

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**Predicción Multi-Horizonte de Capturas de Anchovetas
usando Redes Neuronales MIMO**

Gustavo Alejandro Gómez Contreras

INFORME DE PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO DE EJECUCIÓN EN INFORMÁTICA

Diciembre 2016

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**Predicción Multi-horizonte de capturas de anchovetas
usando Redes Neuronales MIMO**

Gustavo Alejandro Gómez Contreras

Profesor Guía: **Dr. Nivaldo Rodríguez Agurto**

Profesor Co-Referente: **Sr. Aldo Migliario Osorio**

Carrera: **Ingeniería de Ejecución en Informática**

Diciembre 2016

Resumen

Resumen

En este proyecto de título se presenta modelo basado en redes neuronales artificiales, para lograr el pronóstico en el volumen de capturas de anchovetas. La importancia del pronóstico en el tema de la pesca es de mucha relevancia, ya que, se realizan límites anuales para la extracción de la especie, por ende, esto permite mantener la especie. Además, se incluye un estudio sobre las redes neuronales artificiales, centrándose en las redes neuronales MIMO y la red neuronal recurrente Elman. Para la resolución del problema se realizó una transformación de Wavelet en los datos de entrada de la red (los cuales se obtuvieron de Sernapesca), lo cual permitió separar los datos de entrada en alta y baja frecuencia, luego de esto se realizaron las pruebas en ambas redes neuronales con el algoritmo de aprendizaje supervisado backpropagation Levenberg-Marquardt, para poder definir cuál es la mejor topología se utilizaron 3 métricas de rendimiento: RMSE, MAE y el R2. Las pruebas realizadas en las redes neuronales se dividieron en 3 etapas las cuales fueron cada 3 meses comenzando desde los 6 meses de horizonte. A pesar de tener un horizonte de 12 meses las pruebas lograron mantener resultados superiores al 85% de coeficiente de determinación entre la data observada y la data pronosticada.

Palabras Claves: Redes Neuronales, Serie de tiempo, Predicción, MSE, MAE, Redes Neuronales MIMO, Neurona Artificial.

Abstract

In this title project, we present a model based on artificial neural networks, to achieve the forecast in the volume of anchovetas catches. The importance of the forecast in the subject of the fishing is of great relevance, since, annual limits are realized for the extraction of the species, therefore, this allows to maintain the species. In addition, we included a study on artificial neural networks, focusing on the MIMO neural networks and the recurrent neural network Elman. In order to solve the problem, a Wavelet transformation was performed on the input data of the network (which were obtained from Sernapesca), which allowed to separate the input data in high and low frequency, after which the tests were performed in Both neural networks with the Levenberg-Marquardt backpropagation supervised learning algorithm, 3 metrics were used to define the best topology: RMSE, MAE and R2. The tests performed in the neural networks were divided into 3 stages which were every 3 months starting from the 6 months horizon. Despite having a horizon of 12 months, the tests were able to maintain results above 85% of the coefficient of determination between the observed data and the predicted data.

Keywords: Neural Networks, Time series, Forecasting, MSE, MAE, MIMO Neural Network, Artificial Neural.

Índice de Contenido

1. Introducción	1
2. Objetivos	3
2.1. Objetivo General.....	3
2.2. Objetivos Específicos	3
3. Redes Neuronales.....	4
3.1. Redes Neuronales Biológicas	4
4. Historia de las Redes Neuronales.....	5
5. Redes Neuronales Artificiales.....	7
5.1. Ventajas que ofrecen las RN.....	7
5.2. Aprendizaje Adaptativo	7
5.3. Auto-organización	8
5.4. Tolerancia a fallos.....	8
5.5. Elementos básicos de una RN.....	9
5.6. Función de activación	10
5.7. Mecanismos de aprendizaje	11
5.8 Aprendizaje supervisado.....	12
5.9 Aprendizaje No Supervisado	13
5.10 Algoritmo Levenberg-Marquardt.....	15
5.11 Métricas de Rendimiento	15
5.11.1 Root Mean Square Error	15
5.11.2 Mean Absolute Error.....	16
5.11.3 Coeficiente de Determinación.....	16
6. Red Neuronal Elman	17
7. Herramienta a Utilizar.....	18
7.1. Matlab	18
7.2. Interfaz de trabajo	18
7.3. Capturas de la Interfaz	18
8. Pruebas Realizadas.....	20
8.1. Introducción	20
8.2. Seis Meses de horizonte (H = 6).....	20

8.2.1.	Porcentaje de entrenamiento 70% (M = 30)	20
8.2.2.	Porcentaje de entrenamiento 80% (M = 30)	20
8.2.3.	Porcentaje de entrenamiento 70% (M = 32)	21
8.2.4.	Peor Resultado H6	21
8.2.5.	Mejor Resultado H6	23
8.3.	Nueve Meses de horizonte (H = 9)	26
8.3.1.	Porcentaje de entrenamiento 70% (M = 30)	26
8.3.2.	Porcentaje de entrenamiento 80% (M = 30)	26
8.3.3.	Porcentaje de entrenamiento 70% (M = 32)	26
8.3.4.	Peor resultado H9	26
8.3.5.	Mejor resultado H9	29
8.4.	Doce Meses de horizonte (H = 12)	31
8.4.1.	Mejor resultado H12	31
9.	Pruebas de Red Elman	33
9.1.	Seis Meses de Horizonte(H=6)	33
9.1.1.	Mejor Resultado H6	33
9.2.	Nueve Meses de Horizonte (H=9)	35
9.2.1.	Mejor Resultado H9	35
9.3.	Doce Meses de Horizonte (H = 12)	37
9.3.1.	Mejor Resultado H12	37
10.	Comparativa: Red MIMO vs Elman	39
10.1.	Seis Meses de Horizonte	39
10.2.	Nueve Meses de Horizonte	39
10.3.	Doce Meses de Horizonte	40
11.	Conclusión	41
12.	Referencias	42

Índice de Ilustraciones

Ilustración 1-1 Distribución de la Anchoveta.....	2
Ilustración 3-1 Neuronas Biológicas.....	4
Ilustración 5-1 Capas de una RNA.....	9
Ilustración 5-2 Estructura de una neurona artificial.....	9
Ilustración 5-3 Funciones de Activación.....	10
Ilustración 5-4 Función Sigmoidal.....	11
Ilustración 5-5 Función Tangente Hiperbólica.....	11
Ilustración 6-1 Arquitectura Red Elman.....	17
Ilustración 7-1 Interfaz principal de Menú.....	18
Ilustración 7-2 Interfaz de Resultados.....	19
Ilustración 7-3 Interfaz de Indicadores.....	19
Ilustración 8-1 Performance.....	21
Ilustración 8-2 Regresión.....	22
Ilustración 8-3 Output vs Target.....	23
Ilustración 8-4 Performance H6 Mejor.....	23
Ilustración 8-5 Output vs Target Mejor H6.....	24
Ilustración 8-6 Regresión Mejor H6.....	25
Ilustración 8-7 Performance Peor H9.....	27
Ilustración 8-8 Regresión Peor H9.....	28
Ilustración 8-9 Output vs Target Peor H9.....	28
Ilustración 8-10 Performance Mejor H9.....	29
Ilustración 8-11 Output vs Target Mejor H9.....	29
Ilustración 8-12 Regresión Mejor H9.....	30
Ilustración 8-13 Performance Mejor H12.....	31
Ilustración 8-14 Regresión Mejor H12.....	32
Ilustración 8-15 Output vs Target Mejor H12.....	32
Ilustración 9-1 Mejor Performance H6.....	33
Ilustración 9-2 Mejor Regresión H6.....	34
Ilustración 9-3 Mejor Outputs vs Targets H6.....	34
Ilustración 9-4 Mejor Performance H9.....	35
Ilustración 9-5 Mejor Regresión H9.....	36
Ilustración 9-6 Mejor Outputs vs Targets H9.....	36
Ilustración 9-7 Mejor Performance H12.....	37
Ilustración 9-8 Mejor Regresión H12.....	38
Ilustración 9-9 Mejor Outputs vs Targets H12.....	38

Índice de Tablas

Tabla 1 Datos H6 M30 70%	20
Tabla 2 Datos H6 M30 80%	20
Tabla 3 Datos H6 M32 70%	21
Tabla 4 Datos H9 M30 70%	26
Tabla 5 Datos H9 M30 80%	26
Tabla 6 Datos H9 M32 70%	26
Tabla 7 Datos M30 H12 70%	31
Tabla 8 Datos Elman H6	33
Tabla 9 Datos Elman H9	35
Tabla 10 Datos Elman H12	37
Tabla 11 Comparación Redes H6	39
Tabla 12 Comparación Redes H9	39
Tabla 13 Comparación Redes H12	40

Lista de Abreviaturas o Siglas

RNA	: Red Neuronal Artificial
R ²	: Coeficiente de Determinación
RMSE	: Root Mean Squared Error
MSE	: Mean Squared Error
MSA	: Multi-Step Ahead (Multi Horizonte)
MIMO	: Multi Input Multi Output
MAE	: Mean Square Error

1. Introducción

En este caso de estudio, se pretende tomar los datos de volúmenes de captura de Anchoveta y considerarla como una serie temporal, con el fin de lograr predecir su comportamiento. El objetivo será lograr esta predicción en base a Redes Neuronales Artificiales, las cuales pueden definirse como un modelo de procesamiento que intenta emular el sistema nervioso biológico. Estas redes están compuestas por múltiples unidades de proceso, que estarán conectadas conformando una determinada topología. La topología que abordaremos en este caso son las Redes Neuronales MIMO, la cual se refiere a que contiene Multi Inputs and Multi Output (MIMO).

La anchoveta es una especie pelágica que vive en cardúmenes en áreas que pueden estar entre 36 km. y 180 km. de distancia de la costa y se alimenta de zooplankton. Es una especie de aguas superficiales frías, es decir no se las encuentra pegada a las costas, menos en las playas, ni tampoco en aguas profundas. Los cardúmenes de anchovetas se desplazan en aguas superficiales de hasta 50 metros de profundidad en el día y suben en la noche. Se las encuentra entre los 03°30' Sur y los 37°00 Sur. En este espacio hay dos grandes zonas donde se encuentran anchovetas, la franja norte-centro del Perú que es la más importante y la franja del sur del Perú-norte de Chile que es menos poblada.

Cuando las aguas superficiales se calientan, las anchovetas se profundizan hasta cierto límite, porque con la profundidad la salinidad aumenta y también escasea el plancton. Por esa razón, si ocurre algún cambio climático, los cardúmenes se desplazan buscando el hábitat adecuado. Si como ocurrió en 1971, el Fenómeno del Niño es severo recalentando las aguas, los cardúmenes bajan y se alejan, pero como consecuencia de ese desplazamiento, la población disminuye debido a que en ningún otro lugar van a encontrar las condiciones ideales para vivir y reproducirse.

Durante la primavera y el verano normales, la anchoveta se encuentra dentro de una franja costera hasta los 36 a 54 km de la costa; en el otoño e invierno llega a desplazarse hasta los 144 km, y en algunas ocasiones, hasta 180 km de la costa. Se reproduce todo el año, pero especialmente entre julio y setiembre y en menor proporción durante los meses de febrero y marzo. A los seis meses alcanza el tamaño de 8 cm., 10,5 cm. al año de edad y 12 cm. al año y medio.

Para lograr la predicción del volumen de captura de anchovetas, se deberá diseñar un modelo de red neuronal con aprendizaje supervisado backpropagation, además para el ajuste de los pesos se utilizará el algoritmo levenberg-marquardt, el cual, ha mostrado buenos resultados en comparación del algoritmo del gradiente, el cual, tiene una curva de aprendizaje más lenta.

En Perú se capturan cada año millones de toneladas de anchoveta, mayormente con bolicheras industriales que circundan un cardumen por una gran red de cerco que se cierra por debajo. Luego bombean la captura desde la red en la bodega y más tarde de la bodega a las fábricas de harina y aceite. [2]

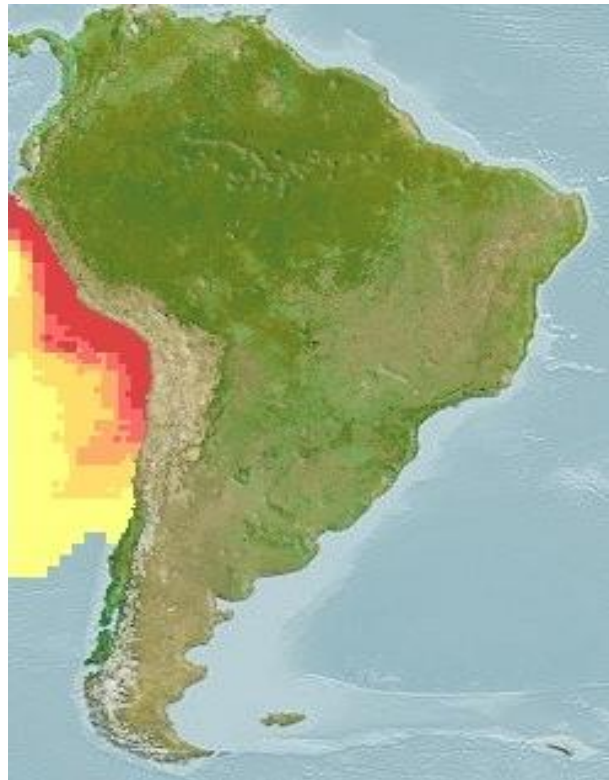


Ilustración 1-Distribución de la Anchoveta.

2. Objetivos

2.1. Objetivo General

Desarrollar dos modelos de redes neuronales MIMO con algoritmo de aprendizaje Levenberg-Marquardt (LM) para pronosticar el volumen de capturas mensuales de anchoveta en la zona centro-sur de Chile

2.2. Objetivos Específicos

- Diseñar y estimar los parámetros de una red neuronal Sigmodal-MIMO con algoritmo LM para pronóstico de capturas de anchovetas
- Diseñar y estimar los parámetros de una red neuronal Elman-MIMO con algoritmo LM para pronóstico de capturas de anchovetas
- Evaluar y comparar el rendimiento de los modelos Sigmodal-MIMO y Elman-MIMO

3. Redes Neuronales

3.1. Redes Neuronales Biológicas

El cerebro es el elemento principal del sistema nervioso humano y está compuesto por un tipo especial de célula llamada neurona. Una neurona es una célula viva y como tal posee todos los elementos comunes de las células biológicas. A su vez, las neuronas tienen características propias que le permiten comunicarse entre ellas, lo que las diferencia del resto de las células biológicas. La figura 4.1 muestra la estructura típica de una neurona biológica.

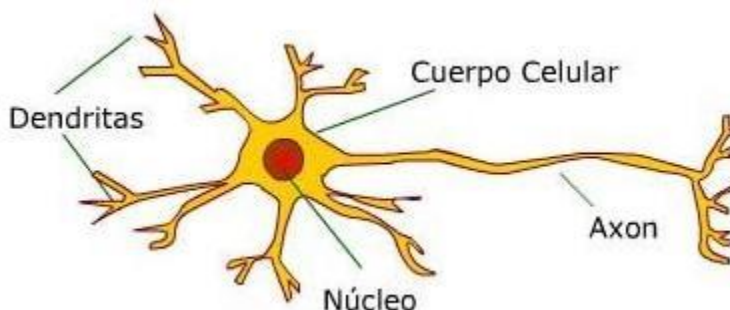


Ilustración 3-1 Neurona Biológica.

De la figura se observa que la neurona biológica está compuesta por un cuerpo celular o soma, del cual se desprende árbol de ramificaciones compuesto por las dendritas. Del soma también parte una fibra tubular, llamada axón, el cual suele ramificarse cerca de su extremo. Las dendritas actúan como un canal de entrada de señales provenientes desde el exterior hacia la neurona, mientras que el axón actúa como un canal de salida. El espacio entre dos neuronas vecinas se denomina sinapsis. En el córtex cerebral se observa una organización horizontal en capas, así como también una organización vertical en columnas de neuronas. La intensidad de una sinapsis no es fija, sino que puede ser modificada en base a la información proveniente del medio. De esta manera la estructura del cerebro no permanece fija, sino que se va modificando por la formación de nuevas conexiones, ya sean excitadoras o inhibitorias, la destrucción de conexiones, la modificación de la intensidad de la sinapsis, o incluso por muerte neuronal.

Desde un punto de vista funcional, las neuronas conforman un procesador de información sencillo. Constan de un subsistema de entrada (dendritas), un subsistema de procesamiento (el soma) y un subsistema de salida (axón) Como se mencionó antes, una de las características principales de las neuronas, y que las distinguen del resto de las células, es su capacidad de comunicarse. Las señales nerviosas pueden ser eléctricas o químicas. La transmisión química se da principalmente en la comunicación entre neuronas, mientras que la eléctrica se produce dentro de una neurona. En general, una neurona recibe información de cientos de neuronas vecinas y la transmite a otras tantas neuronas.^[7]

4. Historia de las Redes Neuronales

1936 - Alan Turing. Fue el primero en estudiar el cerebro como una forma de ver el mundo de la computación. Sin embargo, los primeros teóricos que concibieron los fundamentos de la computación neuronal fueron Warren McCulloch, un neurofisiólogo, y Walter Pitts, un matemático, quienes, en 1943, lanzaron una teoría acerca de la forma de trabajar de las neuronas. Ellos modelaron una red neuronal simple mediante circuitos eléctricos.

1949 - Donald Hebb. Fue el primero en explicar los procesos del aprendizaje (que es el elemento básico de la inteligencia humana) desde un punto de vista psicológico, desarrollando una regla de como el aprendizaje ocurría. Aun hoy, este es el fundamento de la mayoría de las funciones de aprendizaje que pueden hallarse en una red neuronal. Su idea fue que el aprendizaje ocurría cuando ciertos cambios en una neurona eran activados. También intentó encontrar semejanzas entre el aprendizaje y la actividad nerviosa. Los trabajos de Hebb formaron las bases de la Teoría de las Redes Neuronales.^[8]

1950 - Karl Lashley. En sus series de ensayos, encontró que la información no era almacenada en forma centralizada en el cerebro, sino que era distribuida encima de él.

1956 - Congreso de Dartmouth. Este Congreso frecuentemente se menciona para indicar el nacimiento de la inteligencia artificial.

1957 - Frank Rosenblatt. Comenzó el desarrollo del Perceptron. Esta es la red neuronal más antigua; utilizándose hoy en día para aplicación como identificador de patrones. Este modelo era capaz de generalizar, es decir, después de haber aprendido una serie de patrones podía reconocer otros similares, aunque no se le hubiesen presentado en el entrenamiento. Sin embargo, tenía una serie de limitaciones, por ejemplo, su incapacidad para resolver el problema de la función OR-exclusiva y, en general, era incapaz de clasificar clases no separables linealmente.

1959 - Frank Rosenblatt: Principios de Neurodinámica. En este libro confirmó que, bajo ciertas condiciones, el aprendizaje del Perceptron convergía hacia un estado finito (Teorema de Convergencia del Perceptron).^[9]

1960 - Bernard Widroff/Marcian Hoff. Desarrollaron el modelo Adaline (adaptive linear Elements). Esta fue la primera red neuronal aplicada a un problema real (filtros adaptativos para eliminar ecos en las líneas telefónicas) que se ha utilizado comercialmente durante varias décadas.

1961 - Karl Steinbeck: Die Lernmatrix. Red neuronal para simples realizaciones técnicas (memoria asociativa).

1969 - Marvin Minsky/Seymour Papert. En este año casi se produjo la “muerte abrupta” de las Redes Neuronales; ya que Minsky y Papert probaron (matemáticamente) que el Perceptron no era capaz de resolver problemas relativamente fáciles, tales como 7 el aprendizaje de una función no-lineal. Esto demostró que el Perceptron era muy débil, dado que las funciones no-lineales son extensamente empleadas en computación y en los problemas del mundo real.

1974 - Paul Werbos. Desarrolló la idea básica del algoritmo de aprendizaje de propagación hacia atrás (backpropagation); cuyo significado quedó definitivamente aclarado en 1985.

1977 - Stephen Grossberg: Teoría de Resonancia Adaptada (TRA). La Teoría de Resonancia Adaptada es una arquitectura de red que se diferencia de todas las demás previamente inventadas. La misma simula otras habilidades del cerebro: memoria a largo y corto plazo.

1985 - John Hopfield. Provocó el renacimiento de las redes neuronales con su libro: “Computación neuronal de decisiones en problemas de optimización.”

1986 - David Rumelhart/G. Hinton. Redescubrieron el algoritmo de aprendizaje de propagación hacia atrás (backpropagation).^[10]

A partir de 1986, el panorama fue alentador con respecto a las investigaciones y el desarrollo de las redes neuronales. En la actualidad, son numerosos los trabajos que se realizan y publican cada año, las aplicaciones nuevas que surgen (sobre todo en el área de control) y las empresas que lanzan al mercado productos nuevos, tanto hardware como software (sobre todo para simulación). Luego en 1990, se publica una nueva red llamada red de Elman.^[11]

5. Redes Neuronales Artificiales

Nuestro cerebro es un procesador de información capaz de procesar a gran velocidad grandes cantidades de información procedentes de los sentidos, combinarla o compararla con los conocimientos almacenados y dar respuestas adecuadas incluso en situaciones nuevas, características que son dignas de imitar. Aunque todavía se ignora mucho sobre la forma en que el cerebro aprende a procesar la información, se han desarrollado modelos que tratan de mimetizar tales habilidades; denominados Redes Neuronales Artificiales (RNA).

Consisten en un número de procesadores independientes y simples: las neuronas, cuya función es determinada por la estructura de la red. Las neuronas, conectadas en paralelo, se comunican con otras por medio de la fuerza de las conexiones ponderadas, los pesos sinápticos. Cada neurona acumula la entrada que recibe y genera una salida de acuerdo a una función de activación interna.

5.1. Ventajas que ofrecen las RN

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Entre las ventajas se incluyen:

- Aprendizaje Adaptativo. Capacidad de aprender a realizar tareas basadas en un entrenamiento o en una experiencia inicial.
- Auto-organización. Una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- Tolerancia a fallos. La destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.

5.2. Aprendizaje Adaptativo

La capacidad de aprendizaje adaptativo es una de las características más atractivas de redes neuronales. Esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni necesidad de especificar funciones de distribución de probabilidad. Las redes neuronales son sistemas dinámicos auto adaptativos. Son adaptables debido a la capacidad de autoajuste de los elementos procesales (neuronas) que componen el sistema. Son dinámicos, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones.

5.3. Auto-organización

Las redes neuronales emplean su capacidad de aprendizaje adaptativo para auto organizar la información que reciben durante el aprendizaje y/o la operación. Mientras que el aprendizaje es la modificación de cada elemento procesal, la auto organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico. Cuando las redes neuronales se usan para reconocer ciertas clases de patrones, ellas auto organizan la información usada. Por ejemplo, la red llamada backpropagation, creará su propia representación característica, mediante la cual puede reconocer ciertos patrones. Esta auto organización provoca la generalización: facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a las que no había sido expuesta anteriormente. El sistema puede generalizar la entrada para obtener una respuesta. Esta característica es muy importante cuando se tiene que solucionar problemas en los cuales la información de entrada no es muy clara; además permite que el sistema dé una solución, incluso cuando la información de entrada está especificada de forma incompleta.

5.4. Tolerancia a fallos

Las redes neuronales fueron los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Comparados con los sistemas 10 computacionales tradicionales, los cuales pierden su funcionalidad cuando sufren un pequeño error de memoria, en las redes neuronales, si se produce un fallo en un número no muy grande de neuronas y aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina. Hay dos aspectos distintos respecto a la tolerancia a fallos:

- Las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos. Esta es una tolerancia a fallos respecto a los datos.
- Las redes pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

La razón por la que las redes neuronales son tolerantes a los fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento. La mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos almacenan cada pieza de información en un espacio único, localizado y direccionable. En cambio, las redes neuronales almacenan información no localizada. Por lo tanto, la mayoría de las interconexiones entre los nodos de la red tendrán sus valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada.

5.5. Elementos básicos de una RN

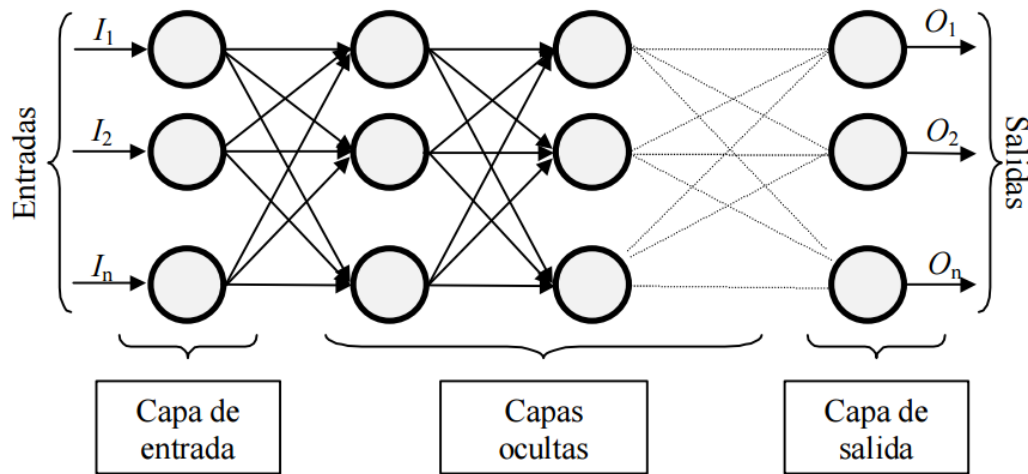


Ilustración 5-1 Capas de una RNA.

La misma está constituida por neuronas interconectadas y arregladas en tres capas (esto último puede variar). Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”. Cabe mencionar que la capa oculta puede estar constituida por varias capas.

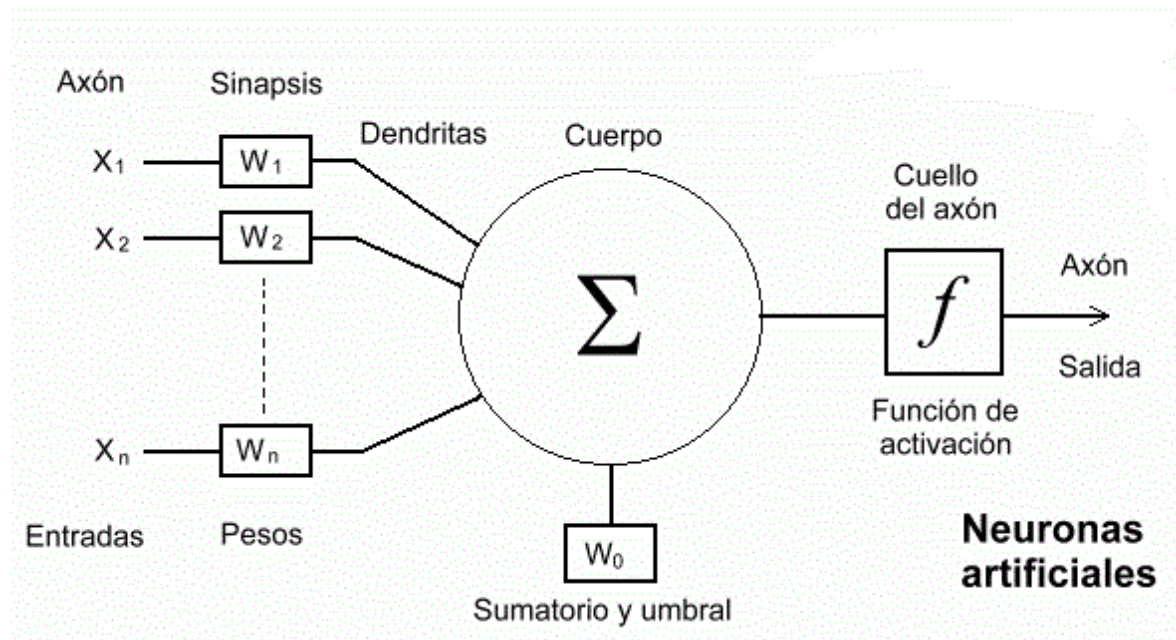


Ilustración 5-2 Estructura de una neurona artificial.

- **Función de red o de propagación:** Es quien se encarga de calcular el valor de base o entrada total a la unidad, generalmente corresponde a una simple suma ponderada de todas las entradas recibidas, es decir, sumatoria de las entradas multiplicadas por el peso o valor de las conexiones.
- **Función de Activación:** Es quizás la característica principal de una neurona artificial, la que mejor define el comportamiento de la misma. Se usan diferentes tipos de funciones, desde funciones simples a funciones no lineales. Este elemento es quien se encargará de calcular el nivel o estado de activación de la neurona en función de la entrada total.
- **Conexiones Ponderadas:** Estas conexiones son las encargadas de cumplir el rol de las conexiones sinápticas, el peso de la conexión equivale a la fuerza o efectividad de la sinapsis. El valor de los pesos y el signo de los mismos definen el tipo excitatorio/inhibitorio y la intensidad de la influencia.
- **Salida:** Se calcula la salida de la neurona a través de su función de activación.

5.6. Función de activación

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada); es decir, que tiene un “estado de activación”. Las neuronas artificiales también tienen diferentes estados de activación; algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado. La función activación calcula el estado de actividad de una neurona; transformando la entrada global en un valor de activación, cuyo rango normalmente va de (0 a 1) o de (-1 a 1). Esto es así, porque una neurona puede estar totalmente inactiva (0 o -1) o activa (1).

Función	Formula	Rango
Identidad	$y = x$	$[-\infty, \infty]$
Escalón	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$	$[0, 1]$
	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$	$[-1, 1]$
Lineal a tramos	$y = \begin{cases} x & \text{si } -1 \leq x \leq 1 \\ +1 & \text{si } x > 1 \\ -1 & \text{si } x < -1 \end{cases}$	$[-1, 1]$
Sigmoidea	$y = \frac{1}{1 + e^{-x}}$	$[0, 1]$
	$y = \tanh(x)$	$[-1, 1]$
Sinusoidal	$y = \text{Sen}(\omega \cdot x + \varphi)$	$[-1, 1]$

Ilustración 5-3 Funciones de Activación.

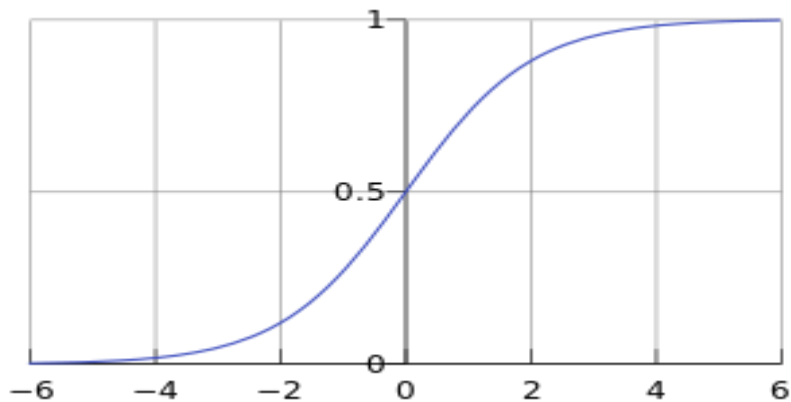


Ilustración 5-4 Función Sigmoide.

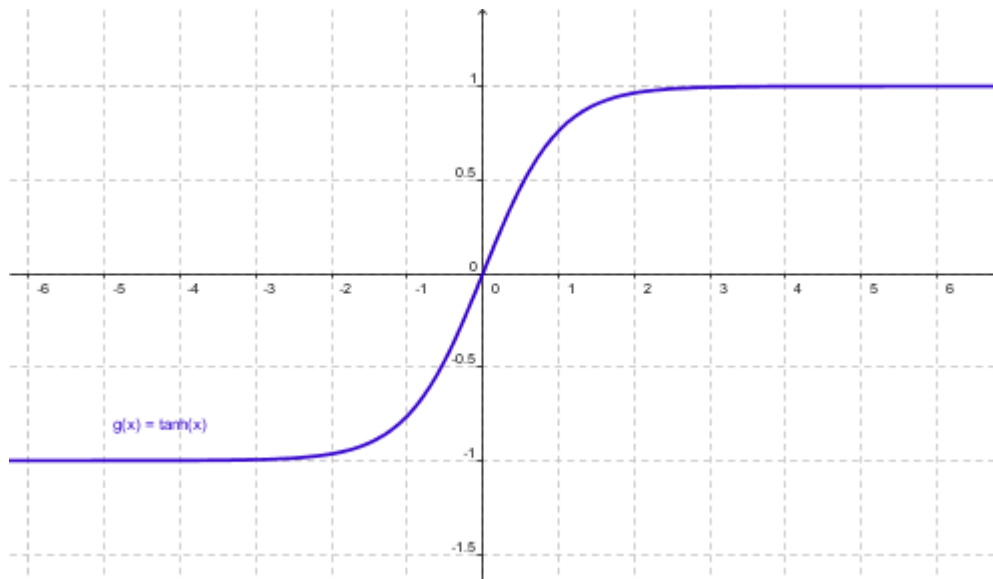


Ilustración 5-5 Función Tangente Hiperbólica.

5.7. Mecanismos de aprendizaje

Se ha visto que los datos de entrada se procesan a través de la red neuronal con el propósito de lograr una salida. También se dijo que las redes neuronales extraen generalizaciones desde un conjunto determinado de ejemplos anteriores de tales problemas de decisión. Una red neuronal debe aprender a calcular la salida correcta para cada constelación (arreglo o vector) de entrada en el conjunto de ejemplos. Este proceso de aprendizaje se denomina: proceso de entrenamiento. El conjunto de datos (o conjunto de ejemplos) sobre el cual este proceso se basa es, por ende, llamado: conjunto de datos de entrenamiento. Si la topología de la red y las diferentes funciones de cada neurona (entrada, activación y salida) no pueden cambiar durante el aprendizaje, mientras que los pesos sobre cada una de las conexiones si pueden hacerlo; el aprendizaje de una red neuronal significa

adaptación de los pesos. En otras palabras, el aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el mismo se reducen a la destrucción, modificación y creación de conexiones entre las neuronas. En los sistemas biológicos existe una continua destrucción y creación de conexiones entre las neuronas. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero. De la misma manera, una conexión se destruye cuando su peso pasa a ser cero. Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, por lo tanto, se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables.

Un aspecto importante respecto al aprendizaje de las redes neuronales es el conocer cómo se modifican los valores de los pesos, es decir, cuáles son los criterios que se siguen para cambiar el valor asignado a las conexiones cuando se pretende que la red aprenda una nueva información. Hay dos métodos de aprendizaje importantes que pueden distinguirse:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.

Otro criterio que se puede utilizar para diferenciar las reglas de aprendizaje se basa en considerar si la red puede aprender durante su funcionamiento habitual o si el aprendizaje supone la desconexión de la red, es decir, su inhabilitación hasta que el proceso termine. En el primer caso, se trataría de un aprendizaje on line, mientras que el segundo es lo que se conoce como off line. Cuando el aprendizaje es off line, se distingue entre una fase de aprendizaje o entrenamiento y una fase de operación o funcionamiento, existiendo un conjunto de datos de entrenamiento y un conjunto de datos de test o prueba, que serán utilizados en la correspondiente fase. Además, los pesos de las conexiones permanecen fijos después que termina la etapa de entrenamiento de la red. Debido precisamente a su carácter estático, estos sistemas no presentan problemas de estabilidad en su funcionamiento.

5.8 Aprendizaje supervisado

Este método es mediante un entrenamiento controlado por un agente externo (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos de las conexiones, con el fin de conseguir que la salida obtenida se aproxime a la deseada. En este tipo de aprendizaje se suelen considerar, a su vez, tres formas de llevarlo a cabo, que dan lugar a los siguientes aprendizajes supervisados:

- Número fijo de ciclos: Se define a priori cierto número de iteraciones, cuando se cumple dicho número se da por terminada la fase de aprendizaje.
- Error descienda por debajo de una cantidad preestablecida: Se define una función de error a nivel de patrón individual y a nivel de la totalidad del conjunto de

entrenamiento. Se determina a priori un valor aceptable de error, cuando la red produzca un valor de error por debajo del prefijado se detiene el entrenamiento.

- Modificación de los pesos sea irrelevante: Cuando se llega a un momento en que las variaciones de los valores de los pesos no sean significativas, se detiene la fase de entrenamiento.

5.9 Aprendizaje No Supervisado

Las redes neuronales artificiales con aprendizaje no supervisado son aquellas que no necesitan de un supervisor externo para realizar su aprendizaje. Estas redes son capaces de ajustar sus parámetros internamente, adaptándose al entorno de la mejor manera posible. La no supervisión conlleva a que la red deba descubrir por sí sola características, regularidades, correlaciones o categorías en los datos de entrada, y se obtengan de forma codificada a la salida.

Las arquitecturas de las redes con aprendizaje no supervisado suelen ser bastante simple. Muchas de ellas tienen una sola capa; siendo en su mayoría Feed-Forward. A las redes con aprendizaje no supervisado también se le conoce como sistemas auto organizados, debido a que la red se ajusta dependiendo únicamente a los valores recibidos como entrada.

Las redes con aprendizaje no supervisado (también conocido como auto supervisado) no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta. Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten en su entrada. Existen varias posibilidades en cuanto a la interpretación de la salida de estas redes, que dependen de su estructura y del algoritmo de aprendizaje empleado. En algunos casos, la salida representa el grado de familiaridad o similitud entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado). En otro caso, podría realizar una clusterización (clustering) o establecimiento de categorías, indicando la red a la salida a qué categoría pertenece la información presentada a la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de las correlaciones entre las informaciones presentadas.

En cuanto a los algoritmos de aprendizaje no supervisado, en general se suelen considerar dos tipos, que dan lugar a los siguientes aprendizajes:

- Aprendizaje hebbiano.
- Aprendizaje competitivo y comparativo.
- Aprendizaje hebbiano.

Esta regla de aprendizaje es la base de muchas otras, la cual pretende medir la familiaridad o extraer características de los datos de entrada. El fundamento es una

suposición bastante simple: si dos neuronas N_i y N_j toman el mismo estado simultáneamente (ambas activas o ambas inactivas), el peso de la conexión entre ambas se incrementa. Las entradas y salidas permitidas a la neurona son: $\{-1, 1\}$ o $\{0, 1\}$ (neuronas binarias). Esto puede explicarse porque la regla de aprendizaje de Hebb se originó a partir de la neurona biológica clásica, que solamente puede tener dos estados: activa o inactiva.

Aprendizaje competitivo y comparativo. Se orienta a la clusterización o clasificación de los datos de entrada. Como característica principal del aprendizaje competitivo se puede decir que, si un patrón nuevo se determina que pertenece a una clase reconocida previamente, entonces la inclusión de este nuevo patrón a esta clase matizará la representación de la misma. Si el patrón de entrada se determinó que no pertenece a ninguna de las clases reconocidas anteriormente, entonces la estructura y los pesos de la red neuronal serán ajustados para reconocer la nueva clase. Elección del conjunto inicial de pesos.

Antes de comenzar el proceso de entrenamiento se debe determinar un estado inicial, lo que significa: escoger un conjunto inicial de pesos para las diversas conexiones entre las neuronas de la red neuronal. Esto puede realizarse por varios criterios; por ejemplo, uno de ellos es otorgar un peso aleatorio a cada conexión, encontrándose los mismos dentro de un cierto intervalo. Generalmente un intervalo del tipo $[-n, n]$, donde n es un número natural positivo. Cabe mencionar que durante el transcurso del entrenamiento los pesos no se encuentran restringidos a dicho intervalo.

Detención del proceso de aprendizaje. Para determinar cuándo se detendrá el proceso de aprendizaje, es necesario establecer una condición de detención. Normalmente el entrenamiento se detiene cuando el cálculo del error cuadrado sobre todos los ejemplos de entrenamiento ha alcanzado un mínimo o cuando para cada uno de los ejemplos dados, el error observado está por debajo de un determinado umbral. Ya que, para controlar este proceso, la mayor parte de las herramientas de las redes neuronales muestran estos errores utilizando gráficos especiales; los cuales no son utilizados para el aprendizaje, sino que solamente para dar un indicio del proceso en sí mismo. Otra condición de detención del aprendizaje puede ser cuando un cierto número de ciclos y/o pasos de entrenamiento hayan sido completamente corridos. Luego de alcanzarse la condición de detención, los pesos no se volverán a cambiar.

Entonces podemos decir que la transformación de los datos de entrada a los de salida está resuelta. Esto se puede interpretar como una función f oculta en el conjunto de la red neuronal. Esta función es exactamente la “instrucción” de cómo la salida será calculada a partir de una constelación (vector) de entrada. El orden en que los ejemplos de entrenamiento se presentan a la red neuronal es otro tema importante. En general se ha observado que en la mayoría de los casos es beneficioso realizarlo en forma aleatoria.

5.10 Algoritmo Levenberg-Marquardt

Se formuló inicialmente en 1944 por Kenneth Levenberg, EE.UU., 1919-1973, y se desarrolló en 1963 por Donald W. Marquardt. El algoritmo es una derivación del método Gauss-Newton para encontrar el mínimo de una función $F(x)$ que es la suma de mínimos cuadrados de funciones no lineales.

$$F(x) = \frac{1}{2} \sum_{i=1}^m [f_i(x)]^2$$

Sea la Jacobiana de $f(x)$ ser denotado $J(x)$, entonces el método de Levenberg-Marquardt busca en la dirección p dada por la solución a las ecuaciones.

$$(J_k^T J_k + \lambda_k I) p_k = -J_k^T f_k$$

Donde λ_k son escalares no negativos y I es la matriz identidad. El método tiene la propiedad que un escalar Δ está relacionado con λ_k donde el vector p_k es la solución local del problema de minimizar $\|J_k p + f_k\|_2^2 / 2$ a $\|p\|_2 \leq \Delta$. [1]

5.11 Métricas de Rendimiento

5.11.1 Root Mean Square Error

El Root Mean Square Error (RMSE) es usado para medir las diferencias entre los valores pronosticados por un modelo y los valores observados realmente. El RMSE representa la desviación estándar de la muestra entre los datos observados y los datos pronosticados. La forma de calcular el RMSE es realizar raíz cuadrada al MSE el cual es la sumatoria del promedio de la diferencia de los datos observados y los pronosticados al cuadrado, la cual se representa por la siguiente formula.

$$RMSE = \sqrt{\sum \frac{(Y_{Pred} - Y_{ref})^2}{N}}$$

5.11.2 Mean Absolute Error

El Mean Absolute Error (MAE) representa la diferencia absoluta entre los datos observados y los datos pronosticados, se calcula de manera muy similar al MSE, o sea, es el promedio de los errores, es decir, la diferencia entre la data observada y la data pronosticada, pero en este caso se realiza en valor absoluto, esto se ve representado por la siguiente formula.

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$

5.11.3 Coeficiente de Determinación

En estadística, el coeficiente de determinación, denominado R^2 y pronunciado “R cuadrado”, es un estadístico usado en el contexto de un modelo estadístico cuyo principal propósito es predecir futuros resultados o probar una hipótesis. El coeficiente determina la calidad del modelo para replicar los resultados, y la proporción de variación de los resultados que puede explicarse por el modelo,

Hay varias definiciones diferentes para R^2 que son algunas veces equivalentes. Las más comunes se refieren a la regresión lineal. En este caso, el R^2 es simplemente el cuadrado del coeficiente de correlación de Pearson, lo cual es sólo cierto para la regresión lineal simple, el R^2 puede tomar valores entre 0 y 1.

$$R^2 = \frac{\theta_{XY}^2}{\theta_X^2 \theta_Y^2}$$

Donde:

- θ_{XY} es la covarianza de (X, Y)
- θ_X es la desviación típica de la variable X.
- θ_Y es la desviación típica de la variable Y.

6. Red Neuronal Elman

La red neuronal de Elman está constituida por una capa de sensores o unidades de entrada, dos capas intermedias de unidades (una capa de unidades ocultas y otra capa de unidades de contexto) y una capa de unidades de salida. Los sensores de entrada y las unidades de salida recogen la información del entorno, es decir, la suministrada por el conjunto de patrones de entrenamiento, mientras que las demás no. Las unidades de salida reciben la señal suministrada por la salida de las unidades de la capa oculta ponderadas por los correspondientes pesos sinápticos y suelen utilizar como función de transferencia la función identidad. Sin embargo, la novedad de este modelo la introducen las unidades de contexto que se utilizan para memorizar las salidas de las unidades ocultas en la etapa anterior, de manera que cada unidad de contexto tiene como salida la salida de la unidad oculta correspondiente en la etapa anterior. Por lo tanto, esta red es sólo parcialmente recurrente y cada unidad de proceso oculta recibe como entrada las salidas de las unidades de contexto y de los sensores de entrada ponderadas por sus pesos sinápticos. De esta manera la salida de la red depende no sólo del patrón de entrada actual sino también de los patrones anteriores a través de las unidades de contexto. Las unidades de proceso de la capa oculta tienen como función de transferencia la función logística, la función tangente hiperbólica o la función identidad.^[11]

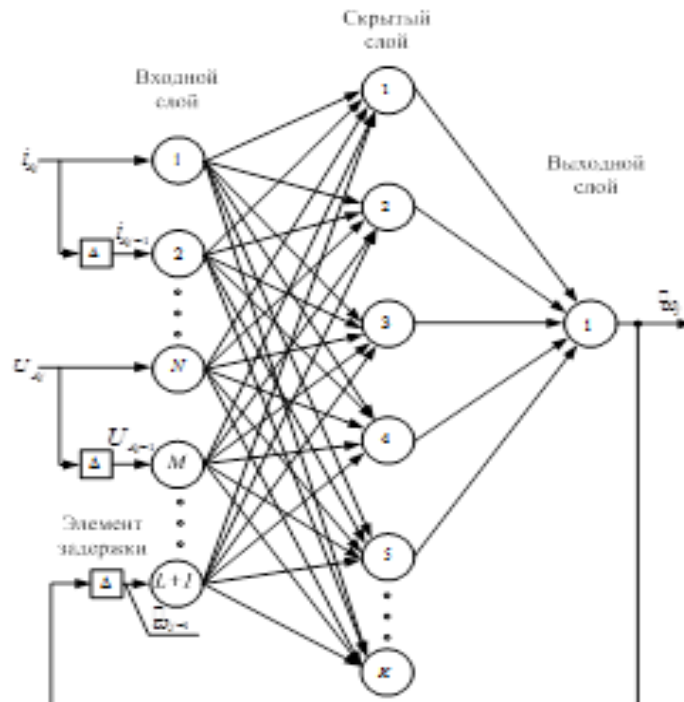


Ilustración 6-1 Arquitectura Red Elman

7. Herramienta a Utilizar

7.1. Matlab

Es una herramienta de software matemático que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio. Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones. Es un software muy usado en universidades y centros de investigación y desarrollo.

7.2. Interfaz de trabajo

Se desarrolló una interfaz la cual permite al usuario abstraerse del código, para que así el usuario solo se dedique comprender sobre la resolución del problema y no del “como” desarrollarla.

7.3. Capturas de la Interfaz

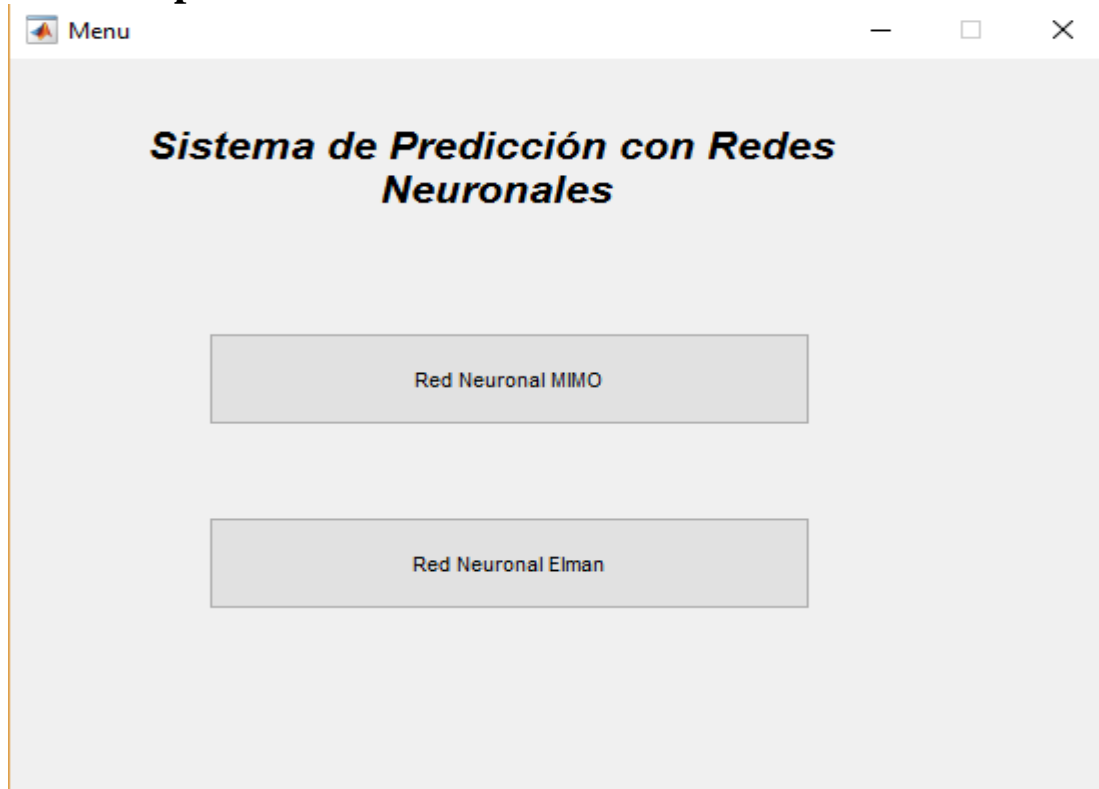


Ilustración 7-1 Interfaz principal de Menú

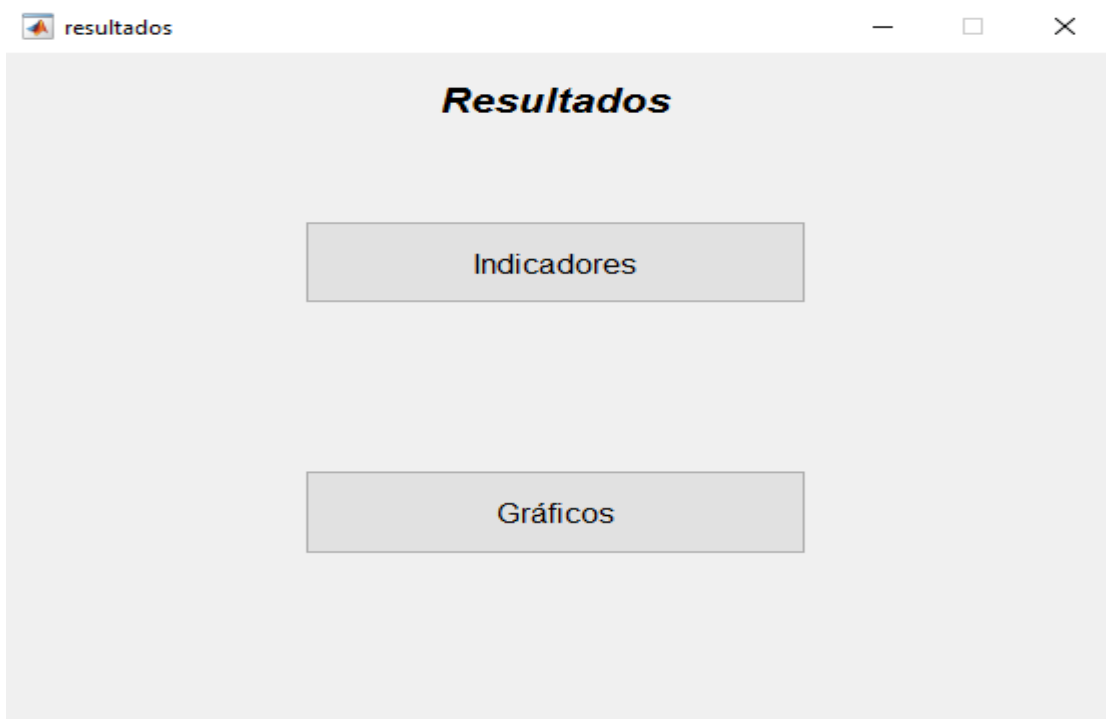


Ilustración 7-2 Interfaz de Resultados

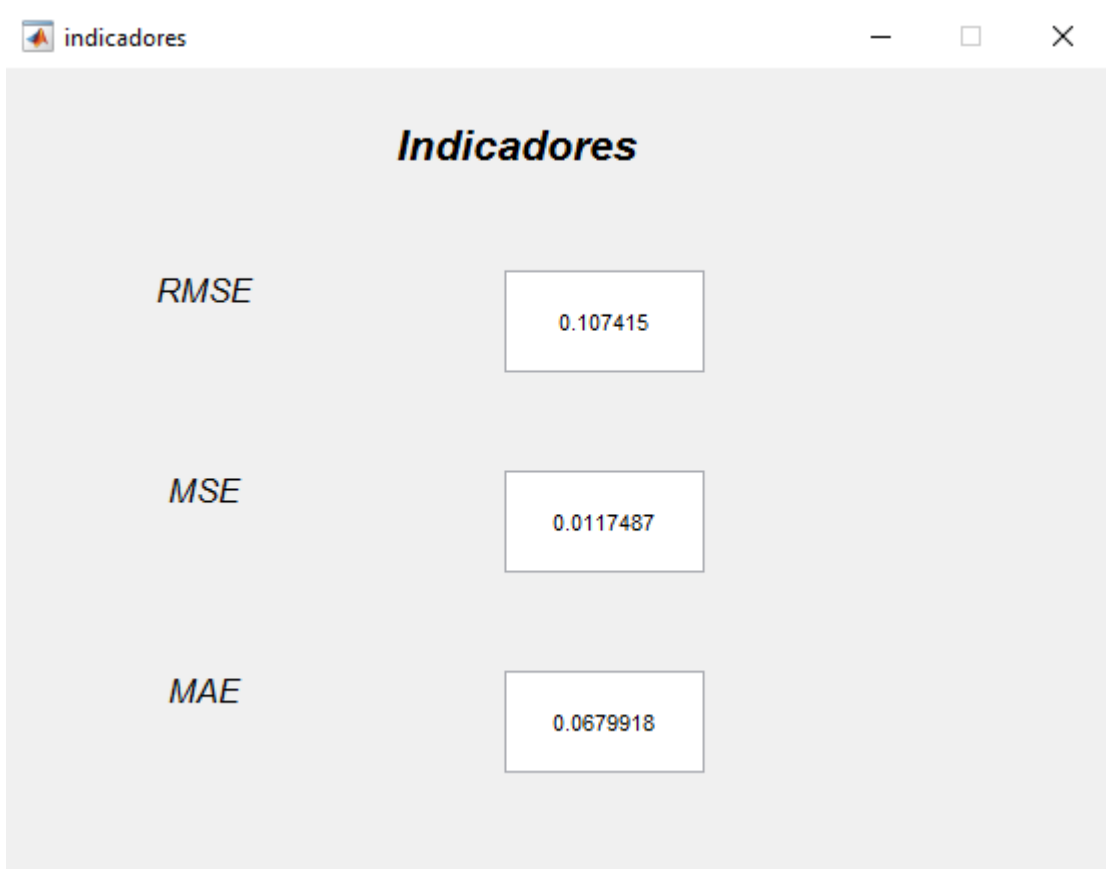


Ilustración 7-3 Interfaz de Indicadores

8. Pruebas Realizadas

8.1. Introducción

Se realizaron pruebas con la interfaz mencionada en el capítulo anterior, en el cual las pruebas se modificaron los porcentajes de entrenamiento, número de neuronas ocultas para ver cuál sería la topología adecuada. En esta entrega se pondrán los resultados en una tabla solo con el MSE y el R2, luego se podrá los gráficos del peor y mejor escenario de las pruebas realizadas

8.2. Seis Meses de horizonte (H = 6)

8.2.1. Porcentaje de entrenamiento 70% (M = 30)

Tabla 1 Datos H6 M30 70%

Neuronas Ocultas	MSE	R2
1	0.01056690	66.29 %
2	0.00476187	80.03%
3	0.00267873	92.25%
4	0.00104246	96.70%
5	0.00042365	98.59%
6	0.00466639	84.93%
7	0.00719679	80.91%

Resultados: En las pruebas realizadas con 70% de datos de entrenamiento se obtuvo un 66.29% de R2 en el peor de los casos, la cual se dio con la topología 30-1-6, por otro lado, el mejor resultado se dio con la siguiente topología 30-5-6, la cual obtuvo un 98.59% de R2.

8.2.2. Porcentaje de entrenamiento 80% (M = 30)

Tabla 2 Datos H6 M30 80%

Neuronas Ocultas	MSE	R2
1	0.01208530	57.74%
2	0.00709963	75.07%
3	0.00248724	91.03%
4	0.00143718	96.57%
5	0.00171005	94.12%
6	0.00076213	96.79%
7	0.00821703	71.04%
8	0.00558940	81.49%

Resultados: En estas pruebas realizadas se utilizó el 80% de los datos para la etapa de entrenamiento de la red, logrando como peor resultado un 57.74% de R2 en la topología 30-1-6, pero llegando a un 96.79% de R2 en su mejor resultado con la topología 30-6-6.

8.2.3. Porcentaje de entrenamiento 70% (M = 32)

Tabla 3 Datos H6 M32 70%

Neuronas Ocultas	MSE	R2
1	0.01093620	61.96%
2	0.00592822	78.48%
3	0.00233397	91.42%

Resultados: En esta etapa se utilizaron el 70% de los datos para la etapa de entrenamiento y además se cambiaron los datos de entrada a 32, llegando al peor escenario el cual fue de 61.96% de R2 en la topología 32-1-6, pero en su mejor escenario logro 91.42% de R2 con la topología 32-3-6.

8.2.4. Peor Resultado H6

De las pruebas realizadas se pidió señalar el peor y el mejor resultado encontrado para el problema, en este caso específico de 6 meses como horizonte de predicción el peor resultado se dio con un porcentaje de entrenamiento de 80% y la topología de la red de 30 nodos de entrada, 1 nodo oculto y 6 de salida (30, 1,6).

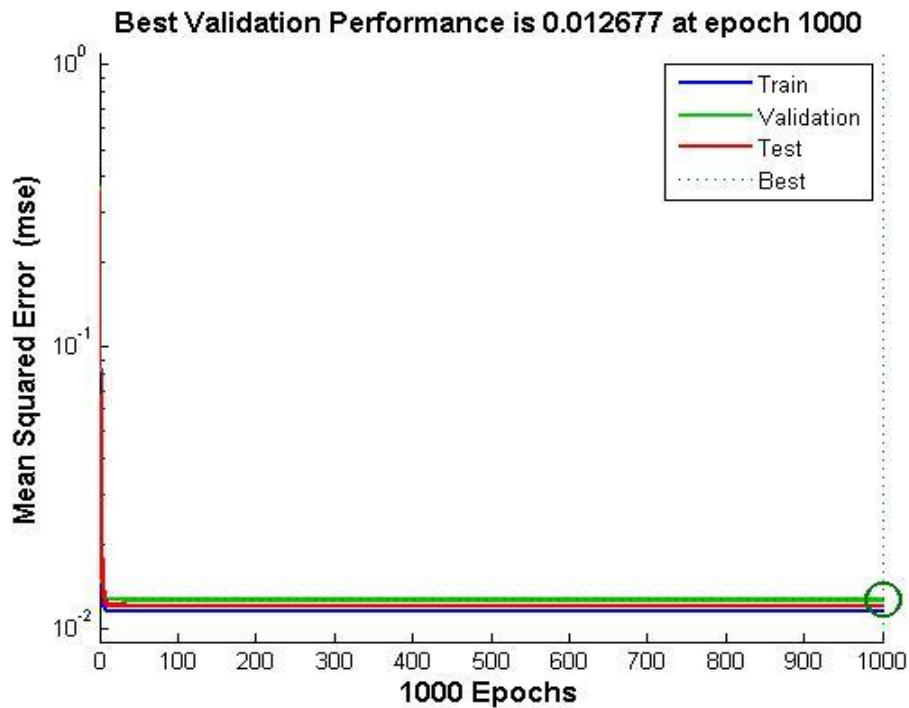


Ilustración 8-1 Performance

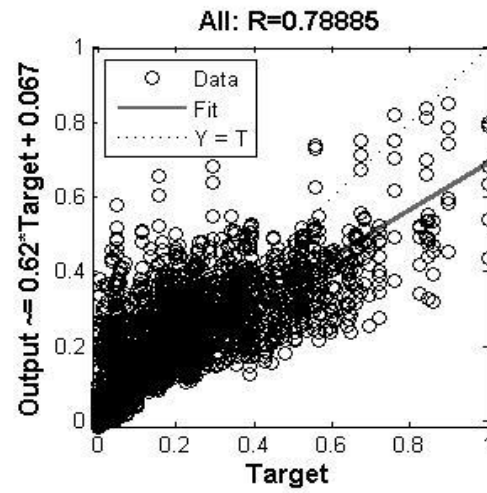
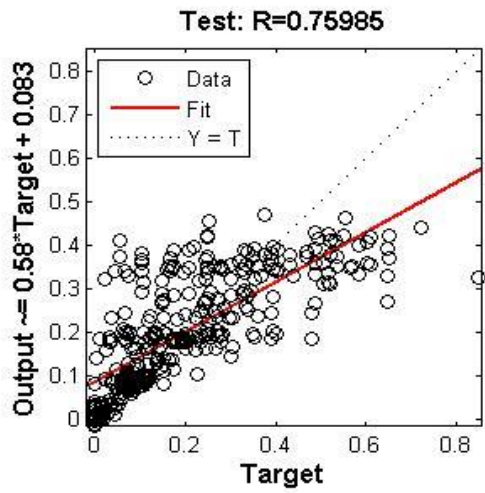
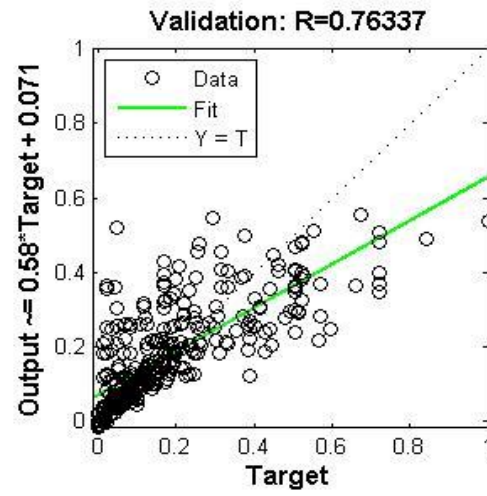
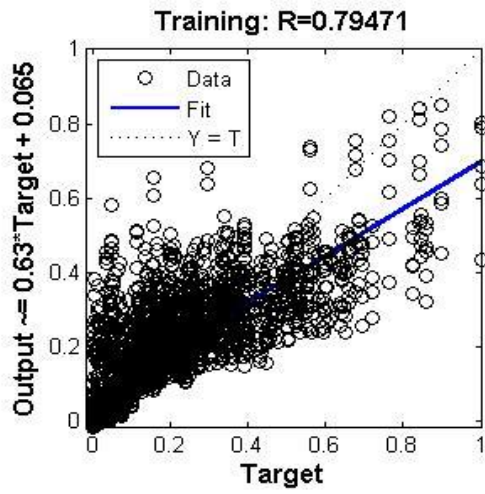


Ilustración 8-2 Regresión

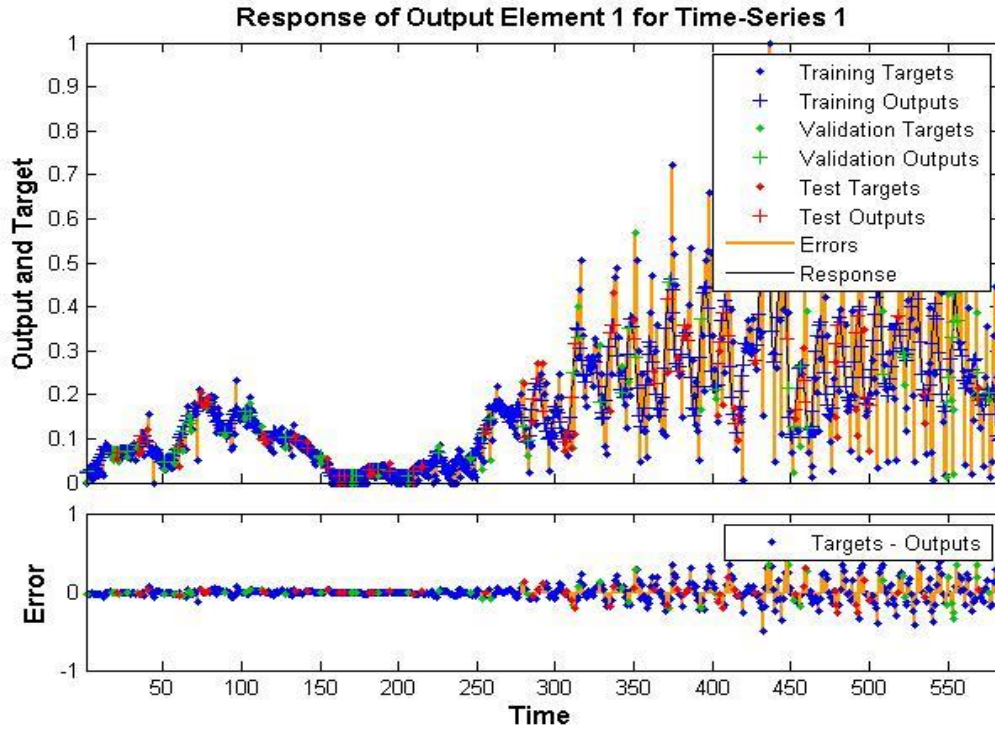


Ilustración 8-3 Output vs Target

8.2.5. Mejor Resultado H6

De las pruebas realizadas se pidió señalar el peor y el mejor resultado encontrado para el problema, en este caso específico de 6 meses como horizonte de predicción el peor resultado se dio con un porcentaje de entrenamiento de 70% y la topología de la red de 30 nodos de entrada, 5 nodo oculto y 6 de salida (30, 5,6).

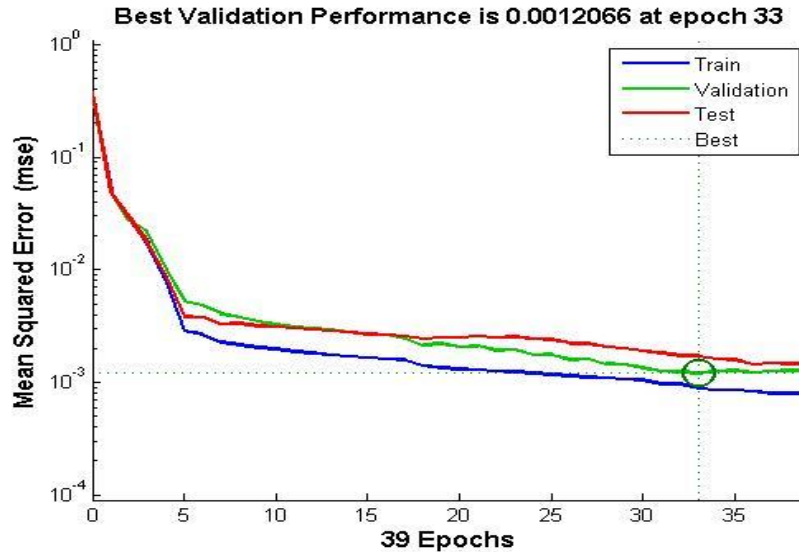


Ilustración 8-4 Performance H6 Mejor

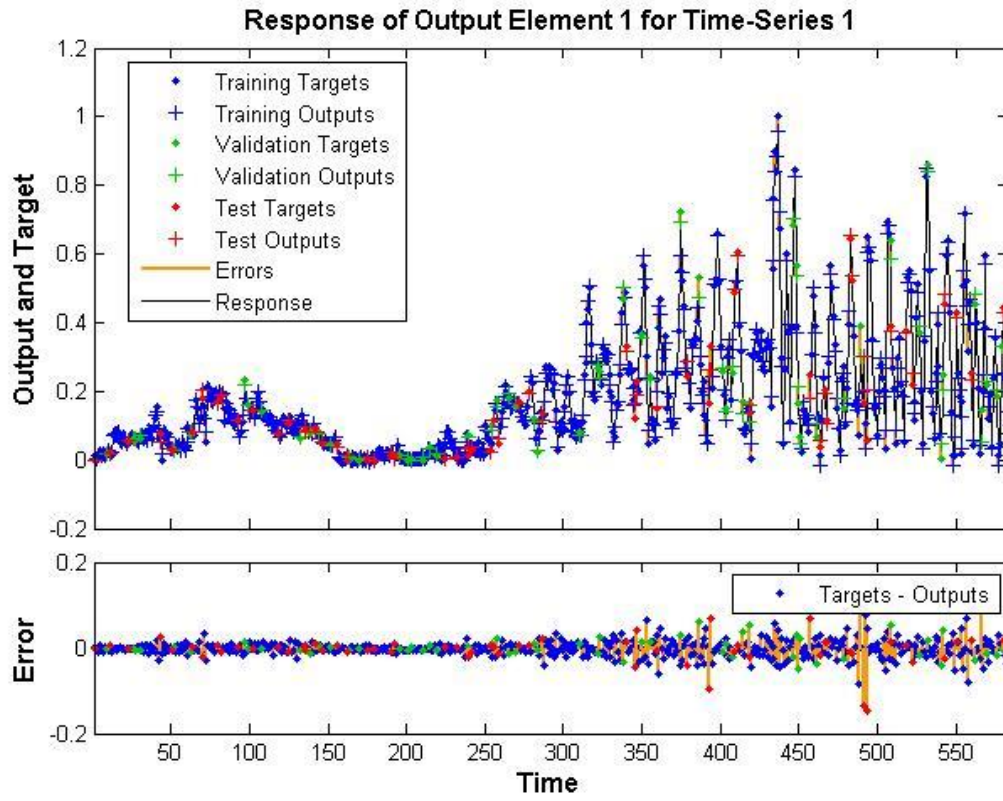


Ilustración 8-5 Output vs Target Mejor H6

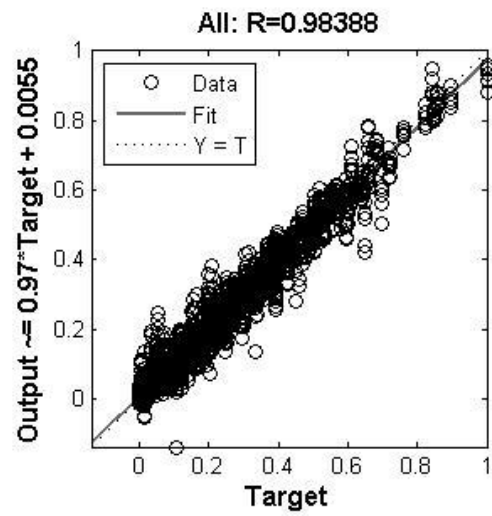
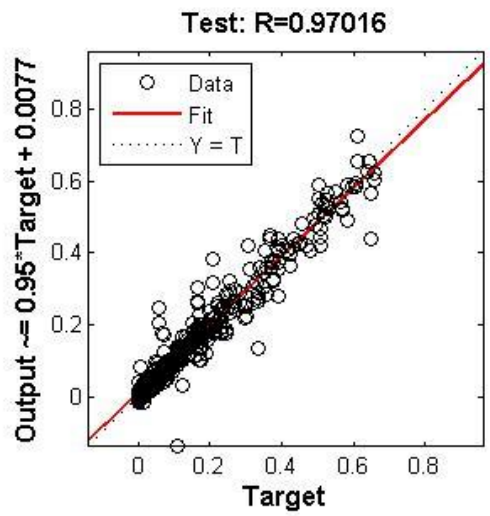
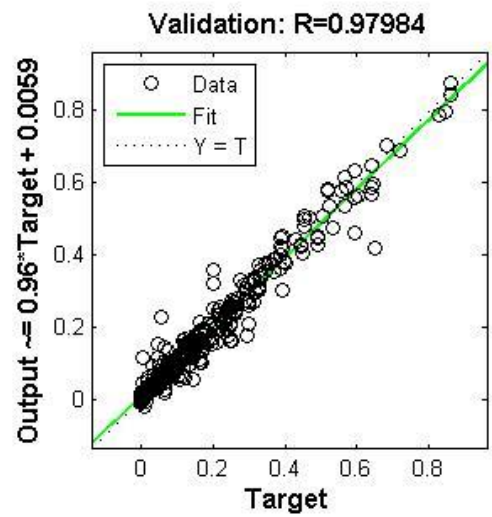
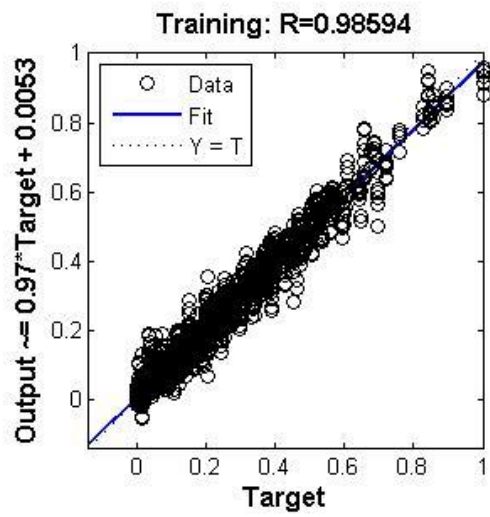


Ilustración 8-6 Regresión Mejor H6

8.3. Nueve Meses de horizonte (H = 9)

8.3.1. Porcentaje de entrenamiento 70% (M = 30)

Tabla 4 Datos H9 M30 70%

Neuronas Ocultas	MSE	R2
1	0.01299340	54.45%
2	0.00951488	73.71%
3	0.00473946	84.38%
4	0.00314565	91.23%
5	0.00175823	93.74%

Resultados: En este caso para 9 meses de horizonte se utilizó el 70% de los datos para etapa de entrenamiento, se llegó a un 54.45% de R2 como peor escenario con la topología 30-1-9, pero logro llegar a 93.74% de R2 en su mejor escenario con la topología 30-5-9.

8.3.2. Porcentaje de entrenamiento 80% (M = 30)

Tabla 5 Datos H9 M30 80%

Neuronas Ocultas	MSE	R2
1	0.01325180	51.19%
2	0.01017640	68.08%
3	0.00602086	80.71%

Resultados: En este caso para 9 meses de horizonte se utilizó el 80% de los datos para etapa de entrenamiento, se llegó a un 51.19% de R2 como peor escenario con la topología 30-1-9, pero logro llegar a 80.71% de R2 en su mejor escenario con la topología 30-3-9.

8.3.3. Porcentaje de entrenamiento 70% (M = 32)

Tabla 6 Datos H9 M32 70%

Neuronas Ocultas	MSE	R2
1	0.01320520	48.26%
2	0.01181190	70.43%

Resultados: En este caso para 9 meses de horizonte se utilizó el 80% de los datos para etapa de entrenamiento y además se cambió los nodos de entrada a 32, se llegó a un 48.26% de R2 como peor escenario con la topología 32-1-9, pero logro llegar a 70.43% de R2 en su mejor escenario con la topología 32-2-9.

8.3.4. Peor resultado H9

Habiendo realizado las pruebas con un horizonte de 9 meses se llegó a que el peor resultado mostrado es un numero de nodos de entrada 32, 70% entrenamiento, 1 nodo oculto y 9 salidas. (32,1,9).

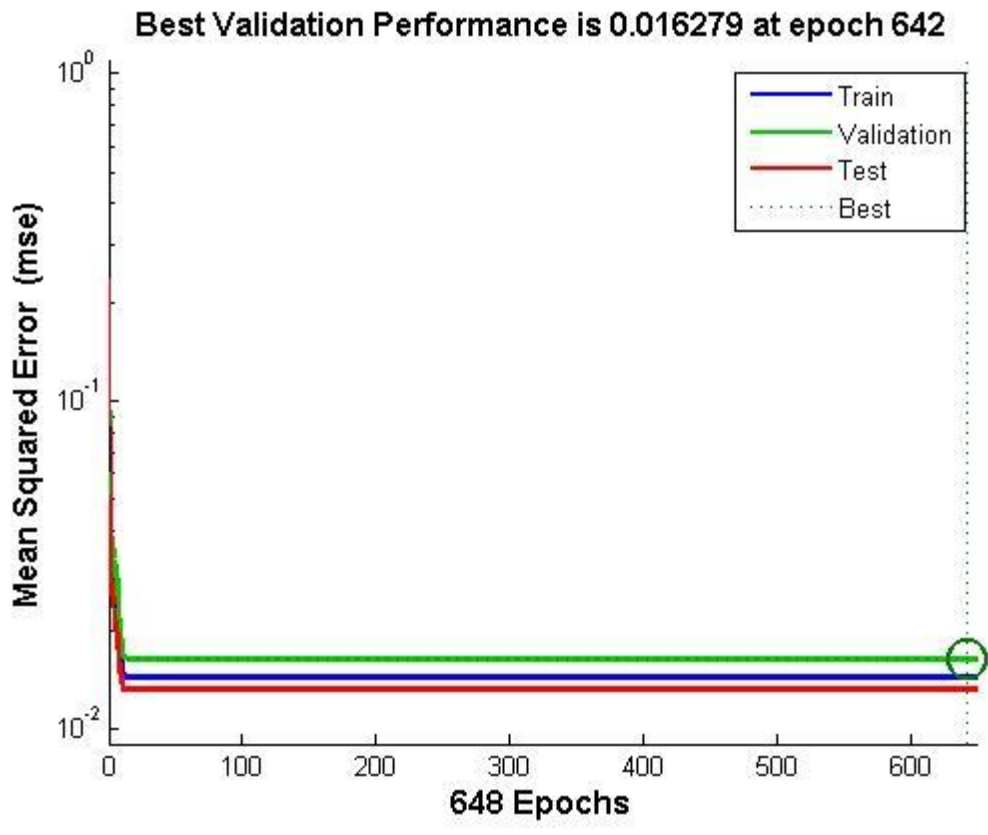


Ilustración 8-7 Performance Peor H9

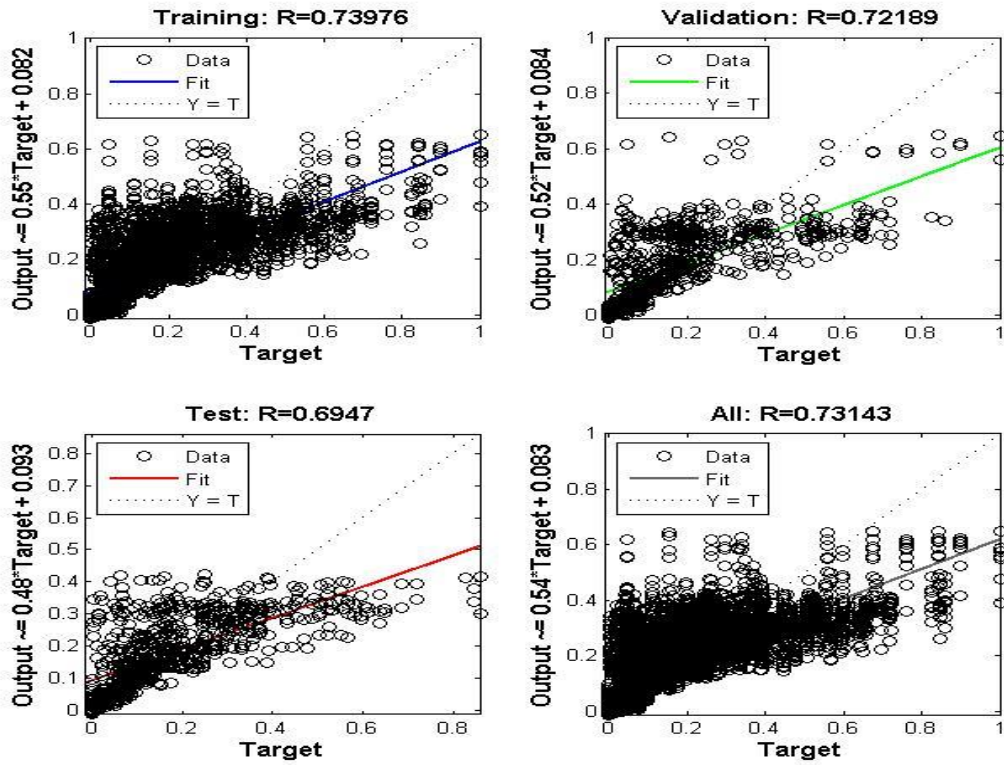


Ilustración 8-8 Regresión Peor H9

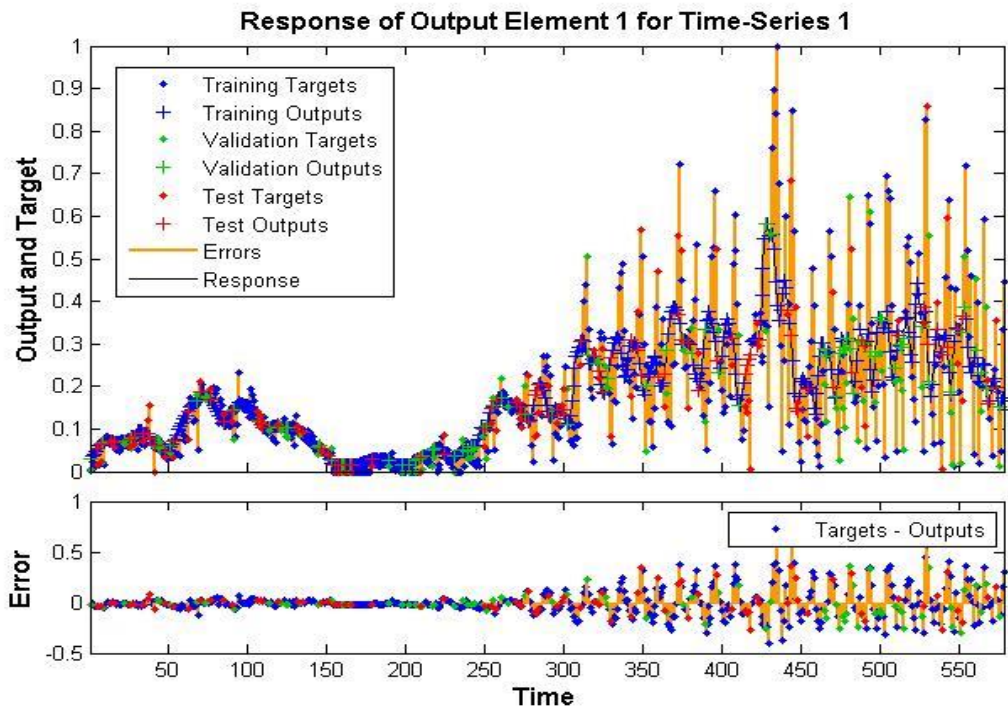


Ilustración 8-9 Output vs Target Peor H9

8.3.5. Mejor resultado H9

En las pruebas realizadas se destacó la siguiente topología 70% de entrenamiento, 30 nodos de entrada, 5 nodos ocultos y 9 salidas.

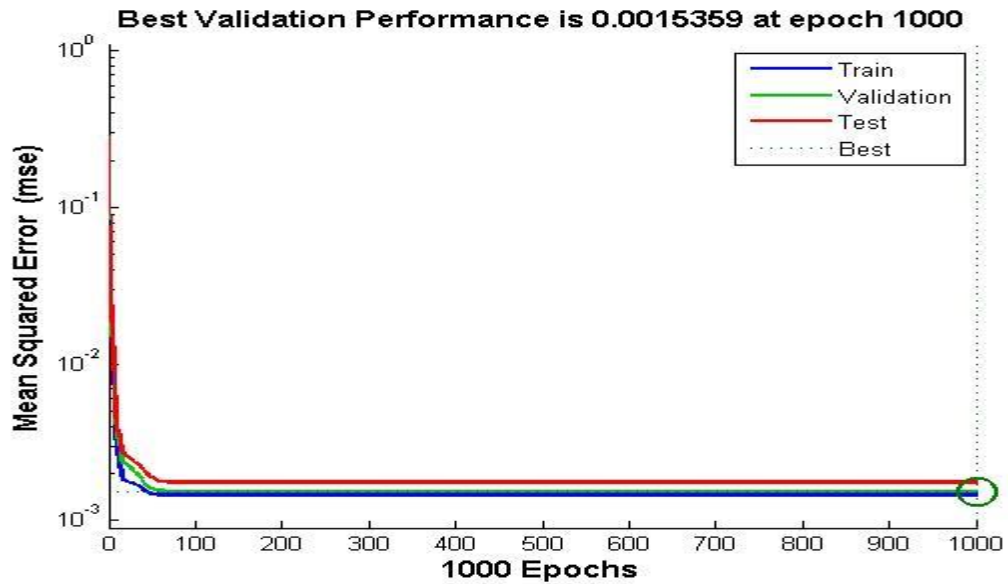


Ilustración 8-10 Performance Mejor H9

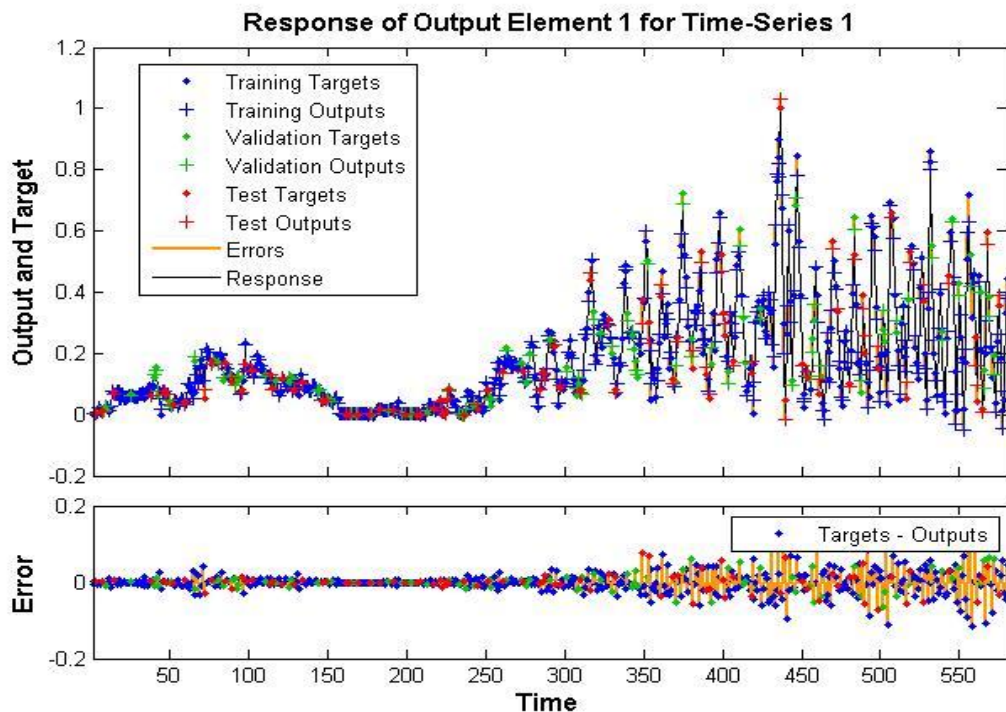


Ilustración 8-11 Output vs Target Mejor H9

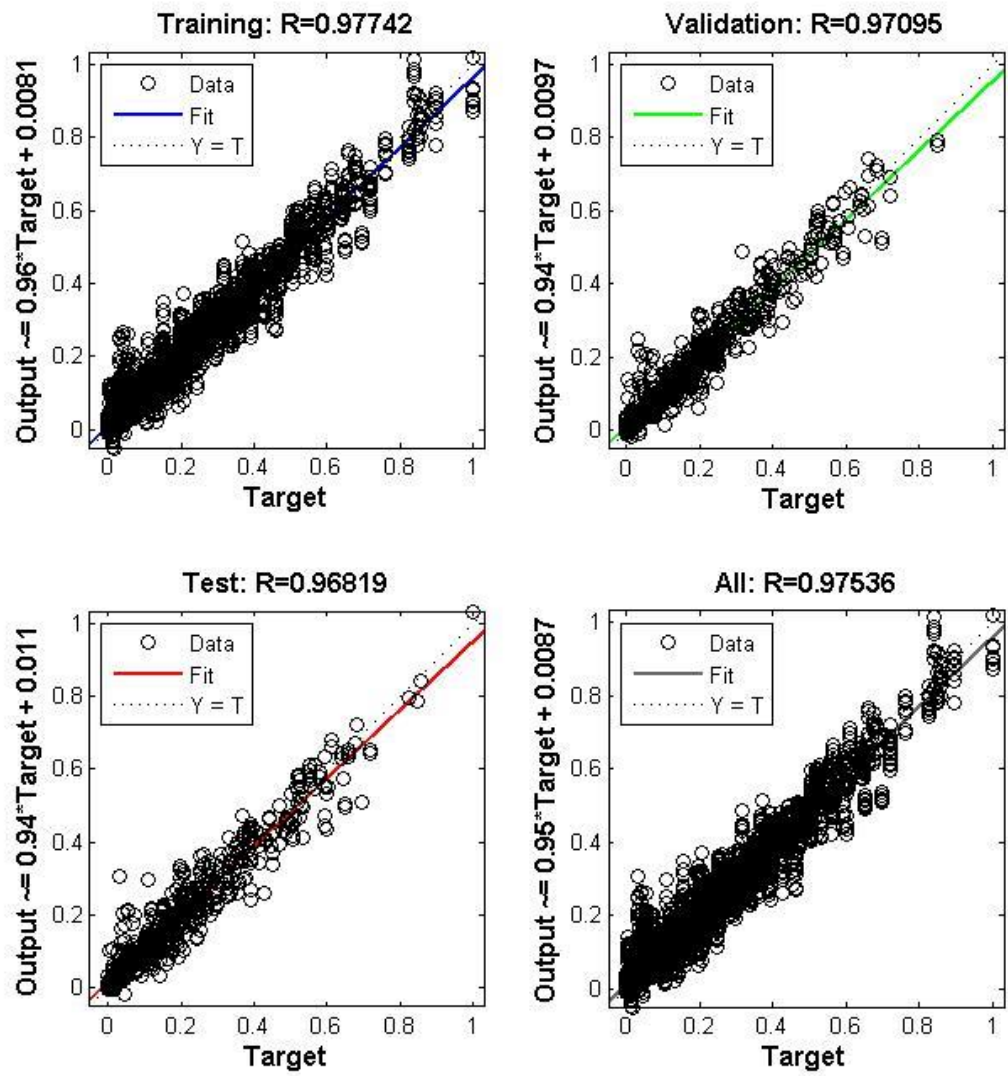


Ilustración 8-12 Regresión Mejor H9

8.4. Doce Meses de horizonte (H = 12)

Tabla 7 Datos M30 H12 70%

Neuronas Ocultas	MSE	R2
1	0.01293030	48.82%
2	0.11397900	60.24%
3	0.00633829	74.28%
4	0.00575773	81.65%

Resultados: En este caso para 12 meses de horizonte se utilizó el 70% de los datos para etapa de entrenamiento y se llegó a un 48.82% de R2 como peor escenario con la topología 30-1-12, pero logro llegar a 81.65% de R2 en su mejor escenario con la topología 30-4-12.

Observaciones: Debido al volumen de datos manejado por la red neuronal en este caso ya que el set de datos era de 60x607 además se le aplico un lag de 30, por lo tanto, estas pruebas se demoraban bastantes más horas que las de menor horizonte.

8.4.1. Mejor resultado H12

La mejor topología encontrada tras realizar las pruebas es de la de 30 nodos de entrada, 70% entrenamiento, 4 nodos ocultos y 12 nodos de salida.

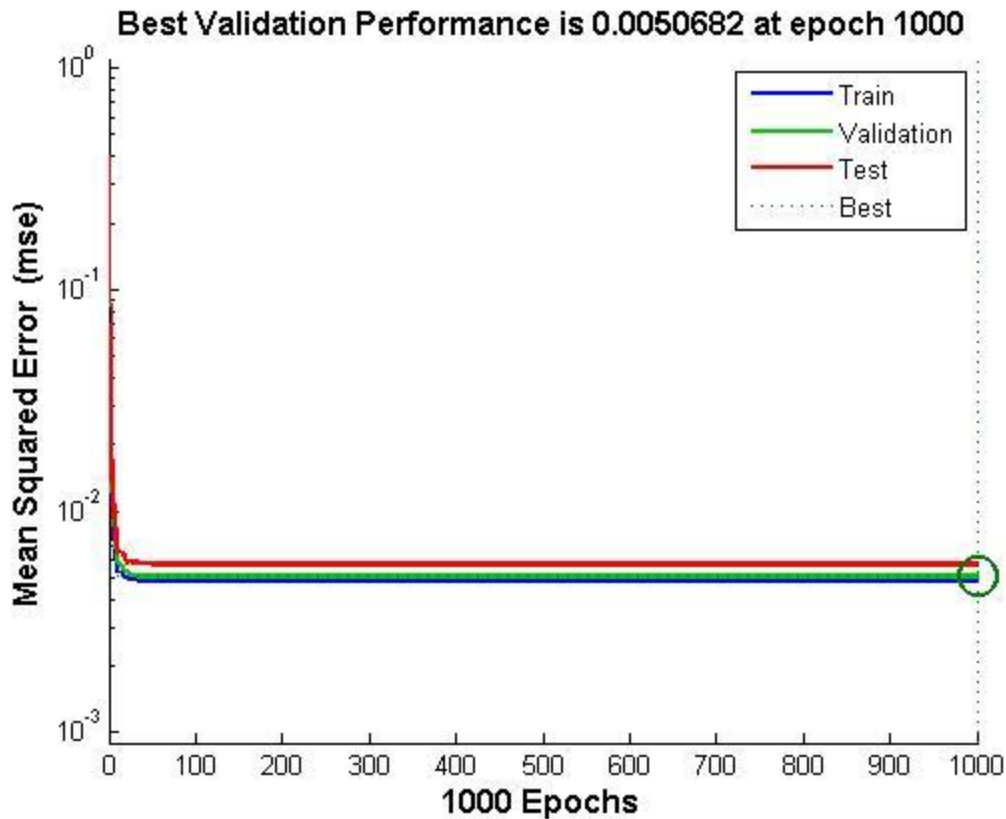


Ilustración 8-13 Performance Mejor H12

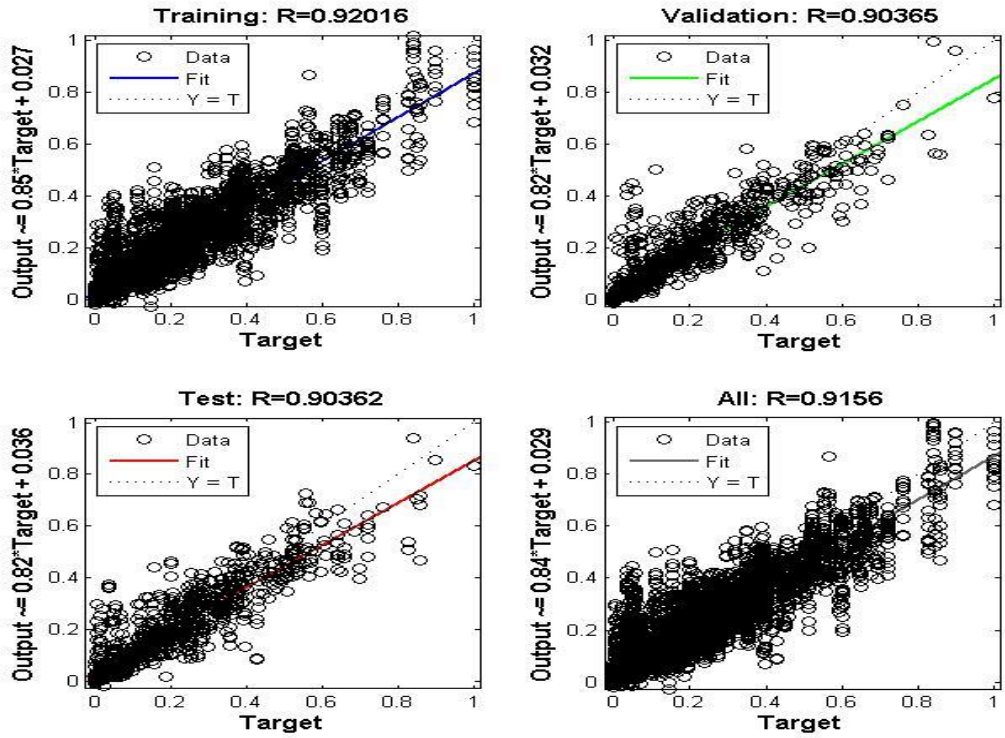


Ilustración 8-14 Regresión Mejor H12

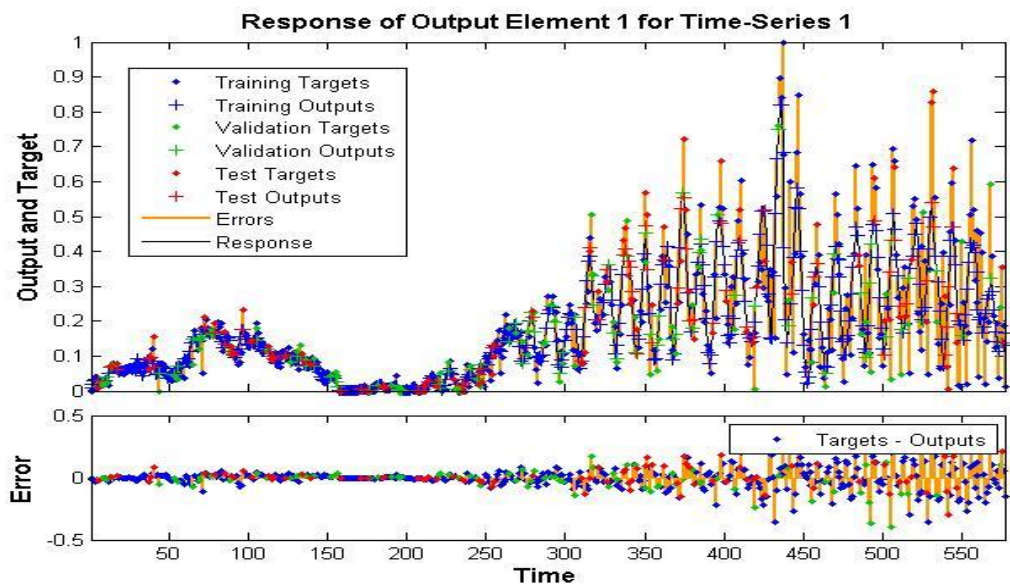


Ilustración 8-15 Output vs Target Mejor H12

9. Pruebas de Red Elman

Las pruebas se realizaron el mismo set de datos que con la red neuronal mimo, de forma de tener igualdad de condiciones previas para ambas redes, la configuración de las pruebas fue de 30 nodos de entradas, 70% de entrenamiento y H nodos de salidas, lo que varía es su capa oculta la cual es determinada por sus nodos ocultos.

9.1. Seis Meses de Horizonte(H=6)

Tabla 8 Datos Elman H6

Neuronas Ocultas	MSE	RMSE	R2	MAE
1	0,0105	0.1025	62.1%	0.0681
2	0.0057	0.0755	81.2%	0.0483
3	0.0032	0.0566	90.7%	0.0371
4	0.0040	0.0632	91.1%	0.0368
5	0.0062	0.0787	88.1%	0.0411
6	0.0031	0.0557	92.3%	0.0339
7	0.0049	0.0700	90.1%	0.0381
8	0.0047	0.0686	89.5%	0.0393
9	0.0029	0.0539	94.3%	0.0294

Resultados: En este caso para 6 meses de horizonte se utilizó el 70% de los datos para etapa de entrenamiento y se llegó a un 62.1% de R2 como peor escenario con la topología 30-1-6, pero logro llegar a 94.3% de R2 en su mejor escenario con la topología 30-9-6.

9.1.1. Mejor Resultado H6

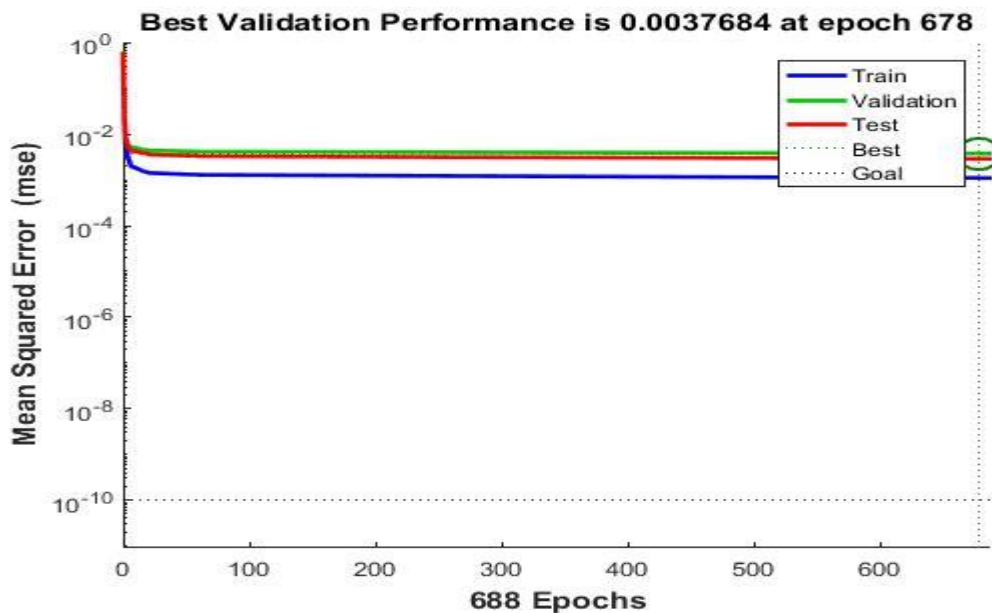


Ilustración 9-1 Mejor Performance H6

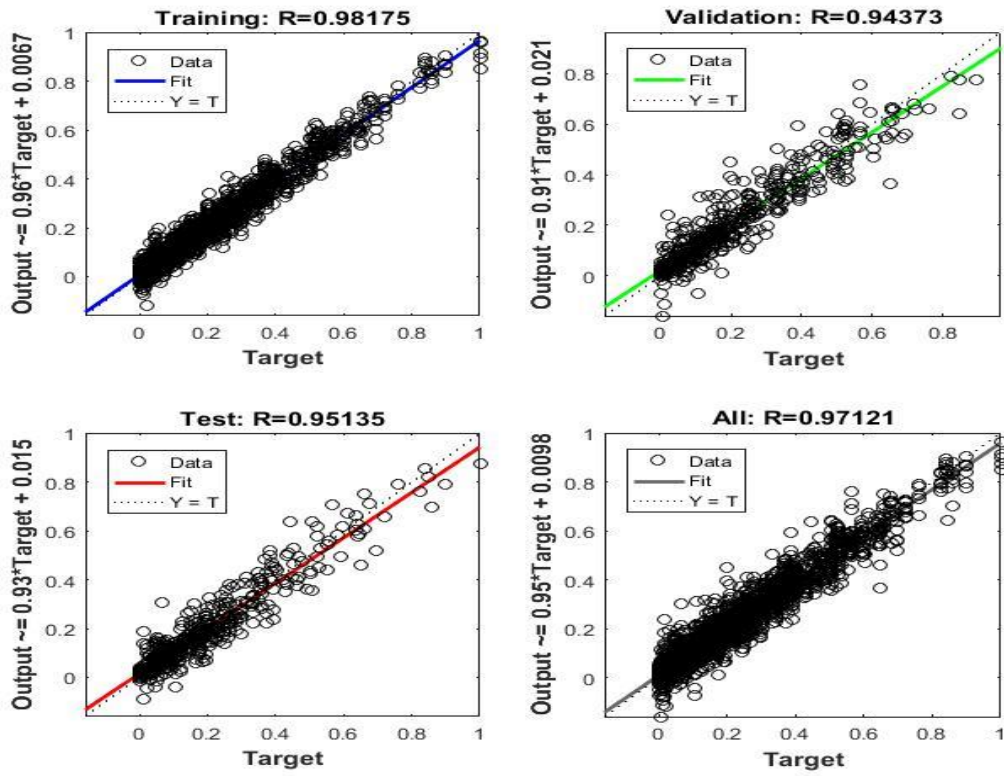


Ilustración 9-2 Mejor Regresión H6

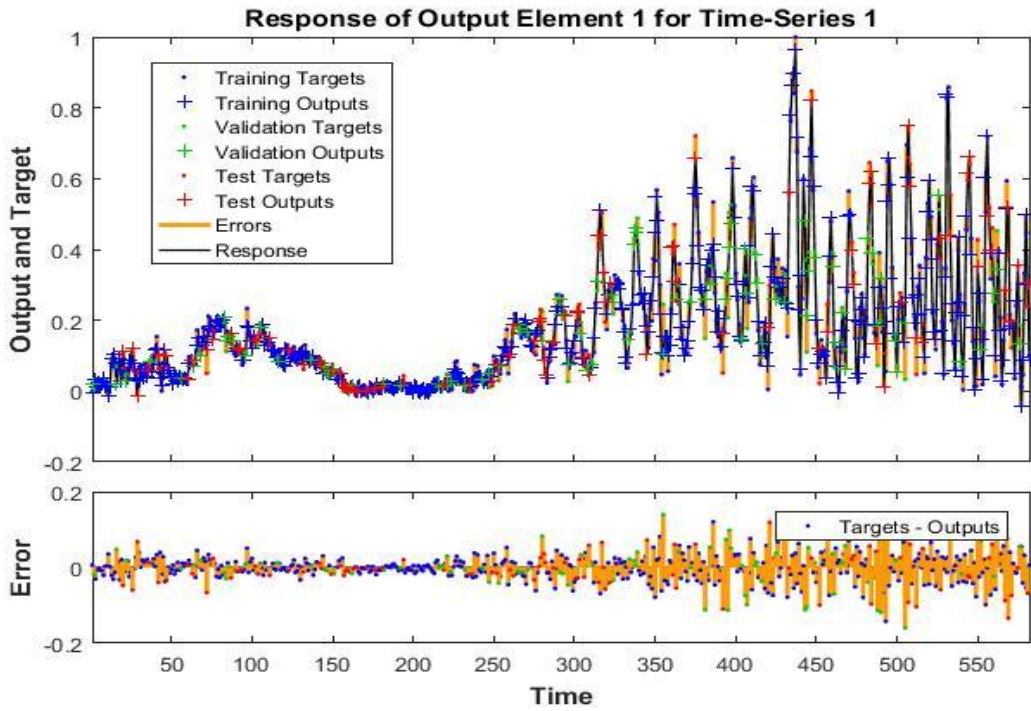


Ilustración 9-3 Mejor Outputs vs Targets H6

9.2. Nueve Meses de Horizonte (H=9)

Tabla 9 Datos Elman H9

Neuronas Ocultas	MSE	RMSE	R2	MAE
1	0.0158	0.1257	53.3%	0.0767
2	0.0087	0.0932	68.8%	0.0622
3	0.0078	0.0883	79.0%	0.0520
4	0.0091	0.0954	76.4%	0.0576
5	0.0051	0.0714	84.5%	0.0465
6	0.0062	0.0787	86.4%	0.0446
7	0.0076	0.0872	82.5%	0.0508
8	0.0037	0.0608	88.1%	0.0415

Resultados: En este caso para 9 meses de horizonte se utilizó el 70% de los datos para etapa de entrenamiento y se llegó a un 53.1% de R2 como peor escenario con la topología 30-1-9, pero logro llegar a 88.1% de R2 en su mejor escenario con la topología 30-8-9.

9.2.1. Mejor Resultado H9

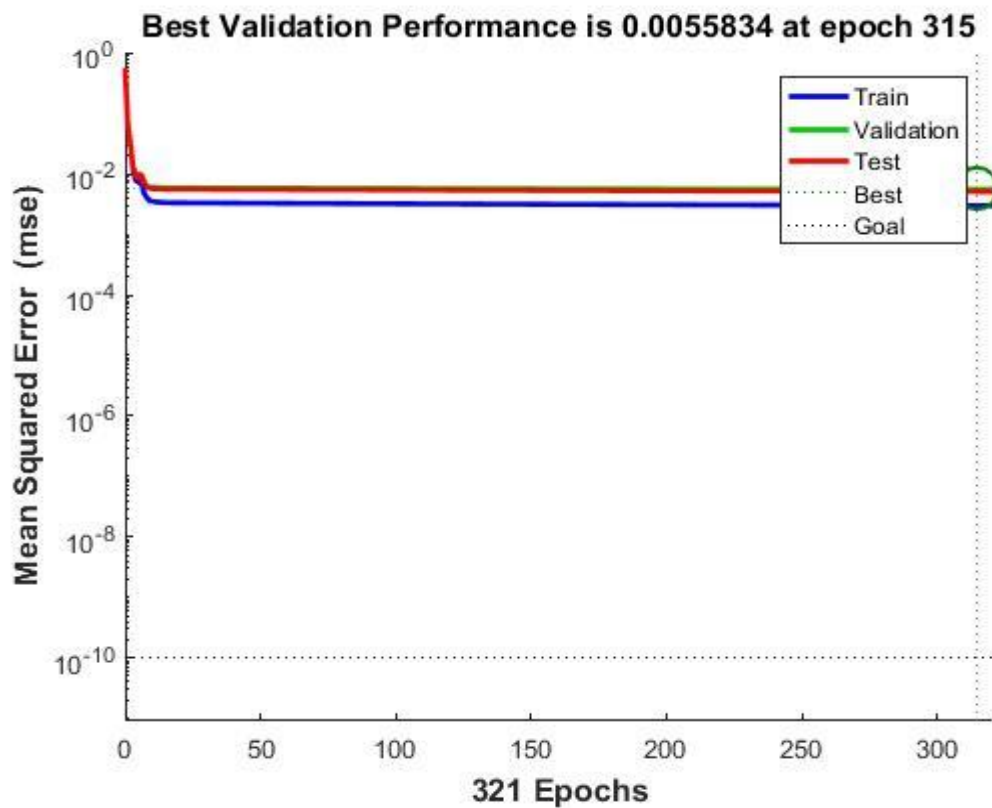


Ilustración 9-4 Mejor Performance H9

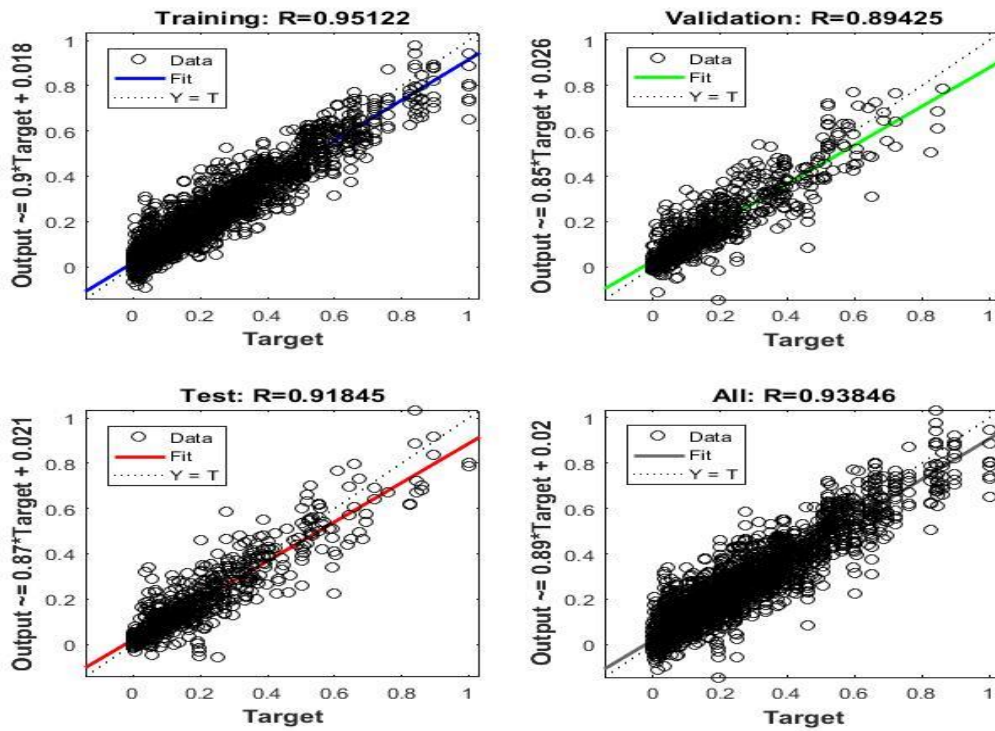


Ilustración 9-5 Mejor Regresión H9

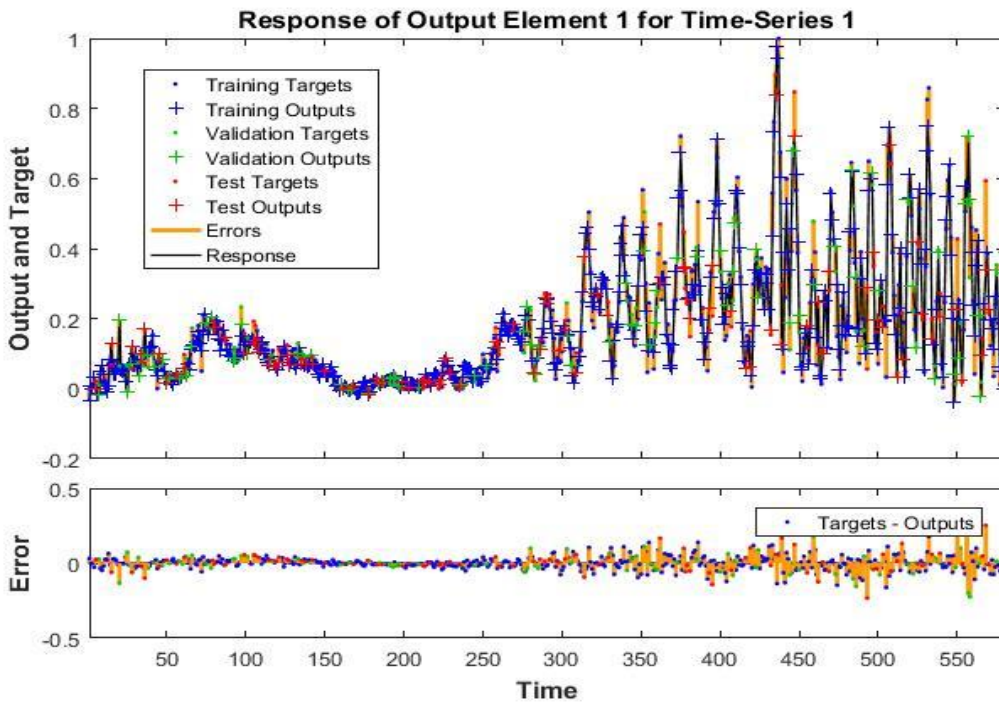


Ilustración 9-6 Mejor Outputs vs Targets H9

9.3. Doce Meses de Horizonte (H = 12)

Tabla 10 Datos Elman H12

Neuronas Ocultas	MSE	RMSE	R2	MAE
1	0.0113	0.1063	48.8%	0.0812
2	0.0137	0.1170	56.8%	0.0763
3	0.0155	0.1245	56.4%	0.0757
4	0.0086	0.0927	73.2%	0.0591
5	0.0094	0.0969	75.8%	0.0584
6	0.0075	0.0866	75.8%	0.0579
7	0.0048	0.0693	85.6%	0.0447
8	0.0098	0.0990	82.0%	0.0510

Resultados: En este caso para 12 meses de horizonte se utilizó el 70% de los datos para etapa de entrenamiento y se llegó a un 48.8% de R2 como peor escenario con la topología 30-1-12, pero logro llegar a 85.6% de R2 en su mejor escenario con la topología 30-7-12.

9.3.1. Mejor Resultado H12

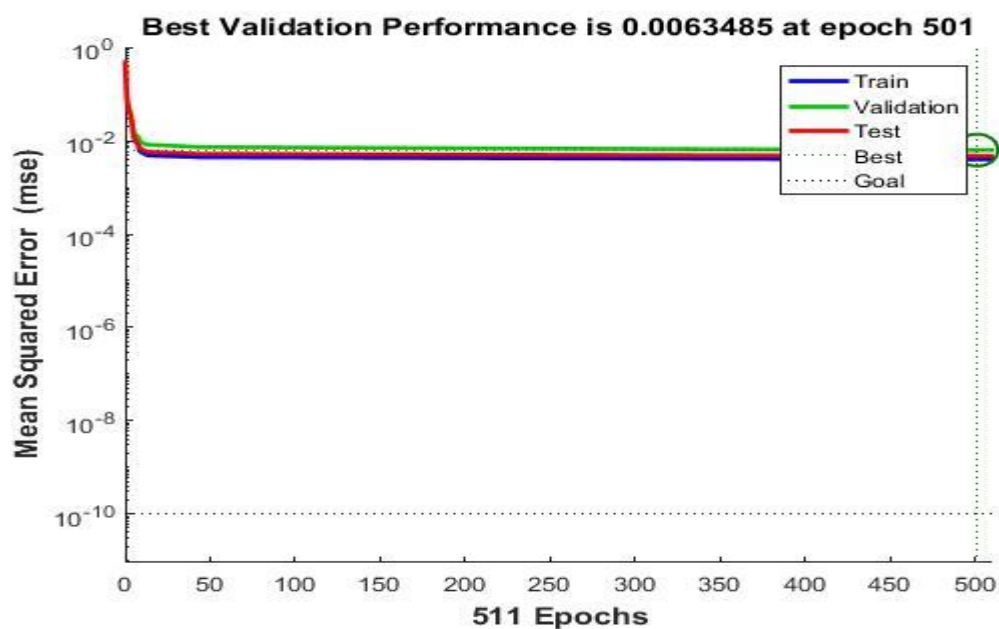


Ilustración 9-7 Mejor Performance H12

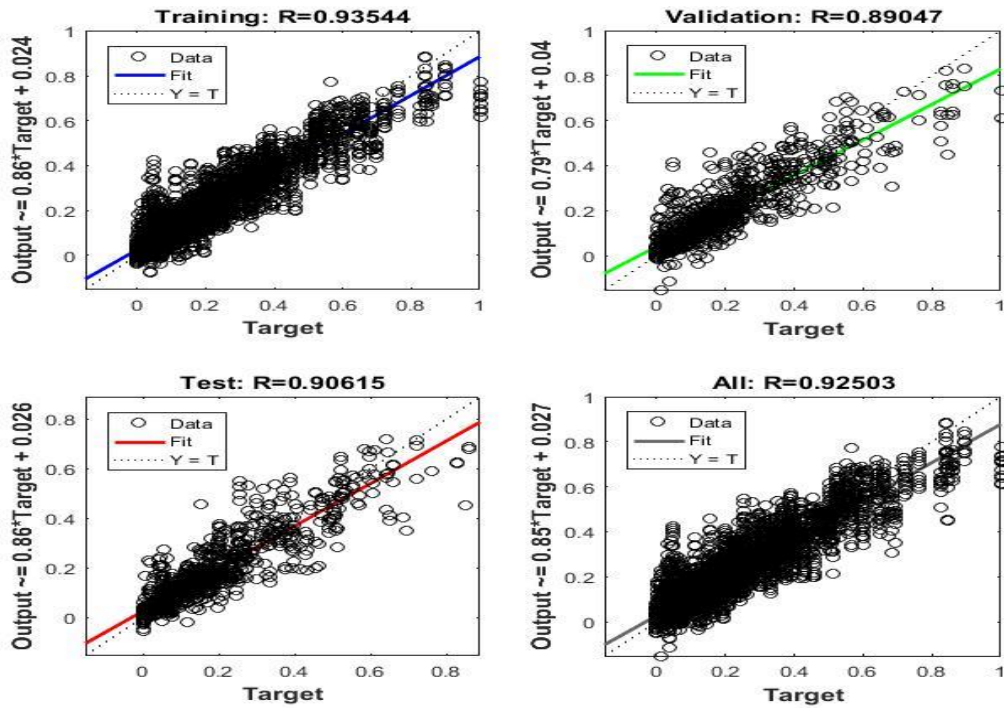


Ilustración 9-8 Mejor Regresión H12

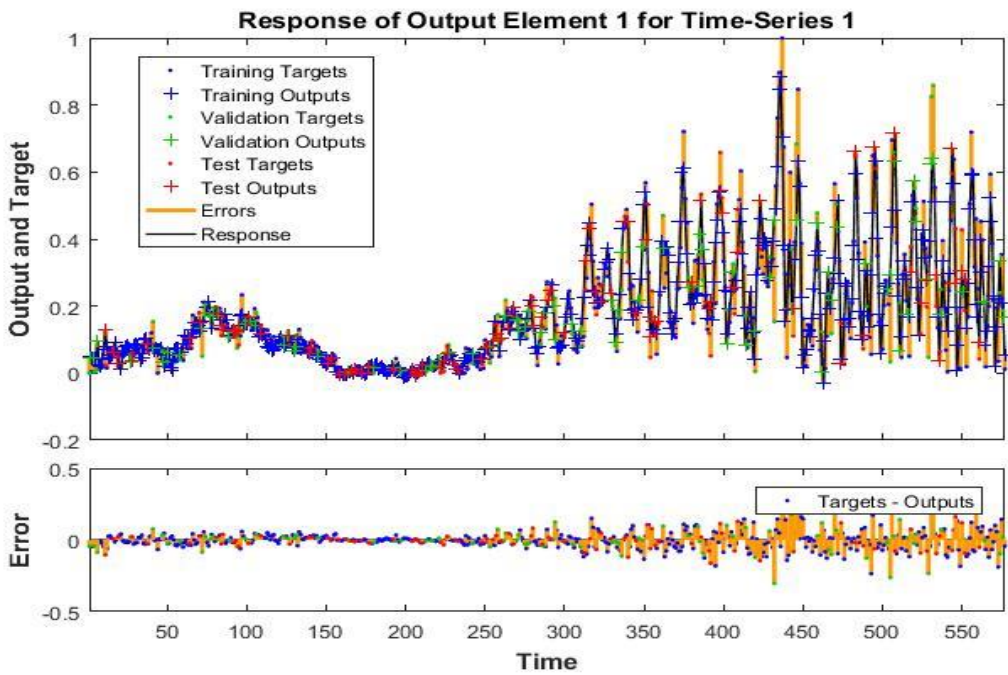


Ilustración 9-9 Mejor Outputs vs Targets H12

10. Comparativa: Red MIMO vs Elman

En este capítulo se compararán los mejores resultados de ambas redes en los distintos horizontes propuestos por el problema, para poder visualizar de mejor manera los datos obtenidos por ambas redes y realizar el análisis correspondiente.

10.1. Seis Meses de Horizonte

Tabla 11 Comparación Redes H6

Indicadores	Red MIMO	Red Elman
MSE	0.0004	0.0029
RMSE	0.0201	0.0539
R2	98.59%	94.3%
MAE	0.0117	0.0294

Observaciones: A pesar de que la Red Neuronal MIMO obtuvo mejores resultados en este caso de 6 meses de horizonte, cabe destacar que el tiempo de ejecución de la red Elman es considerablemente menor, dato a tomar en cuenta si la información es necesaria en un corto plazo.

10.2. Nueve Meses de Horizonte

Tabla 12 Comparación Redes H9

Indicadores	Red MIMO	Red Elman
MSE	0.0018	0.0037
RMSE	0.0423	0.0608
R2	93.74%	88.1%
MAE	0.0244	0.0415

Observaciones: Nuevamente la red neuronal MIMO logro un mejor desempeño en este caso de 9 meses de horizonte, pero lo visto en el punto anterior se vuelve a destacar, la red Elman demora bastante menos a la hora de la ejecución.

10.3. Doce Meses de Horizonte

Tabla 13 Comparación Redes H12

Indicadores	Red MIMO	Red Elman
MSE	0.0058	0.0048
RMSE	0.0759	0.0990
R2	81.65%	85.6%
MAE	0.0437	0.0447

Observaciones: En este caso de 12 meses de horizonte la red Elman logro superar en resultados a la red MIMO, destacando que en la red Elman se realizaron más pruebas con distintas topologías en este horizonte, no así en el caso de la red MIMO, ya que el algoritmo utilizado para esta red demoraba mucho más tiempo en ejecución, al punto de que el manejo de datos no era posible en la computadora utilizada para dichas pruebas.

11. Conclusión

Como todo proyecto relacionado con redes neuronales lo más importante son los datos, ya que, si los datos no están relacionados con el problema o se superponen la red no lograra calibrarse de la mejor manera, por lo tanto, no lograra un buen desempeño, por ende, en este trabajo se realizó una transformación de Wavelet a los datos de entrada, separándolos en datos de alta y baja frecuencia, permitiendo generar un set de datos de 60 datos de entrada para las distintas pruebas realizadas, esto permitió llegar a la topología correcta , necesaria para avalar un excelente modelo de pronóstico para el volumen de captura de la anchoveta.

Dada la naturaleza de las redes neuronales, las cuales conllevan múltiples y complejos cálculos, el tiempo de entrenamiento de la red va a depender del algoritmo utilizado y de la cantidad de neuronas en la capa oculta, por lo tanto, se debe evaluar el hecho de que a mayor cantidad de neuronas ocultas el resultando tiende a ser mejor, pero el tiempo que lleva calibrar una red neuronal con mayor cantidad de neuronas es considerablemente mayor y esto puede afectar a la hora de tomar una decisión en la topología de la red.

Para culminar, cabe destacar que los objetivos específicos 1, 2 y 3 de este proyecto están totalmente cumplidos, entregando la mejor topología para generar el modelo de pronóstico para el volumen de captura de anchovetas. De esta manera se da por finalizado el proyecto, logrando cumplir a cabalidad el objetivo general de este caso de estudio, con resultados satisfactorios capaces de pronosticar el volumen de captura de anchoveta para los horizontes propuestos.

12. Referencias

- [1] Gill, P.R.; Murray, W.; and Wright, M. H. “The Levenberg-Marquardt Method.” in Practical Optimization, pp. 136–137, Academic Press, London, UK, 1981.
- [2][http://www.anchoveta.info/index.php?option=com_content&task=view&id=50&Itemid=76]. Anchoveta.info, 2016. [Consultado: Marzo 2017]
- [3] [www.sernapesca.cl]. Servicio Nacional de Pesca y Acuicultura, 2016. [Consultado: Marzo 2017]
- [4] [Aranguren, S.M. y Muzachiodi, S.L. 2003] Redes neuronales y algoritmos genéticos, Cap. 8: Implicancias del data mining.
- [5] [Isasi y Galván, 2004] Pedro Isasi Viñuela e Inés Galván León, Redes Neuronales Artificiales: Un Enfoque Práctico, Pearson Prentice Hall, 2004.
- [6] [Rosenblatt, 1957] F. Rosenblatt, The perceptron: A perceiving and recognizing automation. Technical Report, Cornell Aeronautical Laboratory, 1957.
- [7] [<http://www.redes-neuronales.com.es/tutorial-redes-neuronales/la-neurona-Biologica.htm>]. redes-neuronales.com.es, 2016. [Consultado: Marzo 2017]
- [8] [Hebb, 1949] Donald Hebb, Organization of Behavior, John Wiley & Sons, 1949.
- [9] [Rosenblatt, 1957] F. Rosenblatt, The perceptron: A perceiving and recognizing automation. Technical Report, Cornell Aeronautical Laboratory, 1957.
- [10] [Hopfield, 1982] J. Hopfield, Neuronal Networks and phisycal systems with emergent collective computational abilities, In proceeding of the National Academy of Science, Vol. 81, 1982.
- [11] [Elman, 1990] J. Elman, Finding structure in time, Vol. 14, Cognitive Science, 1990.