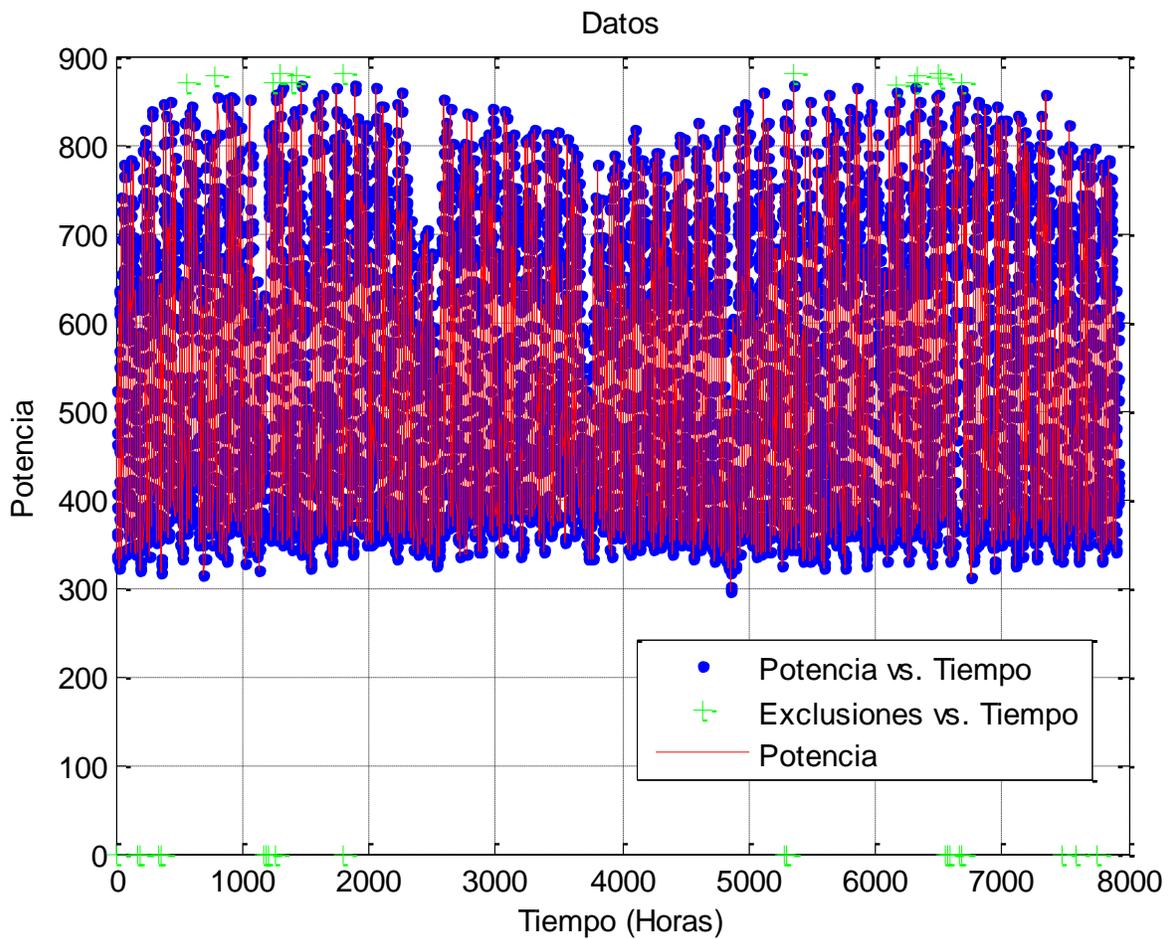


Figura 19.- Datos de potencia activa filtrados.

Los datos se pueden ver en color rojo en rojo, el total de datos interpolados se pueden ver en color azul y los datos que han sido excluidos se pueden ver en color verde, para la potencia reactiva se repite el mismo proceso y los resultados se pueden ver en la figura 19.

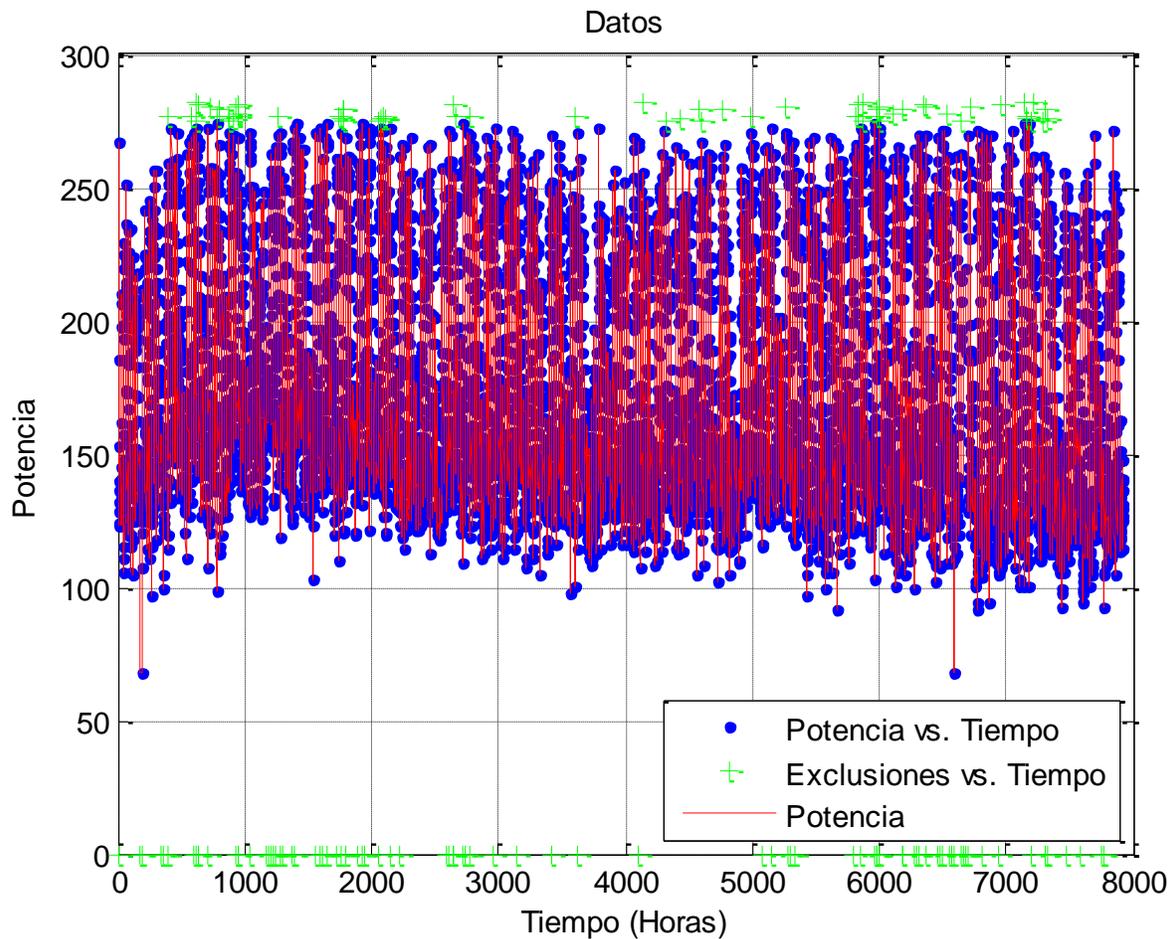


Figura 20.- Datos de potencia reactiva filtrados.

e.1.3. DISEÑO Y PROGRAMACIÓN DE LA RNA

Por la característica y cantidad de los datos obtenidos la arquitectura de RNA escogida para crear el modelo de predicción es una Red No-lineal Autoregresiva o NAR por sus siglas en inglés.

A la serie de datos se la tomó como una serie temporal, cuya métrica de muestreo son las horas, es decir, se tiene los datos máximos de 7920 horas o de 0,9 años, el pronóstico se lo hará tomando en cuenta algunas métricas para asegurar la calidad del modelo de las cuales se hablará en lo posterior.

Para diseñar la RNA se utilizó la función *NAR.m* (ver ANEXO 7) mediante la cual se programó todas las características necesarias. A continuación se detallará el proceso de configuración de la RNA de tipo NAR.

Antes de crear la RNA es necesario realizar el pre-procesamiento de los datos, haciéndolos adaptables a la red, en la etapa de diseño y entrenamiento, para lograr ajustar los datos de utilizó las siguientes líneas de comando:

```
targetSeries = tonndata(PQA1_L, false, false);
```

La función *tonndata* (To Neural Network Data) crea un vector tipo “Cell Array” con los 7920 datos de los máximos por hora de las potencias activas y reactivas.

Para crear la Red No-lineal Autoregresiva se utilizó el siguiente código:

```
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narnet(feedbackDelays, hiddenLayerSize);
```

Esta serie de comandos crean una RNA con 2 retardos en la capa oculta y 10 capas ocultas, estos dos parámetros serán variados en el proceso de creación de las RNA, de modo que se escogerá la que mejor se ajuste a los criterios de diseño. La función *narnet* es la encargada de crear la RNA tipo NAR.

Para el pre-post procesamiento de los datos MatLab ofrece algunas variantes, las funciones escogidas se aplican a la entrada y la salida debido a la característica de la RNA de ser recurrente.

```
net.inputs{1}.processFcns = {'removeconstantrows', 'mapminmax'};
```

En este caso las entradas se procesan a través de dos funciones *removeconstantrows* y *mapminmax*, la primera remueve la componente constante de los datos, dicha componente no posee información alguna a la RNA y causa problemas al algoritmo de

entrenamiento que será implementado posteriormente, la siguiente función mapea los máximos y los mínimos de los datos de entrada normalizándolos a valor de 1 y -1.

Para preparar los datos para el entrenamiento y la simulación se utiliza la función ***preparets***.

```
[inputs, inputStates, layerStates, targets] =  
preparets(net, {}, {}, targetSeries);
```

A través de este código, se crea una serie temporal adaptada de manera particular a las condiciones de la RNA a la cual va a alimentar, realizando desplazamientos temporales mínimos para crear los estados de entrada y los estados de las capas cambiando fácilmente dichos parámetros en función de los retardos que son ajustados al variar las características de la RNA para poder mejorar la red en la etapa de entrenamiento.

Para generar los datos para las diferentes etapas de entrenamiento, validación y testeo se utiliza el siguiente código:

```
net.divideFcn = 'dividerand';  
net.divideMode = 'time';  
net.divideParam.trainRatio = 70/100;  
net.divideParam.valRatio = 15/100;  
net.divideParam.testRatio = 15/100;
```

El método de división es el ***dividerand***, la cual crea grupos de datos escogidos de manera aleatoria del total de las muestras que cumplen el rol de entrada, al tratarse de una serie temporal el modo de división se basa en el tiempo y los grupos se segmentan en proporción de 70% para entrenamiento, 15% para validación y 15% para test.

El siguiente paso consiste en seleccionar la función de entrenamiento para la RNA, en este caso la opción escogida es el algoritmo Levenberg-Marquardt, escogido debido a que una de las métricas de diseño de la RNA es el error cuadrático medio (Men Square Error), el algoritmo de entrenamiento planteado se encarga de reducir la sumatoria de dicho error.

```
net.trainFcn = 'trainlm';  
net.performFcn = 'mse';
```

A través del código mostrado se establece el algoritmo Levenberg-Marquardt que dentro de MatLab se establece mediante la función ***trainlm***, y la métrica para establecer el performance de la RNA es el ***mse***.

Para poder analizar el performance del entrenamiento de la RNA se grafica algunas de las métricas.

```
net.plotFcns = {'plotperform', 'plottrainstate', ...  
               'plotresponse', 'ploterrcorr', 'plotinerrcorr'};
```

En este caso se grafica el performance, los estados del entrenamiento, la respuesta temporal de la RNA, la correlación del error y la correlación del error de entrada. Es necesario aclarar que no todas son analizadas para determinar la calidad de la RNA.

Para entrenar la RNA se utiliza la siguiente línea de comando:

```
[net, tr] = train(net, inputs, targets, inputStates, layerStates);
```

La interface de entrenamiento de la RNA se muestra en la figura 20.

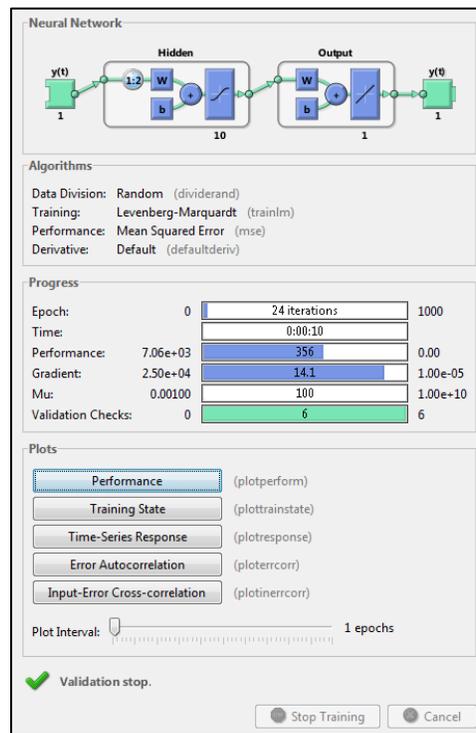


Figura 21.- Interfaz de entrenamiento de RNA en MatLab.

Para realizar el proceso de test de la RNA una vez finalizado el entrenamiento se utiliza el siguiente código:

```
outputs = net(inputs, inputStates, layerStates);
errors = gsubtract(targets, outputs);
performance = perform(net, targets, outputs);
```

Se calcula las salidas y los errores además del performance, el cual se tabulará para cada fase (entrenamiento, test y validación) apoyando al proceso de decisión de la mejor RNA para el modelo de predicción.

e.1.4. ENTRENAMIENTO DE LA RNA

Para cada estructura de red evaluada se realizó un número determinado de entrenamientos, se seleccionó a los 5 mejores resultados del proceso de entrenamiento, escogiendo de ellos el que cumpla con las métricas de diseño establecidas a continuación

- Índice de regresión (R) entre 0,9 y 0,95 como mínimo.
- El menor error cuadrático medio (MSE) posible.
- La correlación del error por debajo del intervalo de confianza (95%).
- Histograma de error en forma de campana lo más cercano a cero.

Para la predicción de la potencia se probó alrededor de 20 tipos de RNA, variando dos parámetros en la arquitectura de las mismas:

- Número de capas ocultas.
- Número de retardos en las capas ocultas.

La arquitectura por default utilizada se muestra en la figura 21.

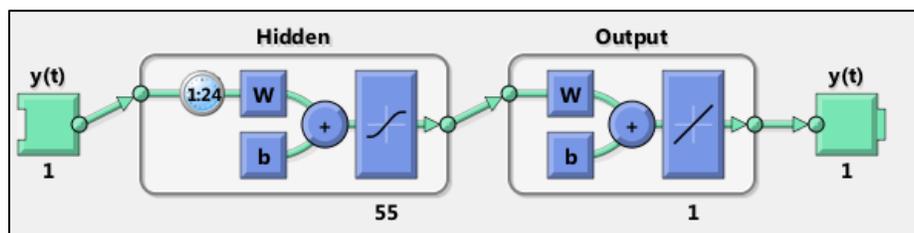


Figura 22.- arquitectura de RNA No-Lineal autoregresiva.

El sistema es de una entrada una salida con un número determinado de capas ocultas o “Hidden Layers”, una capa de salida y las funciones de activación son de tipo “Tansig” para las capas ocultas y “Purelin” para la capa de salida. La estructura de la red no será alterada, se variará los parámetros que ya se mencionó anteriormente.

Previamente se divide los datos para el proceso de entrenamiento, en la tabla 3 se muestra el total de datos a utilizar y su asignación dentro de la fase de entrenamiento.

Tabla 3. Distribución de los datos para el diseño de la RNA.

Distribución de los datos	
Total de datos	7920
Entrenamiento	5544
Validación	1188
Test	1188

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

e.1.4.1. RNA DE PREDICCIÓN DE POTENCIA ACTIVA

Para cada una de las redes probadas se realizó alrededor de 10 procesos de entrenamiento hasta aproximar a los criterios de las métricas planteadas para el diseño.

Cabe señalar que adicional a los criterios también se plantea que la red sea lo más simple posible, para lograr que su entrenamiento se logre en el menor tiempo posible optimizando los recursos computacionales.

A continuación se muestra los 5 mejores entrenamientos de los 10 realizados a las 20 arquitecturas de RNA creadas. Al final se resumirá los resultados obtenidos para realizar el proceso de decisión de la mejor arquitectura de RNA que será implementada como modelo de predicción.

e.1.4.1.1. ENTRENAMIENTO DE RNA POTENCIA ACTIVA 1

La arquitectura de la RNA se muestra en la figura 22.

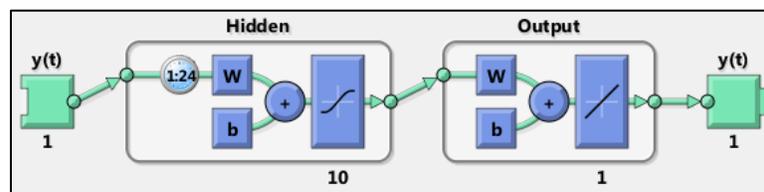


Figura 23.- RNA de potencia activa 1.

La auto-correlación del error para el entrenamiento se muestra en la figura 23.

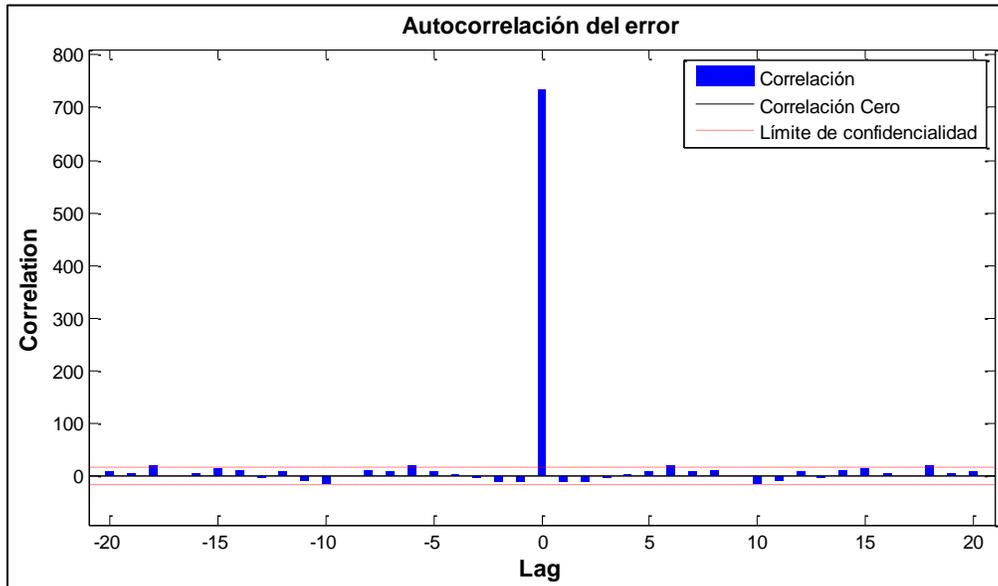


Figura 24.- Auto-correlación del error de la RNA de potencia activa 1.

La auto-correlación del error muestra que el modelo es consistente, ya que los errores se encuentran dentro de los intervalos de confianza y el valor de auto-correlación más alto se encuentra en cero.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 24 y 25 respectivamente.

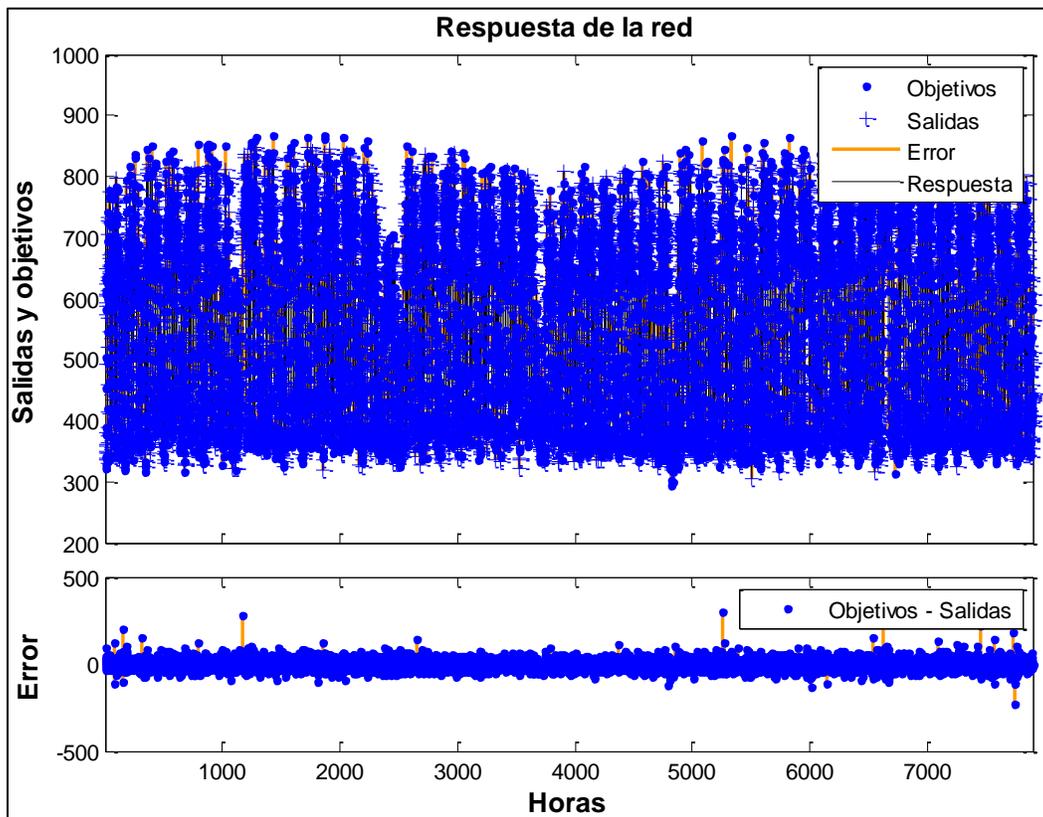


Figura 25.- Respuesta temporal y error de la RNA de potencia activa 1.

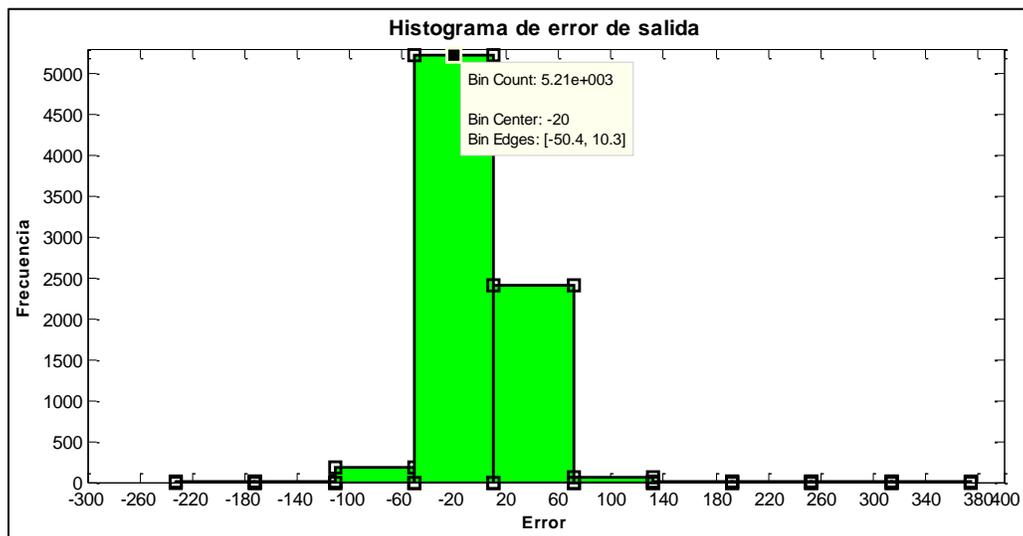


Figura 26.- Histograma del error de salida de la RNA de potencia activa 1.

El valor central del error es de -20 dentro de un intervalo de -50,4 a 10,3 con una frecuencia de repetición alta de 5210, se puede apreciar que el segundo valor de error considerable está en un intervalo entre 20 y 60 con una repetición de aproximadamente 2500, los demás valores de error están por debajo de 500 repeticiones.

e.1.4.1.2. ENTRENAMIENTO DE RNA POTENCIA ACTIVA 2

La arquitectura de la RNA se muestra en la figura 26.

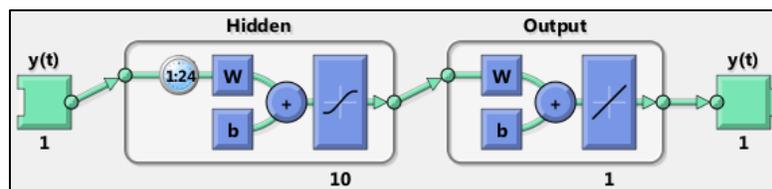


Figura 27.- RNA de potencia activa 2.

La auto-correlación del error para el entrenamiento se muestra en la figura 27.

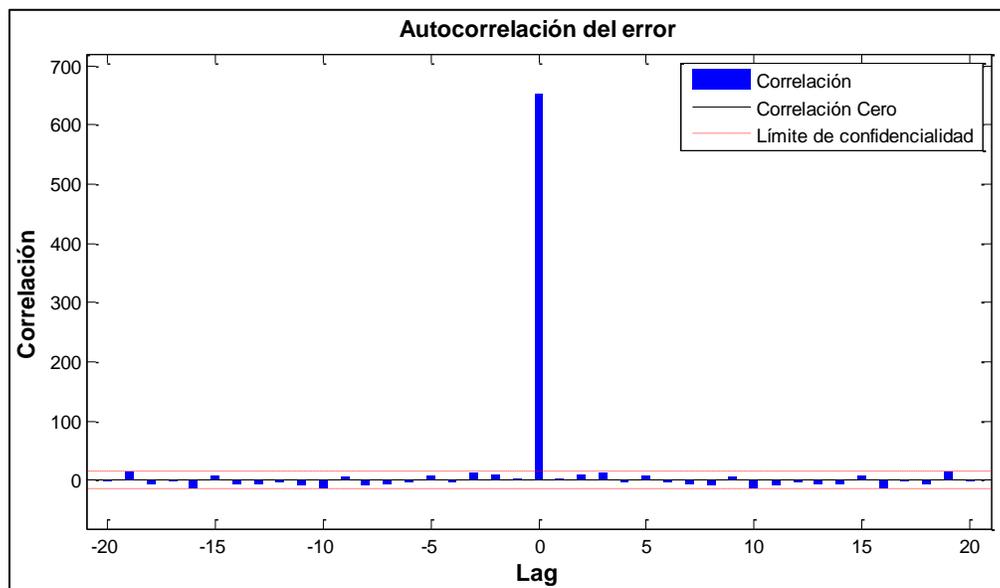


Figura 28.- Auto-correlación del error de la RNA de potencia activa 2.

La auto-correlación del error muestra que el modelo es consistente, ya que los errores se encuentran dentro de los intervalos de confianza y el valor de auto-correlación más alto se encuentra en cero.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 28 y 29 respectivamente.

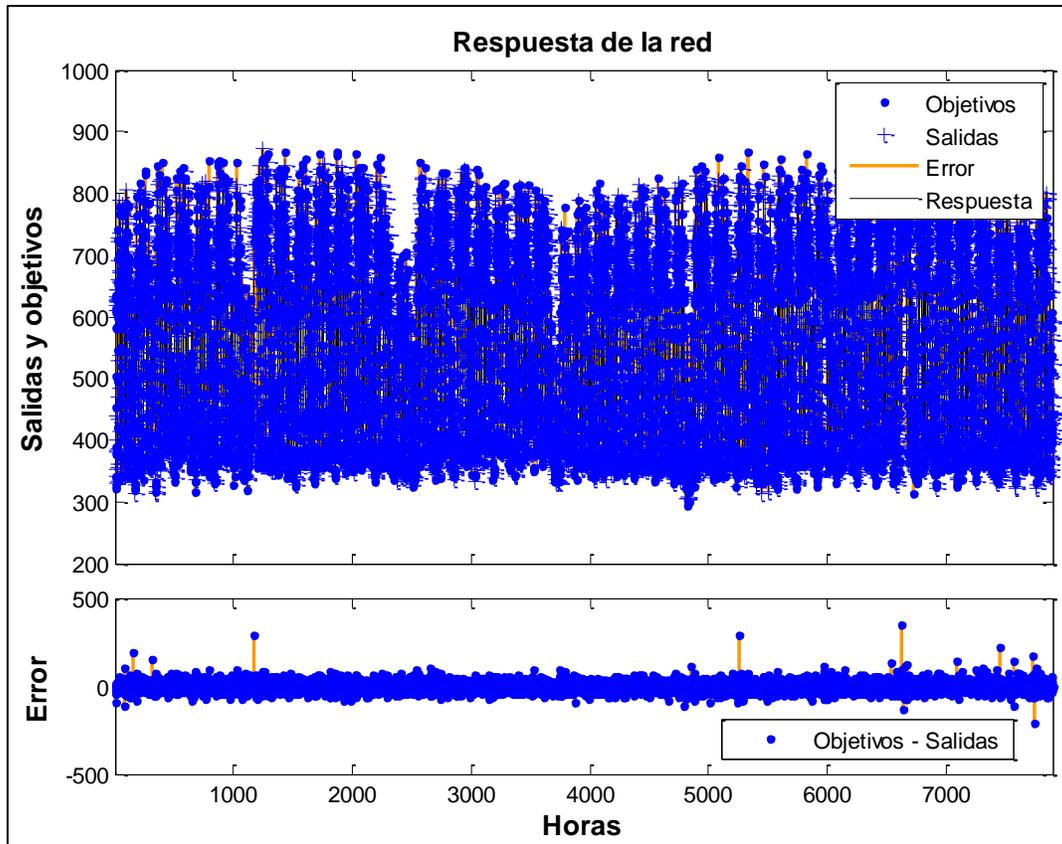


Figura 29.- Respuesta temporal y error de la RNA de potencia activa 2.

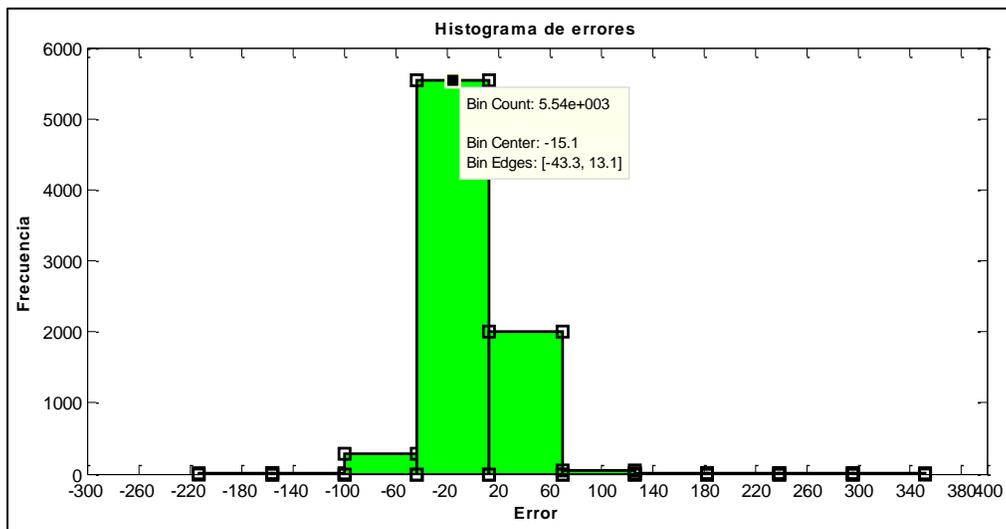


Figura 30.- Histograma del error de salida de la RNA de potencia activa 2.

El valor central del error es de -15,1 dentro de un intervalo de -43,3 a 13,1 con una frecuencia de repetición alta de 5540, se puede apreciar que el segundo valor de error considerable está en un intervalo entre 20 y 60 con una repetición de aproximadamente 2000, los demás valores de error están por debajo de 500 repeticiones.

e.1.4.1.3. ENTRENAMIENTO DE RNA POTENCIA ACTIVA 3

La arquitectura de la RNA se muestra en la figura 30.

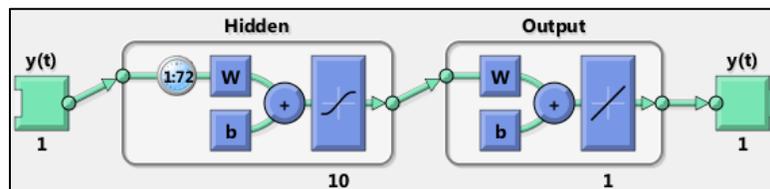


Figura 31.- RNA de potencia activa 3.

La auto-correlación del error para el entrenamiento se muestra en la figura 31.

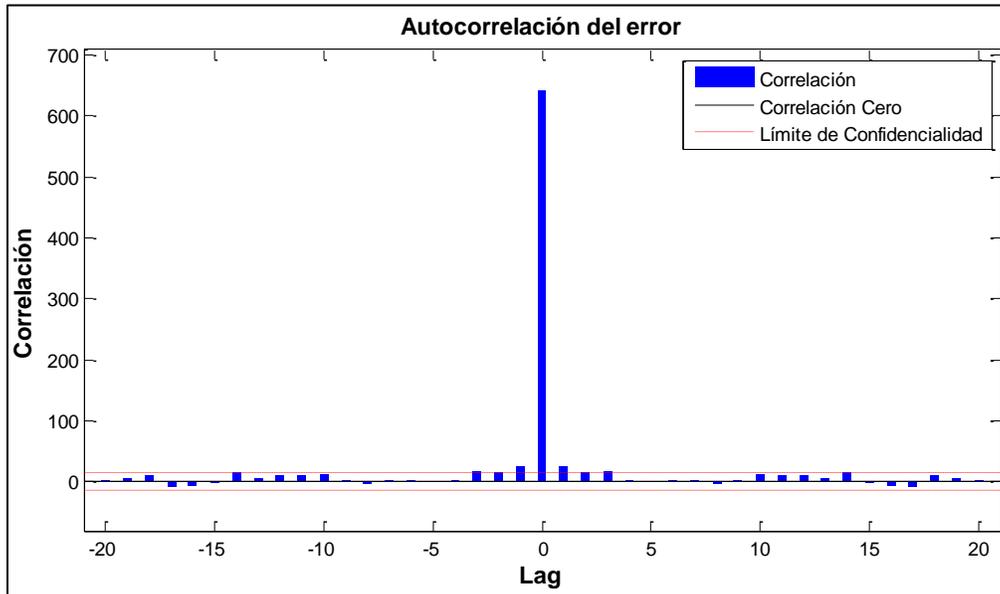


Figura 32.- Auto-correlación del error de la RNA de potencia activa 3.

La auto-correlación del error muestra que el modelo es consistente, aunque los errores superan el intervalo de confianza lo hacen cerca de cero y el valor de correlación más alto está en cero.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 32 y 33 respectivamente.

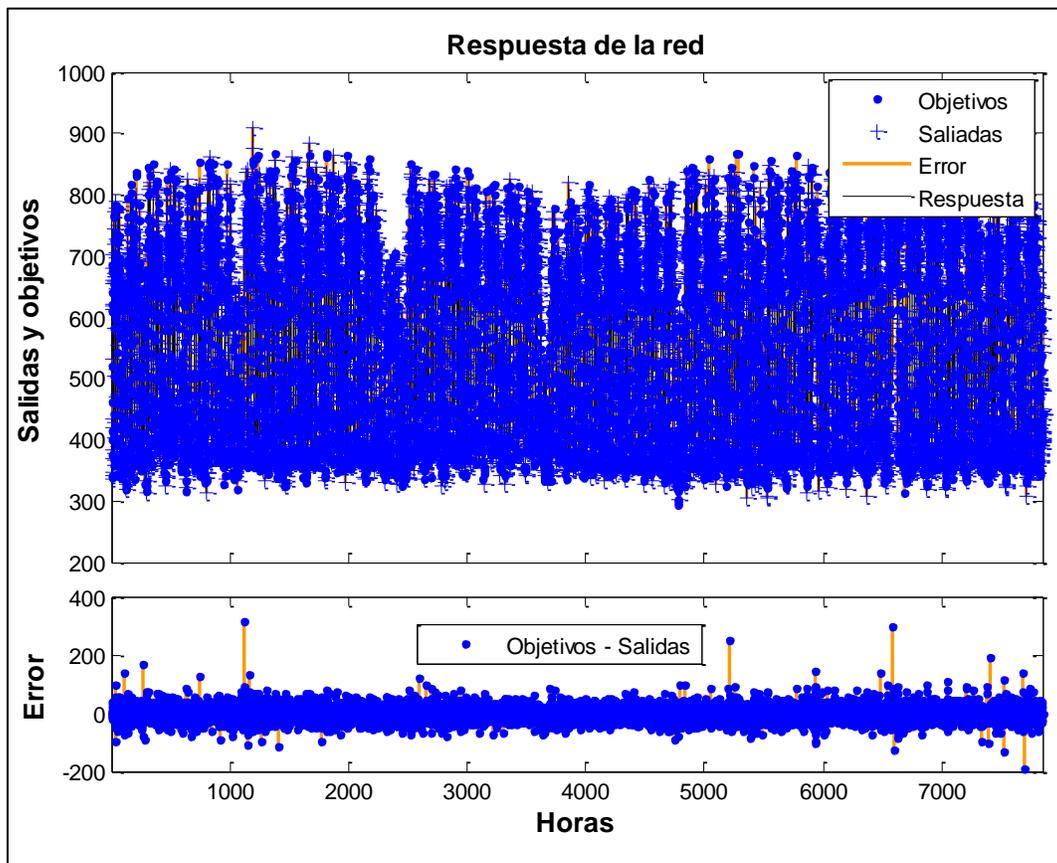


Figura 33.- Respuesta temporal y error de la RNA de potencia activa 3.

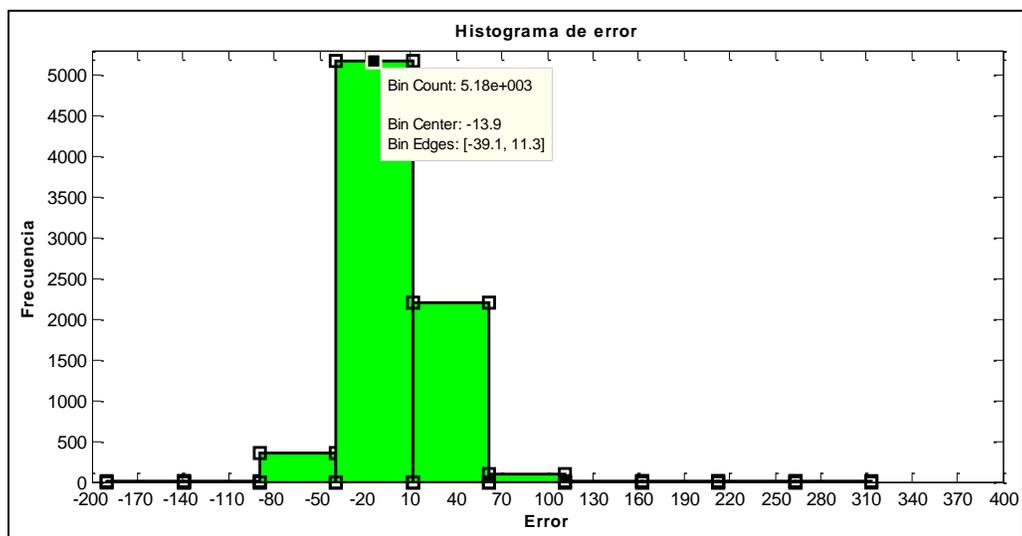


Figura 34.- Histograma del error de salida de la RNA de potencia activa 3.

El valor central del error es de -13,9 dentro de un intervalo de -39,1 a 11,3 con una frecuencia de repetición alta de 5180, se puede apreciar que el segundo valor de error considerable está en un intervalo entre 10 y 70 con una repetición de aproximadamente 2000, los demás valores de error están por debajo de 500 repeticiones.

e.1.4.1.4. ENTRENAMIENTO DE RNA POTENCIA ACTIVA 4

La arquitectura de la RNA se muestra en la figura 34.

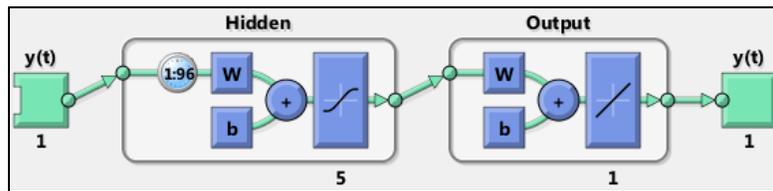


Figura 35.- RNA de potencia activa 4.

La auto-correlación del error para el entrenamiento se muestra en la figura 35.

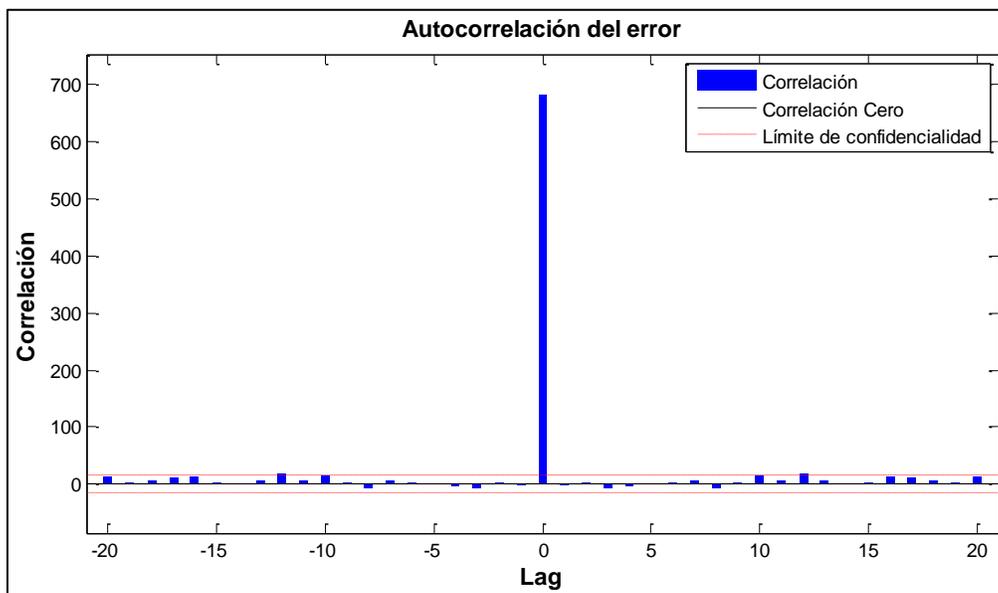


Figura 36.- Auto-correlación del error de la RNA de potencia activa 4.

La auto-correlación del error muestra que el modelo es consistente, ya que los errores se encuentran dentro de los intervalos de confianza y el valor de auto-correlación más alto se encuentra en cero.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 36 y 37 respectivamente.

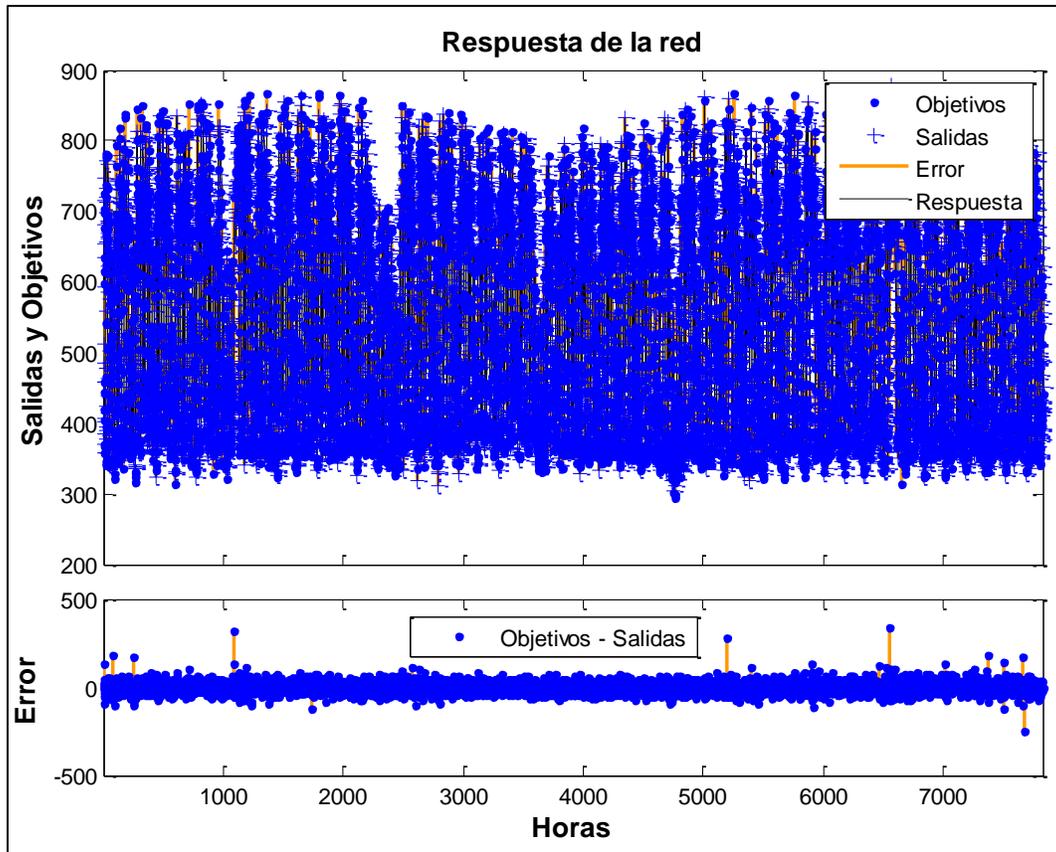


Figura 37.- Respuesta temporal y error de la RNA de potencia activa 4.

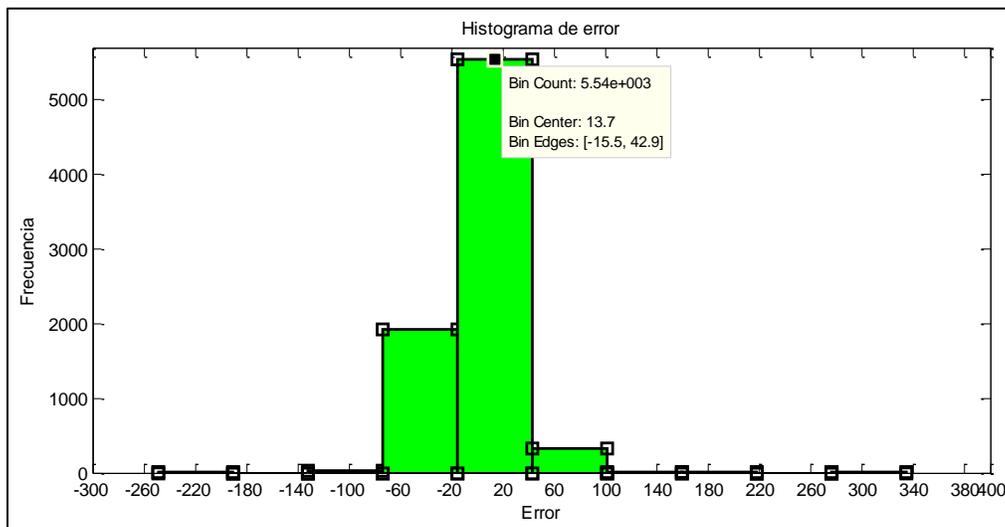


Figura 38.- Histograma del error de salida de la RNA de potencia activa 4.

El valor central del error es de 13,7 dentro de un intervalo de 15,5 a 42,9 con una frecuencia de repetición alta de 5540, se puede apreciar que el segundo valor de error considerable está en un intervalo entre -60 y -20 con una repetición de aproximadamente 2000, los demás valores de error están por debajo de 500 repeticiones.

e.1.4.1.5. ENTRENAMIENTO DE RNA POTENCIA ACTIVA 5

La arquitectura de la RNA se muestra en la figura 38.

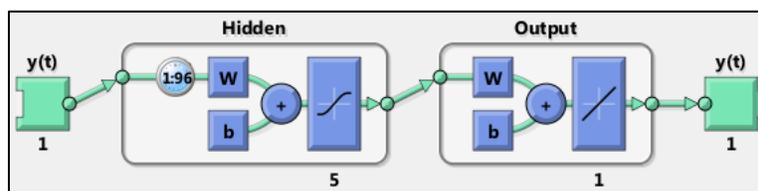


Figura 39.- RNA de potencia activa 5.

La auto-correlación del error para el entrenamiento se muestra en la figura 39.

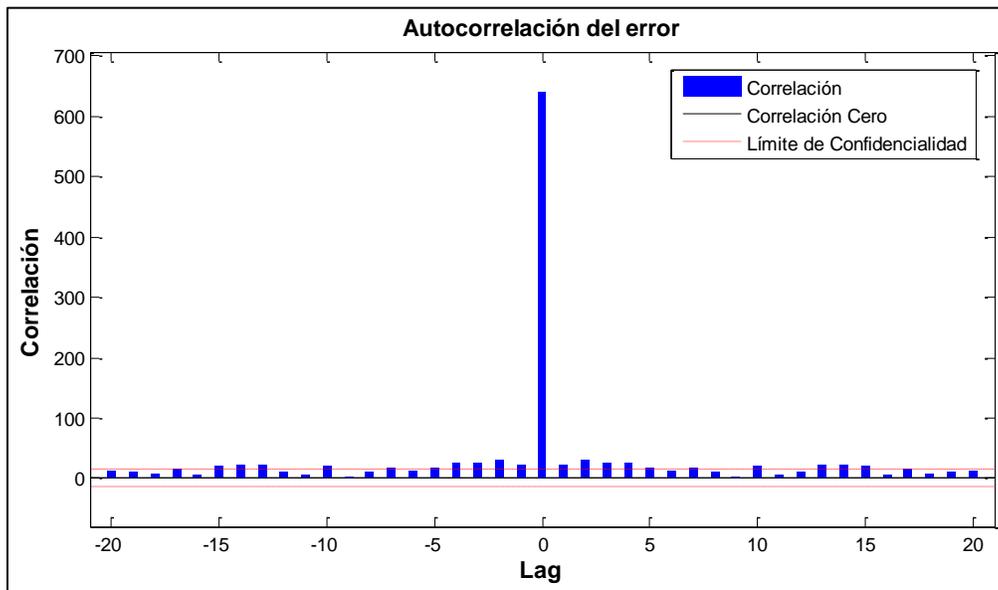


Figura 40.- Respuesta temporal y error de la RNA de potencia activa 5.

En este caso la auto-correlación del error muestra algunos segmentos fuera del intervalo de confianza aplicado para la RNA, lo cual implica la necesidad de aumentar los retardos en las capas ocultas.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 40 y 41 respectivamente.

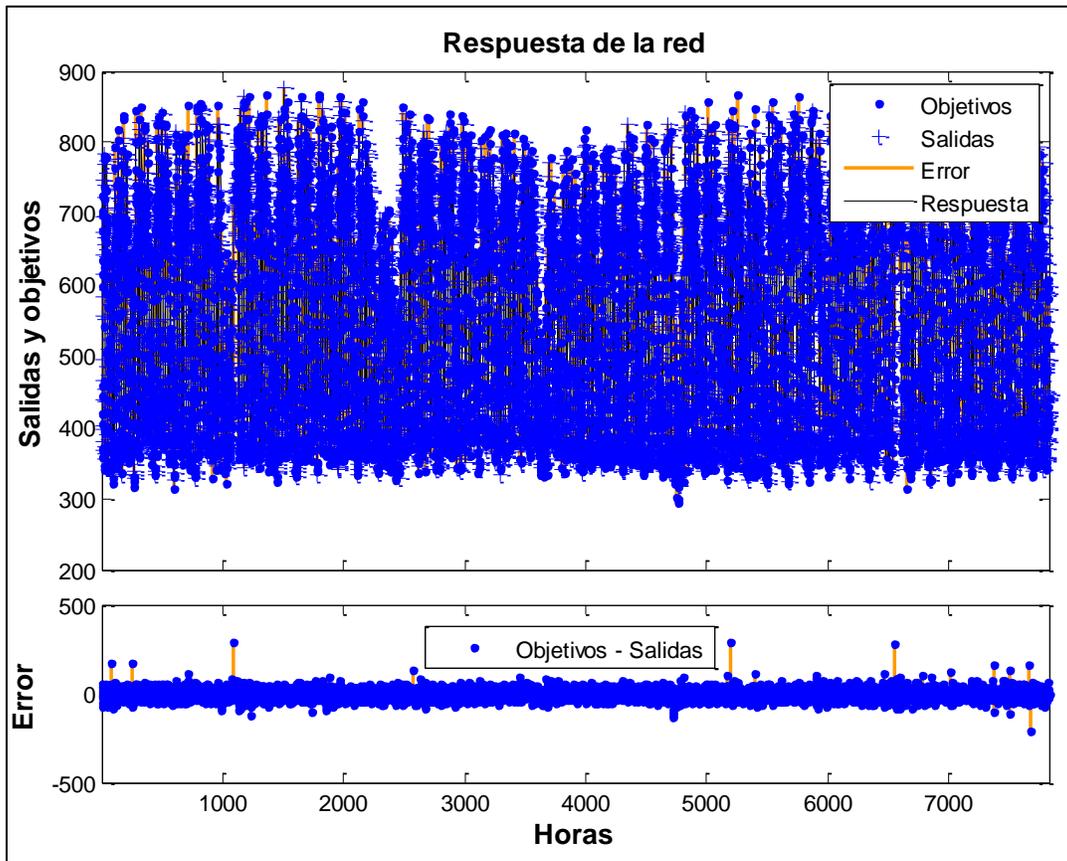


Figura 41.- Respuesta temporal y error de la RNA de potencia activa 5.

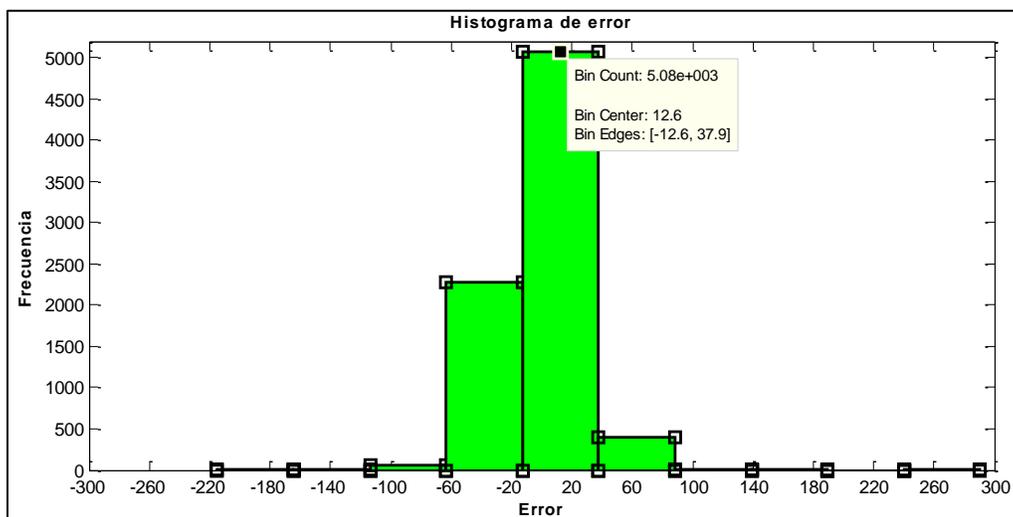


Figura 42.- Histograma del error de salida de la RNA de potencia activa 5.

El valor central del error es de 12,6 dentro de un intervalo de -12,6 a 37,9 con una frecuencia de repetición alta de 5080, se puede apreciar que el segundo valor de error considerable está en un intervalo entre -60 y -20 con una repetición de aproximadamente 2500, los demás valores de error están por debajo de 500 repeticiones.

El resumen de resultados del entrenamiento de la RNA para la predicción de la potencia activa se muestra en la tabla 4, la tabla 5 y la tabla 6:

Tabla 4. Resumen de métricas de entrenamientos de potencia activa RNA.

Parámetros											
Red	Estructura de red					Entrenamiento		Validación		Test	
	N	Re	Ent	Sal	Cap	Performance	Regresión	Performance	Regresión	Performance	Regresión
1	10	24	1	1	2	683,39	0,984	917,69	0,9788	794,93	0,982
2	10	24	1	1	2	640	0,985	675,05	0,984	688,26	0,984
3	10	72	1	1	2	553,21	0,987	799,56	0,98	901,62	0,979
4	5	96	1	1	2	625,9	0,985	722,66	0,982	915,28	0,978
5	5	96	1	1	2	608,34	0,986	686,78	0,983	751,24	0,983

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

Tabla 5. Nomenclatura de la tabla de resumen de entrenamiento.

Abreviatura	Parámetro
N	Neuronas
Re	Retardos
Ent	Entradas
Sal	Salidas
Cap	Capas

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

Tabla 6. Resumen de métricas gráficas de entrenamiento de RNA de potencia activa.

Parámetros gráficos			
Red	Correlación del error	Histograma	
		Error	Repetición
1	Cumple	-20	5210
2	Cumple	-15,1	5540
3	No cumple	-13,9	5180
4	Cumple	13,7	5540
5	No cumple	12,6	5080

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

e.1.4.2. RNA DE PREDICCIÓN DE POTENCIA REACTIVA

Para cada una de las redes probadas se realizó alrededor de 10 procesos de entrenamiento hasta aproximar a los criterios de las métricas planteadas para el diseño.

Cabe señalar que adicional a los criterios también se plantea que la red sea lo más simple posible, para lograr que su entrenamiento se logre en el menor tiempo posible optimizando así mismo los recursos computacionales.

Para el caso de la potencia reactiva, los parámetros de diseño son similares asó como la cantidad de datos que van a ser ingresados en la RNA.

e.1.4.2.1. ENTRENAMIENTO DE RNA POTENCIA REACTIVA 1

La arquitectura de la RNA se muestra en la figura 42.

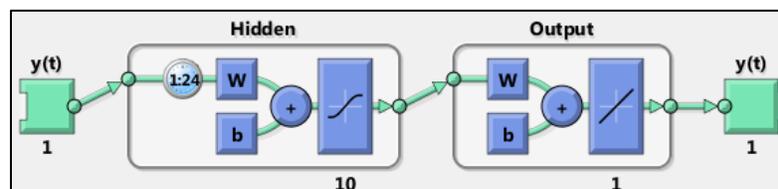


Figura 43.- RNA de potencia reactiva 1.

La auto-correlación del error para el entrenamiento se muestra en la figura 43.

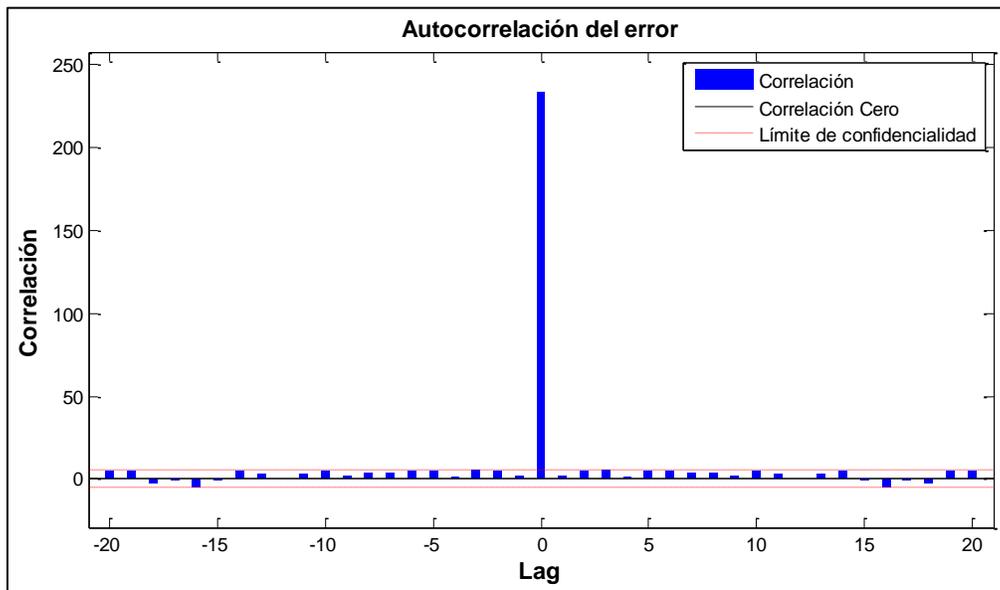


Figura 44.- Auto-correlación del error de la RNA de potencia reactiva 1.

La auto-correlación del error muestra que el modelo es consistente, ya que los errores se encuentran dentro de los intervalos de confianza y el valor de auto-correlación más alto se encuentra en cero.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 44 y 45 respectivamente.

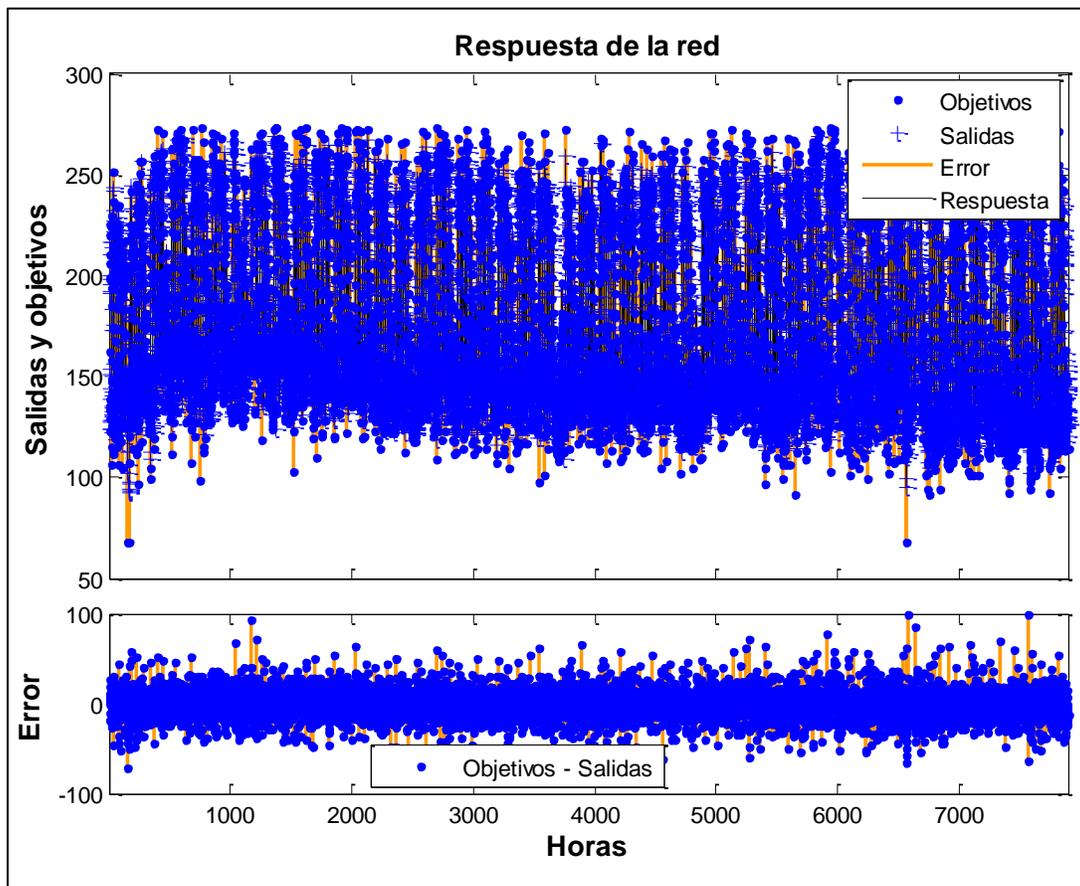


Figura 45.- Respuesta temporal y error de la RNA de potencia reactiva 1.

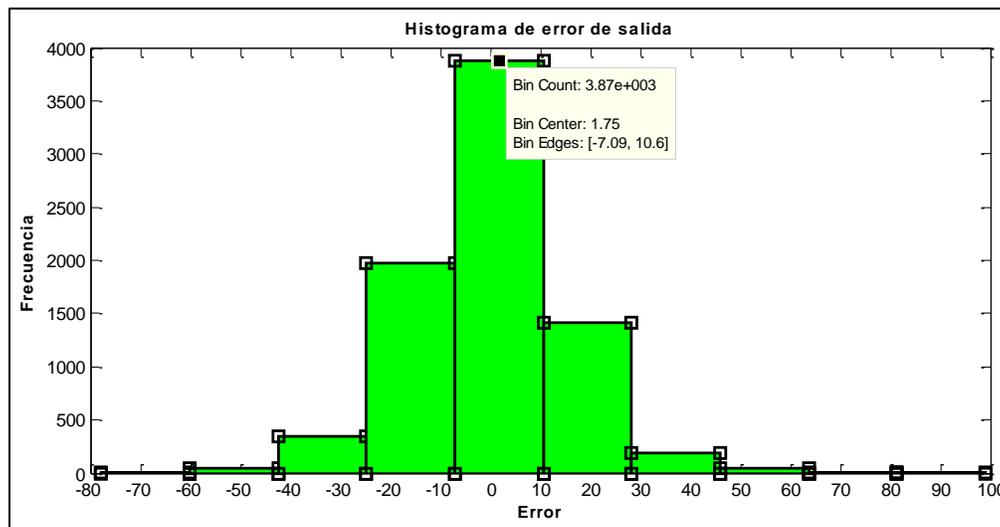


Figura 46.- Histograma del error de salida de la RNA de potencia reactiva 1.

El valor central del error es de 1,75 dentro de un intervalo de -7,09 a 10,6 con una frecuencia de repetición alta de 3870, se puede apreciar que el segundo valor de error considerable está en un intervalo entre -10 y -20 con una repetición de aproximadamente 2000, los demás valores de error están por debajo de 1500 repeticiones.

e.1.4.2.2 ENTRENAMIENTO DE RNA POTENCIA REACTIVA 2

La arquitectura de la RNA se muestra en la figura 46.

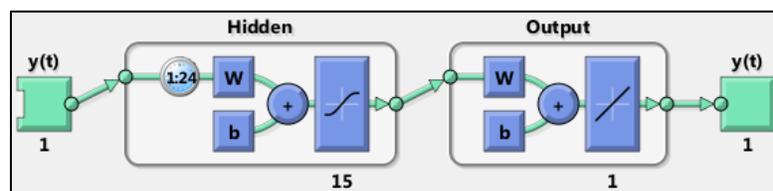


Figura 47.- RNA de potencia reactiva 2.

La auto-correlación del error para el entrenamiento se muestra en la figura 47.

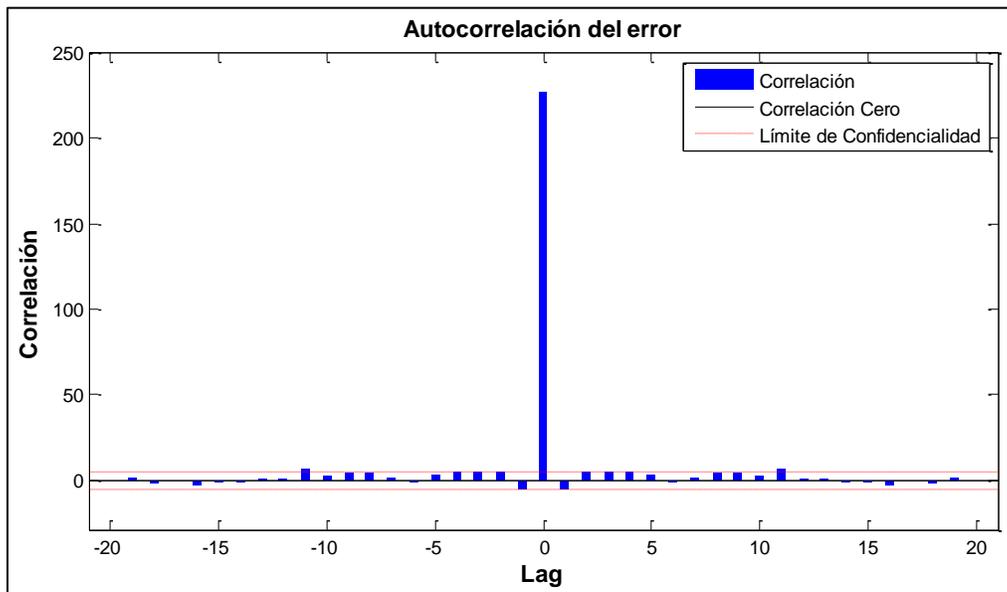


Figura 48.- Respuesta temporal y error de la RNA de potencia reactiva 2.

La auto-correlación del error muestra que el modelo es consistente, ya que los errores se encuentran dentro de los intervalos de confianza y el valor de auto-correlación más alto se encuentra en cero.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 48 y 49 respectivamente.

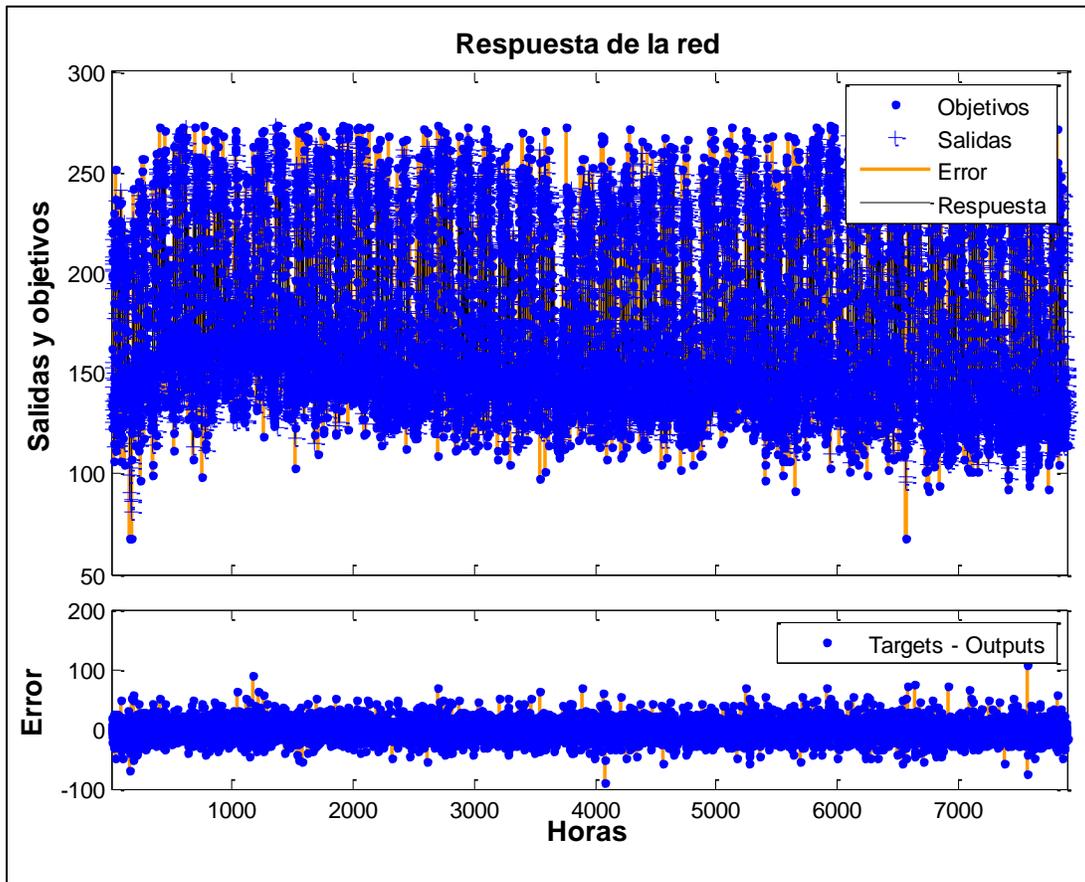


Figura 49.- Respuesta temporal y error de la RNA de potencia reactiva 2.

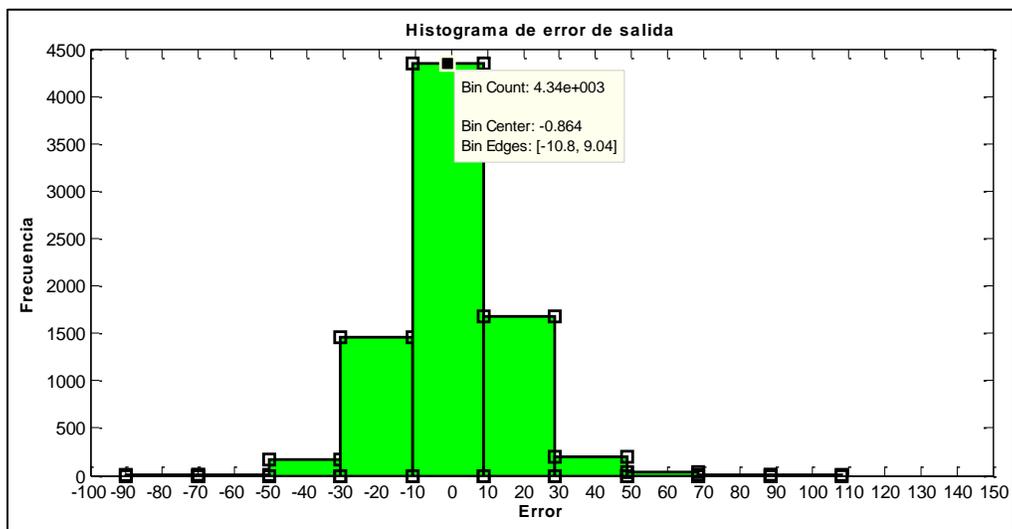


Figura 50.- Histograma del error de salida de la RNA de potencia reactiva 2.

El valor central del error es de $-0,864$ dentro de un intervalo de $-10,8$ a $9,04$ con una frecuencia de repetición alta de 4340 , se puede apreciar que el segundo valor de error considerable está en un intervalo entre 10 y 30 con una repetición de aproximadamente 1500 , los demás valores de error están por debajo de 1500 repeticiones.

e.1.4.2.3. ENTRENAMIENTO DE RNA POTENCIA REACTIVA 3

La arquitectura de la RNA se muestra en la figura 50.

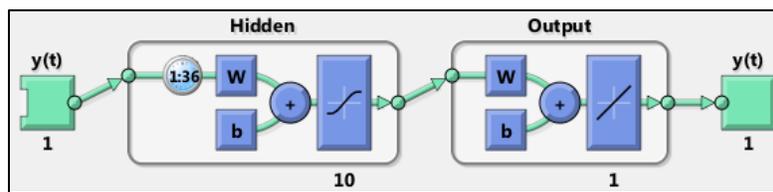


Figura 51.- RNA de potencia reactiva 3.

La auto-correlación del error para el entrenamiento se muestra en la figura 51.

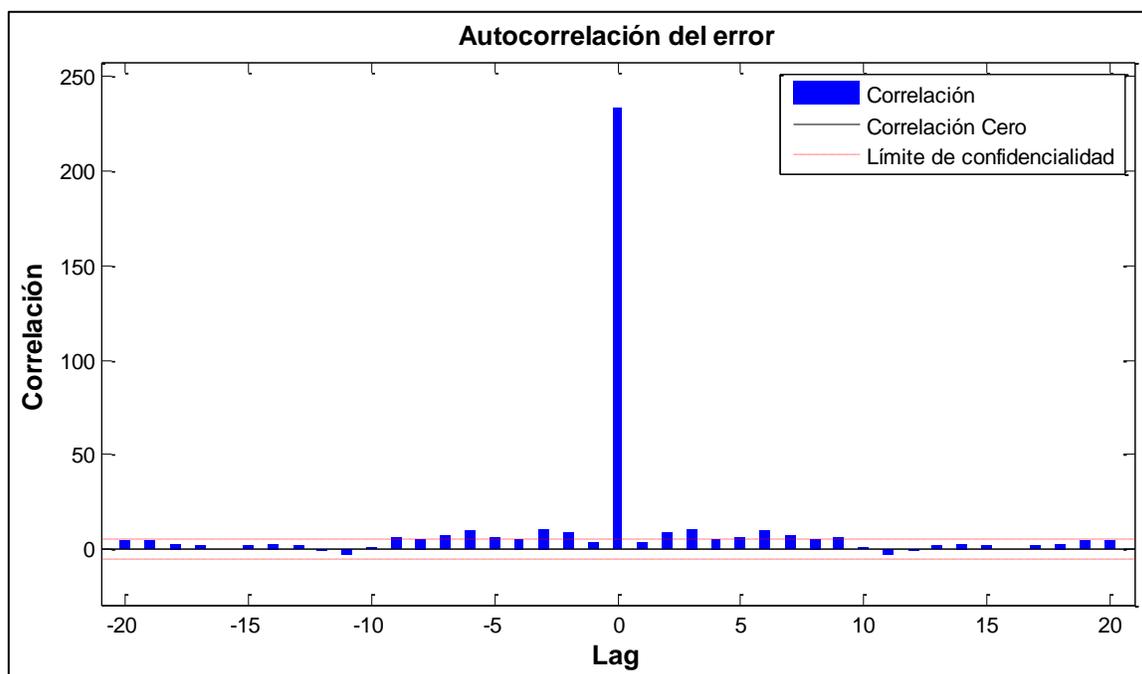


Figura 52.- Auto-correlación del error de la RNA de potencia reactiva 3.

La auto-correlación del error muestra que los errores superan los límites de los intervalos de confianza cerca del valor central (cero), cuya magnitud no es muy elevada, esto indica la necesidad de aumentar el número de retardos.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 52 y 53 respectivamente.

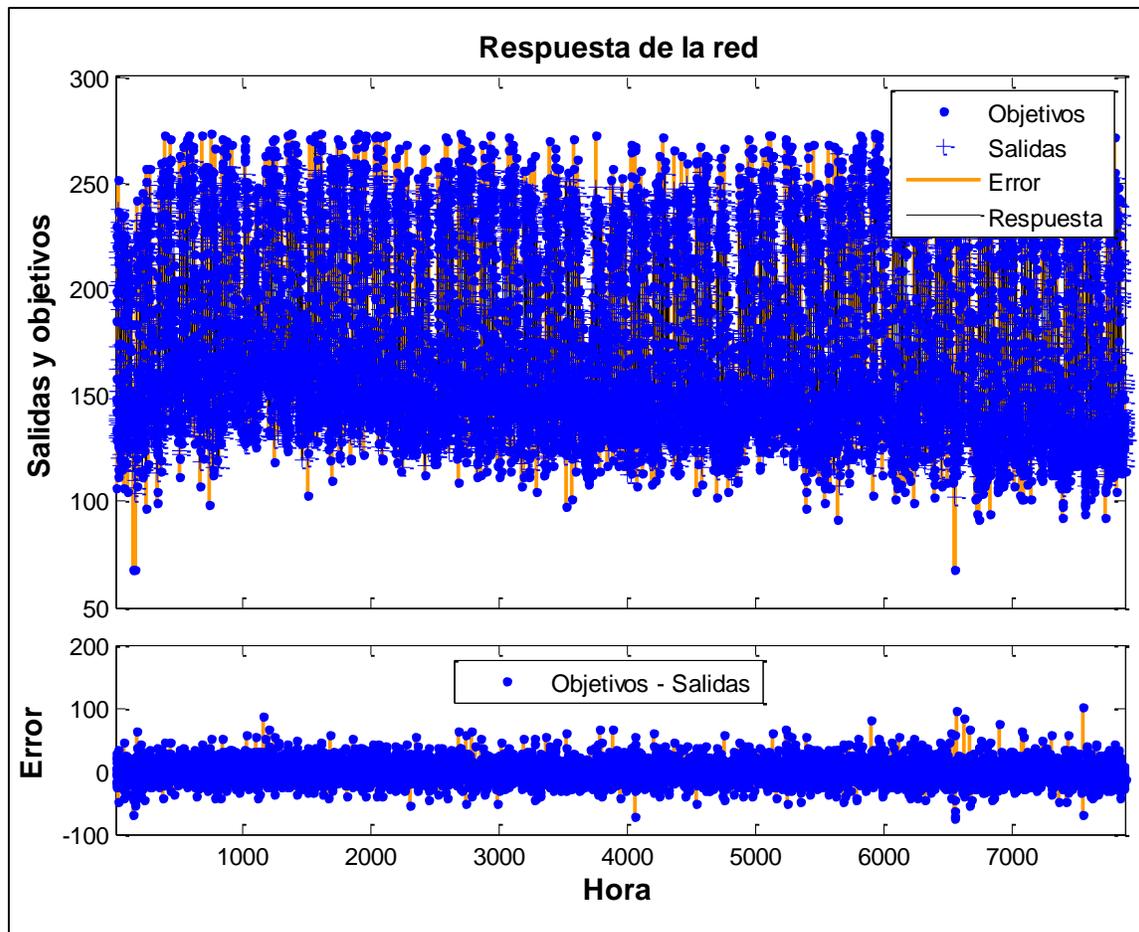


Figura 53.- Respuesta temporal y error de la RNA de potencia reactiva 3.

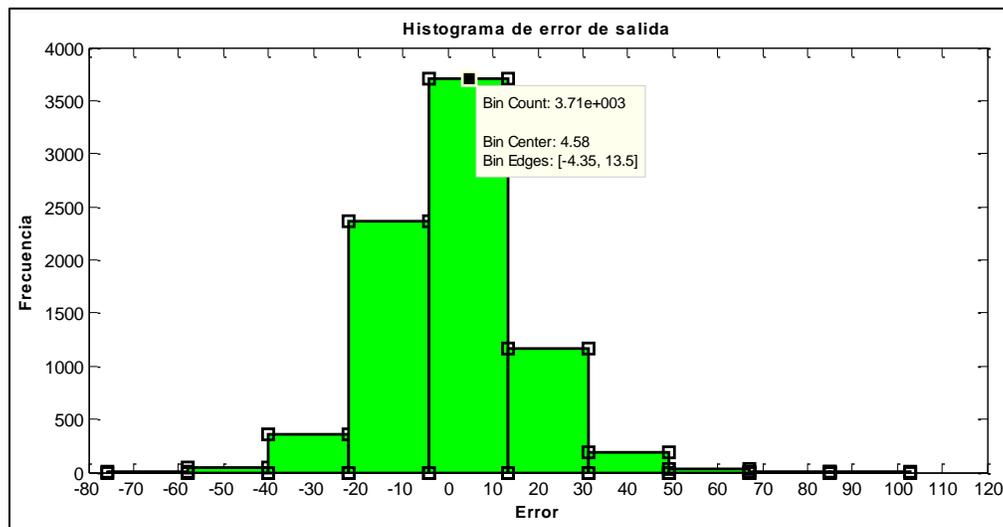


Figura 54.- Histograma del error de salida de la RNA de potencia reactiva 3.

El valor central del error es de 4,58 dentro de un intervalo de -4,35 a 13,5 con una frecuencia de repetición alta de 3710, se puede apreciar que el segundo valor de error considerable está en un intervalo entre -10 y -20 con una repetición de aproximadamente 2500, los demás valores de error están por debajo de 1500 repeticiones.

e.1.4.2.4. ENTRENAMIENTO DE RNA POTENCIA REACTIVA 4

La arquitectura de la RNA se muestra en la figura 54.

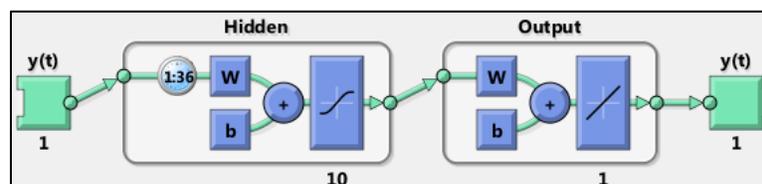


Figura 55.- RNA de potencia reactiva 4.

La auto-correlación del error para el entrenamiento se muestra en la figura 55.

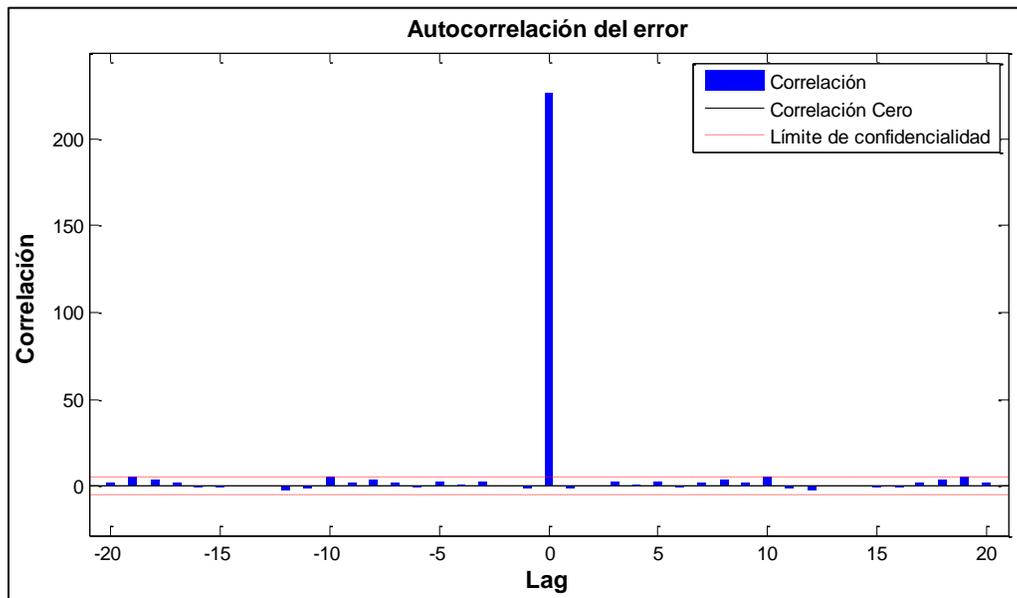


Figura 56.- Auto-correlación del error de la RNA de potencia reactiva 4.

La auto-correlación del error muestra que el modelo es consistente, ya que los errores se encuentran dentro de los intervalos de confianza y el valor de auto-correlación más alto se encuentra en cero.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 56 y 57 respectivamente.

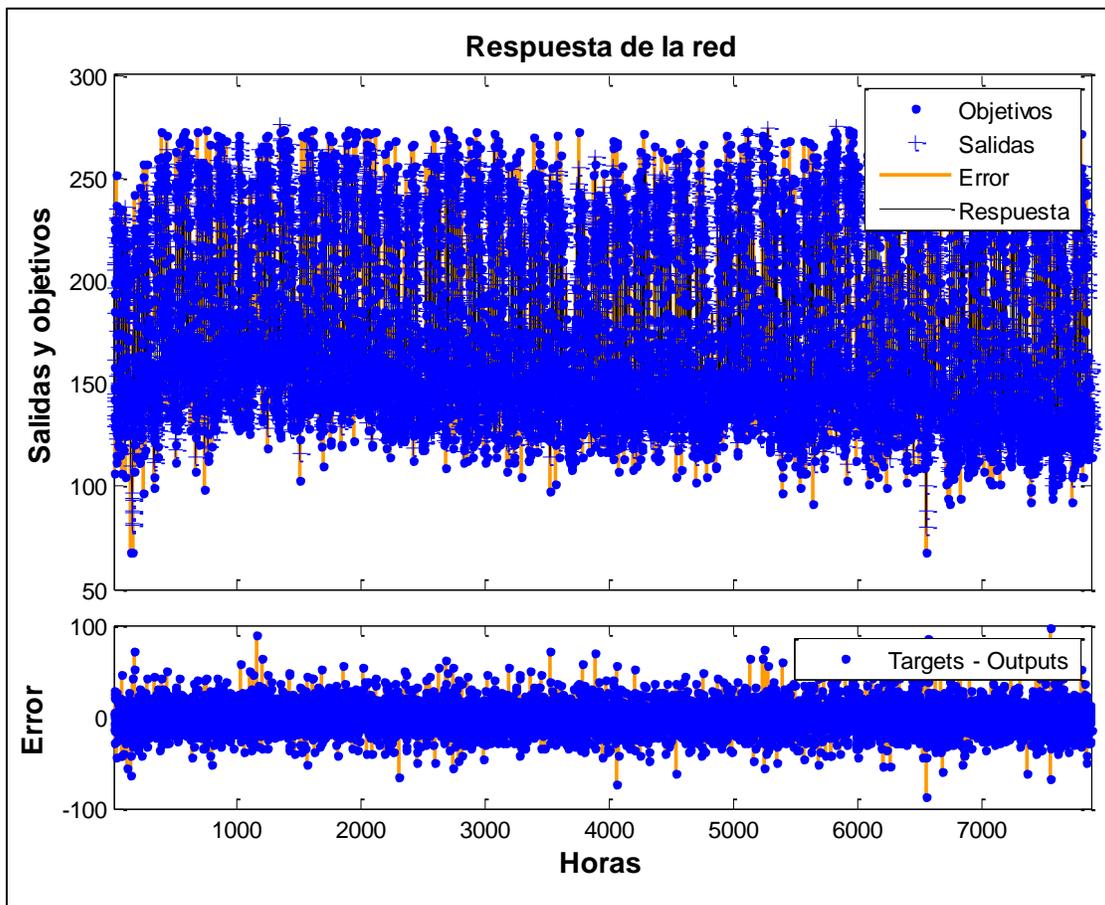


Figura 57.- Respuesta temporal y error de la RNA de potencia reactiva 4.

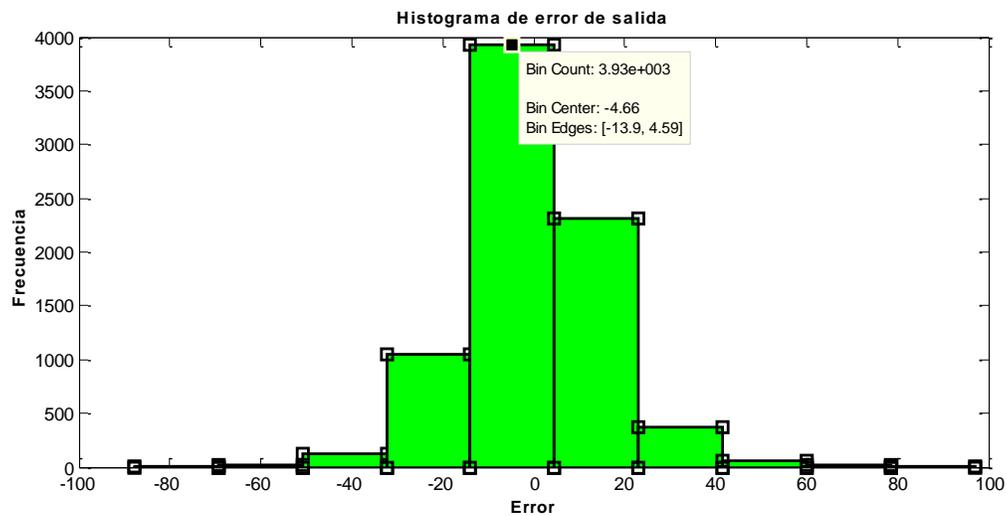


Figura 58.- Histograma del error de salida de la RNA de potencia reactiva 4.

El valor central del error es de -4,66 dentro de un intervalo de -13,9 a 4,59 con una frecuencia de repetición alta de 3930, se puede apreciar que el segundo valor de error considerable está en un intervalo entre 0 y 20 con una repetición de aproximadamente 2500, los demás valores de error están por debajo de 1500 repeticiones.

e.1.4.2.5. ENTRENAMIENTO DE RNA POTENCIA REACTIVA 5

La arquitectura de la RNA se muestra en la figura 58.

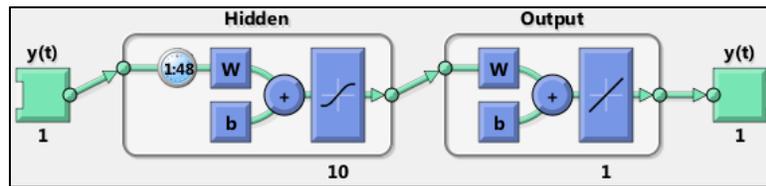


Figura 59.- RNA de potencia reactiva 5.

La auto-correlación del error para el entrenamiento se muestra en la figura 59.

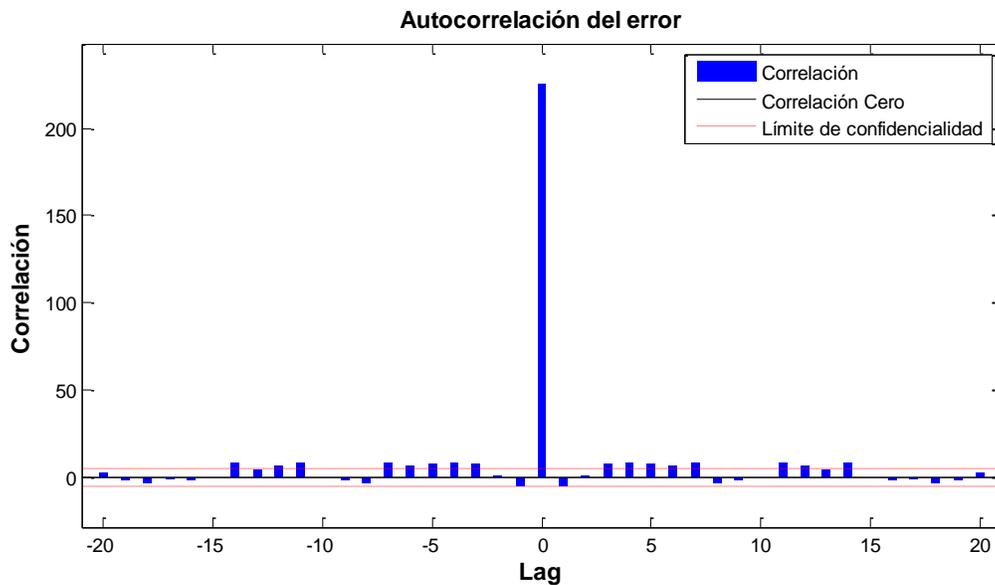


Figura 60.- Auto-correlación del error de la RNA de potencia reactiva 5.

La auto-correlación del error muestra que los errores superan los límites de los intervalos de confianza cerca del valor central (cero), cuya magnitud no es muy elevada, esto indica la necesidad de aumentar el número de retardos.

La respuesta temporal resultante del proceso de validación y testeo de la red entrenada y los errores resultantes se muestran en las figuras 60 y 61 respectivamente.

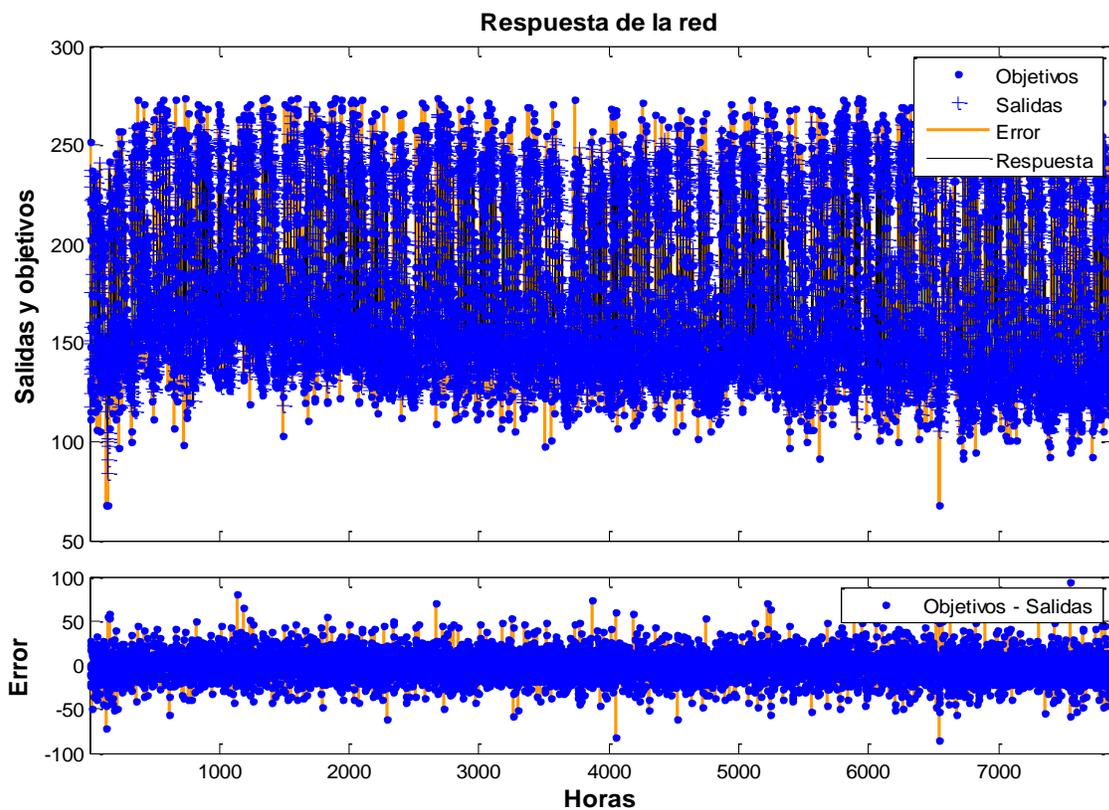


Figura 61.- Respuesta temporal y error de la RNA de potencia reactiva 1.

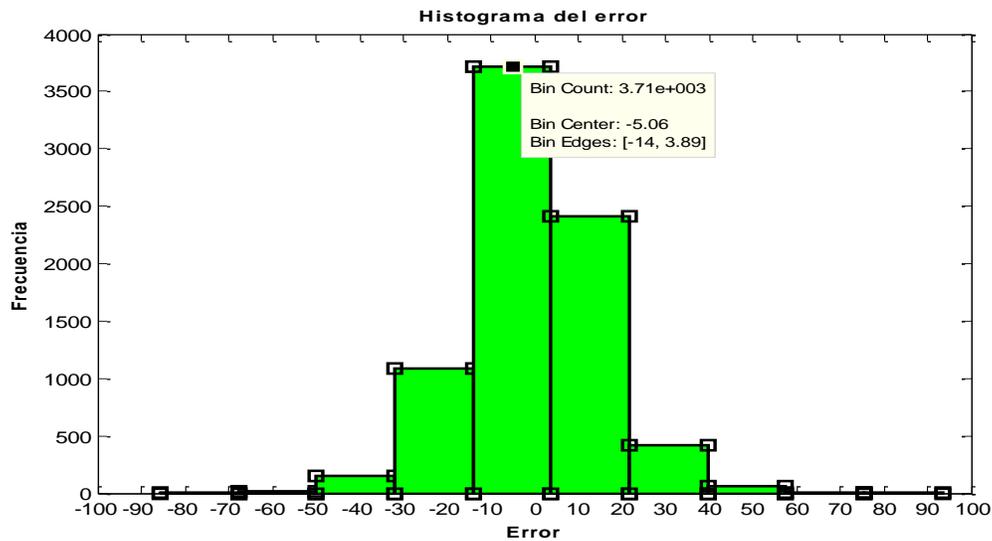


Figura 62.- Histograma del error de salida de la RNA de potencia reactiva 1.

El valor central del error es de -5,06 dentro de un intervalo de -14 a 3,89 con una frecuencia de repetición alta de 3700, se puede apreciar que el segundo valor de error considerable está en un intervalo entre 10 y 20 con una repetición de aproximadamente 2500, los demás valores de error están por debajo de 1500 repeticiones.

El resumen de resultados del entrenamiento de la RNA para la predicción de la potencia reactiva se muestra en la tabla 7, la tabla 5 y la tabla 8:

Tabla 7. Resumen de métricas de entrenamientos de RNA de potencia reactiva.

Parámetros											
Red	Estructura de red					Entrenamiento		Validación		Test	
	N	Re	Ent	Sal	Cap	Performance	Regresión	Performance	Regresión	Performance	Regresión
1	10	24	1	1	2	227,45	0,941	243,03	0,937	254,2	0,936
2	15	24	1	1	2	214,26	0,945	261,71	0,932	252,88	0,934
3	10	36	1	1	2	220,36	0,944	266,5	0,931	262,75	0,93
4	10	36	1	1	2	210,7	0,946	280,55	0,928	246,91	0,934
5	10	48	1	1	2	215,76	0,945	251,5	0,935	244,07	0,937

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

Tabla 8. Resumen de métricas gráficas de entrenamiento de RNA de potencia reactiva.

Parámetros gráficos			
Red	Correlación del error	Histograma	
		Error	Repetición
1	Cumple	1,75	3870
2	Cumple	-0,86	4340
3	No cumple	4,58	3710
4	Cumple	-4,66	3930
5	No cumple	-5,06	3700

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

e.1.4.3. SELECCIÓN DE LA RNA DE PREDICCIÓN

Antes de iniciar la creación y entrenamiento de la RNA se planteó un conjunto de métricas para determinar la consistencia del modelo de predicción, en la sección anterior se resumió los resultados de 5 entrenamientos de cada una de las potencias tanto la activa como la reactiva.

Para la potencia activa se escogió la RNA con los parámetros mostrados en la tabla 9 y la tabla 10.

Tabla 9. Métricas de RNA de potencia activa seleccionada.

Parámetros											
Red	Estructura de red					Entrenamiento		Validación		Test	
	N	Re	Ent	Sal	Cap	Performance	Regresión	Performance	Regresión	Performance	Regresión
2	10	24	1	1	2	640	0,985	675,05	0,984	688,26	0,984

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

Tabla 10. Métricas gráficas de RNA de potencia activa seleccionada.

Parámetros gráficos			
Red	Correlación del error	Histograma	
		Error	Repetición
2	Cumple	-15,1	5540

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

El resumen muestra que tanto en performance dentro de las etapas de entrenamiento, validación y testeo, la RNA a pesar de no tener el menor error, es la más estable en todas las etapas, lo que implica un coeficiente de regresión constante de 0,98 con lo que se cumple con la métrica planteada, la auto-correlación del error está dentro del intervalo de confianza, el error medio es de -15,1 con una alta repetición. En conclusión se ha cumplido con las métricas de diseño planteadas en un inicio.

Para la potencia reactiva se escogió la RNA con los parámetros mostrados en la tabla 11 y la tabla 12.

Tabla 11. Métricas de RNA de potencia reactiva seleccionada.

Parámetros											
Red	Estructura de red					Entrenamiento		Validación		Test	
	N	Re	Ent	Sal	Cap	Performance	Regresión	Performance	Regresión	Performance	Regresión
2	15	24	1	1	2	214,26	0,945	261,71	0,932	252,88	0,934

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

Tabla 12. Métricas gráficas de RNA de potencia reactiva seleccionada.

Parámetros gráficos			
Red	Correlación del error	Histograma	
		Error	Repetición
2	Cumple	-0,86	4340

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

El resumen muestra que tanto en performance dentro de las etapas de entrenamiento, validación y testeo la RNA a pesar de no tener el menor error, es la más estable en todas las etapas, lo cual implica un coeficiente de regresión constante de 0,94 con lo que se cumple con la métrica planteada, la auto-correlación del error está dentro del intervalo de confianza, el error medio es de -0,86 con una alta repetición. En conclusión se cumple con las métricas de diseño planteadas en un inicio.

f. RESULTADOS

Para entrenar la RNA se ingresaron 7920 muestras de los mayores consumos de potencia activa y reactiva por hora del alimentador de la sub estación “San Cayetano” que abastece de energía eléctrica al sector el Calvario de la ciudad de Loja.

Dentro del proceso de creación de la RNA se dividió a los datos en grupos, de entrenamiento, validación y test. La red fue diseñada para predecir la curva de demanda de un día dándole como parámetros de entrada en base a las arquitecturas seleccionadas, los 24 estados de entrada, es decir, la curva de demanda del día anterior al pronóstico y la curva de demanda del día que se desea pronosticar pero de una semana anterior.

Tomando como base los datos dentro del grupo de validación y testeo de la RNA, se obtuvo resultados con los cuales se evaluará el comportamiento de la red.

f.1. EVALUACIÓN Y PREDICCIÓN DE POTENCIA ACTIVA MEDIANTE RNA

f.1.1. EVALUACIÓN DE LA RNA DE PREDICCIÓN DE LA DEMANDA DE POTENCIA ACTIVA

Para evaluar la RNA obtenida del proceso de selección realizado anteriormente, se valorará los resultados mediante el uso de la función *evalnar.m* (ver Anexo 8).

Como estados iniciales se introdujo la curva de demanda de potencia activa perteneciente al día martes 7 de agosto del 2012, como entradas se introdujo la curva de demanda del día miércoles 1 de agosto del 2012, es decir la demanda del mismo día de la semana anterior, la figura 62 y la figura 63 muestran los estados iniciales y las entradas.

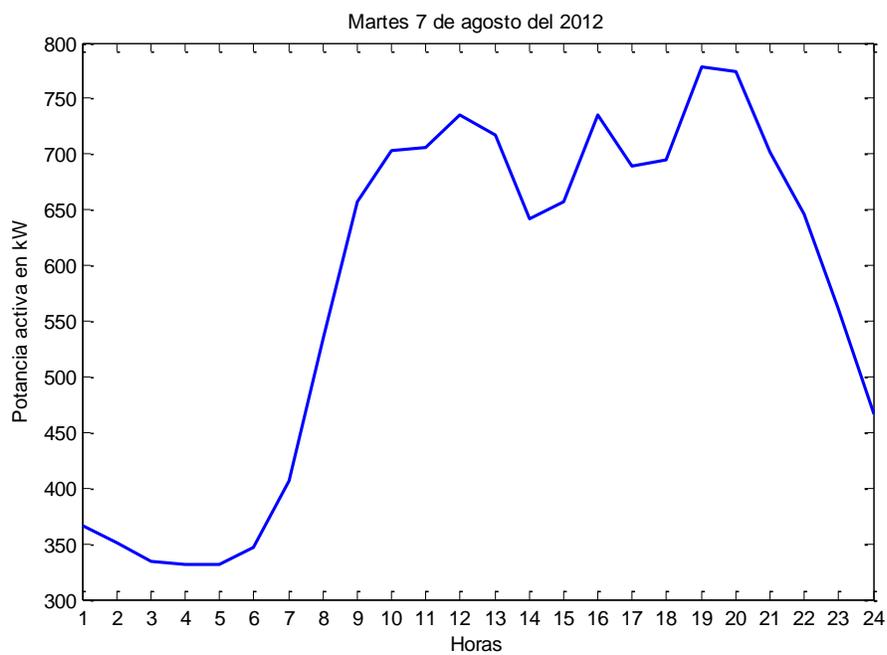


Figura 63.- Curva de demanda de potencia activa martes 07/08/2012.



Figura 64.- Curva de demanda de potencia activa miércoles 01/08/2012.

Los resultados de la evaluación se muestran en la figura 64 y la figura 65.

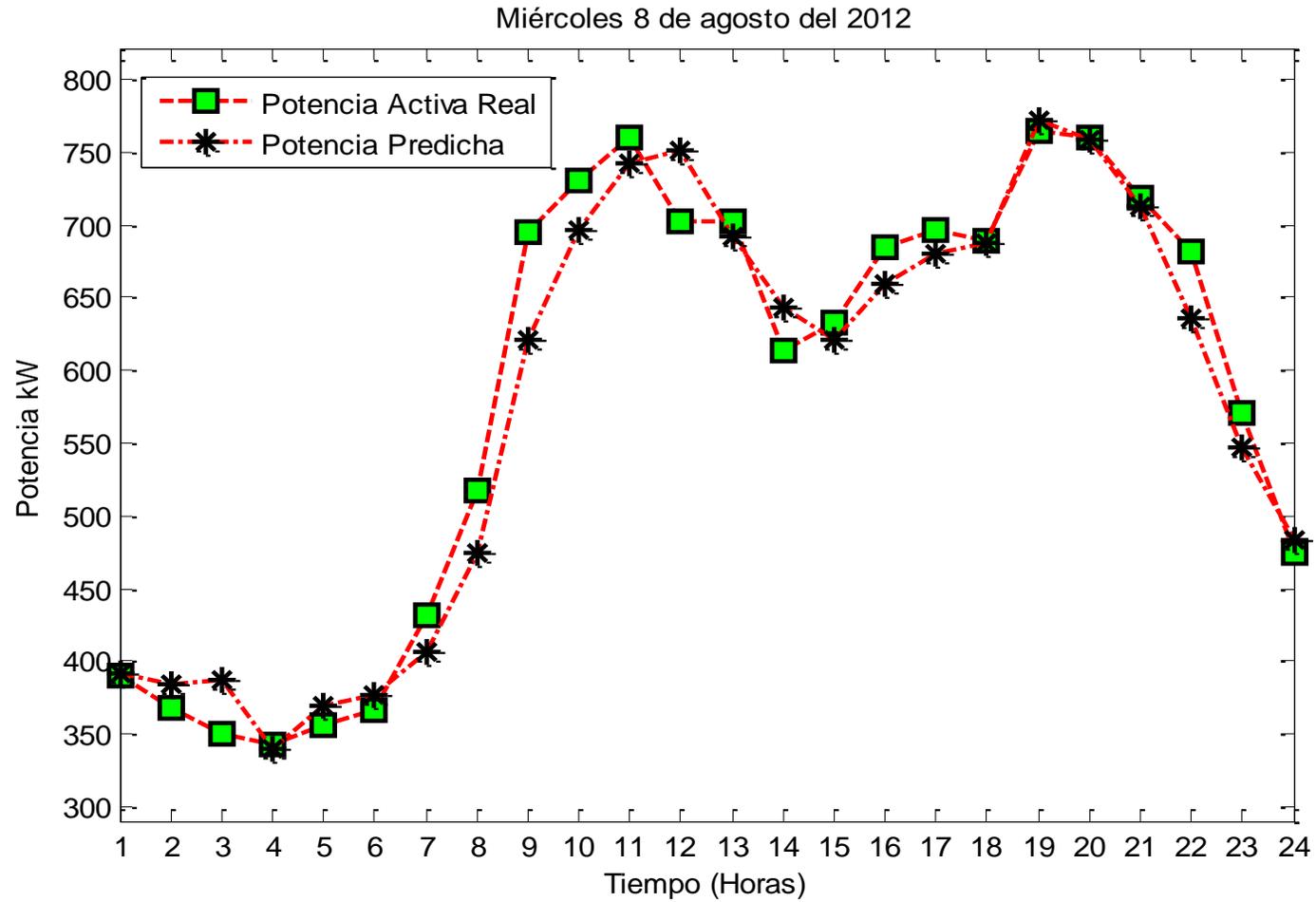


Figura 65.- Curva de demanda real de potencia activa vs. Predicción para el miércoles 08/08/2012.

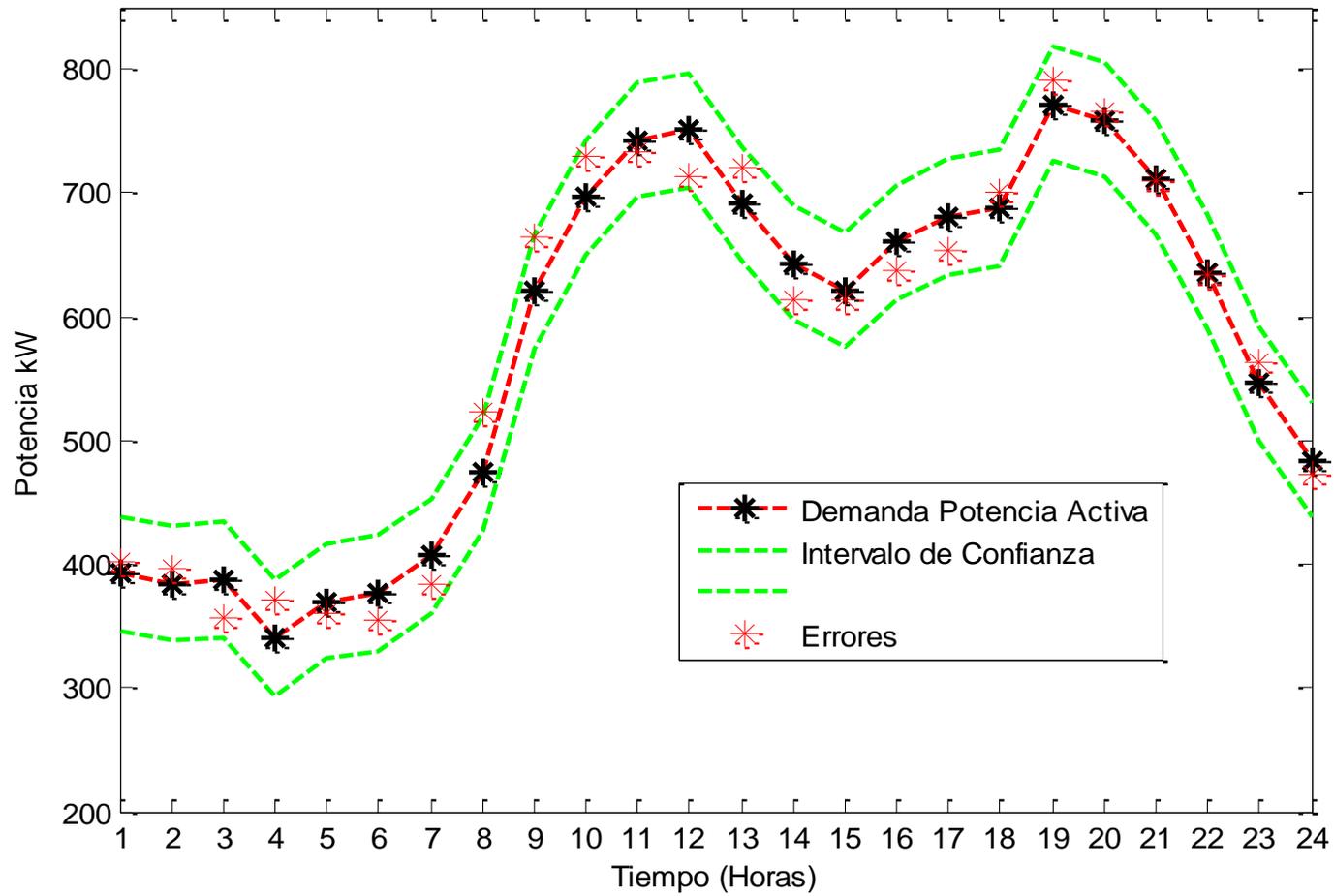


Figura 66.- Evaluación de errores e intervalo de confianza de potencia activa estimada para el 08/08/2012.

Las gráficas de respuesta temporal y de intervalos de confianza (95%) muestran un seguimiento coherente de los resultados de la RNA versus los datos que se están utilizando para validar la RNA, la función además devuelve los valores que se muestran en la tabla 13.

Tabla 13. Datos de salida del proceso de evaluación de la RNA de potencia activa.

TABLA DE DATOS DE SALIDA		
PREDICCIÓN		
Mínimo	Medio	Máximo
717 kW	771,5 kW	826 kW
OBJETIVO		
Mínimo	Medio	Máximo
702 kW	765,2 kW	827,6 kW
MÉTRICAS		
MSE	589,7	
R	0,98	

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

f.1.2. PREDICCIÓN DE POTENCIA ACTIVA MEDIANTE LA RNA

Una vez validado el funcionamiento de la RNA, podemos comenzar a predecir la curva de demanda de potencia activa en base a los parámetros requeridos por el sistema, en este caso la serie temporal de 7920 muestras termina en el domingo 12 de agosto del 2012, se predecirá entonces la curva de demanda del lunes 13 de agosto en base a los estados iniciales mencionados y las entradas del lunes 6 de agosto del 2012, los estados iniciales y la entrada a la RNA se muestra en la figura 66 y la figura 67 respectivamente.

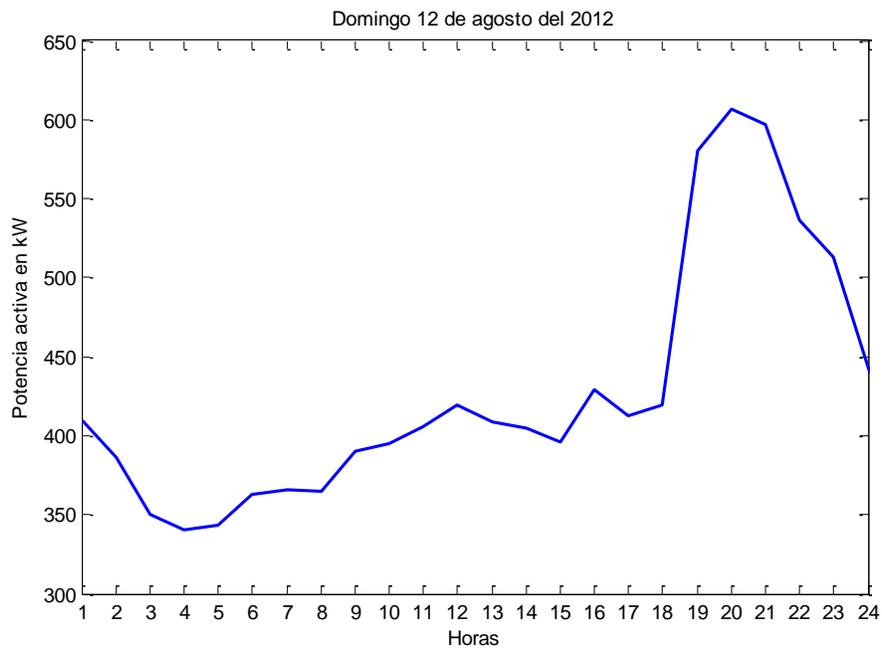


Figura 67.- Curva de demanda de potencia activa domingo 12/08/2012.

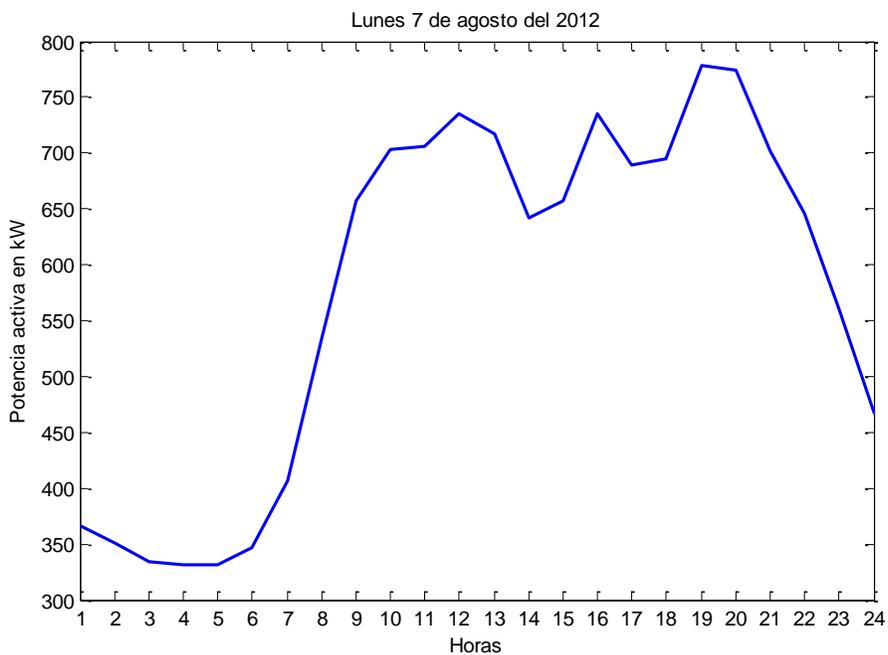


Figura 68.- Curva de demanda de potencia activa lunes 07/08/2012.

La predicción de la curva de demanda de potencia activa usando la RNA escogida se determina mediante la utilización de la función *predictP.m*, los resultados se muestran en la figura 68, en donde se predice la curva de demanda de potencia activa del día Lunes 13 de agosto del 2012, con los datos del día Domingo 12 de agosto del 2012 como estados iniciales y como entrada los datos del día Lunes 7 de agosto del 2012.

Además de la gráfica la función arroja los resultados que se muestran en la tabla 14.

Tabla 14. Datos de salida del proceso de evaluación de la RNA de potencia activa.

TABLA DE DATOS DE SALIDA		
Mínimo	Medio	Máximo
745,3 kW	797,4 kW	849,4 kW

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

Estos valores son referidos a la curva de demanda de potencia activa de un día.

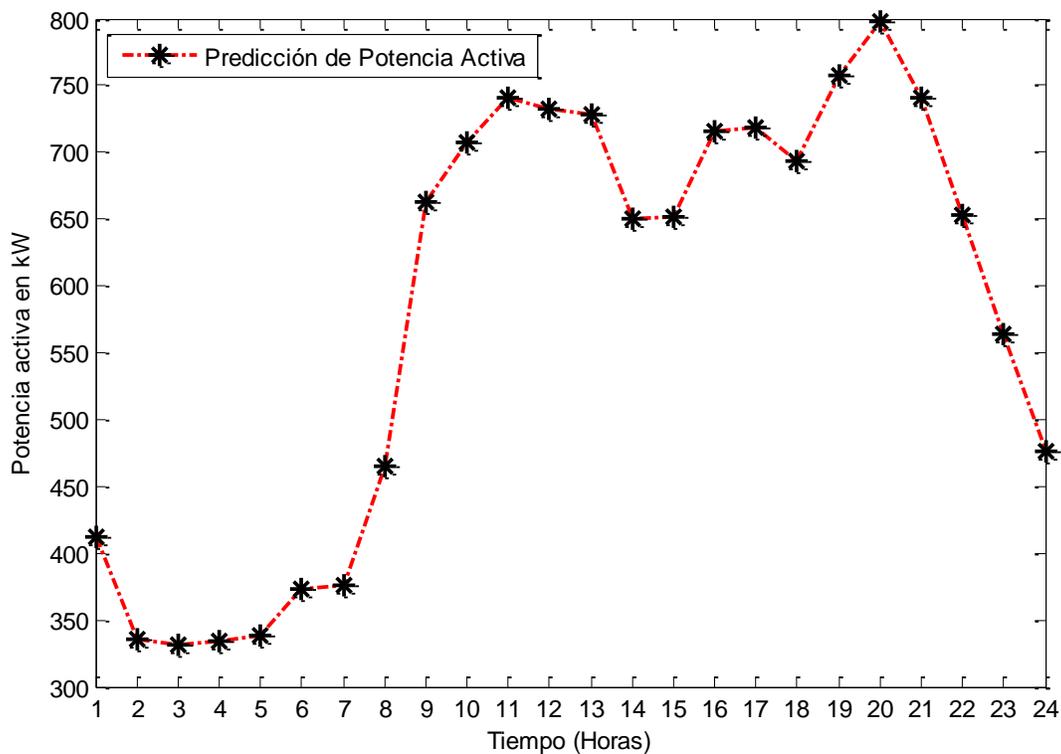


Figura 69.- Curva de demanda de potencia activa pronosticada para el lunes 13/08/2012.

f.1.3. EVALUACIÓN Y PREDICCIÓN DE POTENCIA REACTIVA

Para evaluar la RNA obtenida del proceso de selección realizado anteriormente, se valorará los resultados mediante el uso de la función *evalnar.m*.

Como estados iniciales se introdujo la curva de demanda de potencia reactiva perteneciente al día martes 7 de agosto del 2012, como entradas se introdujo la curva de demanda del día miércoles 1 de agosto del 2012, es decir la demanda del mismo día de la semana anterior, la figura 69 y la figura 70 muestran los estados iniciales y las entradas.

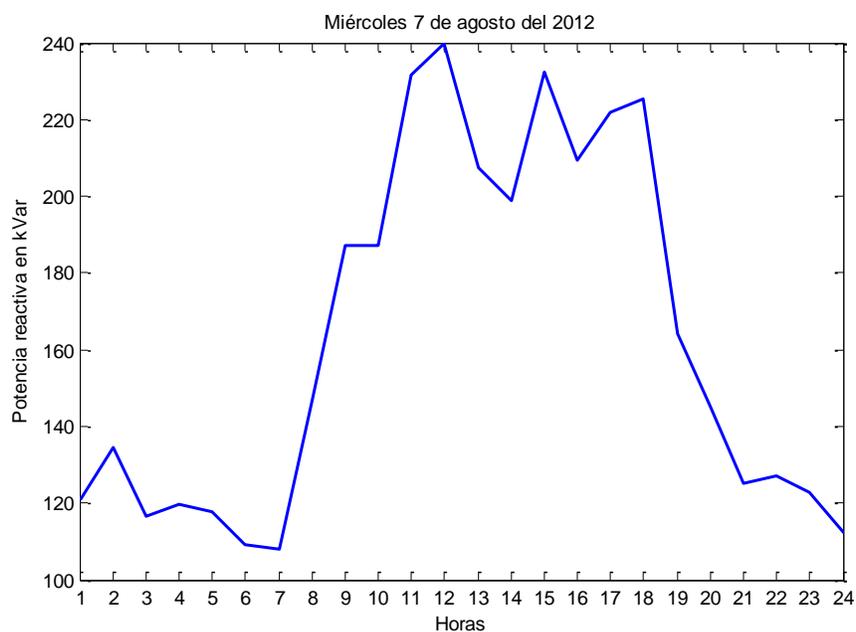


Figura 70.- Curva de demanda de potencia reactiva martes 07/08/2012.

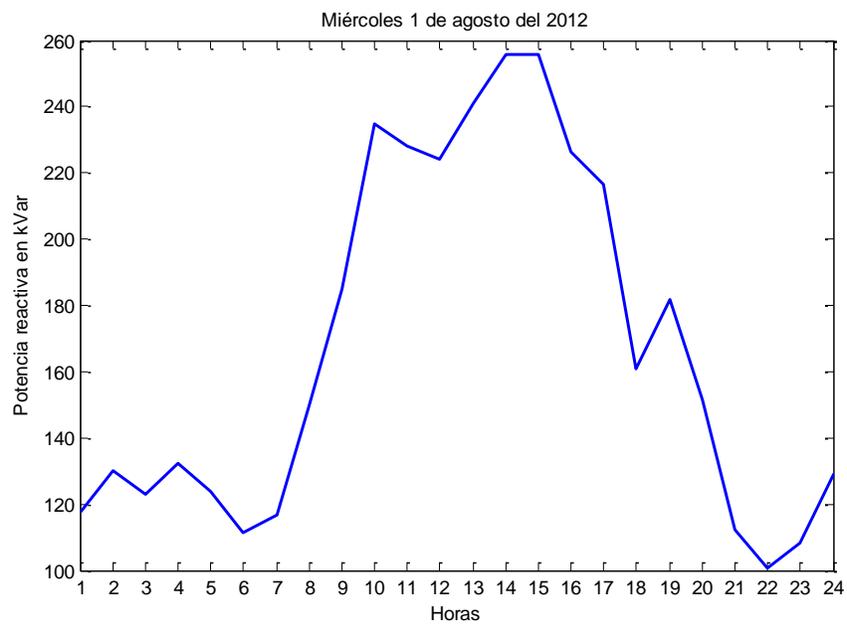


Figura 71.- Curva de demanda de potencia reactiva miércoles 01/08/2012.

Los resultados de la evaluación se muestran en la figura 71 y la figura 72.

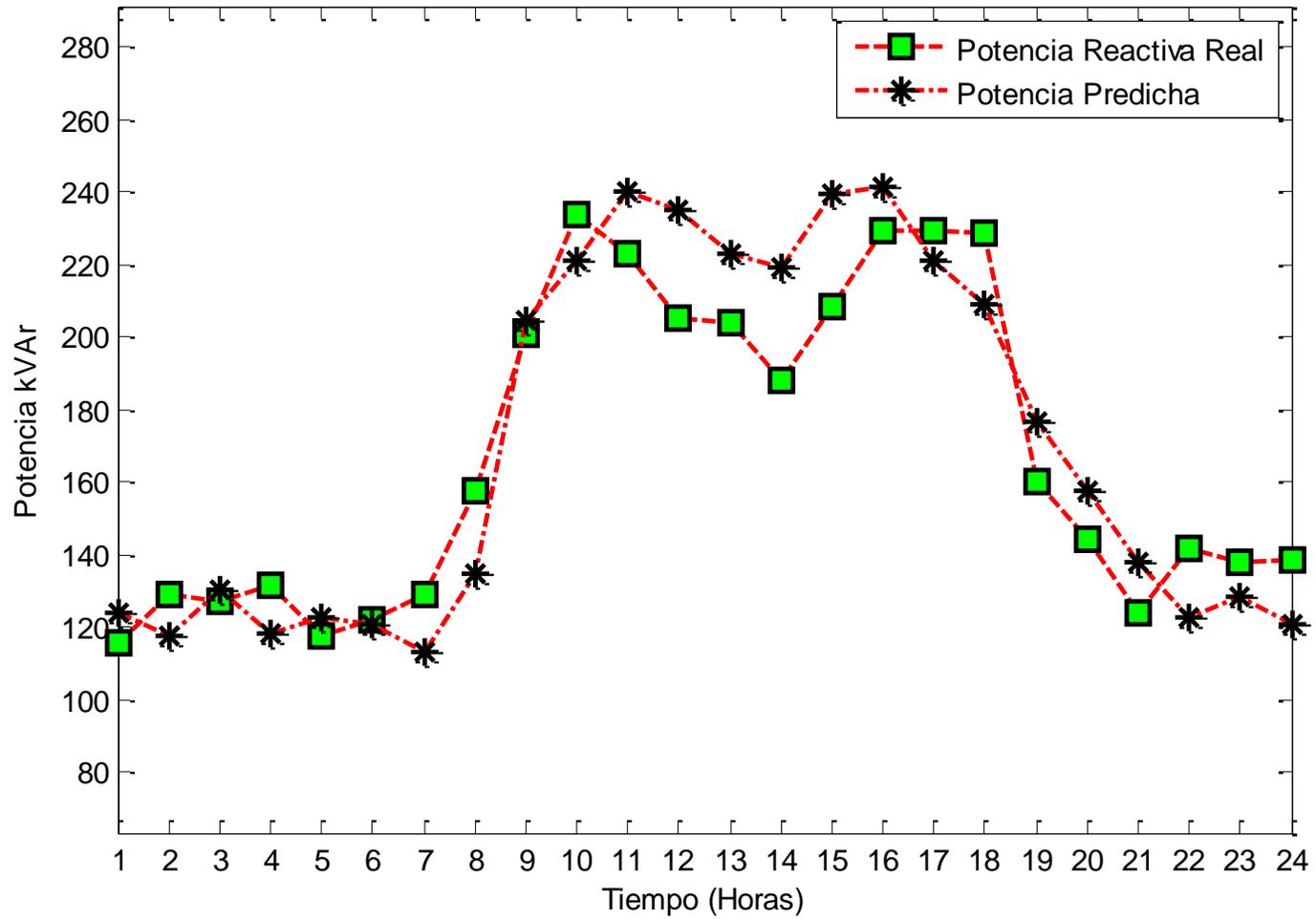


Figura 72.- Curva de demanda real de potencia reactiva vs. Predicción para el miércoles 08/08/2012.

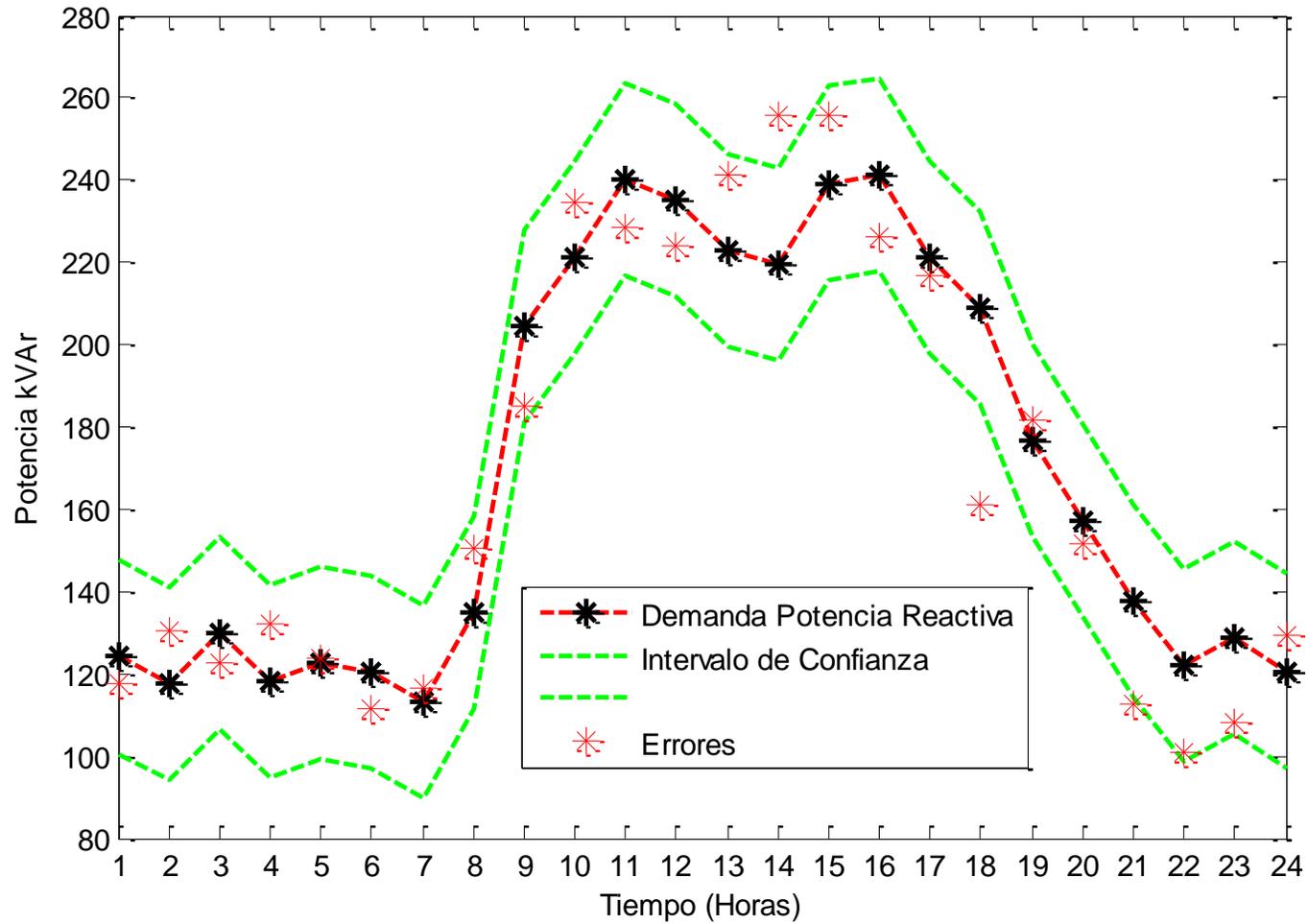


Figura 73.- Evaluación de errores e intervalo de confianza de potencia reactiva predicha para el 08/08/2012.

Las gráficas de respuesta temporal y de intervalos de confianza (95%) muestran un seguimiento coherente de los resultados de la RNA versus los datos que se están utilizando para validar la RNA, la función además devuelve los valores que se muestran en la tabla 15.

Tabla 15. Datos de salida del proceso de evaluación de la RNA de potencia reactiva.

TABLA DE DATOS DE SALIDA		
PREDICCIÓN		
Mínimo	Medio	Máximo
229 kVar	241,2 kVar	253,2 kVar
OBJETIVO		
Mínimo	Medio	Máximo
221,9 kVar	233,6 kVar	245,3 kVar
MÉTRICAS		
MSE	322,6	
R	0,94	

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

f.1.4. PREDICCIÓN DE POTENCIA REACTIVA MEDIANTE RNA

Una vez validado el funcionamiento de la RNA, podemos comenzar a predecir la curva de demanda de potencia reactiva en base a los parámetros requeridos por el sistema, en este caso la serie temporal de 7920 muestras termina en el domingo 12 de agosto del 2012, se predecirá entonces la curva de demanda del lunes 13 de agosto en base a los estados iniciales mencionados y las entradas del lunes 6 de agosto del 2012, los estados iniciales y la entrada a la RNA se muestra en la figura 73 y la figura 74 respectivamente.

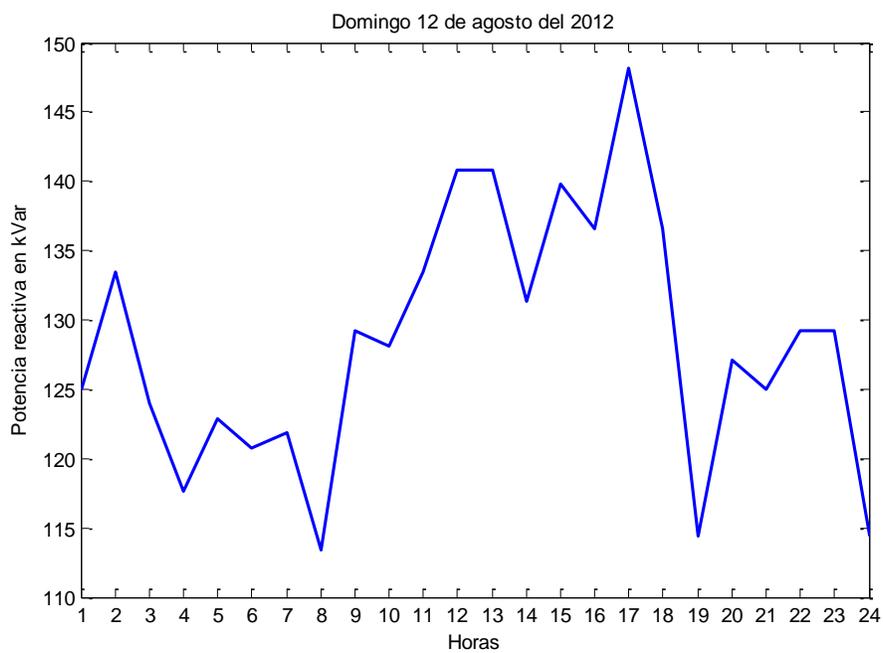


Figura 74.- Curva de demanda de potencia activa domingo 12/08/2012.

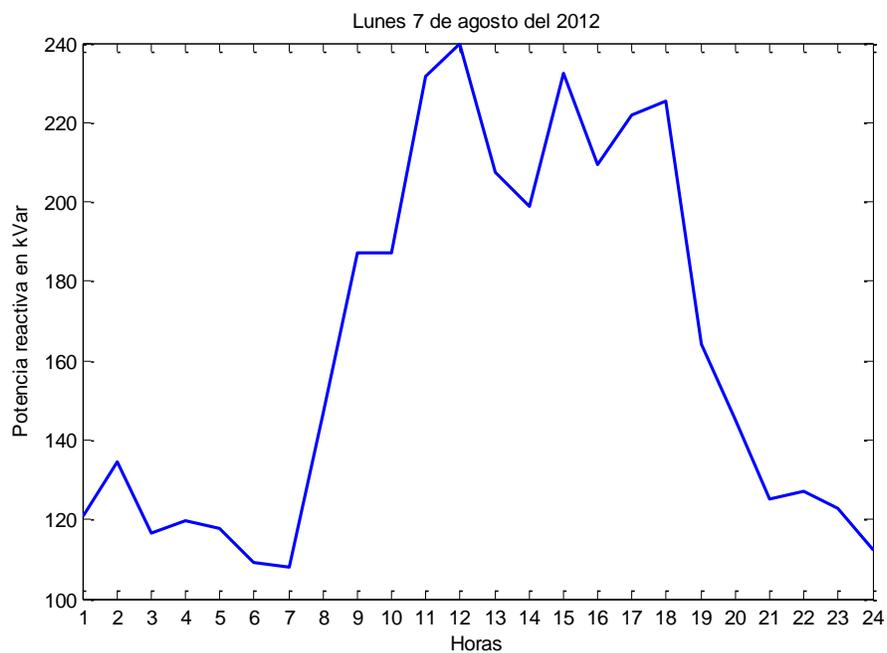


Figura 75.- Curva de demanda de potencia activa lunes 07/08/2012.

La predicción de la curva de demanda de repotencia activa usando la RNA escogida se determina mediante la utilización de la función *predictP.m*, (ver Anexo 9) los resultados se muestran en la figura 75, en donde se predice la curva de demanda de potencia reactiva del día Lunes 13 de agosto del 2012, con los datos del día Domingo 12 de agosto del 2012 como estados iniciales y como entrada los datos del día Lunes 7 de agosto del 2012.

Además de la gráfica la función arroja los resultados que se muestran en la tabla 16.

Tabla 16. Datos de salida del proceso de evaluación de la RNA de potencia reactiva.

TABLA DE DATOS DE SALIDA		
Mínimo	Medio	Máximo
216,6 kVar	233,8 kVar	251 kVar

Fuente: desarrollado por Ing. Juan Pablo Cabrera S.

Estos valores son referidos a la curva de demanda de potencia activa de un día.

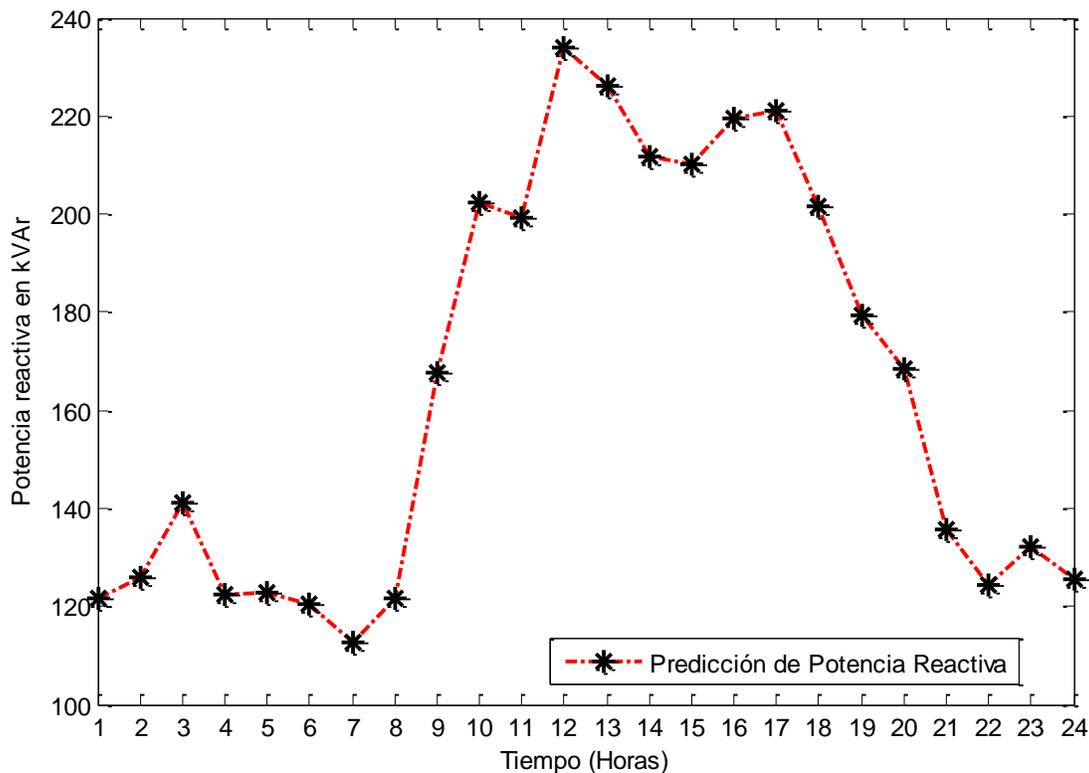


Figura 76.- Curva de demanda de potencia reactiva pronosticada para el lunes 13/08/2012.

Además de la predicción de la demanda de potencia activa y reactiva, también se calcula la demanda de potencia aparente y el factor de potencia, los resultados se muestran en la figura 76 y figura 77 respectivamente.

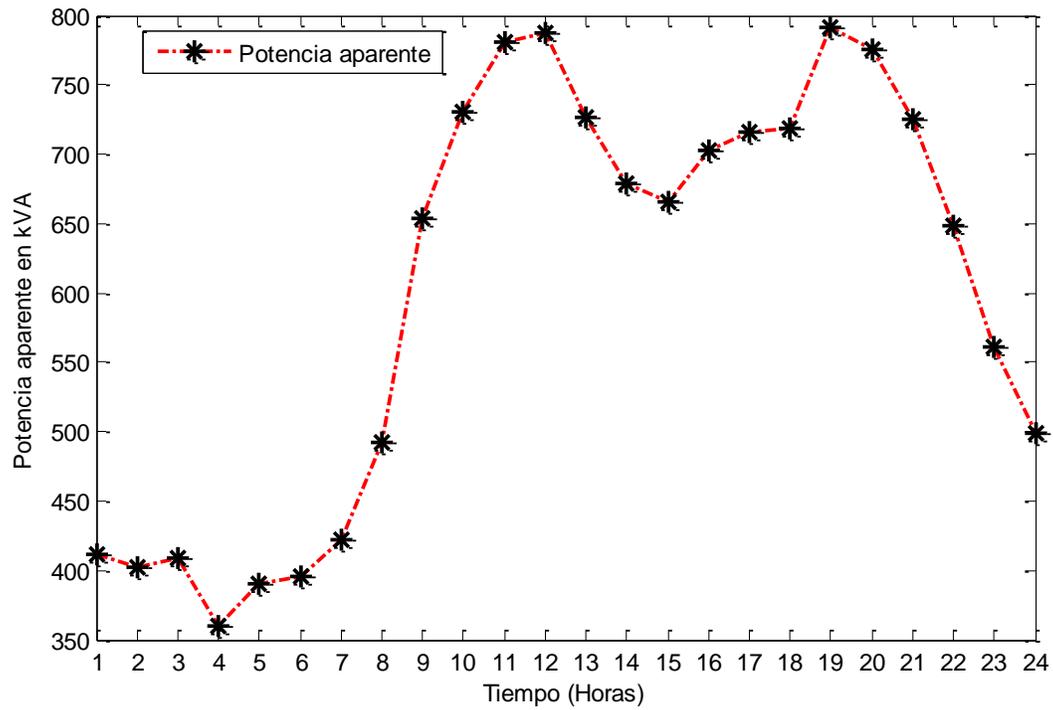


Figura 77.- Curva de demanda de potencia aparente pronosticada para el lunes 13/08/2012.

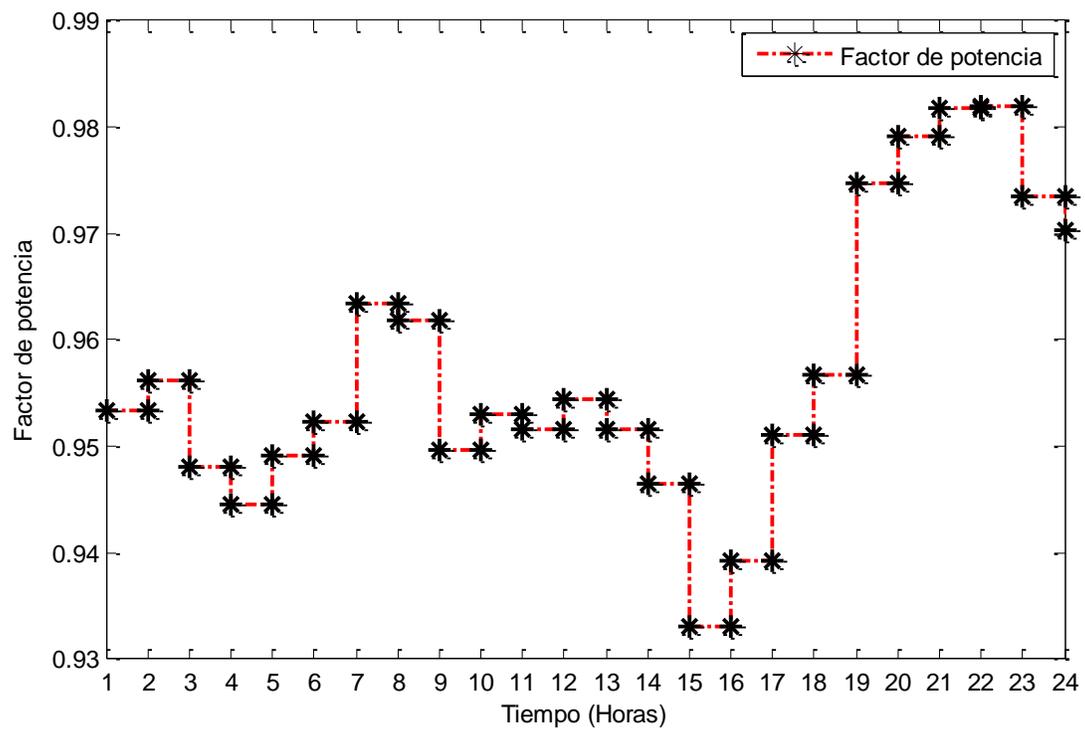


Figura 78.- Factor de potencia para el lunes 13/08/2012.

g. DISCUSIÓN

Los procesos de predicción de demanda de potencia se pueden clasificar en 3 estados:

- Corto plazo
- Mediano plazo
- Largo plazo

En el presente trabajo investigativo se optó por la predicción a corto plazo, es decir la predicción horaria, teniendo como resultado máximo la curva de demanda de potencia de una semana.

Para desarrollar los modelos de predicción se creó una RNA para predecir la potencia activa y otra RNA para predecir la potencia reactiva, la separación de las dos predicciones de potencia reduce la dificultad de procesarlas juntas.

Para el proceso de decisión de la arquitectura de la RNA que se adapte a los requerimientos de la investigación para la predicción, se estableció un conjunto de métricas, todas basadas en el error, una de ellas y por la cual se seleccionó el algoritmo de entrenamiento de la red es el error cuadrático medio (MSE), no se seleccionó las RNA que tengan el menor error cuadrático medio, sino las que mejor se ajusten al conjunto de parámetros planteados al inicio del diseño de la RNA, inclusive en el proceso de validación, se proporciona las métricas referentes al valor de regresión y el error en validación, los cuales son coherentes y cercanos a los planteados al inicio de la etapa de diseño.

La arquitectura de RNA escogida para el desarrollo del modelo de predicción fue una red de tipo No-lineal Autoregresiva, la cual por su característica de ser regresiva, además de los valores de retardos colocados en las diferentes capas ocultas de la RNA, le añaden dinámica al sistema, especialmente cuando los datos fueron tomados como una serie temporal.

Los modelos típicos de pronóstico de demanda a corto plazo se basan en la utilización de una RNA cuya arquitectura es de tipo Perceptrón-multicapa con entradas históricas de los

consumos y variables climáticas (Bastidas, Montoya, & Velásquez, 2008) (Rueda, Velásquez, & Franco, 2011).

En este punto se denota la principal diferencia entre las principales arquitecturas implementadas, según la bibliografía consultada y la empleada en la presente investigación.

En nuestro caso solo se utilizó los datos históricos pertenecientes a un alimentador, en donde a través del pre-procesamiento de los datos se utiliza tendencias, agrupamientos o clusters dentro de los datos de entrada para el entrenamiento de la RNA y así poder determinar tendencias en los datos para realizar el proceso de predicción.

En general los resultados obtenidos por las RNA de predicción, son satisfactorios, se estableció un intervalo de confianza dentro de los cuales se espera que entre la mayoría de los errores inherentes a la predicción, es decir el los errores son del $\pm 5\%$, lo cual se considera aceptable.

Otro de los resultados de la RNA son 3 valores característicos, el mínimo, el medio y el máximo de la curva de demanda de potencia activa y reactiva predicha, pudiendo ser utilizados para la gestión del despacho de demanda de potencia.

La base de datos usada para el entrenamiento de la RNA fue de aproximadamente un año, siendo que la RNA busca patrones en los datos de entrada históricos para poder formar grupos y buscar tendencias sería necesario incrementar los datos de entrada de red, de modo que las métricas mejoren.

En conclusión se determinó dos RNA, una predice la potencia activa y otra la potencia reactiva, ambas poseen un error cuadrático medio relativamente bajo y valores de regresión de 0,98 y 0,94 los cuales son altos, la correlación del error es baja y entra dentro de los límites de los intervalos de confianza así como la respuesta temporal de la curva de demanda de validación.

h. CONCLUSIONES

- La base para el desarrollo del modelo de predicción es la realización de un buen proceso de análisis de la consistencia de los datos, tratando de eliminar en su mayoría las inconsistencias presentes en los mismos, el proceso de depuración de datos fue realizado mediante herramientas estadísticas de filtrado, depuración y suavizado de la curva de demanda, este procedimiento mejora la consistencia de los datos pero añade un margen de error al sistema, en nuestro caso el margen fue de un 5% de error o 95% de intervalo de confianza.
- Las variables de análisis para el desarrollo del modelo fueron exclusivamente potencia y tiempo, es decir, se toma a la totalidad de los datos como una serie temporal, esto diferencia al modelo desarrollado de algunos modelos comparativos encontrados en la literatura. Comúnmente se toma en cuenta variables climáticas o demográficas para la creación de modelos de predicción tratando de determinar cuáles son las mayormente vinculadas al proceso de fluctuación de la demanda, en nuestro caso mediante los antecedentes históricos dados por la serie temporal, se crea patrones y se los agrupa para establecer tendencias y comportamientos típicos, las variables que acompañan a la fluctuación de la demanda son determinadas en las capas ocultas de la arquitectura de la RNA, pero son transparentes al usuario y se optimizan mediante los múltiples entrenamientos.
- Mediante la selección de las variables de afectación al sistema se determinó cual sería la arquitectura de RNA que pudiera ajustarse a las características de diseño y variables participativas, siendo escogida una arquitectura de tipo no lineal autoregresiva con capa de entrada, capas ocultas y capa de salida, funciones de activación de tipo logsig y purelin en las capas ocultas y de salida respectivamente, además de retardos en la capa de entrada lo cual le da la característica de dinámica a la RNA, particularidad idónea debido a que el tiempo es una de las variables sustanciales del modelo.
- Para disminuir la dificultad inherente al análisis se creó dos redes separadas, una para predecir potencia activa y la otra para predecir potencia reactiva. Se crearon 20 arquitecturas de RNA variando el número de capas ocultas y de retardos en la capa

de entrada, cada arquitectura fue entrenada 10 veces y de estos 10 entrenamientos se presentó los 5 mejores, de los cuales se seleccionó a uno para que sea el modelo de predicción, esto dio como resultado una RNA de 10 capas ocultas con 23 retardos para la predicción de la potencia activa y una RNA de 15 capas ocultas con 24 retardos para la predicción de la potencia reactiva, estas dos arquitecturas responden a las métricas de diseño planteados al inicio del proceso de creación de la RNA.

- El modelo desarrollado se clasifica como modelo de predicción a corto plazo, es decir la predicción de potencia se proyecta al día siguiente, esto debido al hecho de que los datos de referencia en la entrada son 24 los cuales representan el valor de máxima potencia por hora en un determinado día, basándose además en la arquitectura de RNA la cual plantea como estados iniciales 24 valores, pudiendo prolongar la salida a los valores que se consideren necesarios.
- El modelo desarrollado podría ser usado para predecir semanas, inclusive años de demanda. La exactitud y consistencia del pronóstico no se garantiza a ese nivel debido principalmente al hecho de que los datos o la serie temporal total es de cerca de un año para el proceso de entrenamiento, validación y testeo. La red se basa en patrones y tendencias de los datos, si los datos son de un año es capaz de predecir la potencia en intervalos de tiempo dentro de dicho año, en el caso de poseer datos de intervalos de tiempo mayores, la RNA sería capaz de proyectar un pronóstico en base a la cantidad de información que maneje tomando en cuenta los patrones de comportamiento de la serie temporal.
- Entrenar la RNA con un número de datos equivalentes a un año, limita la capacidad de proyectar predicciones de demanda de potencia del modelo, causando una limitante en la validez del mismo en el tiempo, es decir, se garantiza un año de predicción en base a la disponibilidad de información, se estima que con una mayor cantidad de años en información la RNA sea capaz de absorber características de crecimiento constantes en la demanda.

i. RECOMENDACIONES

- Es necesario que la Empresa Eléctrica Regional del Sur (EERSSA) realice una verificación al sistema SCADA, esta debido a que encontraron algunas inconsistencias, incongruencias, inclusive vacíos en los datos, lo cual es atribuible a des-calibración en los sistemas de adquisición de datos, errores en el procesamiento o almacenamiento de los mismos o señales de ruido inherentes a todos proceso de transmisión de datos.
- Se recomienda estandarizar el formato de cuantificación del tiempo, ya que en el proceso de creación, entrenamiento y validación de la RNA, en los datos se encontraron diferentes formatos lo cual crea un problema el momentos de realizar el procesamiento informático, así mismo, definir un periodo de muestreo constante lo que facilitará los análisis futuros y minimización de los errores al momento de desarrollar modelos de algún tipo.
- Es recomendable, para cualquier proceso de desarrollo de modelos o cualquier actividad concerniente con bases de datos, realizar un proceso previo de verificación de la calidad de los mismos ya que, la solvencia de los datos garantizará la calidad del modelo de predicción, es decir, si los datos son de pésima o baja calidad el modelo también lo será.
- Al finalizar de todo proceso de predicción es recomendable realizar un análisis de calidad del mismo, a través de procesos de validación y test. Se puede llegar más halla y tratar de determinar la calidad en base a la interpretación más objetiva de resultados, es decir, no todo modelo que tenga las métricas ideales es perfecto, en ocasiones esta exactitud dará como resultado que el modelo prediga inclusive los errores de la serie temporal analizada.
- Es recomendable utilizar una cantidad más elevada de datos, es este caso la base datos fue de 36387 muestras, el equivalente a 7920 horas o 0,9 años, lo ideal sería contar con una base de por lo menos el doble de muestras. La RNA utilizada se basa en tendencias, es decir clasifica, agrupa, datos en función de su evolución temporal,

a mayor conocimiento (más datos) de la serie temporal mayor precisión tendrá la RNA en el momento de realizar la predicción.

- El manejo y procesamiento de bases de datos de gran tamaño, como la utilizada, implica un gran gasto computacional y humano para su análisis, en este punto es recomendable crear un software de pre-procesamiento, el cual, no solo serviría para el desarrollo de modelos, sino, para el análisis mismo de los datos, información que sería más versátil que los mismo datos sin procesar.

j. BIBLIOGRAFÍA

- Arbib, M. (2003). *The Handbook of Brain Theory and Neural Networks: Second Edition*. MIT Press.
- Bao, J. (2000). *Short-term Load Forecasting based on Neural Network and Moving*. Artificial Intelligence Lab, Dept of Computer Science Iowa State.
- Bastidas, L., Montoya, S., & Velásquez, J. (2008). ¿Hacia dónde irán los sectores eléctricos de los países de la región Andina? Tendencias posibles. *Cuadernos de Administración*.
- Bunn, D., & Farmer, E. D. (1985). *Comparative models for electrical load forecasting*.
- Flórez López, R., & Fernández Fernández, J. (2008). *Las Redes Neuronales Artificiales: Fundamentos Teóricos y Aplicaciones Prácticas*. Netbiblo.
- García Delgado, L., Myro, R., & Martínez Serrano, J. (1997). *Lecciones de economías Española (Tercera ed.)*. Civitas.
- Graupe, D. (2007). *Principles of Artificial Neural Networks (2nd Edition)*. World Scientific.
- Haykin, S. (2009). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.
- Hernández López, L. (2006). *Predicción y optimización de emisiones y consumo mediante redes neuronales*. Reverte.
- Kher, L., Sioshanshi, F., & Soroosshian, S. (1987). Energy demand modelling with noisy input – output variables. *The energy journal*.
- Kumar, S. (2004). *Neural Networks: A Classroom Approach*. Tata McGraw-Hill .
- Medina, S., & García, J. (2005). Predicción de demanda de energía en Colombia mediante un sistema de inferencia difuso neuronal. *Revista Energética*.
- Murto, P. (1998). *Neural Network Models for short-term load forecasting*. Helsinki University of Technology: Department of Engineering, Physics and Mathematics.
- Pai, P., & Hong, W. (2005). Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electric Power Systems Research*.

- Priddy, K., & Keller, P. (2005). *Artificial Neural Networks: An Introduction*. SPIE Press.
- Rueda, V. M., Velásquez, J. D., & Franco, C. J. (2011). Avances recientes en la predicción de la demanda de electricidad usando modelos no lineales.
- Shachmurove, Y. (2002). *Applying Artificial Neural Networks to Business*. University of Pennsylvania: Economics and Finance.
- Sivanandam, S., & Deepa, S. (2006). *Intro To Neural Net With Matlab 6.0*. Tata McGraw-Hill Education.
- Tkacz, G., & Hu, S. (1999). *Forecasting GDP growth using Artificial Neural Networks*.
- Valanecia, A., Lozano, C., & Moreno, C. (2007). Modelo de promedios móviles para el pronóstico horario de potencia y energía eléctrica. *El hombre y la máquina*.
- Yegnanarayana, B. (2004). *Artificial Neural Networks*. PHI Learning Pvt. Ltd.

k. ANEXOS

ANEXO 1: código fuente para la lectura de los datos desde un archivo de Excel.

```

%*****Lectura del archivo fuente EXCEL*****
%*****

[~, ~, DataFechas] = xlsread(...,
'C:\Users\CCS USER\MAESTRIA\TESIS\PROYECTOS DE TESIS\MODELOS
DESARROLLO\DATA.xls',...
'Hoja1','A2:A36388');

% Lectura y selección de celdas válidas
DataFechas(cellfun(@(x) ~isempty(x) && isnumeric(x) &&
isnan(x),DataFechas)) = {''};
% conversión a CELLARRAY
[~, ~, raw, dateNums] = xlsread(...,
'C:\Users\CCS USER\MAESTRIA\TESIS\PROYECTOS DE TESIS\MODELOS
DESARROLLO\DATA.xls',...
'Hoja1','B2:K36388','',@convertSpreadsheetDates);

R = ~cellfun(@isequalwiththequalnans,dateNums,raw) & cellfun('isclass',...
raw,'char');
raw(R) = dateNums(R);

% Sustitución de valores no numericos e indeterminados
R = cellfun(@(x) ~isnumeric(x) || isnan(x),raw);
raw(R) = {0.0};
data = cell2mat(raw); % Paso de celda a numérico
Data = data;
clearvars data raw dateNums R columnIndices;

```

ANEXO 2: Función *FechNumVec.m*

```
function [Hora,Dia,Mes] = FechNumVec (DataFechas)

%*****GUIA DE ELEMENTOS DE LA FUNCIÓN*****
%*****
% DataFechas ==> Celda contenedora de información de tiempo de los Datos.
% Hora ==> Vector tipo double con las horas.
% Día ==> Vector tipo double con los días.
% Mes ==> Vector tipo double con los meses.
%*****
%*****

FechasArray = DataFechas; % Definiendo vectores a usar
[fil,~] = size(FechasArray);
for N = 1:fil
    FechasNum = datenum(FechasArray(N,1)); %Fechas codificadas
    FechasVec(N,:) = datevec(FechasNum); % Fechas formato yy mm dd HH MM SS
end
Hora = FechasVec(:,4); %Definiendo vector de horas
Dia = FechasVec(:,3);
Mes = FechasVec(:,2);
```

ANEXO 3: Función *DataMax.m*

```

function [PAMax] = DataMax (Horas,Data)

%*****GUIA DE ELEMENTOS DE LA FUNCIÓN*****
%*****
% Horas ==> Vector de horas de los datos.
% Data ==> Vector de potencia.
% PAMax ==> Vector de salida con los valores máximos por hora.
%*****
%*****

Href = 0;
M = 1;
X = 1;
[fil,~] = size(Horas);

for K = 1:fil
    if Horas(K,1) == Href % Rastreo de valores maximos por hora
        DataVect(M,:) = Data(K,2:3); % Matriz de potencias
        M = M + 1;
    else
        M = 1;
        Vect = DataVect(:,1); % Vector auxiliar de datos por hora
        [C,I] = max(Vect);
        PAMax(X,:) = [Href,DataVect(I,:)]; % Data con los máximos por hora
        X = X+1;
        DataVect(:, :) = 0;
        DataVect(M,:) = Data(K,2:3);
        M = M + 1;
        Href = Href+1;
        if Href > 23
            Href = 0;
        end
    end
end
end

```

ANEXO 4: Función *tstoollog.m*

```
function [PP] = tstoollog2(PAMax)

%*****GUIA DE ELEMENTOS DE LA FUNCIÓN*****
%*****
% PAMax ==> Vector con datos de potencia.
% PP ==> Serie temporal.
%*****
%*****

PP = timeseries(PAMax);
PP.TimeInfo.Format = 'dd-mmm-yyyy HH:MM:SS';
PP.TimeInfo.Units = 'days';
PP.TimeInfo.StartDate = '18-sep-2011 00:00:00';
tsTime = linspace(0.000000,329.958333,7920)';
PP.Time = tsTime;
```

ANEXO 5: Función *MoveOutLi.m*

```
function [c3m_PP] = MoveOutLi(PP)

%*****GUIA DE ELEMENTOS DE LA FUNCIÓN*****
%*****
% c3m ==> Datos filtrados.
% PP ==> Datos en bruto.
%*****
%*****
%% PRE-PROCESAMIENTO DE LOS DATOS
PPmax = PP; % Definiendo vector de trabajo
bin_counts_PP = hist(PPmax); % Frecuencias en la Data
N = max(bin_counts_PP); % Número máximo de frecuencias de la Data
mu_PP = mean(PPmax); % Promedio de la Data
sigma_PP = std(PPmax); % Desviación estandar de la Data
outlierspos_PP = (PPmax - mu_PP) > 2*sigma_PP ; % Límite de control
superior
outliersneg_PP = (PPmax - mu_PP) < -2*sigma_PP; % Límite de control
inferior
zerospos_PP = PPmax == 0; % Ubicación de datos en cero
c3m_PP = PPmax;

%% ESTABLECIMIENTO DE LÍMITES DE EXCLUSIONES
figure( 'Name', 'Análisis estadísitico de los datos' );
subplot(1,2,1),hist(PPmax) % Histograma de la Data
title('Histograma de Potencias'),xlabel('Potencia
kVA'),ylabel('Frecuencia')
hold on
plot([mu_PP mu_PP],[0 N], 'r', 'LineWidth',2) % Posicionando promedio
X = repmat(mu_PP+(1:2)*sigma_PP,2,1); % Estableciendo límite de control
Y = repmat([0;N],1,2);
plot(X,Y, 'g', 'LineWidth',2)
X = repmat(mu_PP-(1:2)*sigma_PP,2,1);
Y = repmat([0;N],1,2);
plot(X,Y, 'g', 'LineWidth',2)
legend('Data', 'Promedio', 'Desviación estandar')
hold off

%% REMOCIÓN DE DATOS ATÍPICOS
c3m_PP(outlierspos_PP) = 0; %Sustitución de outliers positivos
c3m_PP(outliersneg_PP) = 0; %Sustitución de outliers negativos
c3m_PP(zerospos_PP) = 0; %Reposición de valores en cero
% Gráfica de control
subplot(1,2,2),plot(PP),title('Potencia')
xlabel('Tiempo (Horas)'),ylabel('Potencia en kVA')
M = length(PP);
hold on
plot([0 M],[mu_PP mu_PP], '--*r', 'LineWidth',2.5)
plot([0 M],[2*sigma_PP+mu_PP 2*sigma_PP+mu_PP], ':*g', 'LineWidth',2.5)
plot([0 M],[mu_PP-2*sigma_PP mu_PP-2*sigma_PP], ':*g', 'LineWidth',2.5)
legend('Data', 'Promedio', 'Desviación estandar', 'Location', 'SouthEast')
hold off
```

ANEXO 6: *AjusteData.m*

```
function [DataLista,Modelo ,gof] = AjusteData(c3m)

%*****GUIA DE ELEMENTOS DE LA FUNCIÓN*****
%*****
% fitresult ==> Modelo resultante del ajuste de la Data.
% gof ==> Métricas del modelo de ajuste.
%*****
%*****
%% PÁMETROS BÁSICOS DE LA FUNCIÓN
[T,~] = size(c3m);
muestras = 1:T;
muestras = muestras';
mu3 = mean(c3m); % Promedio de la Data
sigma3 = std(c3m);

%% PRE-PROCESAMIENTO DE LOS DATOS
% Preparando datos de las curvas
[xData, yData] = prepareCurveData( muestras, c3m);
% Seteo de las opciones de ajuste.
ft = 'nearestinterp'; % Configurando el modelo de ajuste
opts = fitoptions(ft);
opts.Normalize = 'on';
% Determinación de las exclusiones en los datos
ex = excludedata( xData, yData, 'range',...
    [(-2*sigma3)+mu3 (2*sigma3)+mu3] );
opts.Exclude = ex;

%% DETERMINACIÓN DEL MODELO DE AJUSTE
% Determinación modelo de ajuste (CuveFitingToolBox)
[fitresult, gof] = fit( xData, yData, ft, opts );
% Gráfico de data ajustada vs original y valores excluidos
figure( 'Name', 'Análisis estadístico de los datos' );
h = plot( fitresult, xData, yData, ex );
legend( h, 'Potencia vs. Tiempo', 'Exclusiones vs. Tiempo',...
    'Potencia', 'Location', 'SouthEast' );
title('Datos'),xlabel( 'Tiempo (Horas)' ),ylabel( 'Potencia kVA' );
grid on
% Presentando datos de salida
Modelo = fitresult;
DataLista = feval(fitresult,muestras);
```

ANEXO 7: Función *NAR.m*

```

%*****Red Neuronal Auto-regresiva*****
%*****

% Creación de la data a ser ingresada a la RNA.
targetSeries = tonndata(PQA1_L,false,false);

% Creación de la RNA Auto-Regresiva.
feedbackDelays = 1:2;
hiddenLayerSize = 10;
net = narnet(feedbackDelays,hiddenLayerSize);

% Funciones de Pre/Post-Procesamiento.
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};

% Preparación de los datos para entrenamiento y simulación.
[inputs,inputStates,layerStates,targets] =
preparets(net,{}, {},targetSeries);

% Configuración de los datos para entrenamiento, Validación y Test
net.divideFcn = 'dividerand'; % División aleatoria de los datos
net.divideMode = 'time'; % Modo de división
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Selección del algoritmo de entrenamiento
net.trainFcn = 'trainlm'; % Levenberg-Marquardt

% Selección de la métrica del performance.
net.performFcn = 'mse'; % Error cuadrático medio

% Selección de funciones de graficación.
net.plotFcns = {'plotperform','plottrainstate',...
'plotresponse','ploterrcorr','plotinerrcorr'};

% Entrenamiento de la RNA.
[net,tr] = train(net,inputs,targets,inputStates,layerStates);

% Test de la RNA
outputs = net(inputs,inputStates,layerStates);
errors = gsubtract(targets,outputs);
performance = perform(net,targets,outputs)

% Recalculando Entrenamiento, Validación, Test y Performance.
trainTargets = gmultiply(targets,tr.trainMask);
valTargets = gmultiply(targets,tr.valMask);
testTargets = gmultiply(targets,tr.testMask);
trainPerformance = perform(net,trainTargets,outputs)
valPerformance = perform(net,valTargets,outputs)

```

```
testPerformance = perform(net,testTargets,outputs)

% Visualizar la RNA.
view(net)

% Gráficas de control
figure, plotresponse(targets,outputs)
figure, ploterrcorr(errors)
)
```

ANEXO 8: Función *evalnar.m*

```
function [outputs ys e] = evalnar(net,est,int,targets,tip)

%*****GUÍA DE ELEMENTOS DE LA FUNCIÓN*****
%*****
% outputs => Serie Temporal resultado de evaluar las Red Neuronal.
% ys => Serie resultante de Evaluar la Red de predicción temprana.
% e => Errores de predicción.
% net => Red Neuronal a ser evaluada.
% est => Estados iniciales de la Serie Temporal.
% int => Entradas de valoración para la Red.
% targets => Datos de validación para la Red.
% tip => Elegir Potencia 1 = Activa 0 = Recativa
%*****
%*****

%% PRE-POCESAMIENTO DE LOS DATOS
set = vertcat(est,int); % Establecimiento de vector de entrada completo
% Estableciendo serie-objetivo para ser ingresada a la Red
targetSeries = tonndata(set,false,false);
% Determinando datos necesarios para evaluar la Red
[inputs,inputStates,layerStates] = preparets(net,{}, {}, targetSeries);

%% EVALUACIÓN DE LA RED NEURONAL
outputs = net(inputs,inputStates,layerStates); % Evaluando la Red
outputs = cell2mat(outputs);
% Gráficas de comportamiento de la salida calculada
figure( 'Name', 'Curvas de Potencia Activa, red de predicción temprana'
);
plot(targets,'--rs','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerFaceColor','g',...
      'MarkerSize',10)
axis([1 25 min(outputs-50) max(outputs+50)])
hold on
plot(outputs,'-.*r','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerSize',10)

if tip == 1
    legend('Potencia Activa Real','Potencia Predicha')
    xlabel('Tiempo (Horas)'), ylabel('Potencia kW')
else
    legend('Potencia Reactiva Real','Potencia Predicha')
    xlabel('Tiempo (Horas)'), ylabel('Potencia kVA')
end

%% DETERMINACIÓN DE ERRORES DE PREDICCIÓN

e = gsubtract(int',outputs); % Cálculo de los errores
[R,~,~] = regression(targets',outputs); % Cálculo del valor de regresión
M = mse(e); % Calculo del error cuadrático medio
```

```

%% DETERMINACIÓN DEL INTERVALO DE CONFIANZA
m = bootstrp(24,@mean,outputs); % Calculo del intervalo de confianza
s = std(m,0);
m2 = bootstrp(24,@mean,outputs);
s2 = std(m2,0);

% Gráficas de comportamiento de la salida calculada
figure( 'Name','Gráfica de intervalo de confianza' );
plot(outputs,'--*r','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerSize',10)

hold on
plot(outputs+2*s,'--g','LineWidth',2)
plot(outputs-2*s,'--g','LineWidth',2)
plot(outputs+e,'*r','MarkerSize',10)
if tip == 1
    legend('Demanda Potencia Activa','Intervalo de Confianza de 95%',...
          '','Errores');
    xlabel('Tiempo (Horas)'), ylabel('Potencia kW')
    MaxP = max(outputs);
    disp('*****TABLA DE DATOS DE SALIDA*****')
    disp('*****PREDICCIÓN*****')
    disp('MÍNIMO          MEDIO          MÁXIMO')
    fprintf('%4.1f kW          %4.1f kW          %4.1f kW\n',...
            MaxP-(2*s),MaxP,MaxP+(2*s));
    MaxPn = max(targets);
    disp('*****OBJETIVO*****')
    disp('MÍNIMO          MEDIO          MÁXIMO')
    fprintf('%4.1f kW          %4.1f kW          %4.1f kW\n',...
            MaxPn-(2*s2),MaxPn,MaxPn+(2*s2));
    disp('*****MÉTRICAS*****')
    fprintf('Error cuadrático medio: %f\n',M)
    fprintf('Valor de regresión: : %f\n\n',R)
else
    legend('Demanda Potencia Reactiva','Intervalo de Confianza',...
          '','Errores');
    xlabel('Tiempo (Horas)'), ylabel('Potencia kVar')
    MaxQ = max(outputs);
    disp('*****TABLA DE DATOS DE SALIDA*****')
    disp('*****PREDICCIÓN*****')
    disp('MÍNIMO          MEDIO          MÁXIMO')
    fprintf('%4.1f kVar          %4.1f kVar          %4.1f kVar\n',...
            MaxQ-(0.05*MaxQ),MaxQ,MaxQ+(0.05*MaxQ));
    MaxQn = max(targets);
    disp('*****OBJETIVO*****')
    disp('MÍNIMO          MEDIO          MÁXIMO')
    fprintf('%4.1f kVar          %4.1f kVar          %4.1f kVar\n',...
            MaxQn-(0.05*MaxQn),MaxQn,MaxQn+(0.05*MaxQn));
    disp('*****MÉTRICAS*****')
    fprintf('Error cuadrático medio: %f\n',M)
    fprintf('Valor de regresión: : %f\n\n',R)
end

```

ANEXO 9: Función *predictP.m*

```
function [outputs] = predictP(net,est,int,tip)

%*****GUÍA DE ELEMENTOS DE LA FUNCIÓN*****
%*****
% outputs => Serie Temporal resultado de evaluar las Red Neuronal.
% ys => Serie resultante de Evaluar la Red de predicción temprana.
% net => Red Neuronal a ser evaluada.
% est => Estados iniciales de la Serie Temporal.
% int => Entradas de valoración para la Red.
%*****
%*****

%% PRE-POCESAMIENTO DE LOS DATOS
set = vertcat(est,int); % Establecimiento de vector de entrada completo
% Estableciendo serie-objetivo para ser ingresada a la Red
targetSeries = tonndata(set,false,false);
% Determinando datos necesarios para evaluar la Red
[inputs,inputStates,layerStates] = preparets(net,{}, {},targetSeries);

%% EVALUACIÓN DE LA RED NEURONAL
outputs = net(inputs,inputStates,layerStates); % Evaluando la Red
outputs = cell2mat(outputs);
% Gráficas de comportamiento de la salida calculada
figure( 'Name', 'Curvas de Potencia, red de predicción temprana' );
plot(outputs,'-.*r','LineWidth',2,...
      'MarkerEdgeColor','k',...
      'MarkerSize',10)
m = bootstrp(24,@mean,outputs); % Calculo del intervalo de confianza
s = std(m,0);
m2 = bootstrp(24,@mean,outputs);
s2 = std(m2,0);

if tip == 1
    legend('Predicción de Potencia Activa')
    xlabel('Tiempo (Horas)'), ylabel('Potencia activa en kW')
    MaxP = max(outputs);
    disp('*****TABLA DE DATOS DE SALIDA*****')
    disp('*****')
    disp('MÍNIMO           MEDIO           MÁXIMO')
    fprintf('%4.1f kW           %4.1f kW           %4.1f kW\n',...
            MaxP-(2*s),MaxP,MaxP+(2*s));
else
    legend('Predicción de Potencia Reactiva','Predicción Temprana')
    xlabel('Tiempo (Horas)'), ylabel('Potencia reactiva en kVar')
    MaxQ = max(outputs);
    disp('*****TABLA DE DATOS DE SALIDA*****')
    disp('*****')
    disp('MÍNIMO           MEDIO           MÁXIMO')
    fprintf('%4.1f kVar           %4.1f kVar           %4.1f kVar\n',...
            MaxQ-(2*s2),MaxQ,MaxQ+(2*s2));
End
```