Escuela Politécnica Superior de Linares

Grado en Tecnologías de Telecomunicación

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

ESTIMACIÓN DE VALORES DE CAMPO A PARTIR DE TRAZAS DE MEDIDAS MEDIANTE REDES NEURONALES

Alumno: **Juan Aguilar Pérez**

Tutor: Juan Pedro Roa Gómez

Depto.: Ingeniería de Telecomunicación

Junio, 2018

UNIVERSIDAD DE JAÉN

Escuela Politécnica Superior de Linares

ESTIMACIÓN DE VALORES DE CAMPO A PARTIR DE TRAZAS DE MEDIDAS MEDIANTE REDES NEURONALES

REALIZADO POR:

Juan Aguilar Pérez

DIRIGIDO POR:

Juan Pedro Roa Gómez

El tutor D. Juan Pedro Roa Gómez da el visto bueno para entregar y defender este trabajo fin de grado realizado por el alumno Juan Aguilar Pérez.

Alumno: Juan Aguilar Pérez Tutor: Juan Pedro Roa Gómez

Fdo.: Fdo.:

DEPARTAMENTO DE: Ingeniería de Telecomunicación, Área de Teoría de la Señal y Comunicaciones.

PALABRAS CLAVE: Valores de campo, WI-FI, Redes neuronales.

RESUMEN: Estudio e implementación de redes neuronales para la estimación de señal WI-FI a partir de trazas conocidas.

INDICE

INDICE DE FIGURAS

TABLAS

1 INTRODUCCIÓN AL TRABAJO FIN DE GRADO

1.1 Objetivos del trabajo Fin de Grado

El principal objetivo de este trabajo fin de grado es elaborar una herramienta software, la cual nos permita introducir coordenadas de una ubicación dentro de una zona delimitada y, mediante Redes Neuronales, poder estimar la señal Wi-Fi de esa ubicación.

En este trabajo se aplicarán redes neuronales para estimar la red Wi-Fi, en un cierto punto, a partir de la lógica de las neuronas, de sus métodos y de sus diferentes modelos de entrenamiento y simulación.

Se diseñará un programa en lenguaje de programación MATLAB, que sea capaz de leer datos (coordenadas) y muestre por pantalla los resultados obtenidos para los datos de entrada.

Las pruebas de calidad que se han exigido son las siguientes:

- Que los principios teóricos desarrollados en el diseño del programa sean correctos.
- Que los resultados ofrecidos sean válidos.
- Que el funcionamiento del programa sea el deseado incluso para situaciones límite.

1.2 Contenido de la Memoria

El capítulo 1, que sirve de introducción, expone todos los objetivos que se pretenden conseguir con el trabajo fin de grado.

En el Capítulo 2, se hace un breve resumen del TFG, donde se especifica la necesidad a cubrir por éste.

En el Capítulo 3, se describe el estudio de fenómenos de propagación radioeléctrica, su evolución y la previsión de futuro.

El Capítulo 4 expone, brevemente, cómo se han obtenido los datos de la señal recibida y su posterior análisis.

En el Capítulo 5, se explica qué es una red neuronal, su definición, por qué elementos está constituida, sus características y los diferentes modelos de redes neuronales que se han considerado más importantes.

El capítulo 6 cuenta el estudio realizado con los parámetros de entrada y de salida para la implementación de las redes neuronales desarrolladas en este TFG, los parámetros intermedios a tener en cuenta y los diferentes métodos de implementación para las redes neuronales.

El capítulo 7 describe el entrenamiento y validación de la red neuronal, desarrollada con los diferentes modos y los diferentes casos de entrenamiento.

El capítulo 8 comenta la simulación y estimación del programa realizado mediante varios casos prácticos.

El capítulo 9 muestra las conclusiones sacadas al realizar este trabajo fin de grado.

Por último, en el capítulo 10 se expone la bibliografía utilizada.

2 RESUMEN

Este trabajo fin de grado tiene como finalidad conseguir, mediante el entrenamiento y simulación de redes neuronales, la estimación de señal Wi-Fi en una ubicación dada, dentro de una zona geográfica delimitada.

Dentro de este documento, se estudian los diferentes fenómenos de propagación radioeléctrica, las diferentes redes neuronales con sus correspondientes métodos y su posterior entrenamiento, validación y simulación.

También en este documento, se analizan y se modifican los datos proporcionados por [8], en el documento [9]. En este último, aparecen mediciones en diferentes ubicaciones de redes Wi-Fi medidas en diferentes zonas (anotando en cada momento los datos referentes a la ubicación analizada y a las redes Wi-Fi detectadas).

Se realiza un análisis sobre cuál sería la mejor configuración de los datos de entrada y de salida para la red neuronal, basándonos en los resultados obtenidos de promedio cuadrado de error y promedio absoluto de error (MSE y MEA).

Una vez analizado y modificado las medidas proporcionadas, se procede a crear una base de datos con los que realmente son de utilidad para el objetivo del trabajo fin de grado. Ésta se crea con la finalidad de contener todos los datos necesarios para el entrenamiento, validación y simulación de las redes neuronales implementadas.

La base de datos se realiza con una cierta configuración para que, a la hora de alimentar las redes neuronales, se pueda recoger los datos sin ningún inconveniente.

Una vez recogidos y configurados los datos para las entradas y salidas de las redes neuronales, se procede a entrenar, validar y simularlos con diferentes modelos de redes neuronales.

Se procede a analizar y aplicar el funcionamiento de tres modos de entrenamiento diferentes para la estimación de Wi-Fi, los modos elegidos son los correspondientes a las funciones de MATLAB Traingd, Trainrp, Trainscg.

Para cada modo se realizan tres casos de entrenamiento:

- Caso 1: entrenamiento de red neuronal por defecto, con una sola capa oculta.
- Caso 2: entrenamiento de red neuronal modificando el nº de neuronas por capa.
- Caso 3: entrenamiento de red neuronal modificando el nº de épocas.

Una vez estudiados los tres casos anteriores, se analizan los resultados y se realiza una comparativa acerca de que método y con qué condiciones funcionan mejor las redes neuronales; para la estimación de la red Wi-Fi y mediante una determinada ubicación.

3 ESTUDIO RADIOELÉCTRICO

3.1 Características

Las ondas de radio son ondas electromagnéticas, que poseen una componente eléctrica y una componente magnética.

En condiciones normales, las ondas de radio tienden a desplazarse en línea recta, por lo que habría una línea de vista entre el emisor y el receptor. Este tipo de comunicación es bastante eficiente pero, si se requiere una comunicación de un punto a otro, el cual se encuentra más allá del horizonte, se tiene que tomar en cuenta las condiciones de propagación y las frecuencias para su correcta comunicación. Para realizar comunicaciones seguras, entre dos puntos lejanos y sin salir de la atmósfera, se utilizan frecuencias llamadas altas Frecuencias (High Frequency) o HF que van de 3 Mhz a los 30 Mhz. Estas frecuencias son reflejadas en la atmósfera y regresan a la tierra a grandes distancias.

En comunicaciones por medio de ondas de radio a grandes distancias se tienen que tener en cuenta los fenómenos de refracción, reflexión, dispersión y difracción. Éstos hacen posibles la comunicación entre dos puntos más allá del horizonte [1].

3.2 Conceptos básicos

Como se comentó anteriormente, los fenómenos de refracción, reflexión, dispersión y difracción son de gran importancia para las comunicaciones inalámbricas.

3.2.1 Refracción

Las ondas de radio están expuestas a sufrir una desviación en su trayectoria cuando atraviesan de un medio a otro con densidad distinta. Este efecto sucede cuando las ondas electromagnéticas atraviesan las distintas capas de la atmósfera, variando su trayectoria en un cierto ángulo. La desviación de la trayectoria es proporcional al índice de refractividad, el cual está dado por:

IR=VP/Vm donde IR= Índice de refractividad, VP= velocidad de propagación en el espacio libre, Vm= velocidad de propagación en el medio. [1]

3.2.2 Reflexión

Las ondas de radio atraviesan las diversas capas de la atmósfera, desde la troposfera hasta la ionosfera. Los índices de refractividad de cada una de estas capas son muy diferentes. Estos distintos índices pueden llegar a producir reflexión total.

Las frecuencias de VHF y superiores son las más propensas a esta desviación de trayectoria. [1]

3.2.3 Dispersión

El efecto de la dispersión ocurre cuando las ondas de radio atraviesan alguna masa de electrones o pequeñas gotas de agua en áreas suficientemente grandes. La dispersión de la señal generada por lluvia depende de la comparación del tamaño de la longitud de onda de la señal y el diámetro de la gota de lluvia. Si el diámetro *d*, de la gota de lluvia, es menor a la longitud de onda, la atenuación será pequeña, pero ésta se acrecentará si el diámetro de la gota supera la longitud de onda de la señal. Generalmente la refracción se produce solamente a determinados ángulos. Este efecto es similar al que le ocurre a la luz intentando atravesar la niebla. [1] y [2]

3.2.4 Difracción

Se puede entender la difracción como el esparcimiento de las ondas en los límites de una superficie. Esto quiere decir que, para que exista la difracción, tiene que haber un obstáculo. Así es como este fenómeno permite que parte de la señal llegue al otro lado del objeto. Este fenómeno es de gran utilidad para las zonas de sombra de señal que pueden ser producidas por grandes edificios o montañas. [1] y [2]

3.3 Comunicaciones inalámbricas

Como se comentó anteriormente, gracias a los fenómenos de reflexión, refracción, difracción y dispersión se pueden realizar las comunicaciones inalámbricas a grandes distancias. A continuación se mostrarán las distintas formas de comunicación que existen.

3.3.1 Propagación por onda directa

Para realizar este tipo de propagación es necesario que exista una línea de vista entre el transmisor y el receptor. En este tipo de comunicación se utilizan frecuencias por encima de los 50Mhz. Esto se debe a que las frecuencias altas se ven menos afectadas por los fenómenos atmosféricos, además de que no requiere de antenas grandes para tener una transmisión efectiva de gran directividad, lo que provoca la confiabilidad de que la información llegue a otro lado del transmisor. Este tipo de propagación se utiliza como por ejemplo para la radio FM. A continuación se muestra la propagación por onda directa.

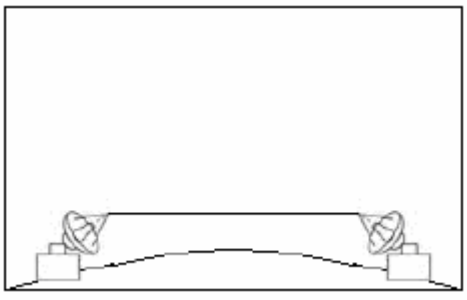

Figura 1: Ejemplo propagación onda directa

3.3.2 Propagación por onda terrestre

Este tipo de propagación es posible gracias a la difracción. Las ondas de radio siguen la curvatura de la tierra por la cual la señal de RF es capaz de alcanzar grandes distancias antes de que la señal sea absorbida por la tierra. Gracias al efecto de la difracción la señal puede sortear edificios y montañas. En la siguiente figura se muestra el efecto que tiene la difracción sobre las señales RF. [1]

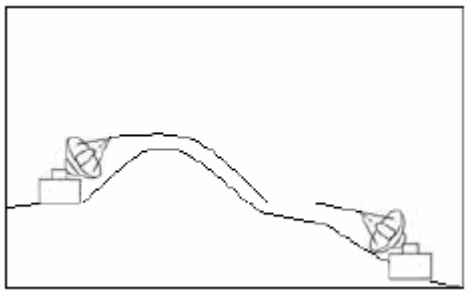

Figura 2: Ejemplo propagación por difracción

La propagación por onda terrestre solo es útil para frecuencias inferiores a los Mega Hertz, siendo esta una de las mejores formas de transmitir una señal de RF de baja frecuencia a largas distancias. Este tipo de propagación es comúnmente usada por las radiodifusoras de media onda y de onda larga. [1]

3.3.3 Propagación por onda refractada o ionosférica.

Este tipo de propagación es de los más importantes. Aquí tiene influencia la atmosfera como refractor y esto a su vez ocurre en la ionósfera. La ionósfera es una capa de la atmosfera que se encuentra entre los 40Km y 320 Km y está formada por un aire altamente ionizado por la radiación solar. Cuando esta capa se encuentra eléctricamente cargada hace que la señal comience a cambiar en un cierto ángulo, esto lo hace sucesivamente hasta que realiza una reflexión total y la señal regresa a tierra. En la siguiente figura se muestra como la señal se refracta en la ionosfera para hacérsela llegar al receptor. Traducción libre de [2]

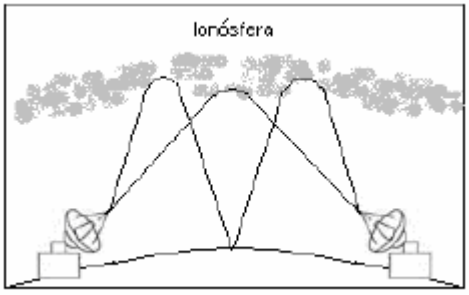

Figura 3: Ejemplo propagación por onda reflejada

Este tipo de propagación puede ser capaz de conectar dos puntos, los cuales no tienen línea de visión directa y se puede transmitir a una distancia de hasta 4000 Km. Si las condiciones de la atmosfera fueran adecuadas se podría conectar un punto a cualquier otro lado del planeta, esto es porque la señal refractada de la ionosfera también puede ser reflejada por la tierra y así sucesivamente. Es importante mencionar que la propagación ionosférica está determinada por la frecuencia utilizada y por el nivel de ionización de la atmosfera. Si se tiene una frecuencia grande de refracción sufrida por la misma ionosfera será menor. Se cuenta con una frecuencia establecida a utilizar a distintas horas del día para poder realizar la comunicación ionosférica, esto se conoce como frecuencia útil máxima, FUM.

3.3.4 Propagación por difracción ionosférica.

Este tipo de propagación se produce cuando las ondas emitidas son superiores a los 30 MHz, debido a su frecuencia la señal no será reflejada por la ionósfera, pero si será difractada, por lo que una pequeña parte de la señal llegará a tierra y solo podrá ser captada por un receptor especialmente sensible. Es por esto que este tipo de transmisión se utiliza muy poco debido a su baja eficiencia. A continuación se muestra el efecto de la difracción de una señal por la ionosfera. [1]

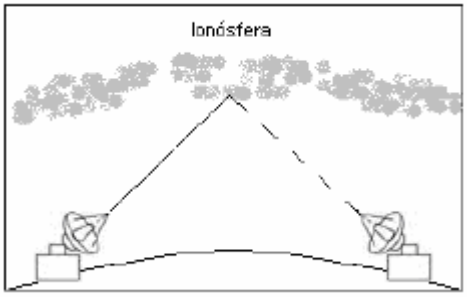

Figura 4: Ejemplo propagación por difracción ionosférica

3.4 Comunicación por tecnología WI-FI

La comunicación inalámbrica o comunicación a distancia, es aquella capaz de enviar una cantidad de datos de un punto a otro (emisor y receptor) sin la necesidad de un agente o un hardware que conecte ambos puntos físicamente. Principalmente el componente que logra el funcionamiento de dicha comunicación es la programación (software). El hardware que se utiliza para el correcto funcionamiento son antenas. [7]

La tecnología inalámbrica utiliza ondas de radiofrecuencia de baja potencia y de una banda específica, de uso libre o privada, para transmitir entre dispositivos. Esta es una de las tecnologías de comunicación inalámbrica mediante ondas más utilizada a día de hoy, también llamada WLAN (wireless Lan, red inalámbrica) o estándar IEEE 802.11.

En la actualidad podemos encontrarnos con dos tipos de comunicación Wi-Fi: 802.11b, que emite a 11 Mb/seg, y 802.11g, más rápida, a 54 MB/seg.

3.4.1 Comunicación Wi-Fi: 802.11b

En julio de 1999, la IEEE expande el 802.11 creando la especificación 802.11b,la cual tiene una velocidad teórica máxima de transmisión de 11 Mbit/s, comparable a una Ethernet tradicional, pero debido al espacio ocupado por la codificación del protocolo CSMA/CD (Carrier Sense Multiple Access / Collision Detect), en la práctica la velocidad máxima de transmisión es de aproximadamente 5.9 Mbit/s para TCP y 7.1 Mbit/s para UDP.[2] La 802.11b utiliza la misma frecuencia de radio que el tradicional 802.11 (2.4GHz). El problema es que al ser esta una frecuencia sin regulación, se podían causar interferencias con hornos microondas, teléfonos móviles y otros aparatos que funcionen en la misma frecuencia. Sin embargo, si las instalaciones 802.11b están a una distancia razonable de otros elementos, estas interferencias son fácilmente evitables. [6]

Este estándar presenta las ventajas de bajo costo, rango de señal muy bueno y difícil de obstruir, y ofrece seguridad de calidad de servicio QOS. También presenta los inconvenientes de baja velocidad máxima, soporte de un número bajo de usuarios a la vez y produce interferencias en la banda de 2.4 GHz, como pueden ser hornos microondas, dispositivos Bluetooth [4], monitores de bebés y teléfonos inalámbricos. También otro gran inconveniente es que este protocolo no es compatibles con los productos de estándar 802.11a por operar en bandas de frecuencia distintas.

3.4.2 Comunicación WI-FI 802.11g

El protocolo de comunicación 802.11g incorporó la técnica de comunicación por radio llamada Multiplexación por división de frecuencia ortogonal (OFDM) que se introdujo originalmente en Wi-Fi con 802.11a ("A"). La tecnología OFDM permitió a 802.11g (y a) lograr un rendimiento de red significativamente mayor que 802.11b.

Por el contrario, 802.11g adoptó la misma gama de frecuencias de comunicación de 2,4 GHz introducida originalmente en Wi-Fi con 802.11b. El uso de esta frecuencia dio a los dispositivos Wi-Fi un rango de señal significativamente mayor de lo que A podría ofrecer.

Hay 14 canales posibles en los que 802.11g puede operar, aunque algunos son ilegales en algunos países. Las frecuencias del canal 1-14 oscilan entre 2.412 GHz y 2.484 GHz.

El protocolo de comunicación 802.11g fue especialmente diseñado para la compatibilidad cruzada. Lo que esto significa es que los dispositivos pueden unirse a redes

inalámbricas incluso cuando el punto de acceso inalámbrico ejecuta una versión Wi-Fi diferente. Incluso el equipo de Wi-Fi 802.11ac más nuevo de la actualidad puede admitir conexiones de clientes G que utilizan estos mismos modos de operación de compatibilidad de 2,4 GHz. [5]

3.4.3 Valor RSSI:

Las mediciones de RSSI representan la calidad relativa de una señal recibida en un dispositivo. RSSI indica el nivel de potencia que se recibe después de cualquier posible pérdida en el nivel de antena y cable. Cuanto mayor sea el valor RSSI, más fuerte será la señal. [4]

Aunque RSSI y dBm son unidades de medida diferentes, ambos indican la intensidad de la señal. El dBm es una relación de potencia de la potencia medida como referencias a un mW (miliwatt). Mientras que dBm es un índice absoluto, el RSSI es relativo.

Para la señal significativa de buena calidad de medición ha de sustraer el ruido de la línea de la potencia de la señal. Una mayor diferencia de señal a ruido significa una mejor intensidad de señal.

3.4.4 RSSI en implementaciones 802.11:

RSSI puede ayudar a determinar cuándo el nivel de energía de radio en el canal es menor que un cierto punto para que la tarjeta de red esté clara para enviar (CTS). En ese momento se puede enviar el paquete de datos. [4]

El estándar 802.11 no da ninguna definición de cómo se relacionan el valor de RSSI y el nivel de potencia en mW o dBm. Los fabricantes de chips y proveedores dan su propia precisión, granularidad y rango para la potencia real (mW o dBm).

4 ANÁLISIS DE SEÑAL RECIBIDA

4.1 Origen y obtención de Señal recibida

En los documentos proporcionados por [8] en el artículo [9] se hace referencia a los datos recogidos y a la forma de obtenerlos en función de las diferentes ubicaciones.

El hardware utilizado para recopilar los datos de estos documentos fue:

- Un ordenador portátil T30
- Tarjeta Orinoco Gold 802.11 con antena externa de 5 dB
- Una unidad GPS Garmin Rhino

En estos documentos se muestran las medidas tomadas dentro del área metropolitana de Seattle en tres zonas geográficas diferentes:

- En el centro de Seattle
- Zona de Revenna
- Kirkland

El primer documento hace referencia a los datos recogidos en el centro de Seattle condado de King, el segundo documento hace referencia a la zona de revenna en la que se encuentra la fraternidad y en el último documento, muestra los registros tomados en la ciudad de kirkland al este de Seattle.

Mencionar que para obtener los datos proporcionados en [2] se utilizó la tarjeta WI-FI para descubrir redes inalámbricas y para escuchar las respuestas de sondeo desde los puntos de acceso cercanos, mientras que el portátil emitía periódicamente una trama de petición de 802.11 y el dispositivo GPS registraba la posición geográfica en la que se realizaba la medida [9]

4.2 Edición y estructura de datos recibidos

La estructura de los datos es muy importante y es de tener en cuenta de cara a la implementación de las redes neuronales.

Por ese motivo, es de vital importancia saber que parámetros son necesarios y cuales no de todas las medidas registradas en los documentos aportados por [1]

Para poder adaptar los datos que tenemos a las necesidades reales de este trabajo fin de grado, se decide crear una nueva base de datos con solamente los datos necesarios para éste trabajo fin de grado.

Se considera que los datos imprescindibles para realizar una estimación de red Wi-Fi en función de una ubicación son:

- Latitud

- Longitud
- Altura antena GPS
- Numero de satélite
- Nombre de la WI-FI
- Ídem
- RSSI de la WI-FI

Por ello, se procede a crear en un documento Excel una base de datos con los campos mencionados anteriormente.

La idea de tener todos los datos almacenados en un documento Excel es para que a la hora de realizar los entrenamientos de las redes neuronales, poder recoger los datos desde una fuente fiable y poder incorporarlos al programa creado en Matlab sin ningún problema.

El documento Excel se compone por tres pestañas, una para cada zona geográfica. En cada pestaña, se encuentran los datos registrados para cada medición.

La primera pestaña contiene los registros de la señal GPS y WI-FI de la zona céntrica de Seattle, la segunda pestaña contiene los registros pertenecientes a la zona de Revena y la tercera pestaña contiene los registros de kirkland.

4.3 Definición de valores registrados.

Aquí se pretende definir los valores registrados en la base de datos, serían los datos mencionados en el apartado 4.2

Latitud:

Longitud:

Altura antena GPS:

Numero de satélite:

Nombre de la WI-FI:

Ídem:

RSSI de la WI-FI:

5 REDES NEURONALES

Capitulo donde se explica que son, como aparecieron y sus características referentes a las redes neuronales.

5.1 Introducción redes neuronales

Las redes neuronales artificiales están compuestas por muchos elementos simples de procesamiento conocimos como neuronas, es decir las redes neuronales están compuestas por neuronas.

En el siglo pasado, la investigación por la computación neuronal tuvo una gran demanda, ya que el hecho de poder crear redes computacionales artificiales con la intención de modelar sistemas artificiales, que estuvieran definidos previamente, llamaba la atención a cualquier investigador o curioso del mundo de la tecnología.

La aplicación de redes neuronales para el ámbito de las telecomunicaciones no es utilizada recientemente, desde finales del siglo pasado se utilizan para resolver problemas de interconexión por multi-capa de redes de datos, reconfiguración de redes en sistemas satélites, para enrutamiento adaptativo en redes, etc. [10]

5.1.1 Definición de red Neuronal y biológica

Una red neuronal artificial es un sistema de procesamiento de información, con características de rendimiento comunes con una red neuronal biológica.

Una red neuronal biológica está compuesta por neuronas biológicas, esta consta de redes en forma de árbol conocidas como dendritas, conectadas al cuerpo de la neurona que contiene el núcleo, por último, desde el cuerpo de la célula se extiende una rama llamada axón (en el grafico posterior se aprecia).[11]

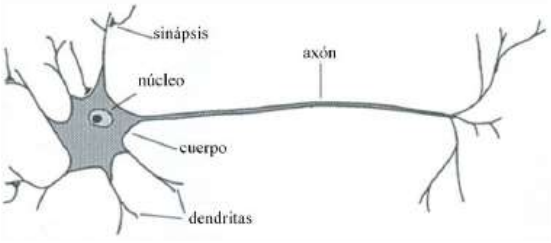

Figura 5: Ejemplo Grafico de neurona biológica

Tanto las redes neuronales como las redes biológicas están basados en los mismos principios:

-En las neuronas es el lugar donde se produce el procesamiento de la información

-Las señales se trasmiten por enlaces de conexión entre las neuronas

-Cada enlace de conexión tiene un peso Wij asociado que multiplica la señal transmitida

-Para producir una salida yi, cada neurona aplica una función f de activación a su entrada neta (es la suma de las señales multiplicadas por su peso

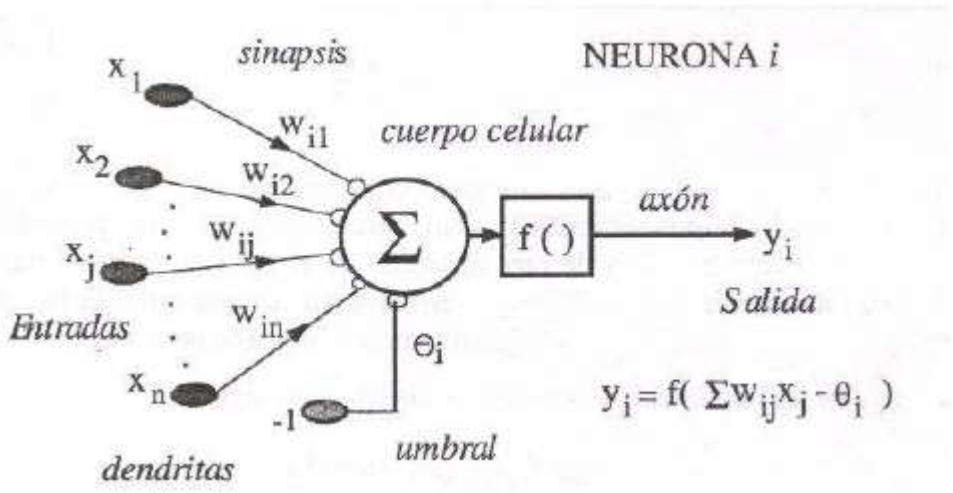

Figura 6: Representación de neurona artificial

La imagen anterior representa las partes de una neurona artificial con sus similitudes con una neurona biológica. Aquí se muestra que los valores típicos de las entradas xj son multiplicados por cada uno de los pesos Wij, con la suma de estos productos, se genera la entrada neta, esta entrada neta es modificada por la función de activación f () y produce la salida yi. [11]

5.1.2 Características Red Neuronal

Una red neuronal es caracterizada por:

1. La arquitectura: Es el número de neuronas en capas, la constitución de las capas y el patrón de conexiones entre las capas. La red mostrada en la figura 5 está compuesta por una capa de entrada, una capa oculta y una capa de salida. Un peso W entre dos neuronas en capas adyacentes, representa la fuerza de conexión que se ejerce entre ellas.

2. El entrenamiento o algoritmo de aprendizaje: El método para la determinación de los pesos entre las neuronas, que puede ser supervisado cuando a la red se le presenta un conjunto de vectores de entrada con una salida asociada y los pesos se ajustan de manera iterativa para que produzcan la salida deseada; o no supervisado cuando solo se presenta un conjunto de vectores de entrada y la red neuronal modifica sus pesos de manera que vectores con características semejantes presenten la misma salida.

3. La función de activación: Que transforma la entrada neta y produce una señal de salida. Algunas funciones de activación comunes son: la función identidad, la función de Heaviside y la función sigmoidea.

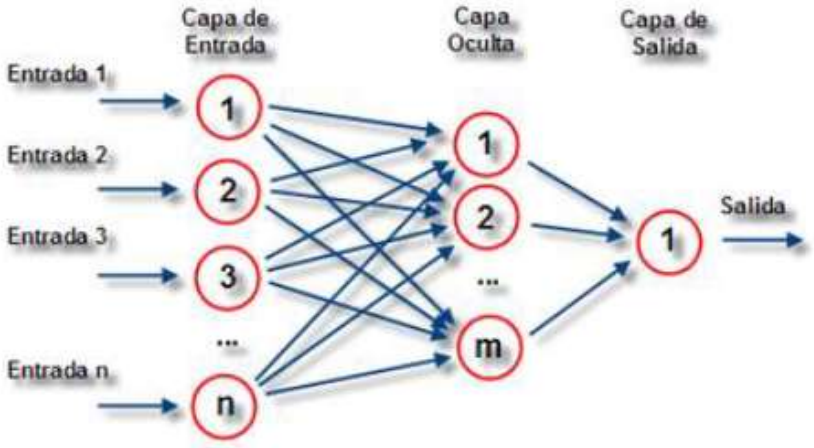

Figura 7: Representación de arquitectura de una red neuronal

En los subcapítulos siguientes se pone a conocimiento los diferentes tipos de redes neuronales que se han considerado más importantes.

5.2 Modelos Redes Neuronales

5.2.1 Modelo Perceptrón

El perceptrón fue el primer modelo de red neuronal, fue creado por Rosenblatt a finales de los años cincuenta. Este modelo está constituido por dos capas de neuronas, una de entrada y otra de salida.

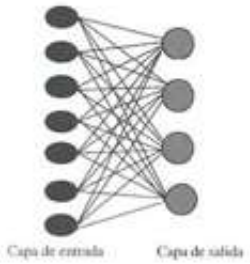

Figura 8: Representación de arquitectura red perceptrón

Es un modelo de red neuronal unidireccional. Los vectores de entrada tienen que ser vectores Binarios y su función de transferencia f(x) viene definida como:

$$f(x = \begin{cases} 1 & si\ w * x - u > 0 \\ 0 & en\ cualquier\ otro\ caso \end{cases}$$

Donde *w* es el vector de pesos reales y *w*x* es el producto escalar, el termino que presenta el grado de inhibición de la neurona es *u*, éste es un término constante que no depende del valor de la entrada.

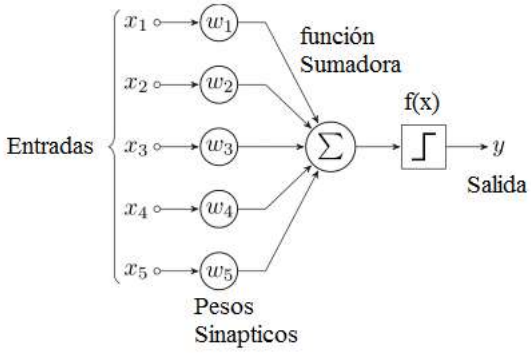

Figura 9: Representación parámetros red perceptrón

El desarrollo del modelo de perceptrón levanto gran interés debido a la capacidad de poder generalizar a partir de los vectores de entrenamiento y poder trabajar con conexiones distribuidas aleatoriamente.

Los pesos y los umbrales pueden ser entrenados para producir un vector de salida correcto correspondiente a cada vector de entrada.

Se demostró que la regla de aprendizaje del perceptrón converge en un tiempo finito si existe la solución.

El perceptrón funciona en función de un conjunto de entrenamiento (vectores de entrada y salida conocidos) que se presenta a la red (cada vector consecuente uno tras otro). Cuando la salida de la red es la correcta, no se produce ningún cambio, sin embargo, los pesos y los umbrales se actualizan en función de la regla de aprendizaje del perceptrón.

Se considera una época cuando es completado un ciclo a través de todos los vectores de entrada de entrenamiento. Hasta que el error de aprendizaje no se considere aceptable, los ciclos se repiten uno tras otro, cuando se alcanza el error aceptable, se puede dar por concluido el entrenamiento.

Una vez concluida la fase de entrenamiento, se podrán hacer simulaciones con un vector que no estuviese definido en el vector de entrenamiento, para así, poder presentarle a la red un nuevo vector y ver realmente su funcionamiento, dicha red tenderá a dar como resultado un valor aproximado al valor generado por los vectores de entrenamiento.

5.2.2 Modelo Hopfield

En 1982 John Hopfield, en su artículo “Neural Networks and Physical Systems with Emergent Colective Computational Abilities” [12] describió un nuevo modelo de redes neurales capaz de realizar tareas computacionales cuyo objetivo inicial era el comportarse

como una memoria de contenido direccional CAM ("Content-Addressable Memory"). Hopfield demostró que este modelo no solo era capaz de funcionar como una memoria, sino que también era capaz de realizar funciones de generalización, reconocimiento de patrones, clasificación, corrección de errores y retención de secuencias temporales.

Como se comentó en apartados anteriores, los sistemas de redes neurales artificiales se basan en el funcionamiento de las redes neurales biológicas.

La Red Hopfield original consta de un sistema de n elementos computacionales (neuronas), interconectados entre sí, por ello, todas y cada una de las neuronas están conectadas con las neuronas restantes pero con ellas mismas no ("full interconnected network"), formando así un plano bidimensional donde no se puede llegar a distinguir una capa de entrada o de salida, lo que provoca que, todas las neuronas están en una única capa y las conexiones entre ellas son bidireccionales (todas las neuronas están conectadas entre sí pero con ellas mismas no).

El hecho de que las conexiones sean bidireccionales implica que sólo hace falta una conexión para unir dos neuronas. Dicha conexión simboliza una unión sináptica.

La red de Hopfield se trata de una red auto asociativa, y con valores de salida representados entre 0 y 1 o entre -1 y 1, ya que, su función de activación es de tipo sigmoidea.

El funcionamiento de la red de Hopfield se basa en que la red almacena una serie de información (patrones) durante la fase de aprendizaje, cuando se presenta a su entrada alguna de la información almacenada, la red tiende a evolucionar hasta estabilizarse, y ofrece a la salida la información almacenada que coincide con la entrada. Cuando la información mostrada a la entrada, no coincide con ninguna información almacenada dentro de la red, la red evoluciona generando como salida el patrón más parecido almacenado.

Para el buen funcionamiento de esta red, es necesario previamente haber codificado y representado como vector (con tantas componentes como neuronas tenga) la información (entrada), esta información se aplica a la única capa de la red (cada neurona recibe un elemento del vector de entrada). Cada neurona recibe como entrada las salidas de las demás neuronas, que serían (inicialmente) el valor de las entradas multiplicadas por los pesos de las conexiones. Por último, la suma de todos estos valores sería la entrada neta de la neurona, a la que se le aplica la función de transferencia y de activación.

Todo este proceso anterior se repite hasta que las salidas de las neuronas se estabilizan (dejen de cambiar de valor), el conjunto de todo valores corresponderá con algún valor almacenado en la etapa de aprendizaje. [13]

5.2.3 Modelo de Boltzman (Maquina de Boltzman)

La máquina de Boltzmann es una red neuronal de tipo estocástico. Las diferentes arquitecturas posibles de este tipo de red pueden resumirse en tres tipos principales de modelos de la máquina Boltzmann, siendo estos las redes "completion network"," input-output network", y la" feedforward network". La información aquí mencionada es traducción libre de [14].

Estas redes se componen por un conjunto de neuronas Z y un conjunto de conexiones C entre sí. Sus neuronas son de forma binarias y por lo normal, solo pueden tener dos valores de salida {0 1} que son conocidos como estados activos e inactivos. Una configuración K de la máquina de boltzman es un vector de longitud |Z|, tal que el elemento K (z) representa el estado de la neurona z. por lo tanto, se define el conjunto K, que contiene todas las configuraciones k posibles para la red. [14]

Cada conexión (Z1, Z2) de C tiene un peso sináptico asociado wzizj = Wzjzi Є R. estas conexiones se consideran activas cuando K (z1) como K (z2) son iguales a 1.

Por lo tanto, se define una función F conocida como función del coseno como una medida cuantitativa para medir la calidad de la configuración K de la red.

$$F(k) = \sum_{(z1,z2)} Wz1z2 * k(z1) * k\,(z2)$$

El objetivo de la máquina de boltzman es la maximización de la función del coseno, por lo que, se puede interpretar de su definición que tiende a las conexiones exitosas (wz1zj positivo) a ser activadas, y por lo contrario, a las conexiones inhibitorias (wzizj negativas) a ser evitadas.

5.2.3.1 Modelo Completion Network

La red completion network tiene una arquitectura parecida a la red Hopfield, con una diferencia fundamental que es la posibilidad de encontrar neuronas tanto visibles como ocultas, es decir que no es una arquitectura monocapa como propuso su creador Hopfield.

Las neuronas ocultas son aquellas no accesibles desde el exterior de la red, lo que implica que no reciben entradas externas al sistema ni son puntos de salida del mismo. Asimismo, las conexiones se establecen bidireccionalmente con pesos simétricos (Wzizj=Wzjzi).

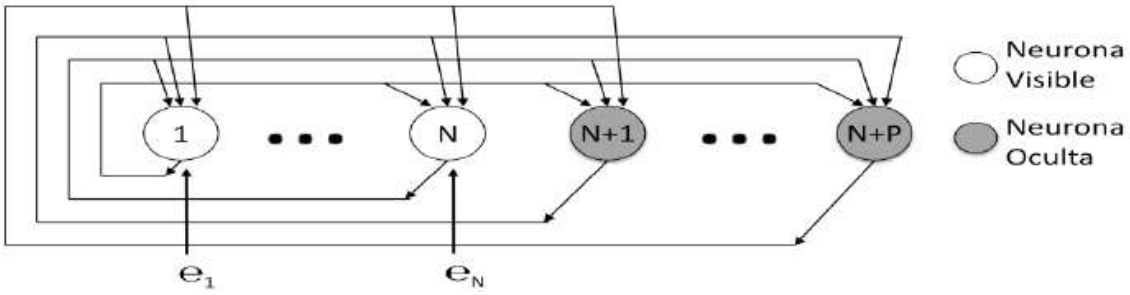

Figura 10: Representación modelo completion Network

5.2.3.2 Modelo input-output network

La red input-output al igual que la completion network tiene neuronas ocultas y visibles, las neuronas visibles en la capa de entrada y salida y las neuronas ocultas en una capa intermedia.

Este tipo de máquina de Boltzmann tiene tres tipos de conexiones. Las conexiones hacia adelante son aquellas entre la capa de entrada y la capa oculta, y entre la capa de entrada y la salida. Las conexiones en ambos sentidos son aquellas entre las neuronas ocultas y las de la capa de salida o de entrada. Finalmente, las conexiones laterales son aquellas entre neuronas de la misma capa. [15]

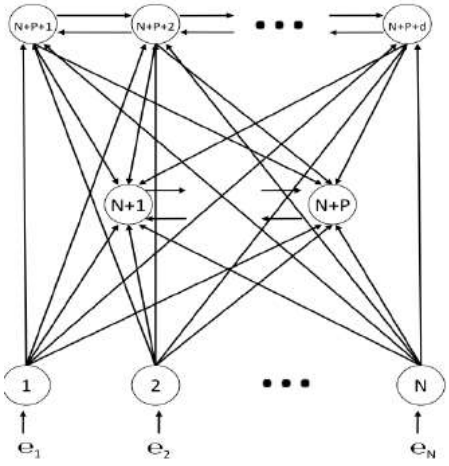

Figura 11: Representación de modelo Input-Output Network

5.2.3.3 Modelo Feed-forward network

Es una máquina de Boltzmann que cuente únicamente con conexiones hacia adelante, similar a un perceptrón de 3 capas.

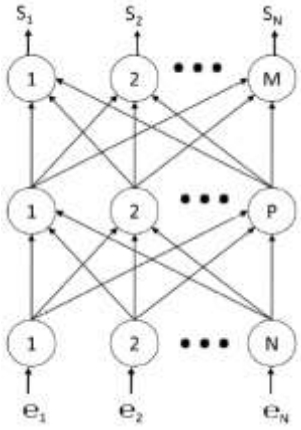

Figura 12: Representación de modelo Feed-forward network

5.2.4 Modelo BackPropagation

La BackPropagation es un tipo de red de aprendizaje supervisado de gradiente descendiente, que emplea un ciclo de propagación adaptación de dos fases.

Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de la capa oculta solo reciben una fracción de la señal total del error basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original.

Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total. Basándose en la señal de error percibida, donde se actualizan los pesos de conexión de cada neurona, para hacer que la red converja hacia un estado que permita clasificar correctamente todos los patrones de entrenamiento. [15]

La importancia de este proceso consiste en su capacidad de auto adaptar los pesos de las neuronas intermedias para aprender la relación que existe entre un conjunto de patrones dados como ejemplo y sus salidas correspondientes.

Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su

entrenamiento. Y a la inversa, las unidades de las capas ocultas tienen una tendencia a inhibir su salida si el patrón de entrada no contiene la característica para reconocer, para la cual han sido entrenadas.

Varias investigaciones han demostrado que, durante el proceso de entrenamiento, la red BackPropagation tiende a desarrollar relaciones internas entre neuronas con el fin de organizar los datos de entrenamiento en clases. Esta tendencia se puede extrapolar, para llegar a la hipótesis consistente en que todas las unidades de la capa oculta de una Back Propagation son asociadas de alguna manera a características específicas del patrón de entrada como consecuencia del entrenamiento. Lo que sea o no exactamente la asociación puede no resultar evidente para el observador humano, lo importante es que la red ha encontrado una representación interna que le permite generar las salidas deseadas cuando se le dan las entradas, en el proceso de entrenamiento. [15]

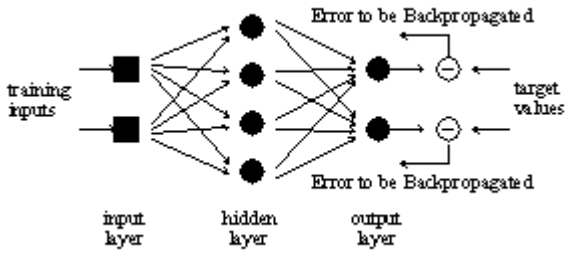

Figura 13: Representación Backpropagated

Esta misma representación interna se puede aplicar a entradas que la red no haya visto antes. La red clasificará estas entradas según las características que compartan con los ejemplos de entrenamiento.

Una variación del algoritmo backpropagation es conocido como Levenberg-Marquardt backpropagation; este muestra la convergencia más eficiente durante el proceso de entrenamiento de propagación hacia atrás porque actúa como un ajuste entre el método de optimización de primer orden (método de la máxima pendiente) y un método de optimización de segundo orden, estable pero de lenta convergencia, conocido como Gauss-Newton. Traducción libre de [15]

6 IMPLEMENTACIÓN DE RED

Para la implementación de la red neuronal es de vital importancia saber qué datos de entrada y qué datos de salida son los más apropiados para minimizar el error cuadrático medio MSE, el error medio absoluto MAE, o, simplemente, para minimizar la diferencia entre los target y los datos de salida.

Para ello, se realiza una comparativa sobre los errores anteriormente mencionados. Con ello podremos comprobar qué parámetros de entrada y de salida son los más apropiados para la implementación de la red. Por tanto, se crean redes neuronales con diferentes tipos de entradas y salidas y se analiza los resultados obtenidos.

Estas redes se definen con parámetros de configuración por defecto, con 1 capa oculta de 100 neuronas y función de transferencia tipo logic (se explica en el apartado 6.5), también mencionar que la capa de salida de la red está compuesta con una función de transferencia tipo plurig (se explica en el apartado 6.5).

Para realizar la comparativa de los errores producidos, se opta por dejar fijo el modo de entrenamiento de red neuronal. El modo elegido por sus características es el modo de entrenamiento trainscg.

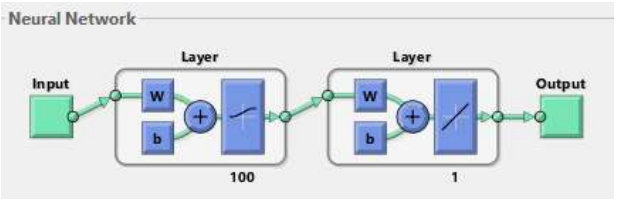

Figura 14: Red definida por defecto

Se decide realizar este análisis con los datos pertenecientes al centro de la ciudad ya que como indica la lógica, donde más dificultades hay para estimar la Wi-Fi es donde más señales de Wi-Fi hay. Por lo que el estudio de datos de entrada y salida solo se realiza para estos datos y la decisión tomada se acata para las demás ubicaciones.

Las posibles configuraciones de redes son las mencionadas a continuación.

6.1 Parámetros de entrada en coordenadas decimales.

Se implementan redes con parámetros de entada en coordenadas decimales. Estas redes se implementan con 2 entradas, el dato de latitud y el dato de longitud de la ubicación, y con diferentes salidas. Las salidas que se consideran para estas redes son los parámetros RSSI, los parámetros Ídem y el conjunto de los dos.

Por lo tanto se crean tres redes neuronales con diferentes salidas para comparar cuanto error se produce y en que entrenamiento se produce.

6.1.1 *Red 2 entradas con datos de entrada en coordenadas decimales y 1 salida con dato de salida RSSI*

Representación gráfica de red:

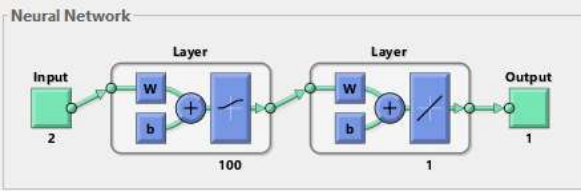

Figura 15: Red con parámetros de entrada en coord. Decimal y salida RSSI

Resultados de simulacion de red:

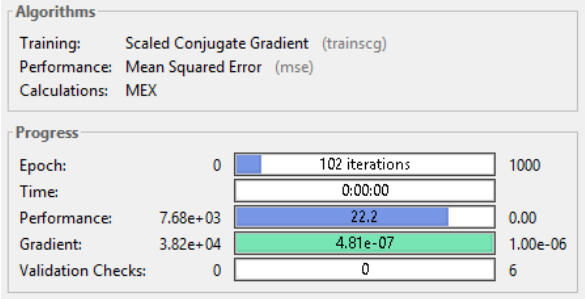

Figura 16: Resultados red con parámetros de entrada en coord. Decimal y salida RSSI

Representacion grafica de salida objetivo en color rojo y salida producida por la red en color azul:

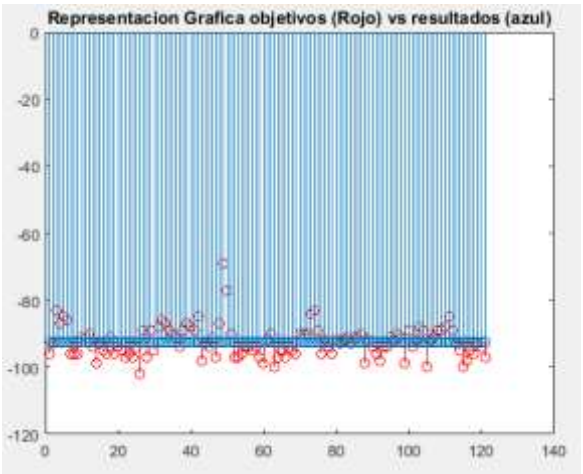

Figura 16: Representación gráfica de salida red coord. Decimal y salida RSSI

Representacion grafica del error cometido por dato:

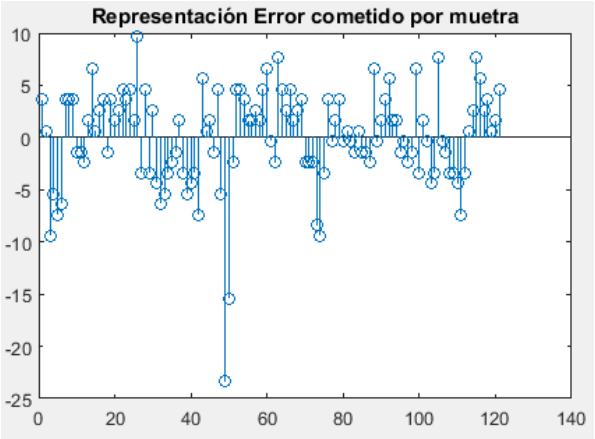

Figura 17: Representación gráfica de error en red coord. Decimal y salida RSSI

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red de 2 entradas, en formato decimal y 1 salida con objetivo RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Conclusión: se aprecia que rápidamente se alcanza el gradiente minimo y se detiene la simulación, esto provoca que el promedio cuadrado del error es muy alto y que los resultados de entrenamiento no sean satisfactorios.

6.1.2 *Red 2 entradas con datos de entrada en coordenadas decimales y 1 salida con dato de salida ídem*

Representación de la red:

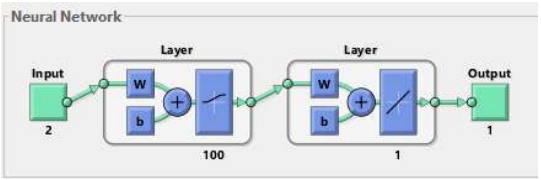

Figura 18: Red con parámetros de entrada en coord. Decimal y salida Ídem

Resultados de simulacion de red:

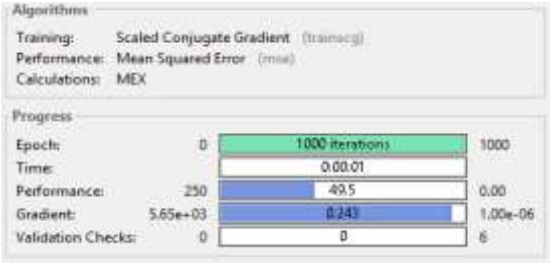

Figura 19: Resultados red con parámetros de entrada en coord. Decimal y salida Ídem

Representacion grafica de salida objetivo en color rojo y salida producida por la red en color azul:

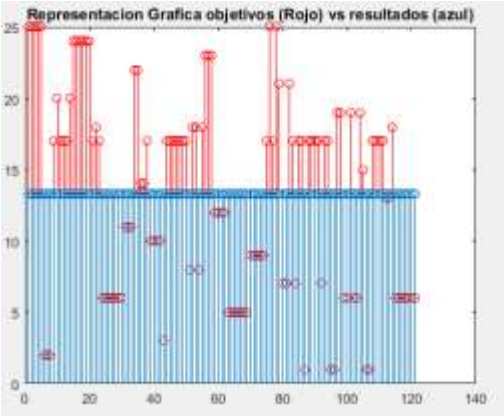

Figura 20: Representación gráfica de salida red coord. Decimal y salida Ídem

Representacion grafica del error cometido por dato:

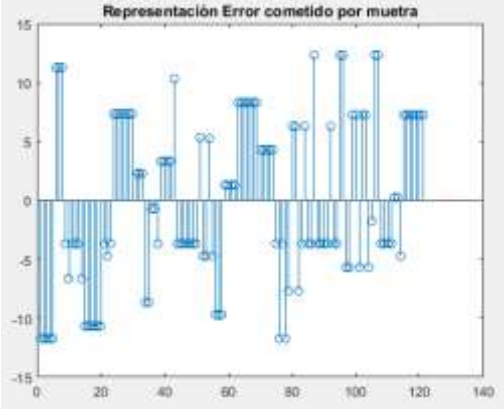

Figura 21: Representación gráfica de error en red coord. Decimal y salida Ídem

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red de 2 entradas, en formato decimal y 1 salida con objetivo idem
    Resultado MSE(Promedio cuadrado del error): 49.47
    Resultado MEA(Promedio absoluto del error): 6.25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Conclusiones: se aprecia que la simulación se detiene porque se alcanza el número de épocas establecido. El promedio cuadrado del error es de un nivel en torno a 50 y con este promedio se observa que los resultados obtenidos no son satisfactorios.

6.1.3 Red 2 entradas con datos de entrada en coordenadas decimales y 2 salidas con datos de salida RSSI e ídem.

Representación de la red:

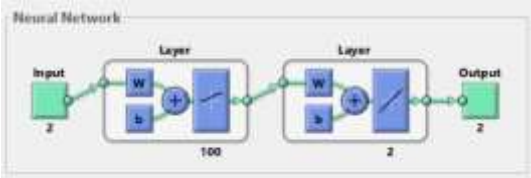

Figura 22: Red con parámetros de entrada en coord. Decimal y salidas RSSI e Ídem

Resultados de simulacion de red:

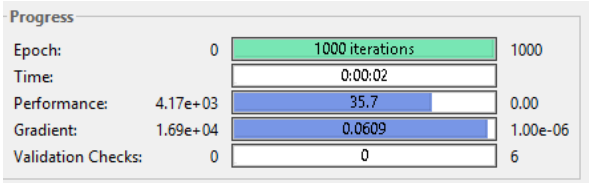

Figura 23: Resultados red con parám. de entrada en coord. Dec y salidas RSSI e Ídem

Representacion grafica de salida objetivo en color rojo y salida producida por la red en color azul:

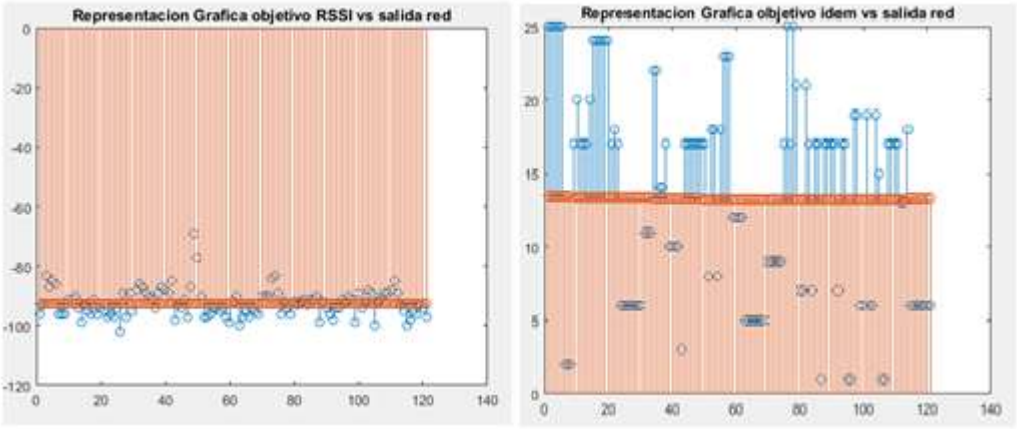

Figura 24: Representación gráfica de salida red coord. Decimal y salida Ídem

Representacion grafica del error cometido por dato:

RSSI e Idem

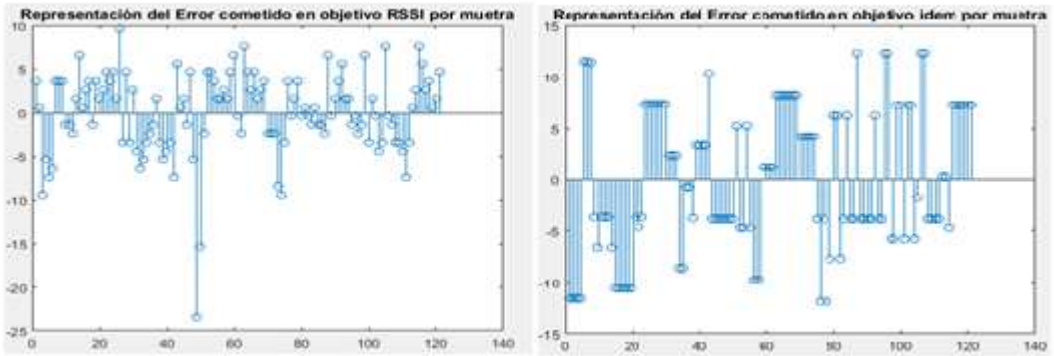

Figura 25: Representación gráfica de error en red coord. Dec y salidas RSSI e Ídem

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red de 2 entradas, en formato decimal y 2 salidas con objetivos RSSI e idem
    Resultado MSE(Promedio cuadrado del error): 35.71
    Resultado MEA(Promedio absoluto del error): 4.92
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Conclusiones: el entrenamiento acaba porque se alcanza el número de épocas máximo establecido produciéndose un promedio cuadrado de error bastante alto, en torno a un nivel 35 por lo que los resultados de simulación no son satisfactorios.

Como conclusión global del entrenamiento con datos de entrada en coordenadas decimales y variando los diferentes objetivos o datos de salida, se obtiene que se produce un error MSE bastante alto. Se cree que una de las posibles causas de que este error sea tan alto es la cercanía de los datos de entrada, ya que los datos de entrada no tienen variación hasta la quinta cifra significativa del valor. Otra causa podría ser las variables de entrada a la red, hasta ahora se está entrenando con 2 valores de entrada a la red, estos podrían ser muy pocas variables de entrada para una red con tan poca variación de datos.

6.2 Parámetros de entrada en coordenadas grados, minutos y segundos.

Para introducir los parámetros de entada en coordenadas grados, minutos y segundos se necesitan 6 entradas a la red, la entrada de latitud en grados, latitud en minutos, latitud en segundos, longitud en grados, longitud en minutos y longitud en segundos de la ubicación.

Como parámetros de salida se tienen la señal RSSI y el ídem de la Wi-Fi.

6.2.1 Red 6 entradas con datos de entrada en coordenadas decimales y 1 salida con dato de salida RSSI

Representación gráfica de red:

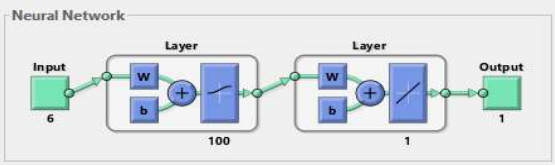

Figura 26: Red con parámetros de entrada en coord. y salida Grados RSSI

Resultados de simulacion de red:

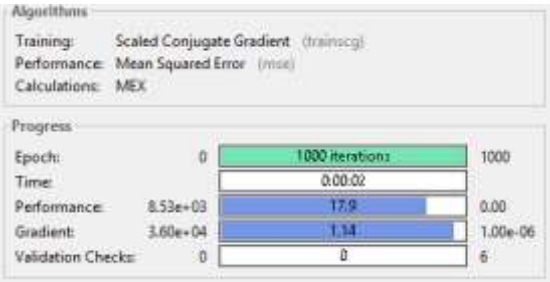

Figura 27: Resultados red con parám. de entrada en coord. Grados y salidas RSSI

Representacion grafica de salida objetivo en color rojo y salida producida por la red en color azul:

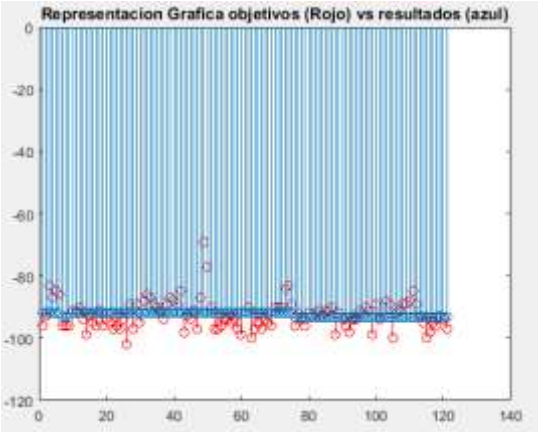

Figura 28: Representación gráfica de salida red coord. Grados y salida RSSI

Representacion grafica del error cometido por dato:

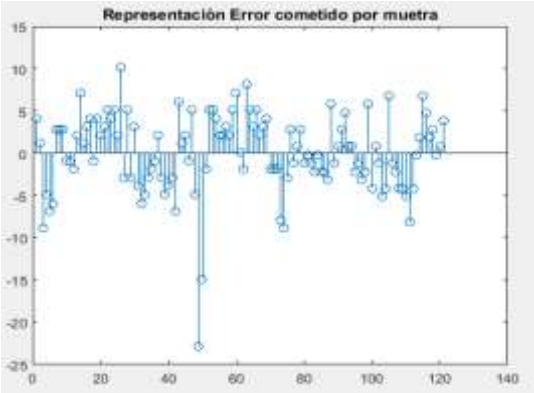

Figura 29: Representación gráfica de error en red coord. Grados y salida RSSI

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red de 6 entradas, en formato Grados y 1 salida con objetivo RSSI
    Resultado MSE(Promedio cuadrado del error): 17.89
    Resultado MEA(Promedio absoluto del error): 3.17
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Conclusiones: se observa que sigue habiendo promedio cuadrado de error alto, aunque menor que en los resultados proporcionados en 6.1.1., también el error promedio absoluto es menor en este entrenamiento y la simulación se detiene llegando al número de épocas contemplado.

6.2.2 Red 6 entradas con datos de entrada en coordenadas decimales y 1 salida con dato de salida ídem

Representación gráfica de red:

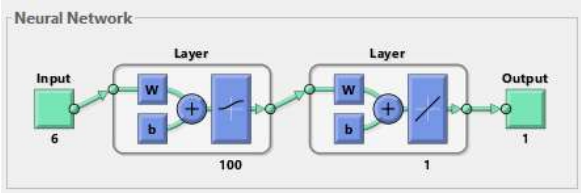

Figura 30: Red con parámetros de entrada en coord. y salida Grados Ídem

Resultados de simulacion de red:

Figura 31: Resultados red con parám. de entrada en coord. Grados y salidas Ídem

Representacion grafica de salida objetivo en color rojo y salida producida por la red en color azul:

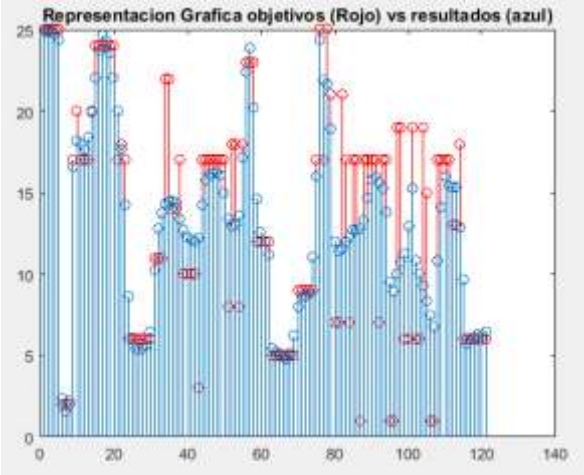

Figura 32: Representación gráfica de salida red coord. Grados y salida Ídem

Representacion grafica del error cometido por dato:

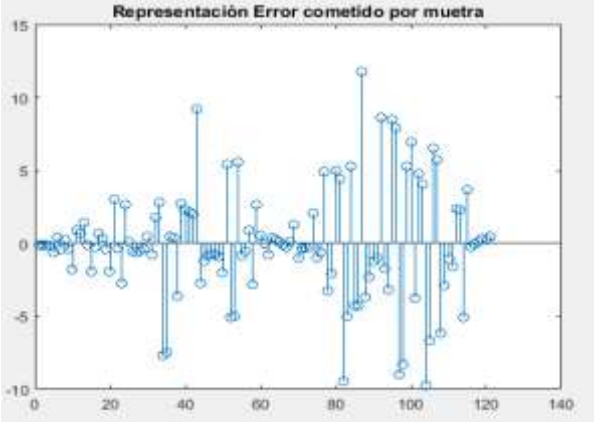

Figura 33: Representación gráfica de error en red coord. Grados y salida Ídem

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red de 6 entradas, en formato Grados y 1 salida con objetivo idem
    Resultado MSE(Promedio cuadrado del error): 12.64
    Resultado MEA(Promedio absoluto del error): 2.42
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Conclusiones: se observa que se tiene un promedio de error bastante menor que el producido en el apartado 6.1.2, también se observa que se produce un promedio absoluto de error algo inferior y que la simulación se detiene por llegar al número de épocas definido.

6.2.3 Red 2 entradas con datos de entrada en coordenadas decimales y 2 salidas con datos de salida RSSI e ídem.

Representación gráfica de red:

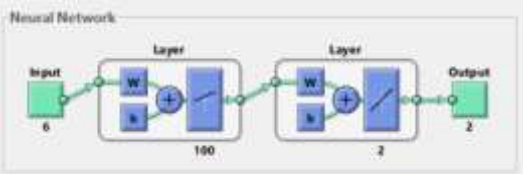

Figura 34: Red con parámetros de entrada en coord. Grad y salidas RSSI e Ídem

Resultados de simulacion de red:

Figura 35: Resultados red con parám. de ent. en coord. Grados y salidas RSSI e Ídem

Representacion grafica de salida objetivo en color rojo y salida producida por la red en color azul:

RSSI e Idem:

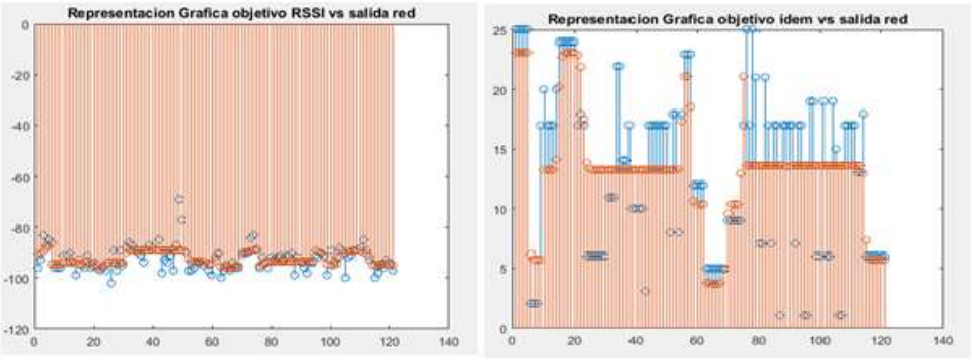

Figura 36: Representación gráfica de salida red coord. Grad y salidas RSSI e Ídem

Representacion grafica de salida de objetivos en color rojo y salida producida por la red en color azul:

Para RSSI e Idem

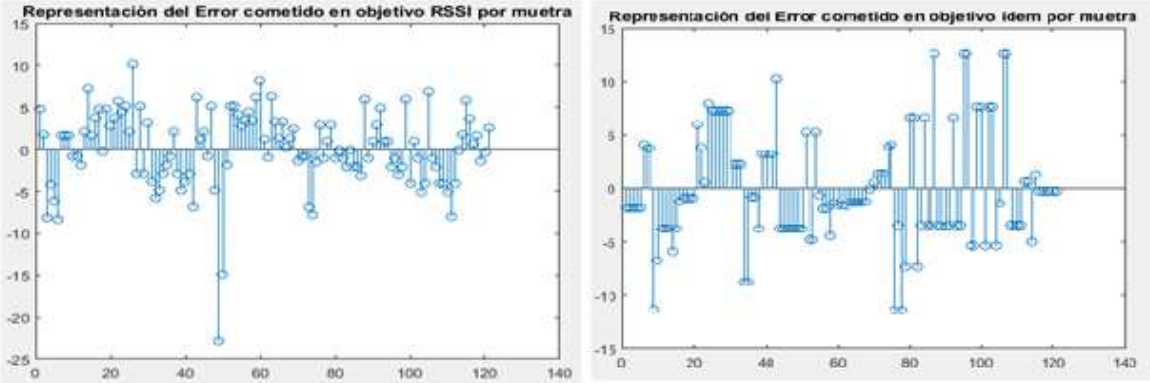

Figura 37: Representación gráfica de error en red coord. Dec y salidas RSSI e Ídem

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red de 6 entradas, en formato Grados y 2 salida con objetivo idem
    Resultado MSE(Promedio cuadrado del error): 20.38
    Resultado MEA(Promedio absoluto del error): 3.39
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
>>
```

Conclusiones: se observa que el promedio cuadrado de error producido en este entrenamiento es inferior al producido para el apartado 6.1.3 y que el error promedio absoluto es también algo inferior. Aunque los errores hayan disminuido, los resultados obtenidos no son satisfactorios.

6.3 Cuadro comparativo en función de datos de entrada, datos de salida y errores producidos

A continuación se representa en forma de tabla todos los resultados comentados en los apartados anteriores:

Entradas	Objetivos	MSE	MEA
2 Entradas (coordenadas decimales)	RSSI	49,47	6.25
2 Entradas (coordenadas decimales)	Ídem	22,17	3,60
2 Entradas (coordenadas decimales)	RSSI e Ídem	35,71	4,92
6 Entradas (coordenadas Grados, minutos, segundos)	RSSI	17,89	3,17
6 Entradas (coordenadas Grados, minutos, segundos)	Ídem	12,64	2,42
6 Entradas(coordenadas Grados, minutos, segundos)	RSSI e Ídem	20,38	3,39

Tabla 1: Conclusiones en función de datos y configuración de entrada y salida

Se aprecia que el mínimo valor de error producido tanto para el promedio cuadrado como para el promedio absoluto se produce cuando se consideran 6 parámetros de entrada a la red neuronal y como parámetro de salida 1 objetivo. Los seis parámetros de entrada serían las coordenadas en grados, minutos y segundos, y el objetivo sería el valor de ídem de la red Wi-Fi.

6.4 Conclusiones para implementación de algoritmo.

Tras analizar todos los entrenamientos en los apartados anteriores y ver con qué tipo de dato de entrada y que tipo de dato de salida se consigue tener mejores resultados, se llega a la conclusión de que para realizar el algoritmo de estimación de señal Wi-Fi con redes neuronales es necesario tener los datos de entrada en coordenadas de tipo grados, segundos y minutos.

Para minimizar lo máximo posible el error, se decide implementar y evaluar por separado los objetivos de RSSI e Ídem dentro del mismo algoritmo. Es decir con los mismos datos de entrada, realizar dos entrenamientos y simulaciones, uno para obtener el valor de RSSI y otro para obtener el ídem de la señal Wi-Fi a estimar.

6.5 Parámetros intermedios a tener en cuenta en las redes

Los parámetros intermedios a tener en cuenta para las redes neuronales son los pesos, las funciones de transferencia y el umbral.

Para la implementación de las redes, se optó por la utilización del algoritmo Blackpropagation de tipo feedforward, ya que de las opciones contempladas en el apartado 5.2.es la que mejores condiciones y prestaciones ofrecía.

La arquitectura feedforward de múltiples capas (multiplayer) ofrece diferentes posibilidades de funciones de transferencia.

Función de trasferencia logarítmica logsig (Figura 38), genera salidas entre 0 y 1, las entradas de las neuronas puede ir desde infinitamente negativo a infinitamente positivo (cualquier valor es posible).

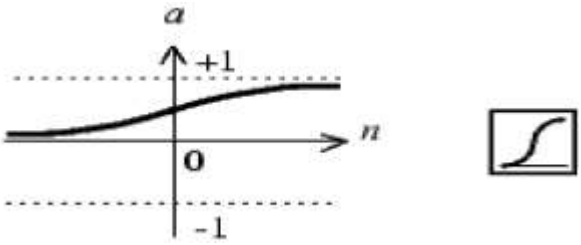

Figura 38: Función de transferencia logsig

Otra función que se permite utilizar es la función de transferencia tangencial o tasing (Figura 39).

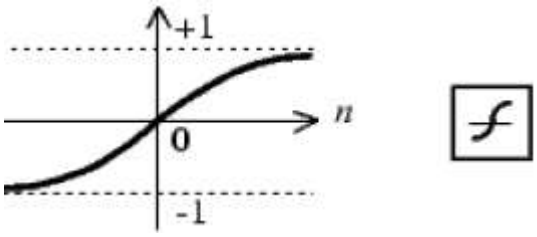

Figura 39: Función de transferencia tasing

Y por último, la función de traslado lineal o purelin (Figura 40). Esta función es muy utilizada cuando se quiere que las salidas de la red sean de forma lineal, es decir, que puedan tomar cualquier valor. Las anteriores funciones escalarían las salidas de la red a un rango más pequeño.

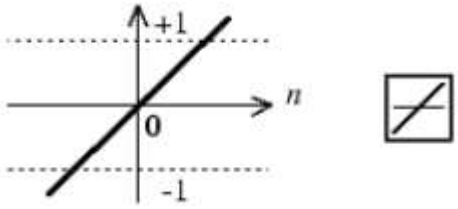

Figura 40: Función de transferencia pureling

Las redes utilizadas en este TFG se han implementado con las funciones de transferencia logsig para las capas ocultas, y purelin para la capa de salida.

Al utilizar la función logsig en las capas ocultas y purelin para la capa de salida, la red puede aprender relaciones lineales y no lineales entre la entrada y la salida, y al utilizar en la capa de salida la función purelin, se pueden producir valores fuera del umbral +1 y -1.

Una vez creada la red e inicializados los pesos y el umbral (tanto los pesos como el umbral se inicializan de forma automática), se procede a su entrenamiento.

Durante el entrenamiento, los pesos y el umbral son iterativamente ajustados para minimizar la función de desempeño o como aparece en las representaciones mse, esta función se define como el promedio cuadrado del error entre los rendimientos de la red y los rendimientos designados (objetivos).

Todos los algoritmos que se mencionaran en capítulos posteriores, utilizan la función de desempeño para la determinación del ajuste de los pesos y del umbral.

6.6 Implementación Red BackPropagation y sus funciones

Una vez descrito en el apartado anterior como se obtienen los pesos, umbrales y las funciones de transferencia a utilizar, se describe como se implementan las redes.

En primer lugar se utiliza la función newff para crear una red feedforward, esta función requiere cuatro parámetros de entrada. El primer parámetro consta de una matriz del mínimo y máximo valor para cada uno de los elementos del vector de entrada. El segundo parámetro es un array que contiene los tamaños de cada capa, es decir, la cantidad de neuronas por capa. El tercer parámetro es un array que contiene los nombres de las funciones de transferencia que serán usadas para cada capa. El último parámetro contiene el nombre de la función de entrenamiento que se utilizará en la red.

Utilizando la función newff de la forma anterior descrita, se crea la red y también se inicializan los pesos y el umbral de la red. De esta forma la red estaría lista para ser entrenada.

Como se comentó anteriormente, durante el entrenamiento los pesos y el umbral son ajustados para minimizar la función de desempeño. La función de desempeño que utilizamos es el promedio cuadrado del error MSE. Hay dos maneras para las que este algoritmo de descenso de gradiente puede llevarse a cabo; por modo incremental o el modo por lote.

En el modo incremental el gradiente se computa y los pesos se actualizan después de cada entrada que se aplica a la red.

En el modo por lote todas las entradas se aplican a la red antes que los pesos se actualicen, es decir, se actualizan solo los pesos y el umbral de la red después de que el entrenamiento entero haya sido aplicado a la red y se suman los gradientes cuadrados en cada entrenamiento para determinar el cambio de los pesos y el umbral.

En este TFG se van a realizar tres diferentes entrenamientos, para cada entrenamiento se utilizará una diferente función de entrenamiento. Entre todas las funciones existentes para el entrenamiento de redes neuronales, se han elegido las tres funciones que se explican a continuación:

-Algoritmo descenso de gradiente por entrenamiento incremental (traingd)

-Algoritmo descenso de gradiente con técnicas heurísticas; algoritmo de rezago (trainrp)

-Algoritmo de gradiente conjugado escalado (trainscg)

Con cada una de estas tres funciones se realiza un estudio en función de sus parámetros de configuración (capas ocultas, numero de neuronas, numero de épocas, etc...) Para obtener cual es la mejor opción de función de entrenamiento para estimar redes Wi-Fi mediante redes neuronales.

6.6.1 Algoritmo de descenso de gradiente por entrenamiento incremental (traingd)

El entrenamiento de disminución de gradiente o por pasos se ejecuta con la función traingd. En este entrenamiento se actualizan los pesos y el umbral en dirección negativa al gradiente de la función de desempeño.

Hay siete parámetros de entrenamiento asociados al traingd: epochs, show, goal, time, min_grad, max_fail, y mc. El mc da como resultado la proporción de aprendizaje, este se obtiene multiplicando el tiempo por el negativo del gradiente para determinar los cambios a los pesos y el umbral. Si la proporción de aprendizaje se hace demasiado grande, el algoritmo se vuelve inestable. Si la proporción de aprendizaje se fija demasiado pequeña, el algoritmo toma un largo tiempo para converger.

El estado de entrenamiento se muestra para cada iteración. Los demás parámetros determinan cuando finaliza el entrenamiento. El entrenamiento se detiene si el número de iteraciones excede "épocas", si los decrementos de la función de desempeño están por debajo del nivel "goal," si la magnitud del gradiente está en menos del mingrad, o si el tiempo de entrenamiento es más largo que el tiempo establecido.

6.6.2 Algoritmo descenso de gradiente con técnicas heurísticas; algoritmo de rezago (trainrp)

Este tipo de algoritmos de descenso de gradiente con técnicas heurísticas forman parte de los entrenamientos rápidos. Es decir son una versión de los entrenamientos con descenso de gradiente con velocidad adquirida. Estos pueden converger hasta 10 veces más rápidos que los anteriores. Las técnicas heurísticas desarrollan un análisis para la optimización del algoritmo de descenso de gradiente normal.

En las redes multicapa se usan funciones de transferencia sigmoideas en la capa oculta, estas funciones comprimen un rango de la entrada infinito en uno finito. Las funciones sigmoideas son caracterizadas por que tienden a acercarse a ceros cuando la entrada es más grande. Esto es un problema cuando se usan algoritmos de descenso para entrenar una red multicapa con funciones sigmoideas, porque el gradiente puede tener una magnitud muy pequeña y por consiguiente, causar cambios pequeños en los pesos y el umbral, aunque los pesos y el umbral están lejos de sus valores óptimos.

El propósito del algoritmo de entrenamiento de rezago (Rprop) es eliminar los efectos de las magnitudes de los derivados parciales. La señal del derivado se usa para determinar la dirección de la actualización del peso, la magnitud del derivado no tiene efecto en la actualización del peso. El tamaño del cambio de peso es determinado por un valor de actualización aparte. El valor de actualización para cada peso y el umbral es aumentado por

un factor delt_inc, siempre que el derivado de la función de desempeño con respecto a ese peso tenga la misma señal para dos iteraciones sucesivas. El valor de actualización es disminuido por un factor delt_dec siempre que el derivado con respeto que el peso cambie la señal de la iteración anterior. Si el derivado es cero, entonces los restos de valor de actualización son los mismos. Siempre que los pesos estén oscilando el cambio de peso se reducirá. Si el peso continúa cambiando en la misma dirección para varias iteraciones, entonces la magnitud del cambio de peso se aumenta.

6.6.3 Algoritmo de gradiente conjugado escalado (trainscg)

El algoritmo del backpropagation básico ajusta los pesos en la dirección de descenso (negativo del gradiente). Ésta es la dirección en la que la función de desempeño disminuye rápidamente. Aunque la función disminuya rápidamente a lo largo del negativo del gradiente, esto necesariamente no produce una convergencia más rápida.

En los algoritmos de gradiente conjugado una búsqueda se realiza a lo largo de direcciones conjugadas que producen generalmente una convergencia más rápida que las direcciones de descenso.

En la mayoría de los algoritmos de gradiente conjugado, el tamaño del paso se ajusta en cada iteración. Una búsqueda se hace a lo largo de la dirección de gradiente conjugado para determinar el tamaño del paso que minimiza la función de desempeño.

Hay cuatro variaciones diferentes de algoritmos de este tipo; actualización de fletcher-Reeves (traincgf), actuaización de Polak-Ribiare (traincgp), restablecimiento de Powell-beale (traincgb) y por último y el que se va a utilizar el gradiente conjugado escalado (trainscg).

Para el entrenamiento con gradiente conjugado se necesita un alto nivel computacional debido a que el gradiente conjugado requiere una búsqueda lineal en cada iteración. El algoritmo de gradiente conjugado escalado (SCG) fue diseñado para evitar la búsqueda lineal. La idea básica de este algoritmo es combinar la aproximación de región de modelo-confianza con el gradiente conjugado.

Para la creación de la red con este algoritmo de gradiente escalado se necesitan definir los parámetros de entrenamiento de epochs, show, goal, time, min_grad, max_fail, sigma, lambda. El parámetro sigma determina el cambio en el peso para la segunda aproximación derivativa. El parámetro lambda regula la indecisión.

7 ENTRENAMIENTO Y VALIDACIÓN DE RED NEURONAL DESARROLLADA

En el capítulo seis se demostró que los datos de entrada para minimizar el error ocasionado a la hora de implementar la red neuronal tendrían que ser los datos en coordenadas grados, minutos y segundos. También se demostró que para la estimación de red WI-FI, los resultados son más satisfactorios si se entrenaban por separado los objetivos buscados.

En este capítulo se pretende realizar un estudio de cómo afecta el cambio en la función de entrenamiento a los errores evaluados, es decir, se busca ver que función de entrenamiento para redes neuronales ofrece menores errores y por lo consiguiente, mejores resultados para la estimación de red WI-FI. Para continúan con la metodología contemplada en capítulos anteriores, se utilizan como valores de entrada los referentes a los datos del centro de la ciudad de Seattle y como objetivos o datos de salida los valores de RSSI e Ídem.

El funcionamiento de las tres funciones que se pretende analizar son las siguientes:

-Traingd (entrenamiento incremental)

-Trainrp (técnica heurística; algoritmo de rezago)

-Trainscg (Algoritmo de gradiente conjugado escalado)

Estas tres funciones fueron descritas en el capítulo 6 tanto su forma de implementarlas como su funcionalidad. Para cada función se realizaran tres variaciones en sus parámetros de configuración. Estas tres variaciones son:

-número de capas ocultas

-número de neuronas por capa

-número de épocas.

Para realizar la comparativa de errores producidos por los diferentes métodos se evalúan los errores de promedio cuadrático de error MSE, el error promedio absoluto MAE, y también la diferencia entre los target y los datos de salida.

Una vez realizado todos estos entrenamientos y ver que función y con que parámetros se producen menos errores, se procederá a implementar la red neuronal definitiva con la que se realizaran las simulaciones para la estimación de red WI-FI, todo esto se contempla en el capítulo siguiente.

7.1 Estudio comparativo con función Traingd

Se procede a realizar los entrenamientos y comparaciones de los resultados obtenidos partiendo de una red neuronal creada por defecto y con la función de entrenamiento Traingd.

Es necesario mantener los mismos parámetros de entrada y de salida para todas las variaciones de entrenamientos realizadas. Se entrenará por separado los diferentes objetivos, es decir, habrá dos redes, una para los objetivos de RSSI y otra para los objetivos de Ídem. Los parámetros para cada red son:

NettraingdRSSI:

Entradas; Lat_Grados, Lat_min, Lat_seg, Lon_Grados, Lon_min, Lon_seg

Salida: RSSI

Nettraingdidem:

Entradas; Lat_Grados, Lat_min, Lat_seg, Lon_Grados, Lon_min, Lon_seg

Salida: ídem

7.1.1 Caso 1 de entrenamiento variando capas ocultas

Con los datos de entrada mencionados anteriormente se procede a realizar simulaciones de las redes neuronales. La primera simulación se realiza con todos los parámetros por defecto menos el número de capas ocultas, que se modificará con los valores de 1, 2 y 3 capas ocultas en dicha red.

7.1.1.1 A continuación se muestran los resultados obtenidos con función traingd con 1 capa oculta y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

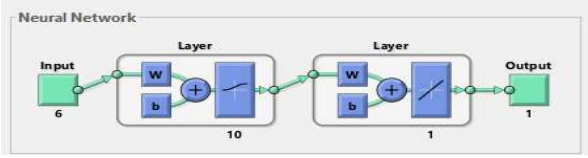

Figura 41: Red 1 capa oculta objetivo RSSI, función traingd

Resultados de simulacion de red:

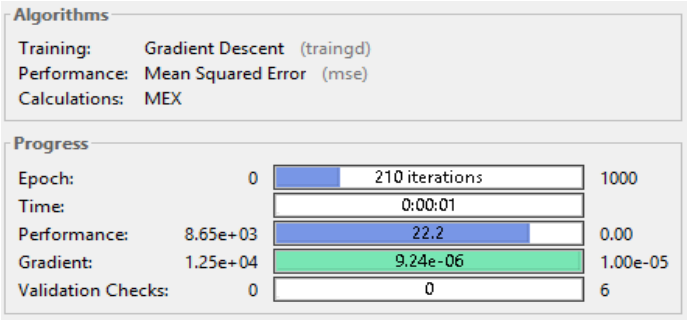

Figura 42: Resultados red 1 capa oculta objetivo RSSI, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

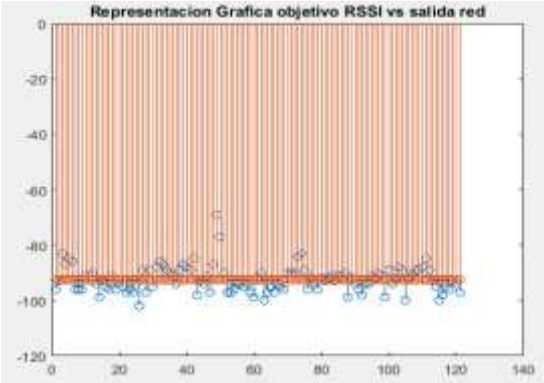

Figura 43: Representación salida red 1 capa oculta objetivo RSSI, función traingd

Representacion grafica del error cometido por dato:

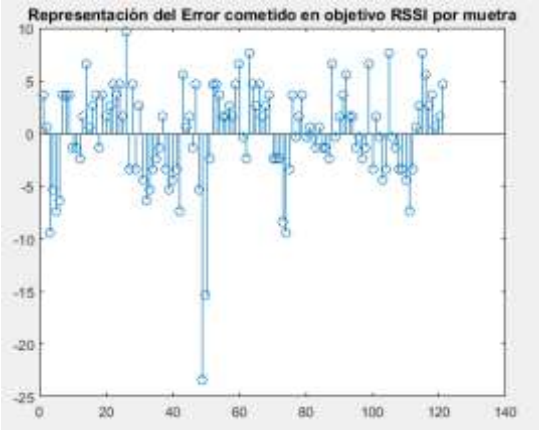

Figura 44: Representación error red 1 capa oculta objetivo RSSI, función traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

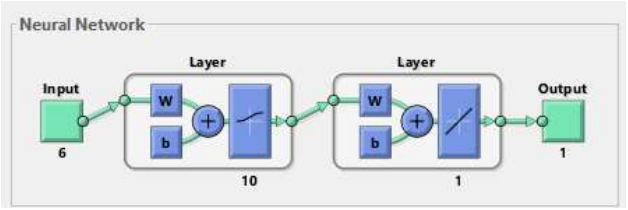

Figura 45: Red 1 capa oculta objetivo Ídem, función traingd

Resultados de simulacion de red:

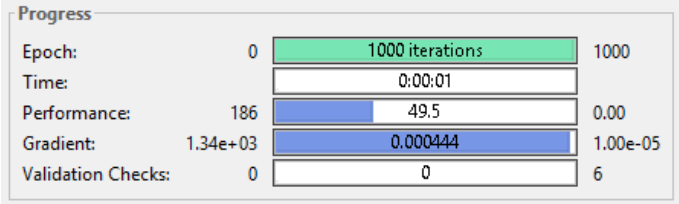

Figura 46: Resultados red 1 capa oculta objetivo Ídem, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

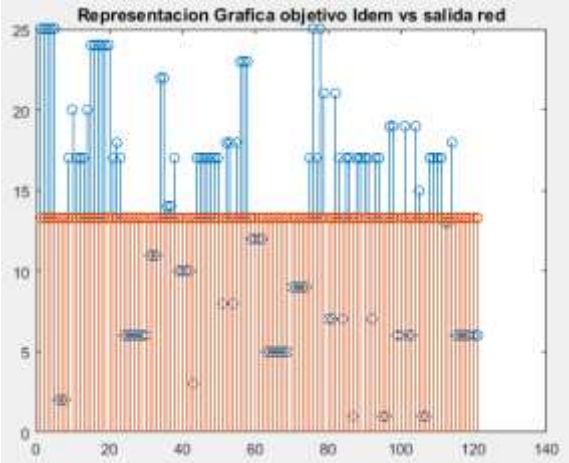

Figura 47: Representación salida red 1 capa oculta objetivo Ídem, función traingd

Representacion grafica del error cometido por dato:

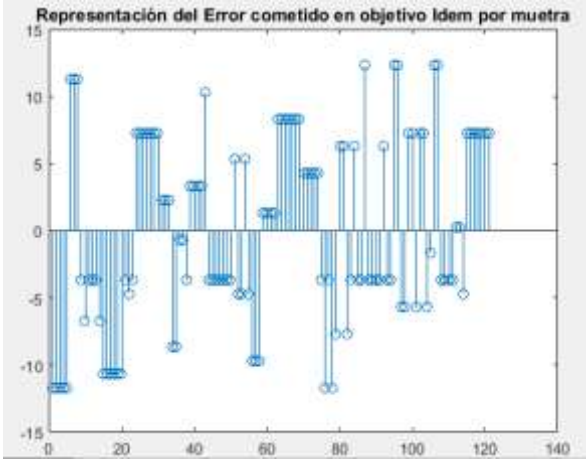

Figura 48: Representación error red 1 capa oculta objetivo Ídem, función traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 49.50
    Resultado MEA(Promedio absoluto del error): 6.25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.1.2 A continuación se muestran los resultados obtenidos con función traingd con 2 capas ocultas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

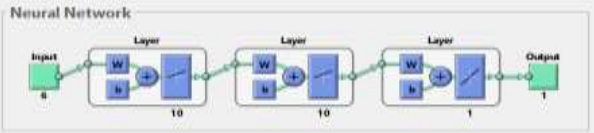

Figura 49: Red 2 capas ocultas objetivo RSSI, función traingd

Resultados de simulacion de red:

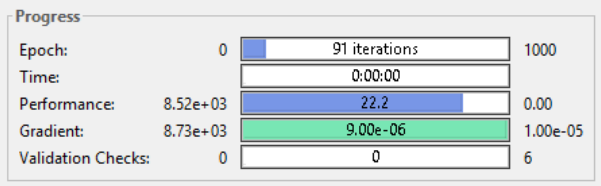

Figura 50: Resultados red 2 capas ocultas objetivo RSSI, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

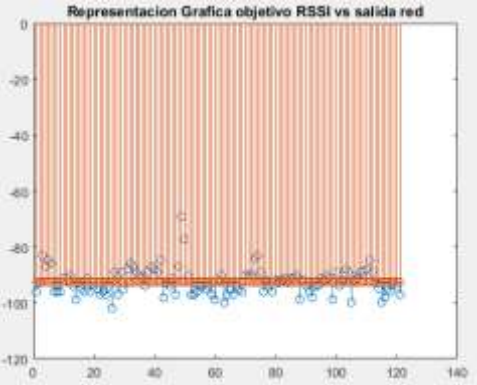

Figura 51: Representación salida red 2 capas ocultas objetivo RSSI, función traingd

Representacion grafica del error cometido por dato:

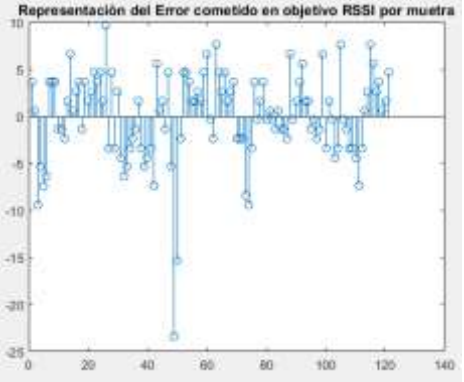

Figura 52: Representación error red 2 capas ocultas objetivo RSSI, función traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 2 capa oculta. Salida RSSI:
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

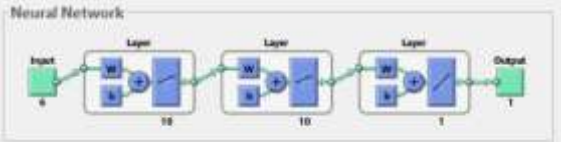

Figura 53: Red 2 capas ocultas objetivo Ídem, función traingd

Resultados de simulacion de red:

Figura 54: Resultados red 2 capas ocultas objetivo Ídem, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

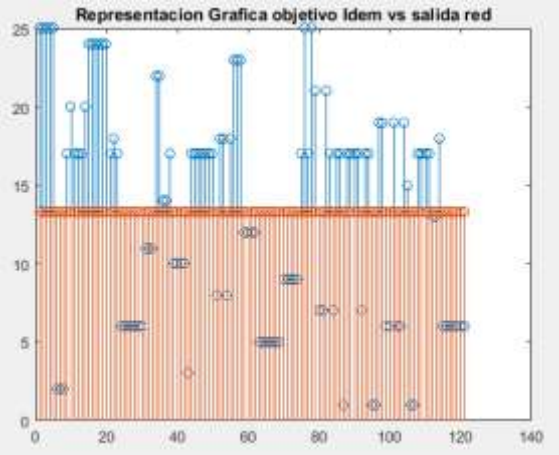

Figura 55: Representación salida red 2 capas ocultas objetivo Ídem, función traingd

Representacion grafica del error cometido por dato:

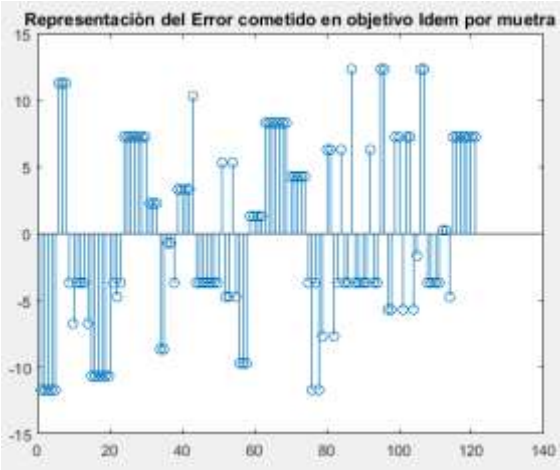

Figura 56: Representación error red 2 capas ocultas objetivo Ídem, función traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 2 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 49.50
    Resultado MEA(Promedio absoluto del error): 6.25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.1.3 A continuación se muestran los resultados obtenidos con función traingd con 3 capas ocultas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

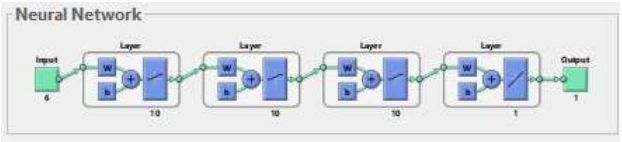

Figura 57: Red 3 capas ocultas objetivo RSSI, función traingd

Resultados de simulacion de red:

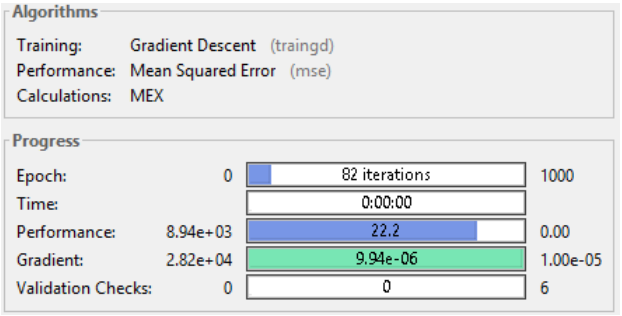

Figura 58: Resultados red 3 capas ocultas objetivo RSSI, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

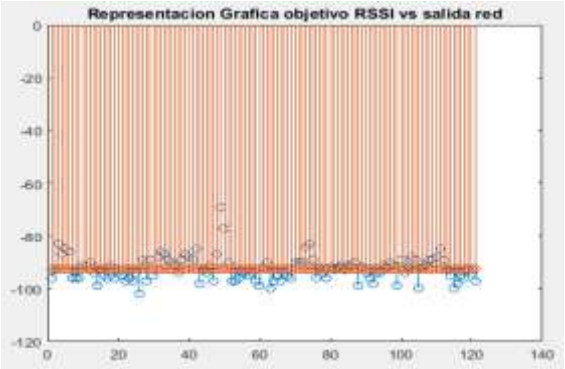

Figura 59: Representación salida red 3 capas ocultas objetivo RSSI, función traingd

Representacion grafica del error cometido por dato:

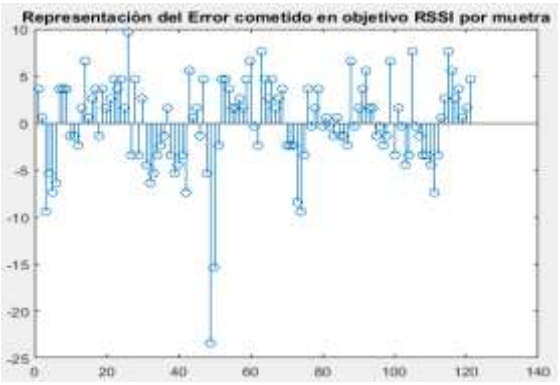

Figura 60: Representación error red 3 capas ocultas objetivo RSSI, función traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 3 capa oculta. Salida RSSI:
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

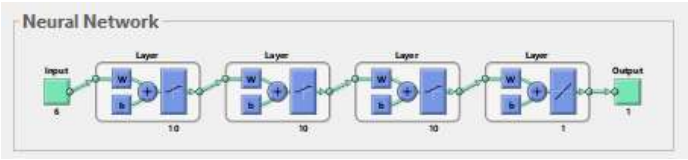

Figura 61: Red 3 capas ocultas objetivo Ídem, función traingd

Resultados de simulacion de red:

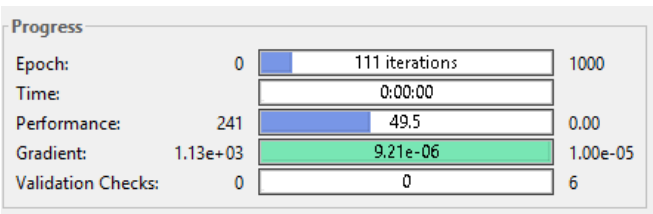

Figura 62: resultados red 3 capas ocultas objetivo Ídem, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

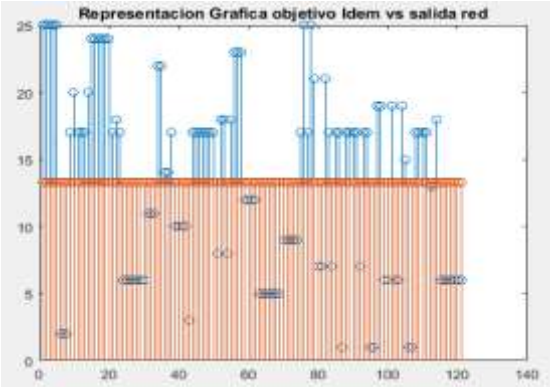

Figura 63: Representación salida red 3 capas ocultas objetivo Ídem, función traingd

Representacion grafica del error cometido por dato:

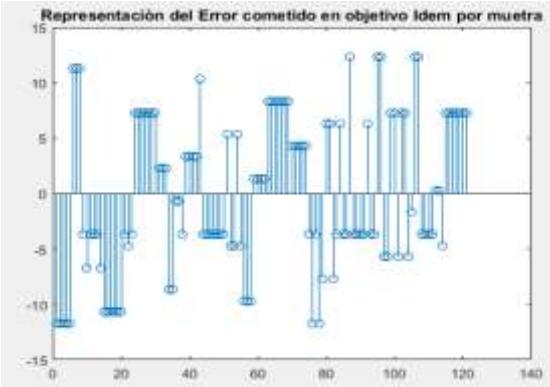

Figura 64: Representación error red 3 capas ocultas objetivo RSSI, función traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 3 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 49.50
    Resultado MEA(Promedio absoluto del error): 6.25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.1.4 Conclusiones y cuadro comparativo

Capas Ocultas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	RSSI	22.17	3.60
1	Idem	49.50	6.25
2	RSSI	22.17	3.60
2	Idem	49.50	6.25
3	RSSI	22.17	3.60
3	Idem	49.50	6.25

Tabla 2: Conclusiones según nº capas ocultas y función Traingd

Se observa que para este metodo de entrenamiento con nuestras caracteristicas no afecta el numero de capas ocultas a la hora de implementar la red. Por lo que se decide continuan con las pruebas con una capa oculta para las redes.

7.1.2 Caso 2 de entrenamiento variando nº de neuronas por capa

Como se acaba de comentar en el apartado anterior, se continúa con la realización de las pruebas con una red con 1 capa oculta.

En este apéndice se pretende variar el número de neuronas por capa oculta. Se van a realizar tres entrenamientos con 100 neuronas, 200 neuronas y 500 neuronas.

7.1.2.1 A continuación se muestran los resultados obtenidos con función traingd con 1 capa oculta, 100 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

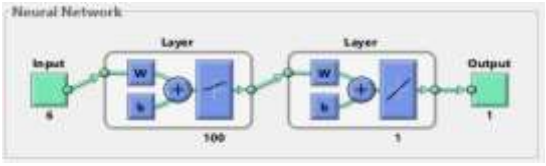

Figura 65: Red 1 capa oculta, 100 neuronas, objetivo RSSI, función traingd

Resultados de simulacion de red:

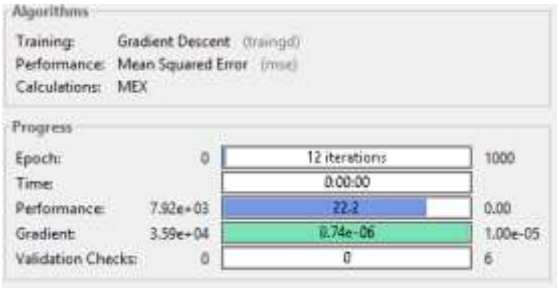

Figura 66: Resultados red 1 capa oc. 100 neuronas, objetivo RSSI, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

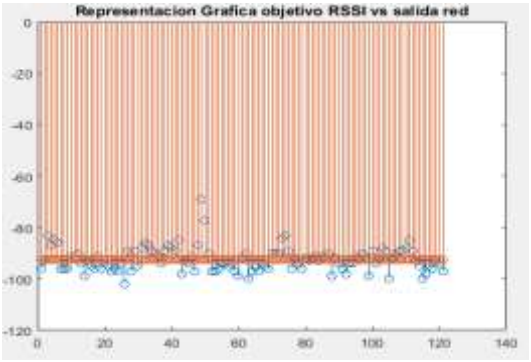

Figura 67: Representación salida red 1 cap. oc.100 neuronas obj. RSSI, func. traingd

Representacion grafica del error cometido por dato:

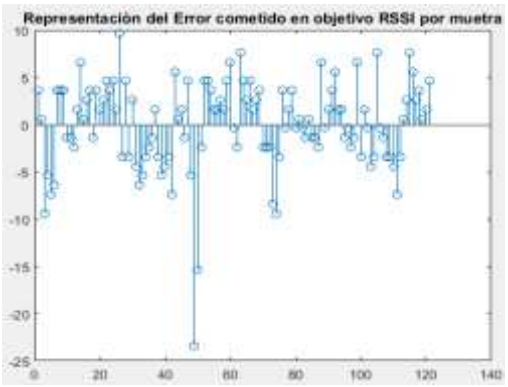

Figura 68: Representación error red 1 cap. oc.100 neuronas obj. RSSI., func. traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 100 neuronas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

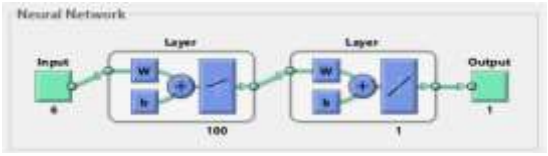

Figura 69: Red 1 capa oculta, 100 neuronas, objetivo Ídem, función traingd

Resultados de simulacion de red:

Figura 70: Resultados red 1 capa oculta, 100 neuronas, objetivo Ídem, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

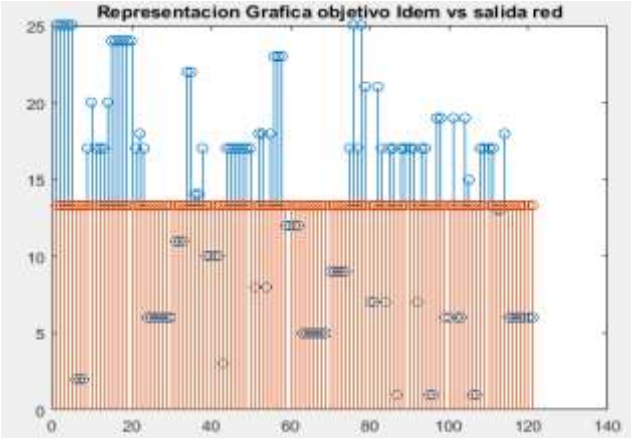

Figura 71: Representación salida red 1 cap. oc.100 neuronas obj. Ídem, func. traingd

Representacion grafica del error cometido por dato:

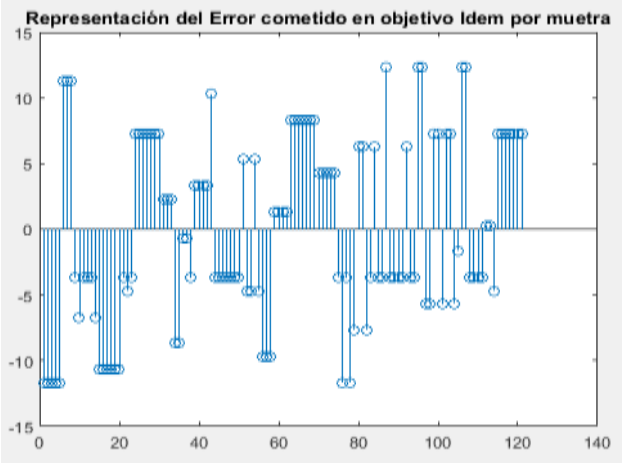

Figura 72: Representación error red 1 cap. oc.100 neuronas obj. Ídem, func. traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 100 neuronas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 49.50
    Resultado MEA(Promedio absoluto del error): 6.25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.2.2 A continuación se muestran los resultados obtenidos con función traingd con 1 capa oculta, 200 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

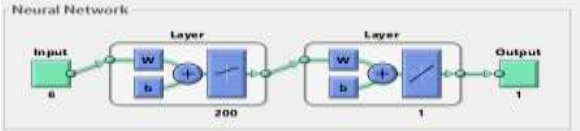

Figura 73: Red 1 capa oculta, 200 neuronas, objetivo RSSI, función traingd

Resultados de simulacion de red:

Figura 74: Resultados red 1 capa oculta, 200 neuronas, obj. RSSI, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

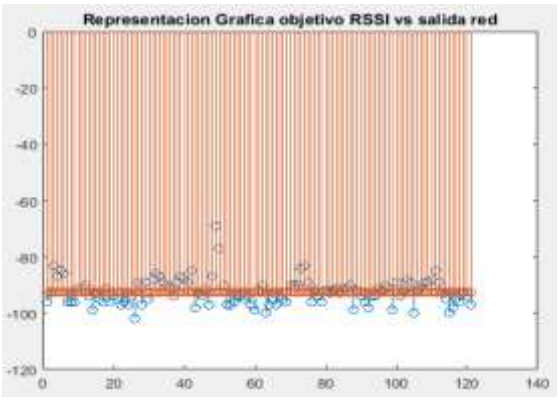

Figura 75: Representación salida red 1 cap. oc.200 neuronas obj. RSSI, func. traingd

Representacion grafica del error cometido por dato:

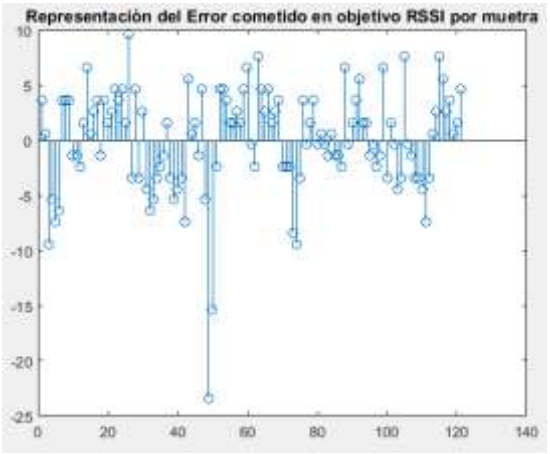

Figura 76: Representación error red 1 cap. oc.200 neuronas obj. RSSI, func. traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 200 neuronas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

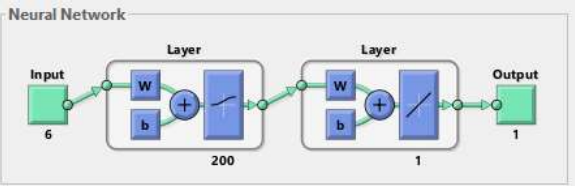

Figura 77: Red 1 capa oculta, 200 neuronas, objetivo Ídem, función traingd

Resultados de simulacion de red:

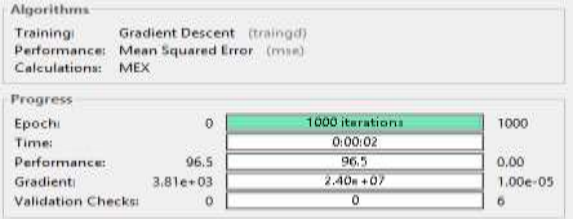

Figura 78: Resultados red 1 capa oculta, 200 neuronas, objetivo Ídem, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

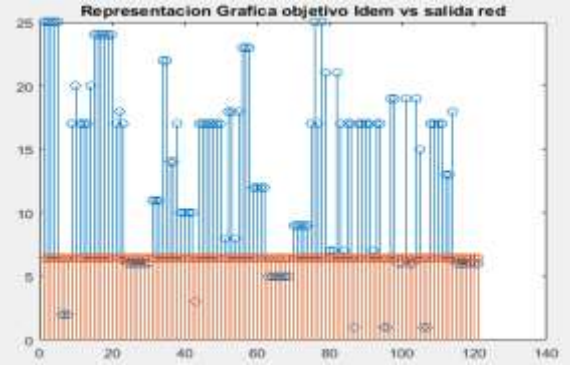

Figura 79: Representación salida red 1 cap. oc.200 neuronas obj. Ídem, func. traingd

Representacion grafica del error cometido por dato:

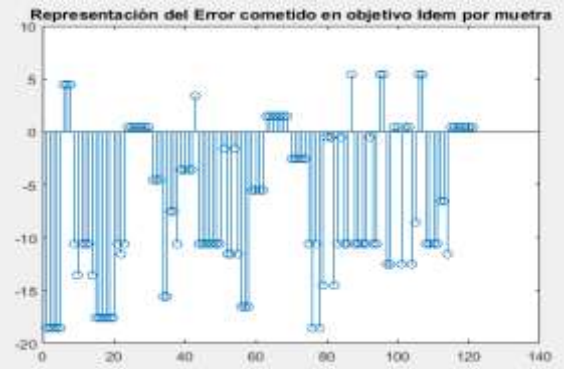

Figura 80: Representación error red 1 cap. oc.200 neuronas obj. Ídem, func. Traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 200 neuronas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 96.52
    Resultado MEA(Promedio absoluto del error): 7.88
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.2.3 A continuación se muestran los resultados obtenidos con función traingd con 1 capa oculta, 500 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

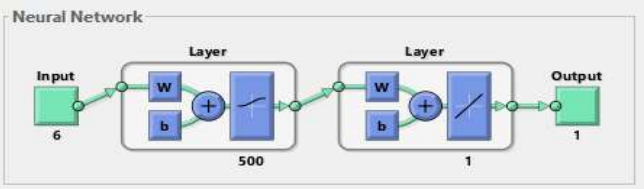

Figura 81: Red 1 capa oculta, 500 neuronas, objetivo RSSI, función traingd

Resultados de simulacion de red:

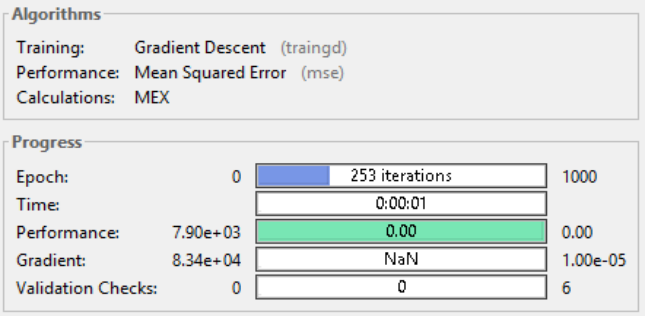

Figura 82: Red 1 capa oculta, 500 neuronas, objetivo RSSI, función traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 500 neuronas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): NaN
    Resultado MEA(Promedio absoluto del error): NaN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

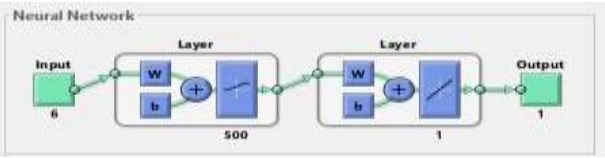

Figura 83: Red 1 capa oculta, 500 neuronas, objetivo RSSI, función traingd

Resultados de simulacion de red:

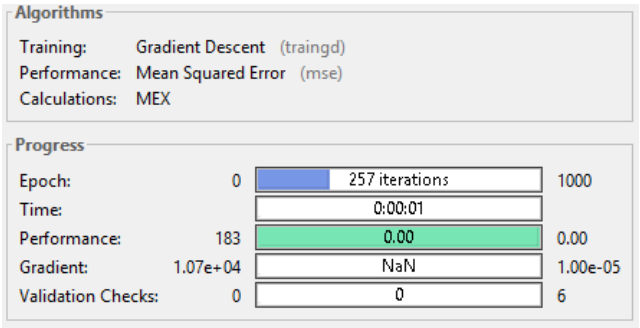

Figura 84: Red 1 capa oculta, 500 neuronas, objetivo RSSI, función traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 500 neuronas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): NaN
    Resultado MEA(Promedio absoluto del error): NaN
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.2.4 Conclusiones y cuadro comparativo

Nº capas ocultas	Nº neuronas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	100	RSSI	22.17	3.60
1	100	Idem	49.50	6.25
1	200	RSSI	22.17	3.60
1	200	Idem	96.52	7.58
1	500	RSSI	NaN	NaN
1	500	Idem	NaN	NaN

Tabla 3: Conclusiones según nº de neuronas y función Traingd

Se aprecia que para este metodo de entrenamiento y con los parametros de entrada y salida que tenenmos en nuestro entrenamiento la variacion del numero de neuronas por capa oculta no mejora el resultado obtenido ni disminulle los errores producidos. Por lo tanto continuamos con el entrenamiento con 1 capa oculta y 100 neuronas.

7.1.3 Caso 3 de entrenamiento variando nº de épocas.

Como se acaba de comentar en el apartado anterior, se continúa con la realización de las pruebas con una red con 1 capa oculta y 100 neuronas.

En este apartado se pretende variar el número de épocas por capa. Se van a realizar tres entrenamientos con 500 épocas, 1500 épocas y 3000 épocas.

7.1.3.1 A continuación se muestran los resultados obtenidos con función traingd con 1 capa oculta, 100 neuronas por capa, 500 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

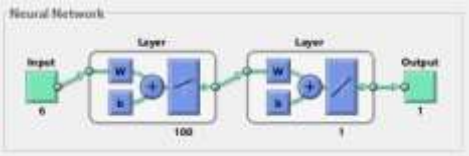

Figura 85: Red 1 capa oculta, 100 neur. 500 épocas objetivo RSSI, función traingd

Resultados de simulacion de red:

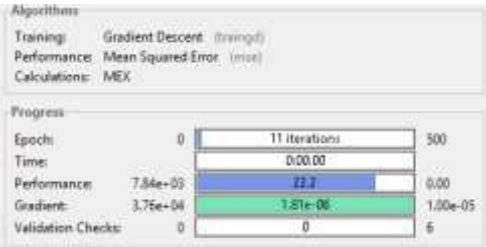

Figura 86: Resultados red 1cap. 100 neur. 500 épocas obj. RSSI, función traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

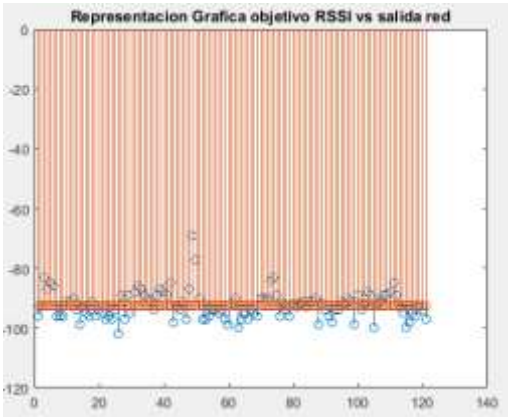

Figura 87: Repre. salida red 1 cap. oc.100 neu, 500 épocas obj. RSSI, func. traingd

Representacion grafica del error cometido por dato:

Figura 88: Repre. error red 1 cap. oc.100 neu, 500 epocas obj. RSSI, func. traingd

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta, 100 neuronas y 500 epocas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

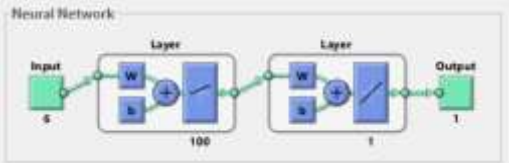

Figura 89: Red 1 cap. oc.100 neu, 500 epocas obj. Ídem, func. traingd

Resultados de simulacion de red:

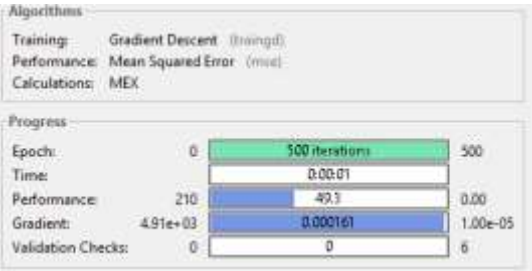

Figura 90: Resultados Red 1 cap. oc.100 neu, 500 epocas obj. Ídem, func. traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

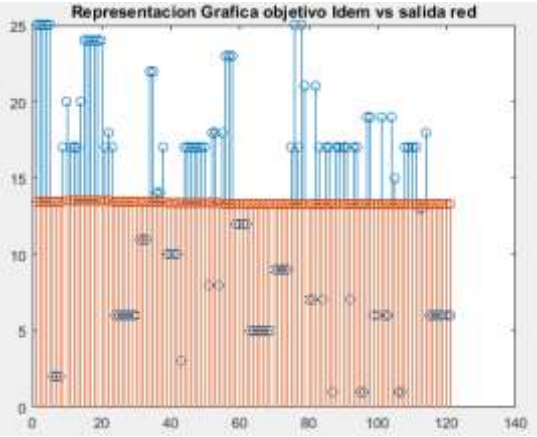

Figura 91: Repre. salida red 1 cap. oc.100 neu, 500 epocas obj. Ídem, func. traingd

Representacion grafica del error cometido por dato:

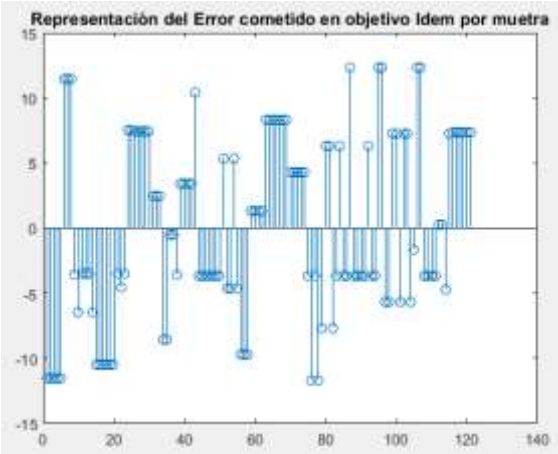

Figura 92: Repre. error red 1 cap. oc.100 neu, 500 epocas obj. Ídem, func. traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 100 neuronas y 500 epocas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 49.25
    Resultado MEA(Promedio absoluto del error): 6.24
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.3.2 A continuación se muestran los resultados obtenidos con función traingd con 1 capa oculta, 100 neuronas por capa, 1500 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

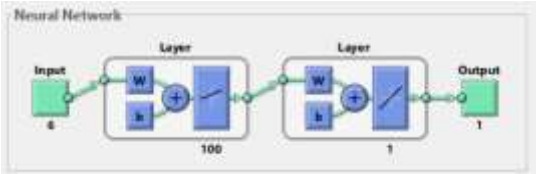

Figura 93: Red 1 cap. oc.100 neu, 1500 epocas obj. RSSI, func. traingd

Resultados de simulacion de red:

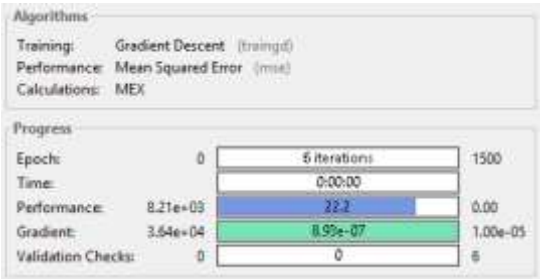

Figura 94: Resultados red 1 cap. oc.100 neu, 1500 epocas obj. RSSI, func. traingd

Resultados de simulacion de red:

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

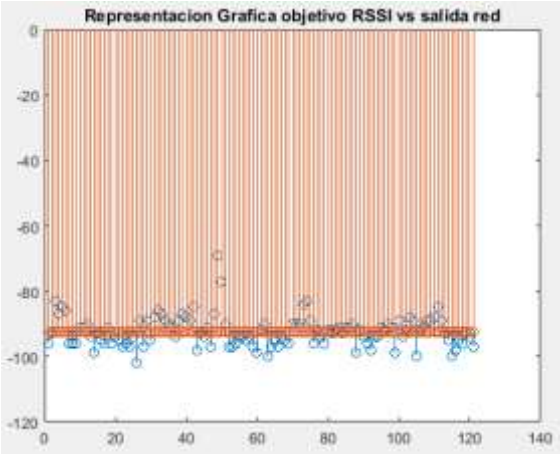

Figura 95: Represent. salida 1cap.oc.100 neu, 1500 epocas obj. RSSI, func. traingd

Representacion grafica del error cometido por dato:

Figura 96: Represent. error 1cap.oc.100 neu, 1500 epocas obj. RSSI, func. traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta, 100 neuronas y 1500 epocas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

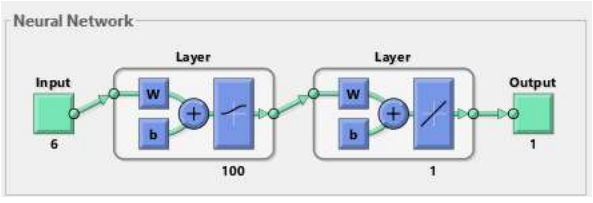

Figura 97: Resultados red 1 cap. oc.100 neu, 1500 epocas obj. Ídem, func. traingd

Resultados de simulacion de red:

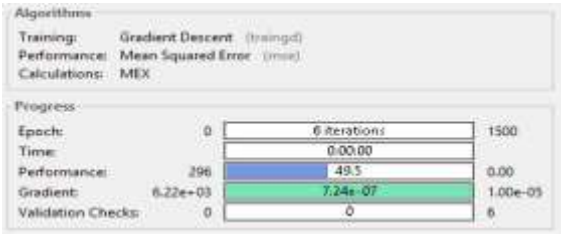

Figura 98: Resultados red 1 cap. oc.100 neu, 1500 epocas obj. Ídem, func. traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

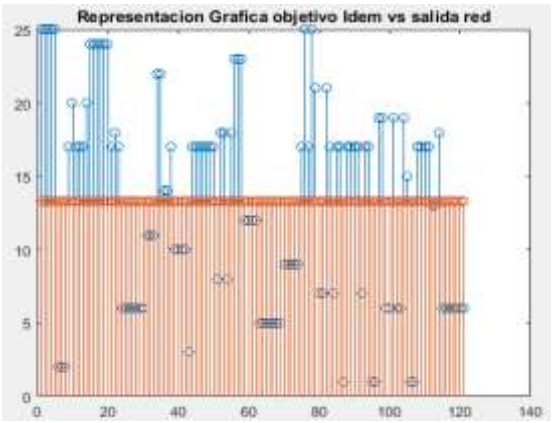

Figura 99: Represent. salida 1cap.oc.100 neu, 1500 epocas obj. Ídem, func. traingd

Representacion grafica del error cometido por dato:

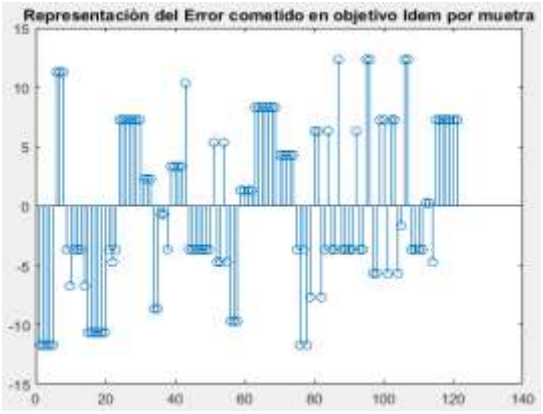

Figura 100: Represent. error 1cap.oc.100 neu, 1500 epocas obj. Ídem, func. traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 100 neuronas y 1500 epocas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 49.50
    Resultado MEA(Promedio absoluto del error): 6.25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.3.3 A continuación se muestran los resultados obtenidos con función traingd con 1 capa oculta, 100 neuronas por capa, 30000 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

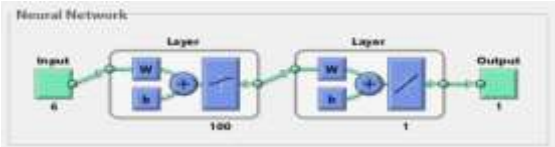

Figura 101: Red 1 cap. oc.100 neu, 3000 epocas obj. RSSI, func. traingd

Resultados de simulacion de red:

Figura 102: Resultados red 1 cap. oc.100 neu, 3000 epocas obj. RSSI, func. traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

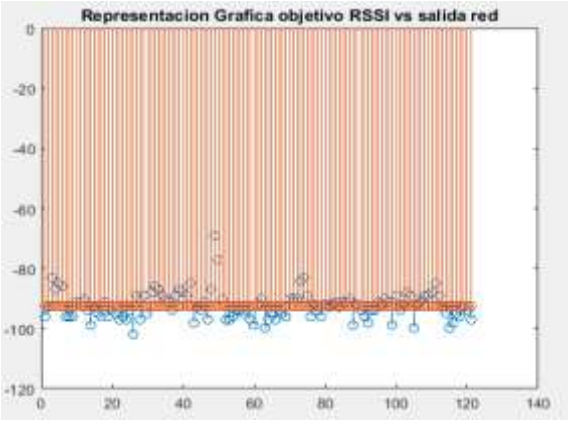

Figura 103: Represent. salida 1cap.oc.100 neu, 3000 epocas obj. RSSI, func. traingd

Representacion grafica del error cometido por dato:

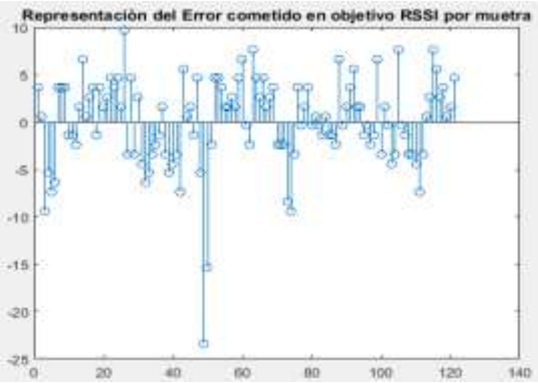

Figura 104: Represent. error 1cap.oc.100 neu, 3000 epocas obj. RSSI, func. traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta, 100 neuronas y 3000 epocas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

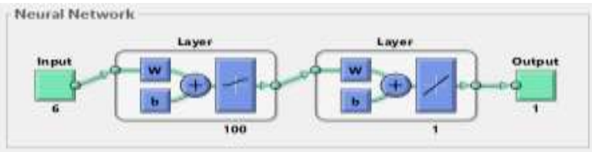

Figura 105: Resultados red 1 cap. oc.100 neu, 3000 epocas obj. RSSI, func. traingd

Resultados de simulacion de red:

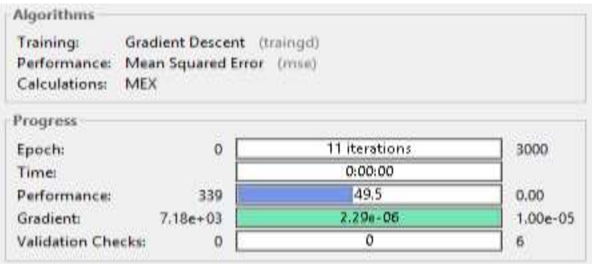

Figura 106: Resultados red 1 cap. oc.100 neu, 3000 epocas obj. RSSI, func. traingd

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

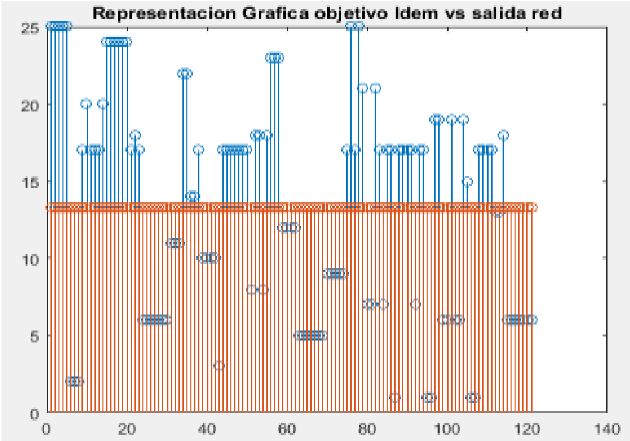

Figura 107: Represent. error 1cap.oc.100 neu, 3000 epocas obj. RSSI, func. traingd

Representacion grafica del error cometido por dato:

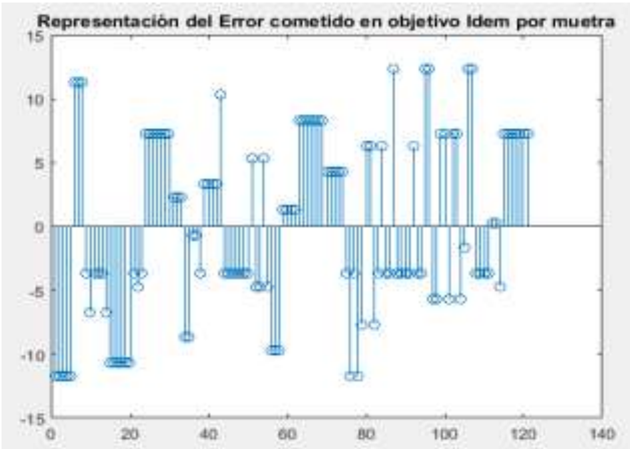

Figura 108: Represent. error 1cap.oc.100 neu, 3000 epocas obj. Ídem, func. traingd

Resultados mostrados según programa implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINGD y 1 capa oculta y 100 neuronas y 3000 epocas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 49.50
    Resultado MEA(Promedio absoluto del error): 6.25
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.1.3.4 Conclusiones y cuadro comparativo

Nº capas ocultas	Nº neuronas	Nº épocas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	100	500	RSSI	22.17	3.60
1	100	500	Idem	49.50	6.25
1	100	1500	RSSI	22.17	3.6
1	100	1500	Idem	49.50	6.25
1	100	3000	RSSI	22.17	3.6
1	100	3000	Idem	49.50	6.25

Tabla 4: Conclusiones según nº de epocas y función Traingd

Se aprecia que para este metodo de entrenamiento y con los parametros de entrada y salida que tenemos, la variacion del numero de neuronas no mejora el resultado obtenido ni disminulle los errores producidos.

Despues de realizar todas las modificaciones en la configuracion de la red con la funcion de entrenamiento traingd se observa que este metodo no es apropiado para estimar valores de señal WI-FI . Por lo que concluimos que el entrenamiento con Traingd no aporta resultados correctos en los entrenamientos.

7.2 Estudio comparativo con función Trainrp

Se procede a realizar los entrenamientos y comparaciones de los resultados obtenidos partiendo de una red neuronal creada por defecto y con la función de entrenamiento Trainrp. Es necesario mantener los mismos parámetros de entrada y de salida para todas las variaciones de entrenamientos realizadas. Se entrenará por separado los diferentes objetivos, es decir, habrá dos redes, una para los objetivos de RSSI y otra para los objetivos de Ídem. Los parámetros para cada red son:

NettraingdRSSI:

Entradas; Lat_Grados, Lat_min, Lat_seg, Lon_Grados, Lon_min, Lon_seg

Salida: RSSI

Nettraingdidem:

Entradas; Lat_Grados, Lat_min, Lat_seg, Lon_Grados, Lon_min, Lon_seg

Salida: ídem

7.2.1 Caso 1 de entrenamiento variando capas ocultas

Con los datos de entrada mencionados anteriormente se procede a realizar simulaciones de las redes neuronales. La primera simulación se realiza con todos los parámetros por defecto menos el número de capas ocultas, que se modificará con los valores de 1, 2 y 3 capas ocultas en dicha red.

7.2.1.1 A continuación se muestran los resultados obtenidos con función trainrp con 1 capa oculta y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

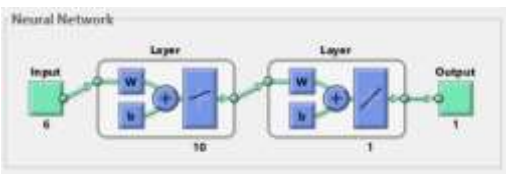

Figura 109: Red 1 capa oculta objetivo RSSI, función trainrp

Resultados de simulacion de red:

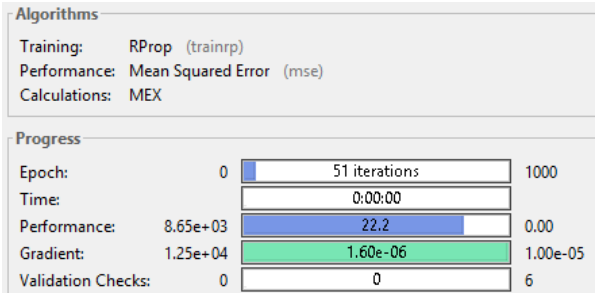

Figura 110: Resultados Red 1 capa oculta objetivo RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

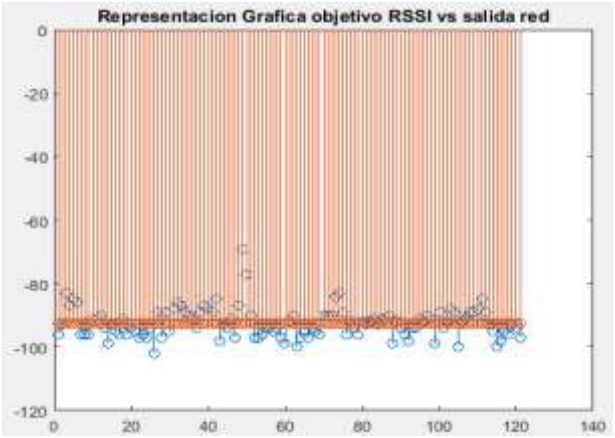

Figura 111: Representación salida red 1 capa oculta objetivo RSSI, función trainrp

Representacion grafica del error cometido por dato:

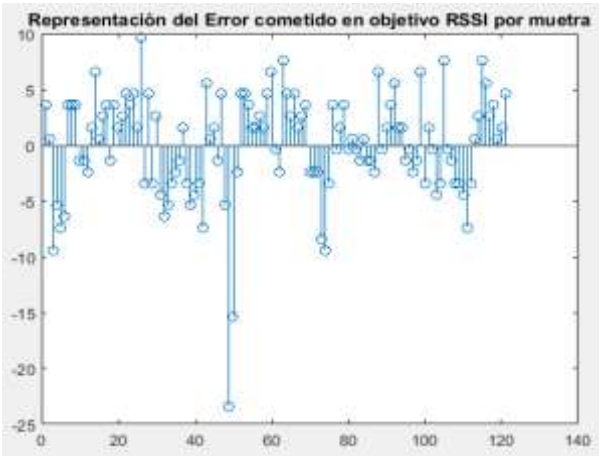

Figura 112: Representación error red 1 capa oculta objetivo RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 1 capa oculta. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

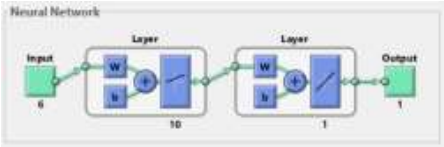

Figura 113: Red 1 capa oculta objetivo Ídem, función trainrp

Resultados de simulacion de red:

Figura 114: Resultados Red 1 capa oculta objetivo Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

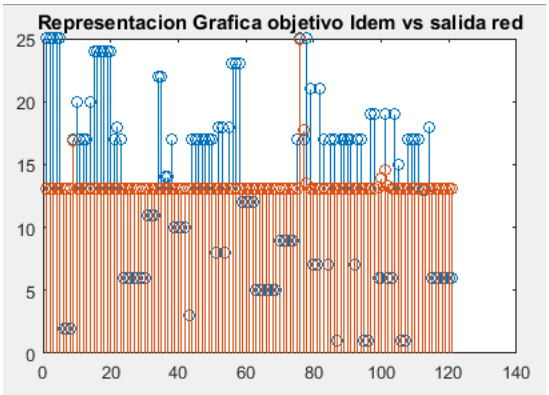

Figura 115: Representación salida red 1 capa oculta objetivo Ídem, función trainrp

Representacion grafica del error cometido por dato:

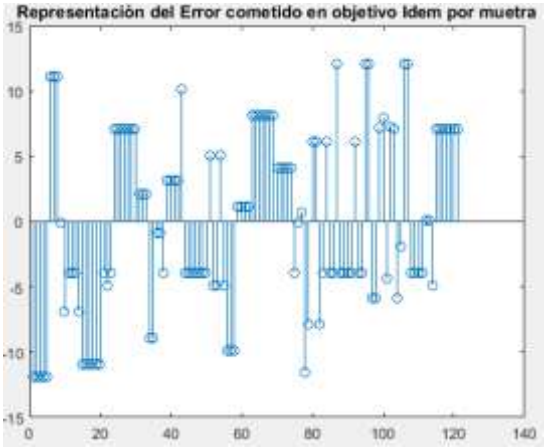

Figura 116: Representación error red 1 capa oculta objetivo Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 1 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 48.06
    Resultado MEA(Promedio absoluto del error): 6.10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.1.2 A continuación se muestran los resultados obtenidos con función trainrp con 2 capa oculta y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

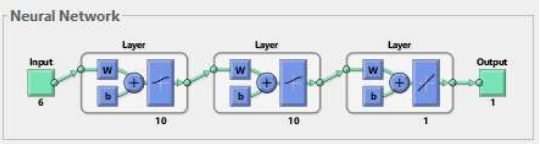

Figura 117: Red 2 capa oculta objetivo RSSI, función trainrp

Resultados de simulacion de red:

Figura 118: Resultados Red 2 capa oculta objetivo RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

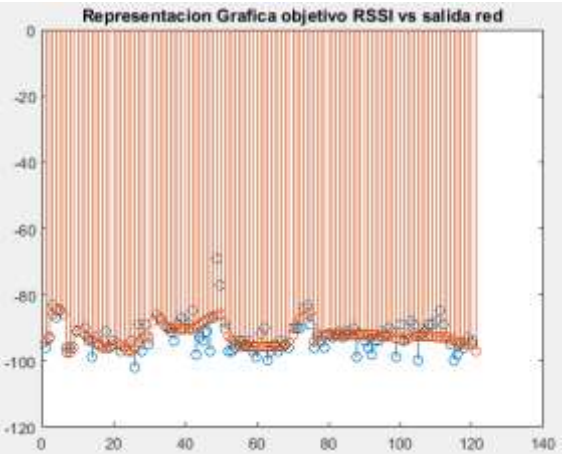

Figura 119: Representación salida red 2 capa oculta objetivo RSSI, función trainrp

Representacion grafica del error cometido por dato:

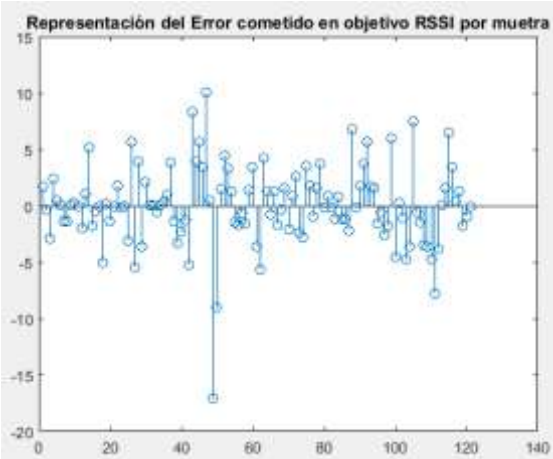

Figura 120: Representación salida red 2 capa oculta objetivo RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 2 capa oculta. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 12.46
    Resultado MEA(Promedio absoluto del error): 2.48
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

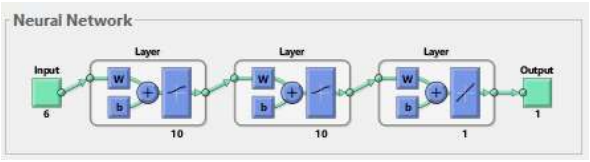

Figura 121: Red 2 capa oculta objetivo Ídem, función trainrp

Resultados de simulacion de red:

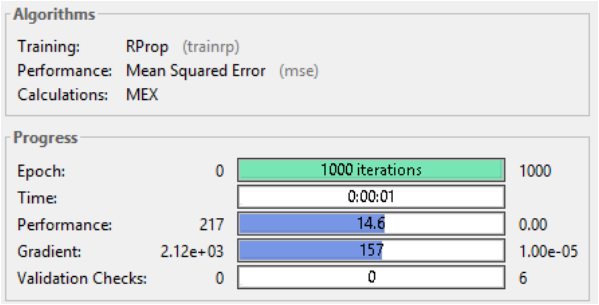

Figura 122: Resultados Red 2 capa oculta objetivo Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

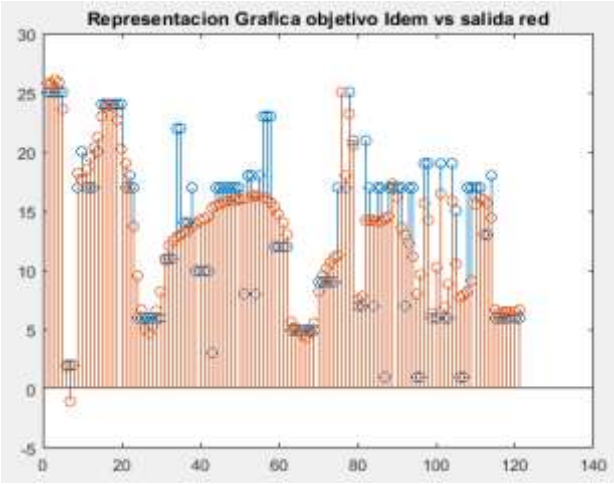

Figura 123: Representación salida red 2 capa oculta objetivo Ídem, función trainrp

Representacion grafica del error cometido por dato:

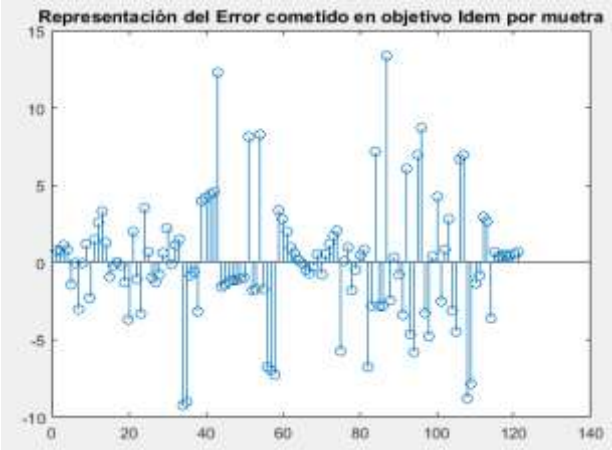

Figura 124: Representación error red 2 capa oculta objetivo Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 2 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 14.63
    Resultado MEA(Promedio absoluto del error): 2.68
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.1.3 A continuación se muestran los resultados obtenidos con función trainrp con 3 capa oculta y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

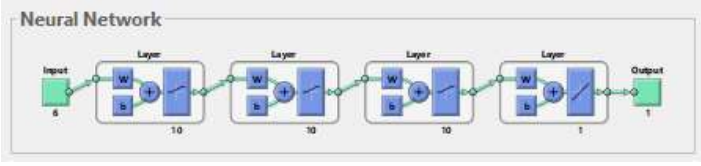

Figura 125: Red 3 capa oculta objetivo RSSI, función trainrp

Resultados de simulacion de red:

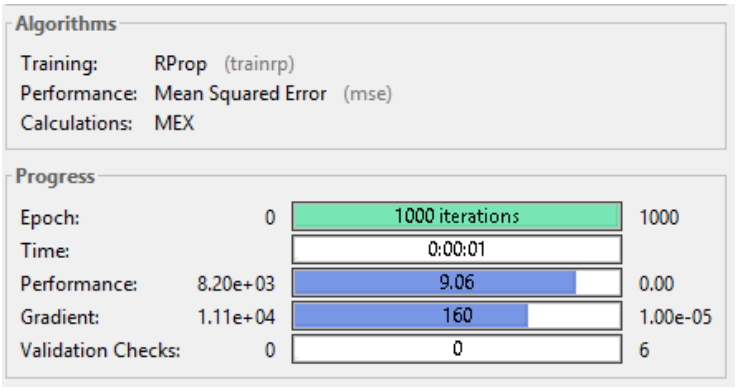

Figura 126: Resultados red 3 capa oculta objetivo RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

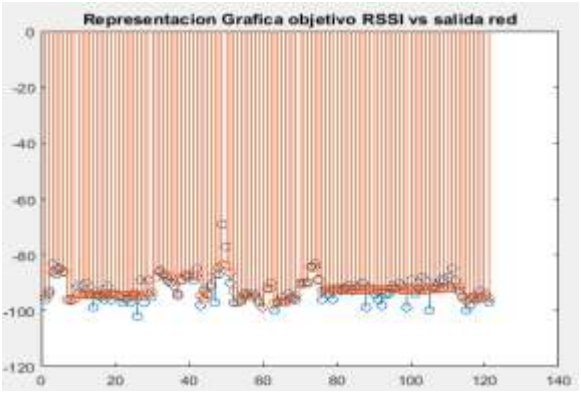

Figura 127: Representación salida red 3 capa oculta objetivo RSSI, función trainrp

Representacion grafica del error cometido por dato:

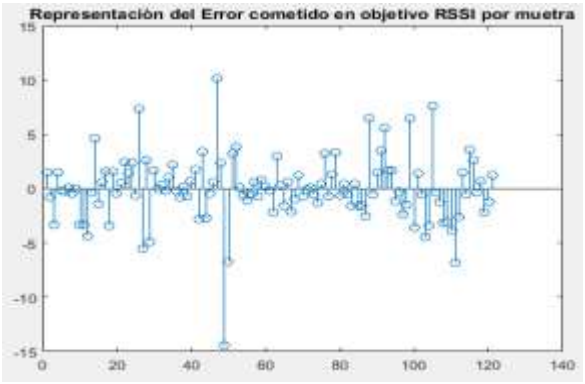

Figura 128: Representación error red 3 capa oculta objetivo RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 9.06
    Resultado MEA(Promedio absoluto del error): 2.05
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

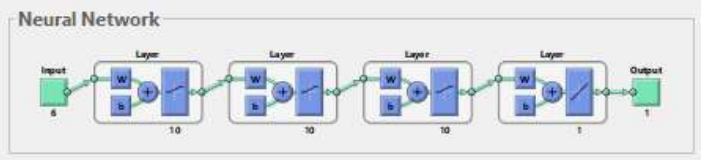

Figura 129: Red 3 capa oculta objetivo Ídem , función trainrp

Resultados de simulacion de red:

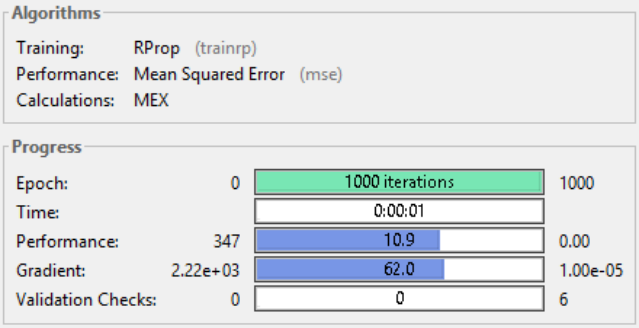

Figura 130: Resultados red 3 capa oculta objetivo Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

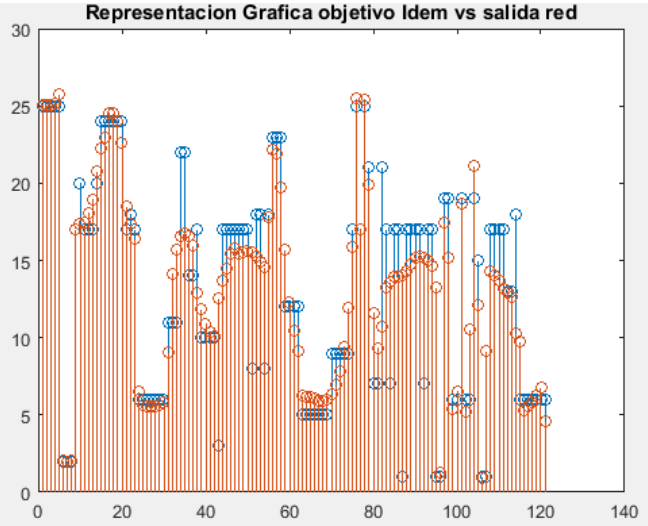

Figura 131: Representación salida red 3 capa oculta objetivo Ídem, función trainrp

Representacion grafica del error cometido por dato:

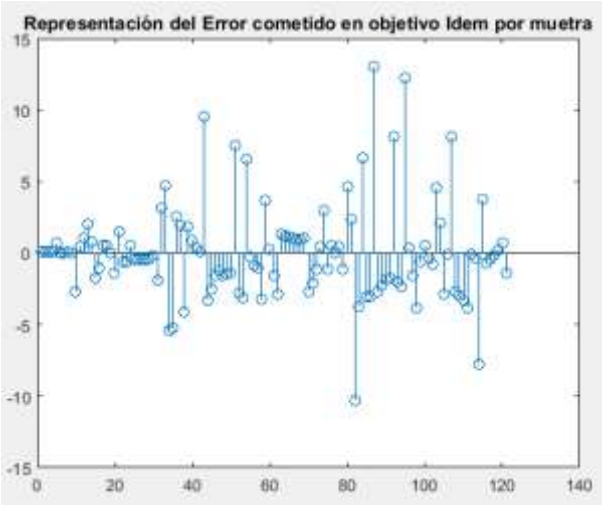

Figura 132: Representación error red 3 capa oculta objetivo Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 10.88
    Resultado MEA(Promedio absoluto del error): 2.16
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.1.4 Conclusiones y cuadro comparativo

Nº capas ocultas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	RSSI	22.17	3.6
1	Idem	48.08	6.10
2	RSSI	12.46	2.48
2	Idem	14.63	6.68
3	RSSI	9.06	2.05
3	Idem	10.9	2.16

Tabla 5: Conclusiones según nº de capas ocultas y función Trainrp

Se observa que para este entrenamiento y con las caracteristicas conforme lo hemos definido se ajusta mejor la utilizacion de 3 capas ocultas, ya que produce menor pormedio cuadrado de error y menor promedio absoluto de error. Por lo que se decide continuar con las pruebas con 2 capas ocultas para las redes.

7.2.2 Caso 2 de entrenamiento variando nº de neuronas por capa

Como se acaba de comentar en el apartado anterior, se continúa con la realización de las pruebas con una red con 3 capa oculta con entrenamiento trainrp

En este apéndice se pretende variar el número de neuronas por capa oculta. Se van a realizar tres entrenamientos con 100 neuronas, 200 neuronas y 500 neuronas.

7.2.2.1 A continuación se muestran los resultados obtenidos con función trainrp con 3 capas ocultas, 100 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

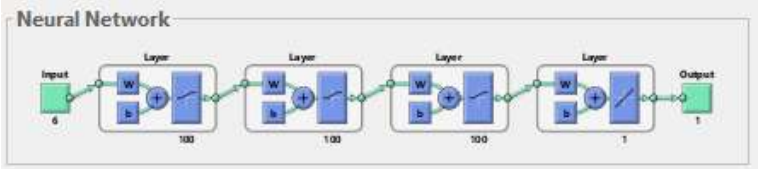

Figura 133: Red 3 capa oculta, 100 neuronas objetivo RSSI, función trainrp

Resultados de simulacion de red:

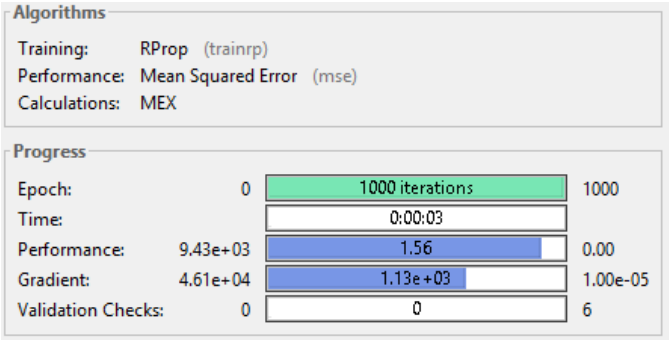

Figura 134: Resultados Red 3capa oculta, 100 neuronas obj. RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

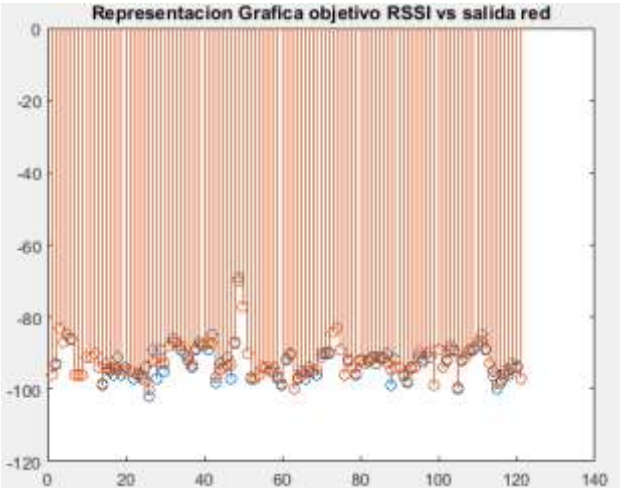

Figura 135: Representación salida 3cap.oc., 100 neuronas obj. RSSI, función trainrp

Representacion grafica del error cometido por dato:

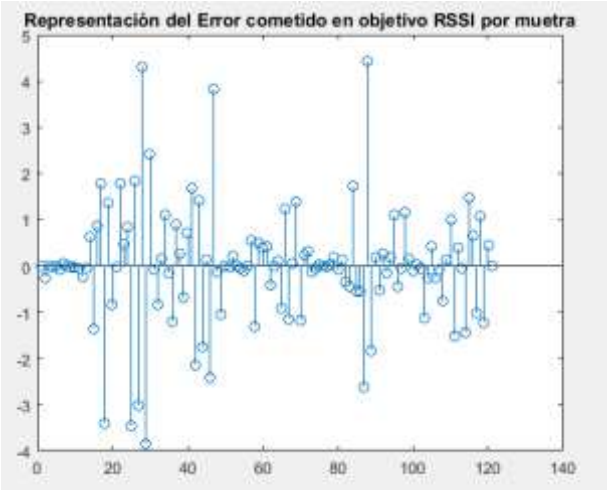

Figura 136: Representación error 3cap.oc., 100 neuronas obj. RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta. 100 neuroas Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 1.56
    Resultado MEA(Promedio absoluto del error): 0.79
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

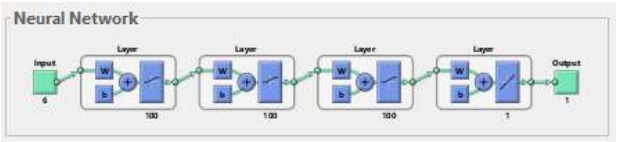

Figura 137: Red 3 capa oculta, 100 neuronas objetivo Ídem, función trainrp

Resultados de simulacion de red:

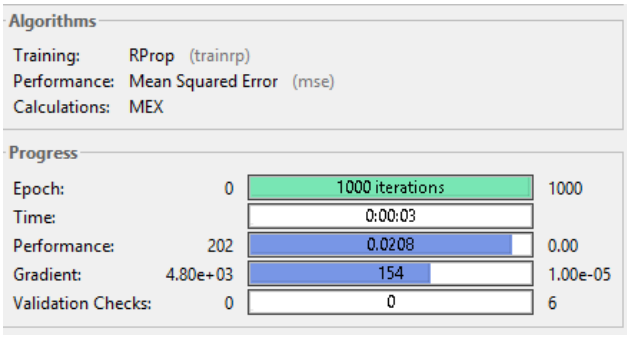

Figura 138: Resultados Red 3 cap. oc, 100 neuronas objetivo Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

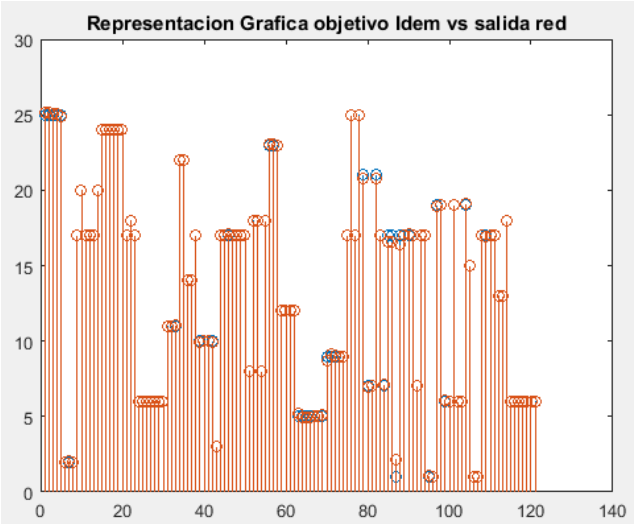

Figura 139: Represen. salida Red 3cap. oc, 100 neuronas obj. Ídem, función trainrp

Representacion grafica del error cometido por dato:

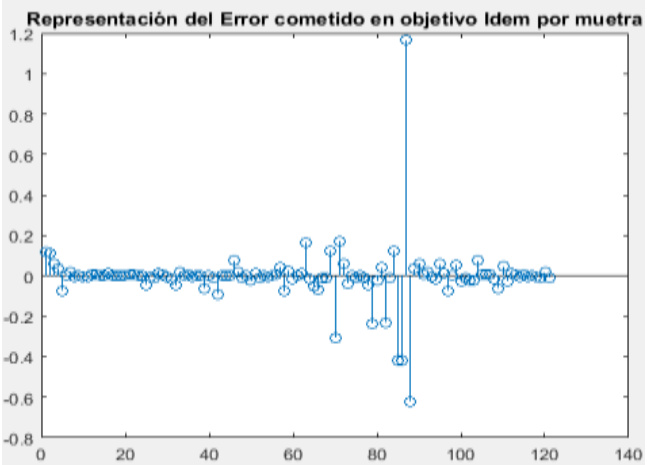

Figura 140: Represen. salida Red 3cap. oc, 100 neuronas obj. Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta, 100 neuronas Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 0.02
    Resultado MEA(Promedio absoluto del error): 0.05
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.2.2 A continuación se muestran los resultados obtenidos con función trainrp con 3 capas ocultas, 300 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

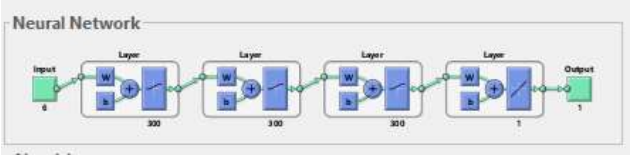

Figura 141: Resultados Red 3 cap. oc, 300 neuronas objetivo RSSI, función trainrp

Resultados de simulacion de red:

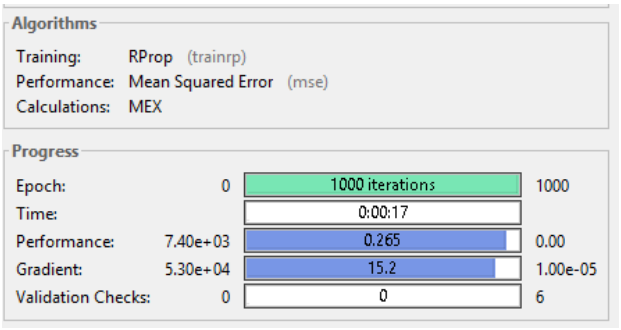

Figura 142: Resultados Red 3 cap. oc, 300 neuronas objetivo RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

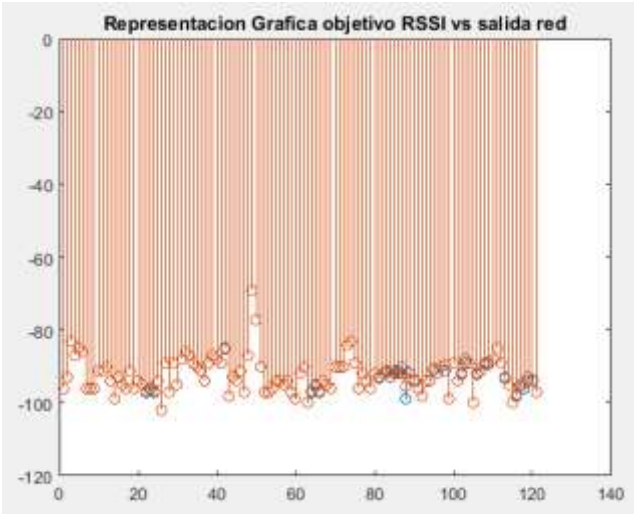

Figura 143: Represen. salida Red 3cap. oc, 300 neuronas obj. RSSI, función trainrp

Representacion grafica del error cometido por dato:

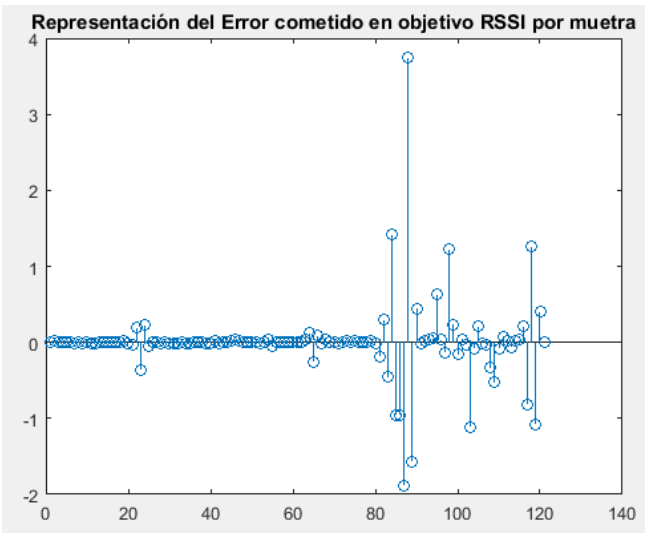

Figura 144: Represen. error Red 3cap. oc, 300 neuronas obj. RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta. 300 neuroas Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 0.26
    Resultado MEA(Promedio absoluto del error): 0.19
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

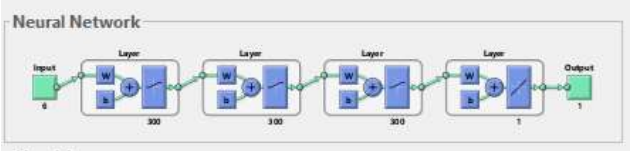

Figura 145: Resultados Red 3 cap. oc, 300 neuronas objetivo Ídem, función trainrp

Resultados de simulacion de red:

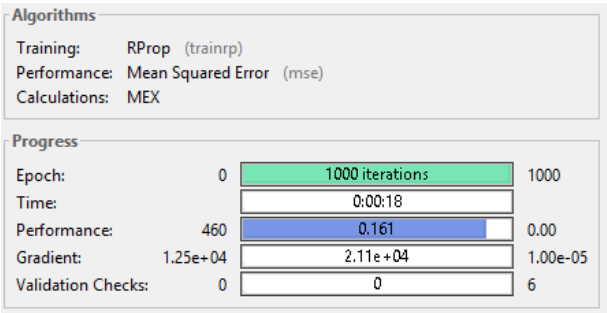

Figura 146: Resultados Red 3 cap. oc, 300 neuronas objetivo Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

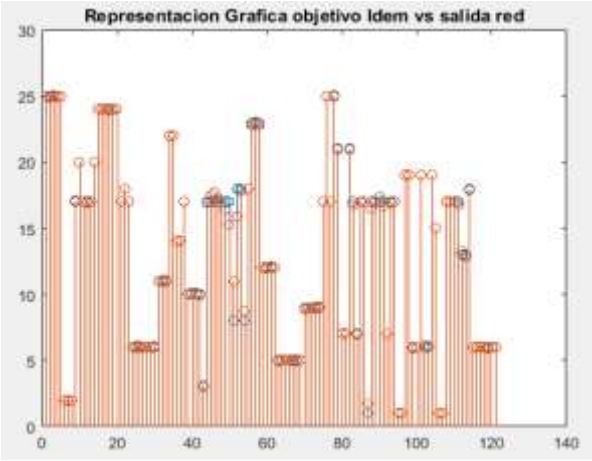

Figura 145: Represen. salida Red 3cap. oc, 300 neuronas obj. Ídem, función trainrp

Representacion grafica del error cometido por dato:

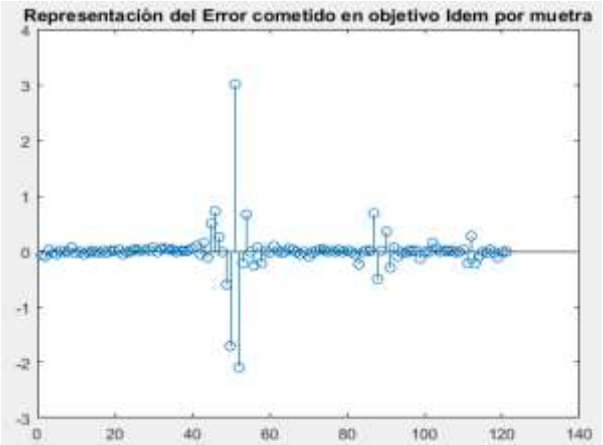

Figura 146: Represen. Red 3rror cap. oc, 300 neuronas obj. Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta, 300 neuronas Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 0.16
    Resultado MEA(Promedio absoluto del error): 0.14
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.2.3 A continuación se muestran los resultados obtenidos con función trainrp con 3 capas ocultas, 500 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

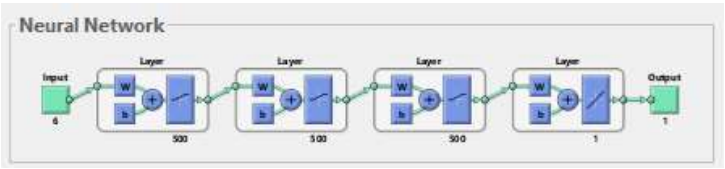

Figura 147: Red 3 cap. oc, 500 neuronas objetivo RSSI, función trainrp

Resultados de simulacion de red:

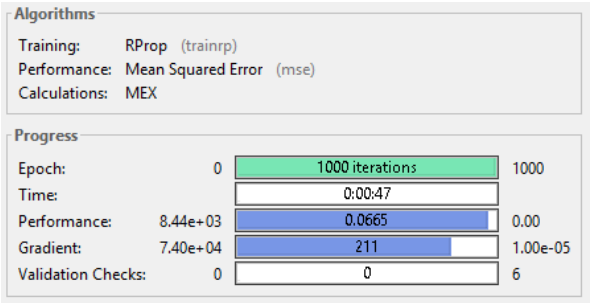

Figura 148: Resultados Red 3 cap. oc, 500 neuronas objetivo RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

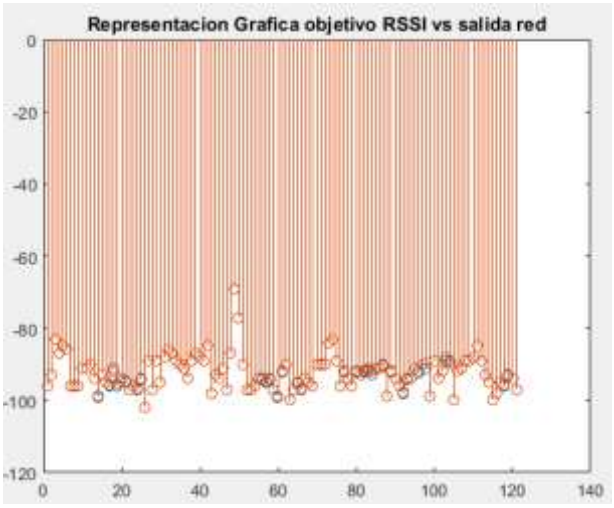

Figura 149: Represen. salida Red 3cap. oc, 500 neuronas obj. RSSI, función trainrp

Representacion grafica del error cometido por dato:

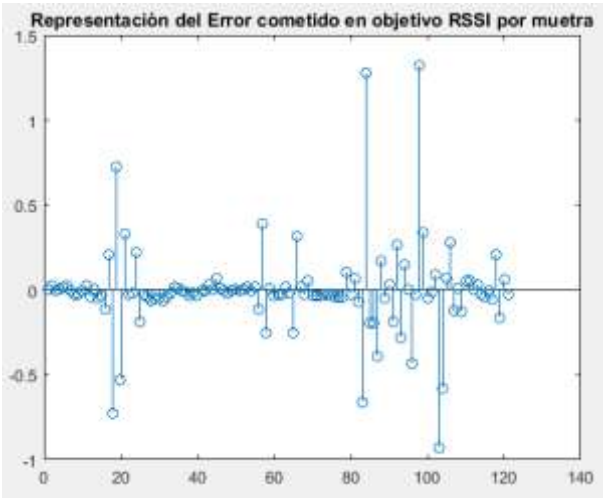

Figura 150: Represen. Error Red 3cap. oc, 500 neuronas obj. RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta. 500 neuroas Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 0.07
    Resultado MEA(Promedio absoluto del error): 0.13
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

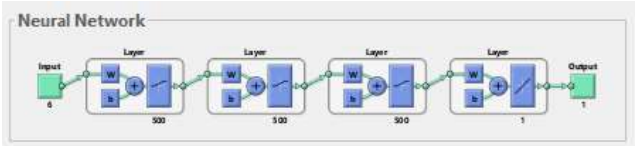

Figura 151: Red 3 cap. oc, 500 neuronas objetivo Ídem, función trainrp

Resultados de simulacion de red:

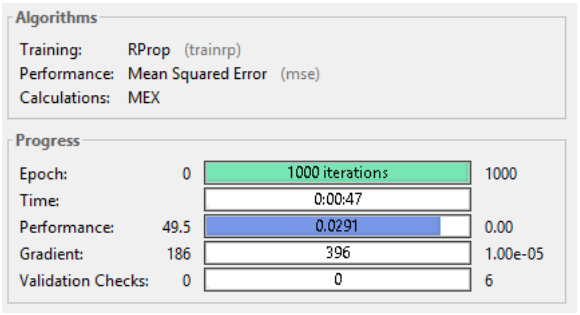

Figura 152: Resultados Red 3 cap. oc, 500 neuronas objetivo Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

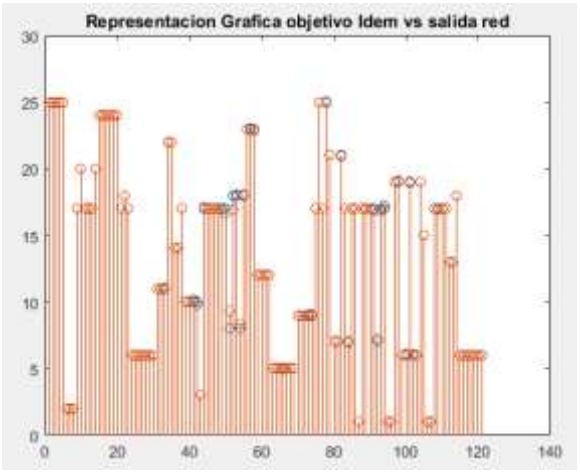

Figura 153: Represen. salida Red 3cap. oc, 500 neuronas obj. Ídem, función trainrp

Representacion grafica del error cometido por dato:

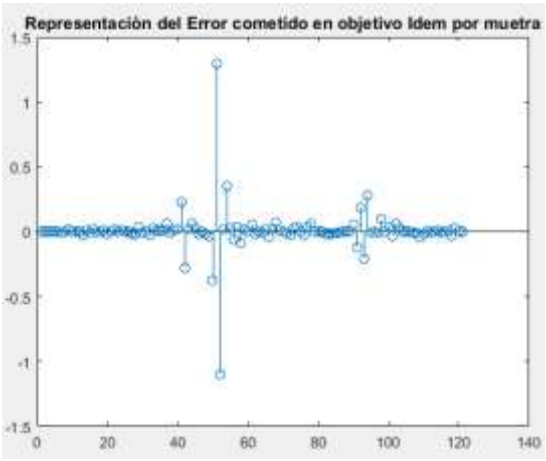

Figura 154: Represen. error Red 3cap. oc, 500 neuronas obj. Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP y 3 capa oculta, 500 neuronas Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 0.03
    Resultado MEA(Promedio absoluto del error): 0.05
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.2.4 Conclusiones y cuadro comparativo

Nº capas ocultas	Nº neuronas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
3	100	RSSI	1.56	0.79
3	100	Idem	0.02	0.05
3	300	RSSI	0.26	0.19
3	300	Idem	0.16	0.14
3	500	RSSI	0.07	0.13
3	500	Idem	0.03	0.05

Tabla 6: Conclusiones según nº de neuronas y función Trainrp

Se observa que para este entrenamiento y con las caracteristicas conforme lo hemos definido se ajusta mejor la utilizacion de 3 capas ocultas con 500 neuronas por capa, ya que produce menor pormedio cuadrado de error y menor promedio absoluto de error. Por lo que se decide continuar con las pruebas con redes de 3 capas ocultas con 500 neuronas por capa.

7.2.3 Caso 3 de entrenamiento variando nº de épocas.

Como se acaba de comentar en el apartado anterior, se continúa con la realización de las pruebas con redes con 3 capas ocultas y 500 neuronas.

En este apartado se pretende variar el número de épocas por capa. Se van a realizar tres entrenamientos con 500 épocas, 1500 épocas y 3000 épocas.

7.2.3.1 A continuación se muestran los resultados obtenidos con función trainrp con 3 capas ocultas, 500 neuronas por capa, 500 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

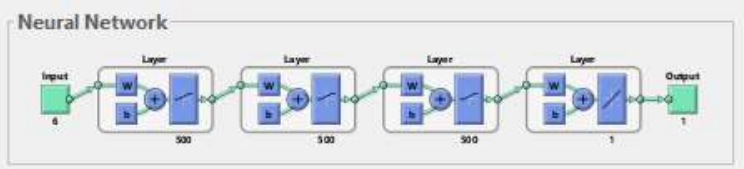

Figura 155: Red 3 cap. oc, 500 neu. 500 epocas objetivo RSSI, función trainrp

Resultados de simulacion de red:

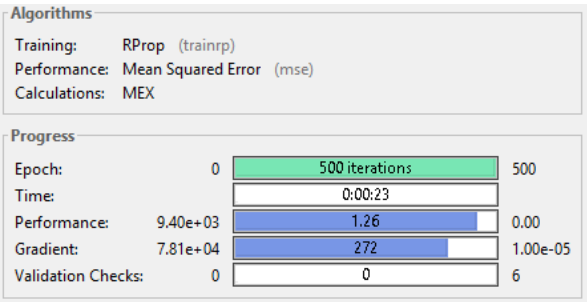

Figura 156: Resultados Red 3 cap. oc, 500 neu. 500 epocas obj. RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

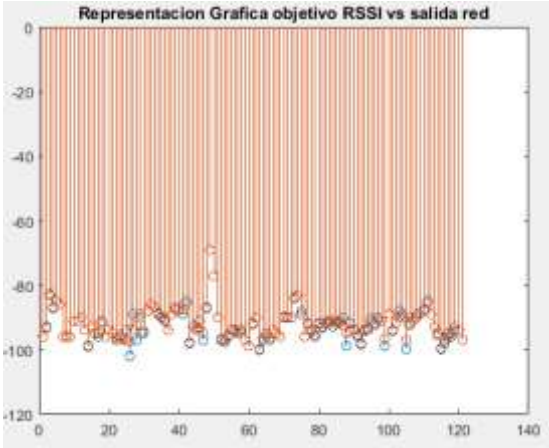

Figura 157: Represent. salida 3cap.oc, 500 neu. 500 epocas obj. RSSI, función trainrp

Representacion grafica del error cometido por dato:

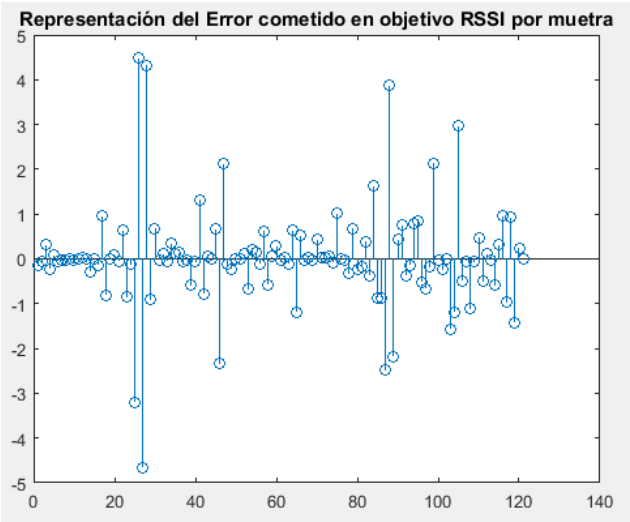

Figura 158: Represent. error 3cap.oc, 500 neu. 500 epocas obj. RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP 3 capa oculta 500 neuroas 500 epocas Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 1.26
    Resultado MEA(Promedio absoluto del error): 0.62
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

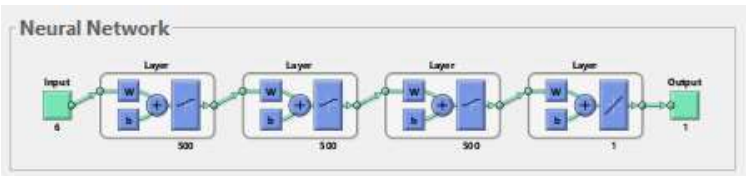

Figura 159: Red 3 cap. oc, 500 neu. 500 epocas objetivo Ídem, función trainrp

Resultados de simulacion de red:

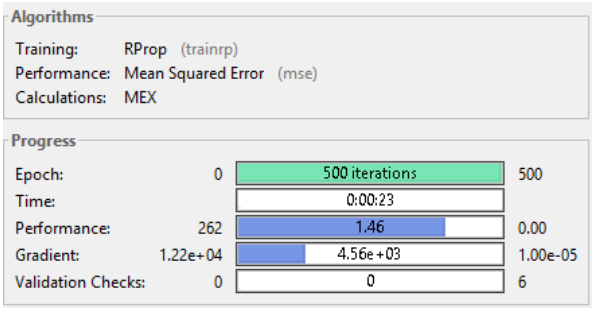

Figura 160: Resultados Red 3 cap. oc, 500 neu. 500 epocas obj. Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

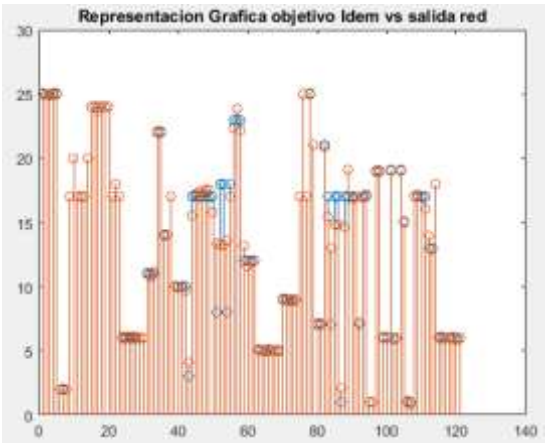

Figura 161: Represent. salida 3cap.oc, 500 neu. 500 epocas obj. Ídem, función trainrp

Representacion grafica del error cometido por dato:

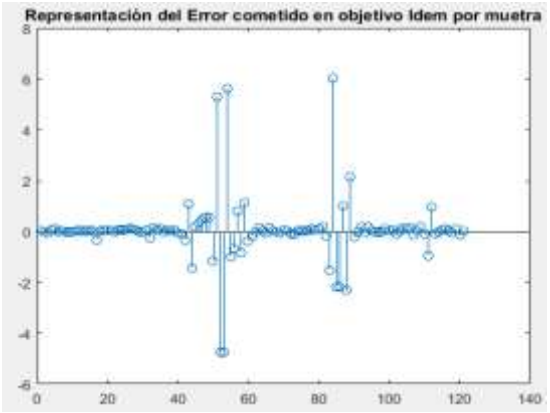

Figura 162: Represent. error 3cap.oc, 500 neu. 500 epocas obj. Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP 3 capa oculta 500 neuronas 500 epocas Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 1.46
    Resultado MEA(Promedio absoluto del error): 0.47
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.3.2 A continuación se muestran los resultados obtenidos con función trainrp con 3 capas ocultas, 500 neuronas por capa, 1500 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

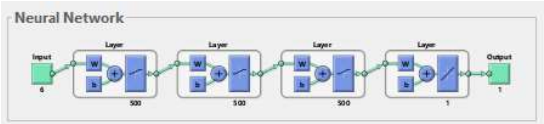

Figura 163: Red 3 cap. oc, 500 neu. 1500 epocas objetivo RSSI, función trainrp

Resultados de simulacion de red:

Figura 164: Resultados Red 3 cap. oc, 500 neu. 1500 epocas obj RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

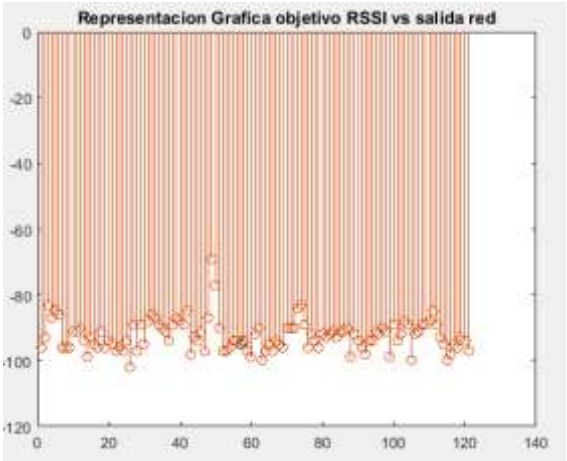

Figura 165: Represent. salida 3cap.oc, 500 neu.1500 epoc obj. RSSI, función trainrp

Representacion grafica del error cometido por dato:

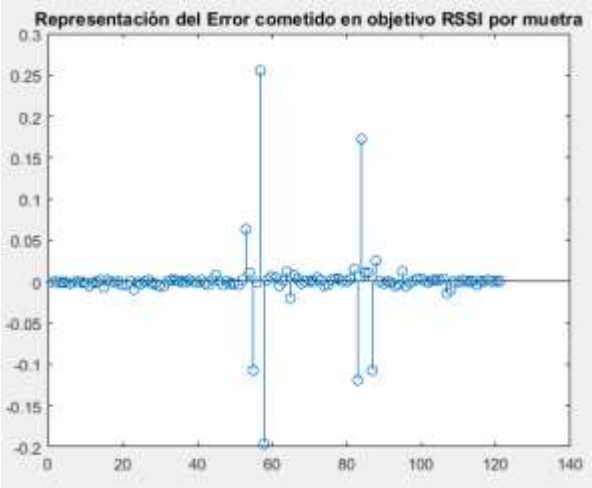

Figura 166: Represent. salida 3cap.oc, 500 neu.1500 epoc obj. RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP 3 capa oculta 500 neuroas 1500 epocas Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 0.00
    Resultado MEA(Promedio absoluto del error): 0.01
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

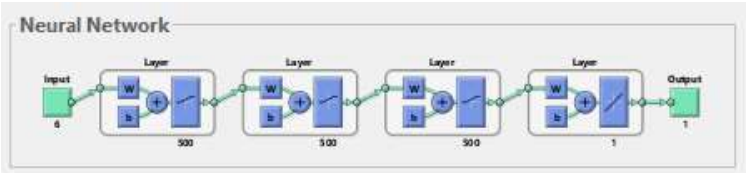

Figura 167: Red 3 cap. oc, 500 neu. 1500 epocas objetivo Ídem, función trainrp

Resultados de simulacion de red:

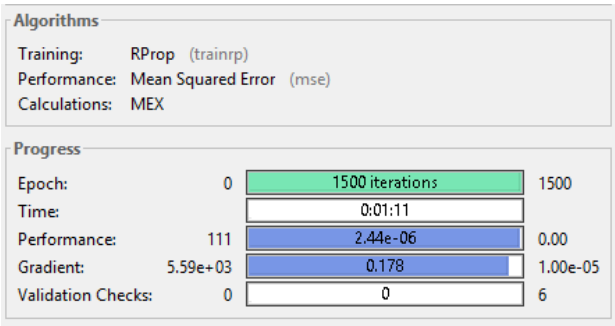

Figura 168: Resultados Red 3 cap. oc, 500 neu. 1500 epocas obj Ídem función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

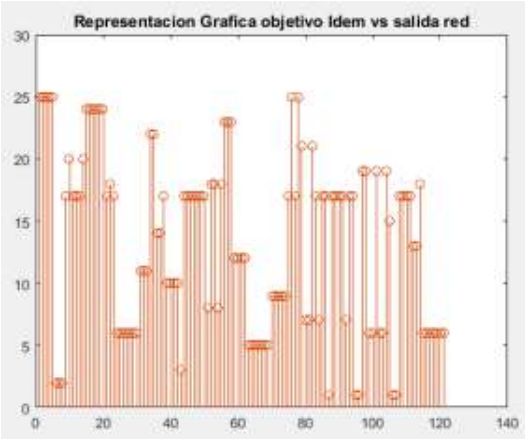

Figura 169: Represent. salida 3cap.oc, 500 neu.1500 epoc obj. Ídem, función trainrp

Representacion grafica del error cometido por dato:

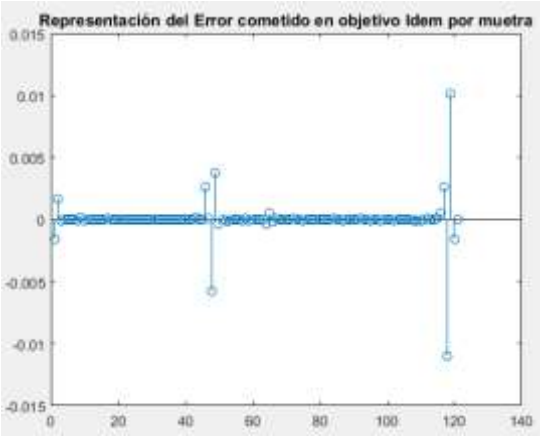

Figura 170: Represent. error 3cap.oc, 500 neu.1500 epoc obj. Ídem, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP 3 capa oculta 500 neuronas 1500 epocas Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 0.00
    Resultado MEA(Promedio absoluto del error): 0.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.3.3 A continuación se muestran los resultados obtenidos con función trainrp con 3 capas ocultas, 500 neuronas por capa, 3000 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

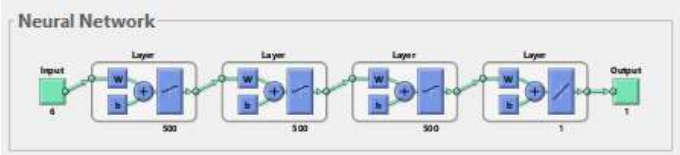

Figura 171: Red 3 cap. oc, 500 neu. 3000 epocas objetivo RSSI, función trainrp

Resultados de simulacion de red:

Figura 172: Resultados Red 3 cap. oc, 500 neu. 3000 epocas obj RSSI, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

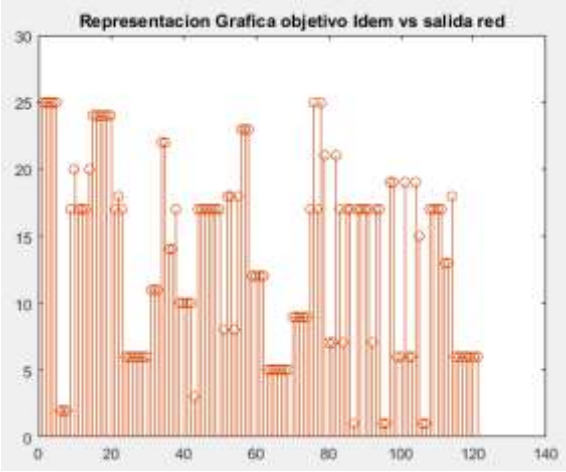

Figura 173: Represent. salida 3cap.oc, 500 neu.3000 epoc obj. RSSI, función trainrp

Representacion grafica del error cometido por dato:

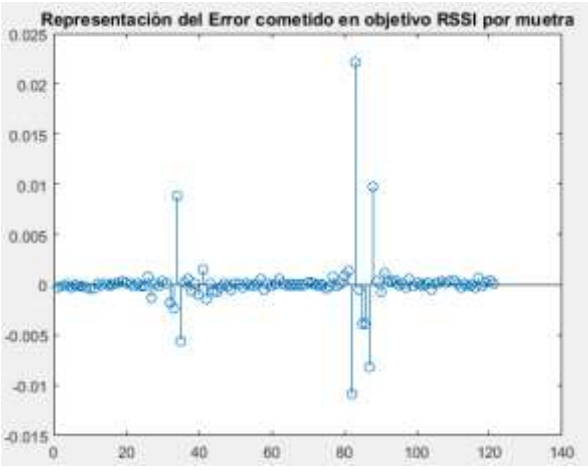

Figura 174: Represent. error 3cap.oc, 500 neu.3000 epoc obj. RSSI, función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP 3 capa oculta 500 neuroas 3000 epocas Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 0.00
    Resultado MEA(Promedio absoluto del error): 0.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

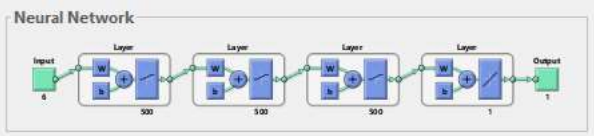

Figura 175: Red 3 cap. oc, 500 neu. 3000 epocas objetivo Ídem, función trainrp

Resultados de simulacion de red:

Figura 176: Resultados Red 3 cap. oc, 500 neu. 3000 epoc obj. Ídem, función trainrp

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

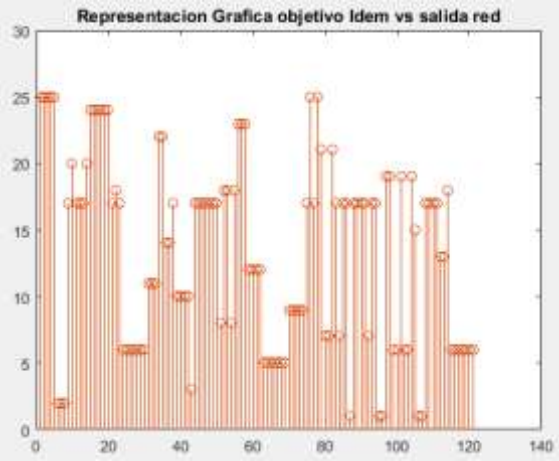

Figura 177: Represent.salida 3cap.oc, 500 neu. 3000epocas obj Ídem función trainrp

Representacion grafica del error cometido por dato:

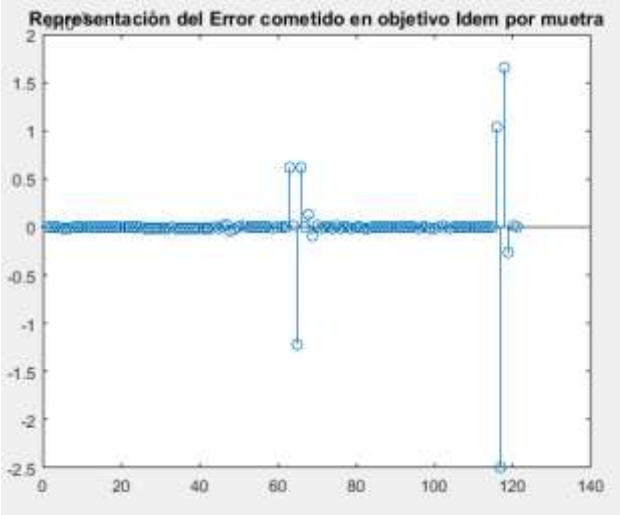

Figura 178: Represent. error 3cap.oc, 500 neu. 3000epocas obj Ídem función trainrp

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINRP 3 capa oculta 500 neuronas 3000 epocas Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 0.00
    Resultado MEA(Promedio absoluto del error): 0.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.2.3.4 Conclusiones y cuadro comparativo

Nº capas ocultas	Nº neuronas	Nº epocas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
3	500	500	RSSI	1.26	0.42
3	500	500	Idem	1.46	0.47
3	500	1500	RSSI	0	0.01
3	500	1500	Idem	0	0
3	500	3000	RSSI	0	0
3	500	3000	Idem	0	0

Tabla 7: Conclusiones según nº de epocas y función Trainrp

Se observa que a partir de un cierto numero de epocas de simulacion los errores producidos son cercanos a 0 y lo unico que se incrementa es el tiempo de simulacion.

Por lo tanto se concluye esta prueba con buenos resultados. La red que mejor resultados ofrece seria la red con entrenamiento trainrp con 3 capas ocultas, 500 neuonas por capa y 1500 epocas. Con esta red se tendria un error de entrenamiento de aproximadamente 0.

7.3 Estudio comparativo con función Trainscg

Se procede a realizar los entrenamientos y comparaciones de los resultados obtenidos partiendo de una red neuronal creada por defecto y con la función de entrenamiento Trainscg. Es necesario mantener los mismos parámetros de entrada y de salida para todas las

variaciones de entrenamientos realizadas. Se entrenará por separado los diferentes objetivos, es decir, habrá dos redes, una para los objetivos de RSSI y otra para los objetivos de Ídem. Los parámetros para cada red son:

NettraingdRSSI:

Entradas; Lat_Grados, Lat_min, Lat_seg, Lon_Grados, Lon_min, Lon_seg

Salida: RSSI

Nettraingdidem:

Entradas; Lat_Grados, Lat_min, Lat_seg, Lon_Grados, Lon_min, Lon_seg

Salida: ídem

7.3.1 Caso 1 de entrenamiento variando capas ocultas

Con los datos de entrada mencionados anteriormente se procede a realizar simulaciones de las redes neuronales. La primera simulación se realiza con todos los parámetros por defecto menos el número de capas ocultas, que se modificará con los valores de 1, 2 y 3 capas ocultas en dicha red.

7.3.1.1 A continuación se muestran los resultados obtenidos con función trainscg con 1 capa oculta y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

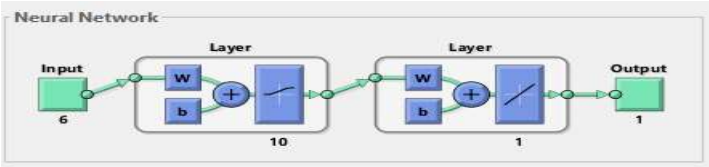

Figura 179: Red 1 capa oculta objetivo RSSI, función trainscg

Resultados de simulacion de red:

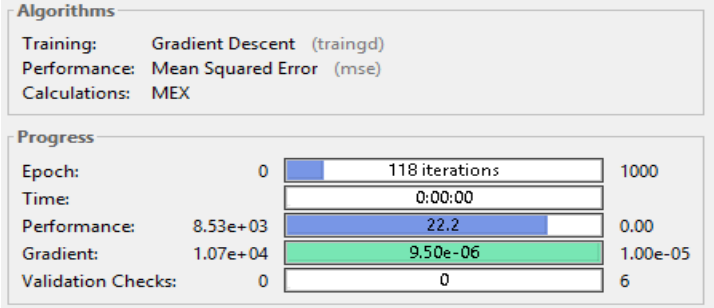

Figura 180: Resultados Red 1 capa oculta objetivo RSSI, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

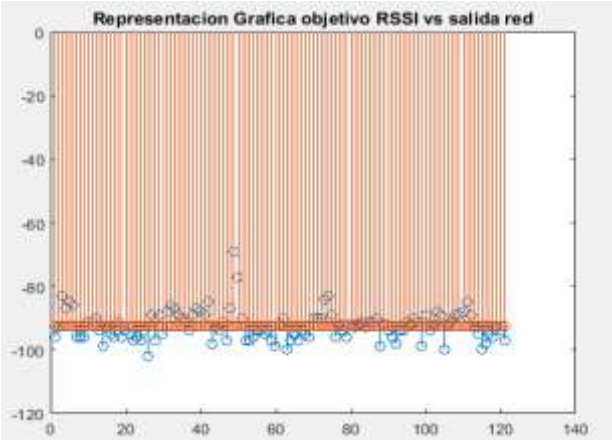

Figura 181: Representación salida 1 capa oculta objetivo RSSI, función trainscg

Representacion grafica del error cometido por dato:

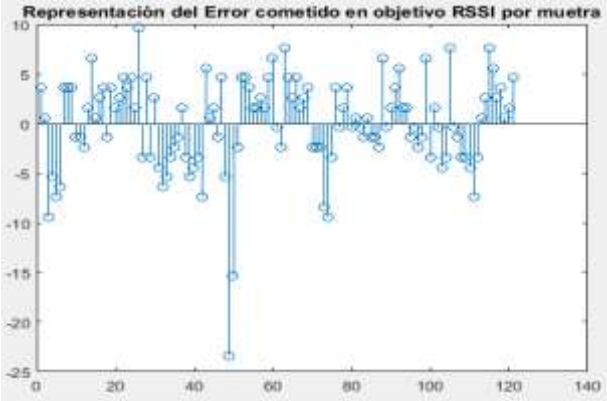

Figura 182: Representación error 1 capa oculta objetivo RSSI, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 1 capa oculta. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

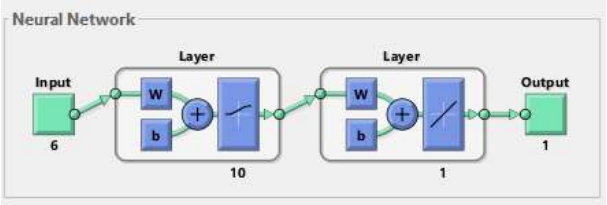

Figura 183: Red 1 capa oculta objetivo Ídem, función trainscg

Resultados de simulacion de red:

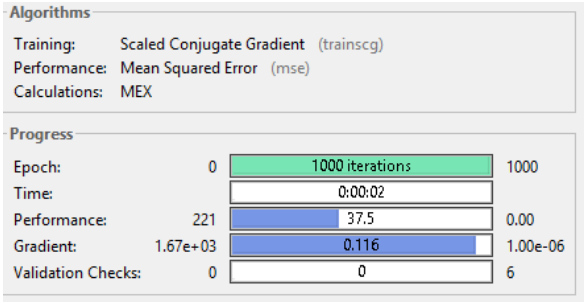

Figura 184: Resultados Red 1 capa oculta objetivo Ídem, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

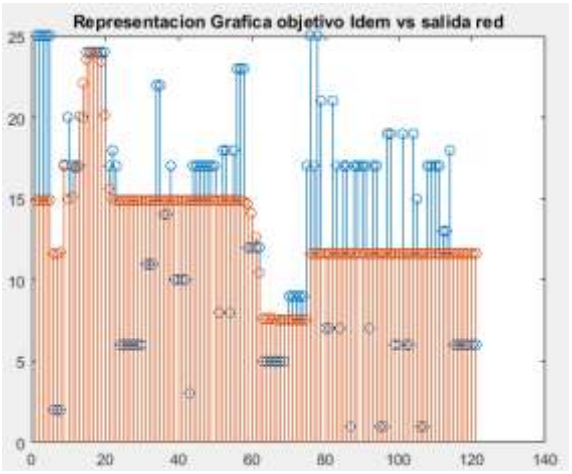

Figura 185: Representación Salida Red 1 capa oculta objetivo Ídem, función trainscg

Representacion grafica del error cometido por dato:

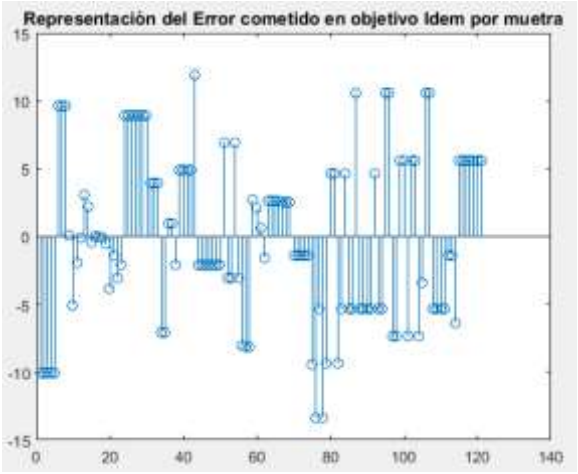

Figura 186: Representación error Red 1 capa oculta objetivo Ídem, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 1 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 37.45
    Resultado MEA(Promedio absoluto del error): 5.18
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.1.2 A continuación se muestran los resultados obtenidos con función trainscg con 2 capas ocultas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

Figura 187: Red 2 capa oculta objetivo RSSI, función trainscg

Resultados de simulacion de red:

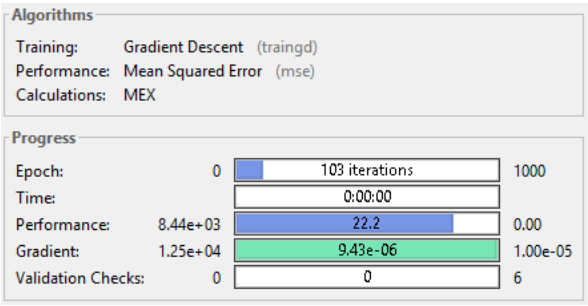

Figura 188: Resultados Red 2 capa oculta objetivo RSSI, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

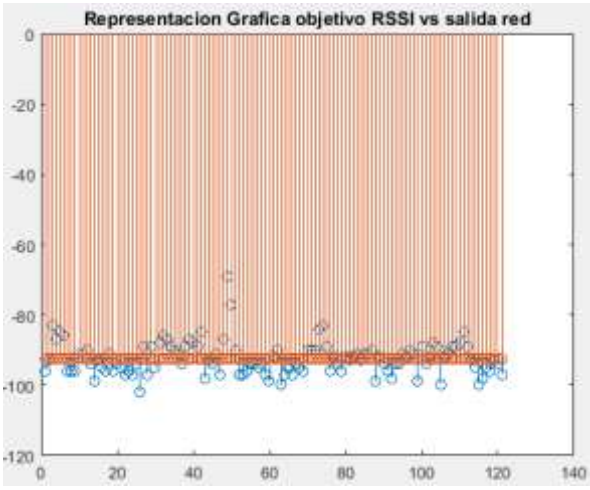

Figura 189: Representación Salida 2 capa oculta objetivo RSSI, función trainscg

Representacion grafica del error cometido por dato:

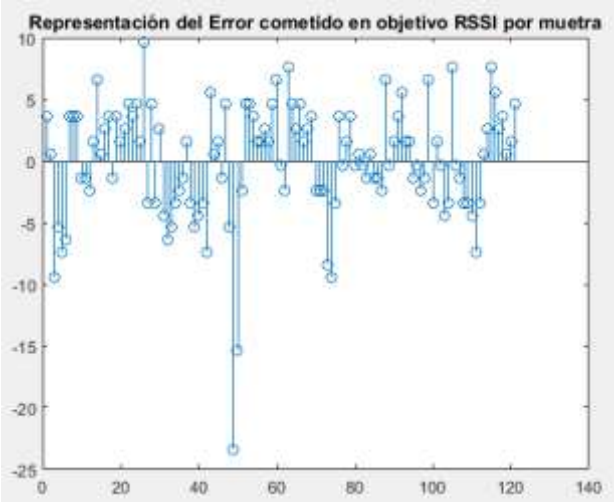

Figura 190: Representación Error 2 capa oculta objetivo RSSI, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

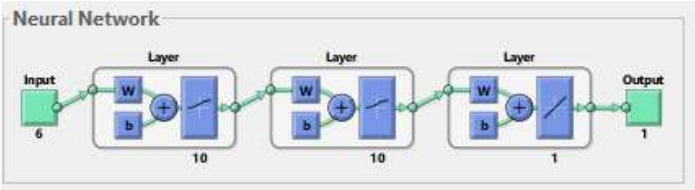

Figura 191: Red 2 capa oculta objetivo Ídem, función trainscg

Resultados de simulacion de red:

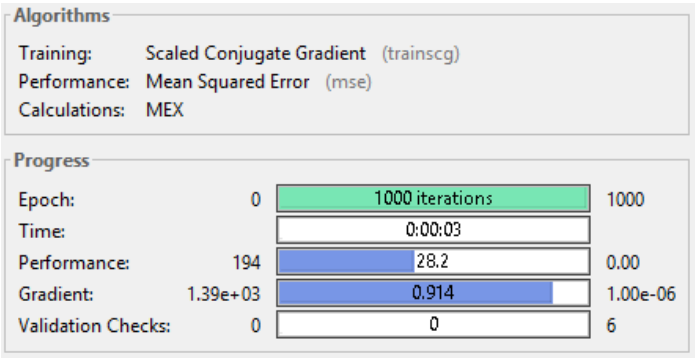

Figura 192: Resultados Red 2 capa oculta objetivo Ídem, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

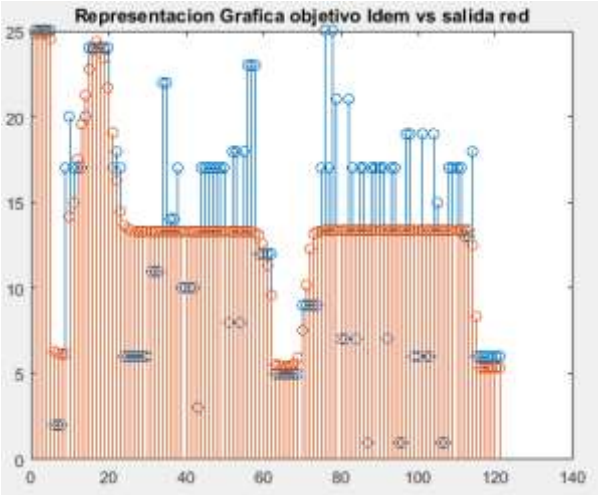

Figura 193: Representación Salida 2 capa oculta objetivo Ídem, función trainscg

Representacion grafica del error cometido por dato:

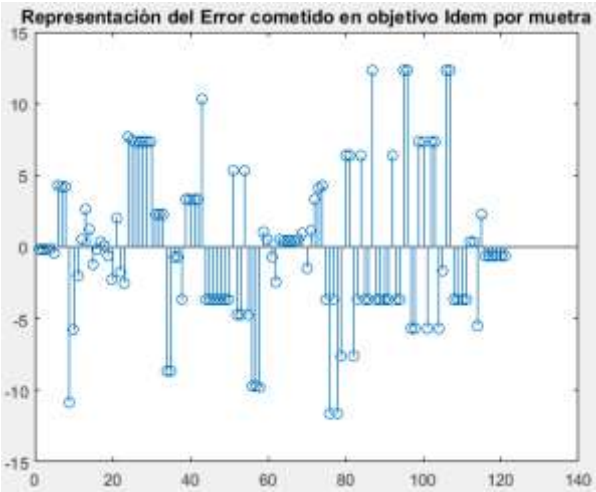

Figura 194: Representación error 2 capa oculta objetivo Ídem, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 28.19
    Resultado MEA(Promedio absoluto del error): 4.14
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.1.3 A continuación se muestran los resultados obtenidos con función trainscg con 3 capas ocultas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

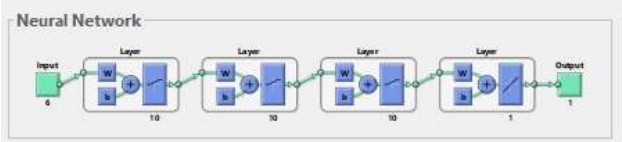

Figura 195: Red 3 capa oculta objetivo RSSI, función trainscg

Resultados de simulacion de red:

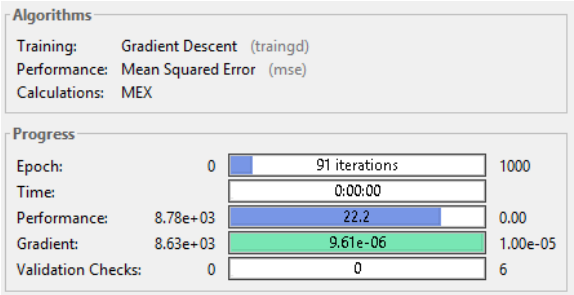

Figura 196: Resultados red 3 capa oculta objetivo RSSI, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

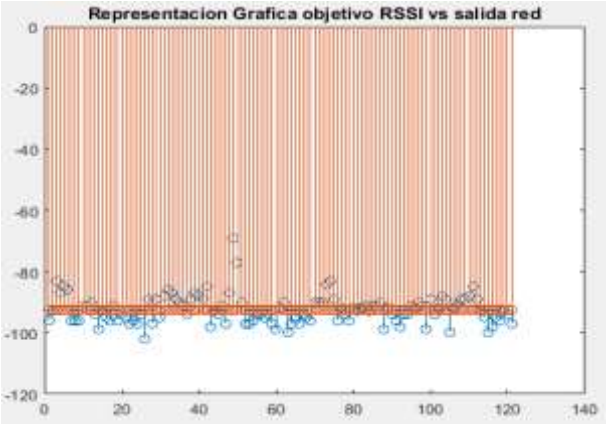

Figura 197: Representación salida 3 capa oculta objetivo RSSI, función trainscg

Representacion grafica del error cometido por dato:

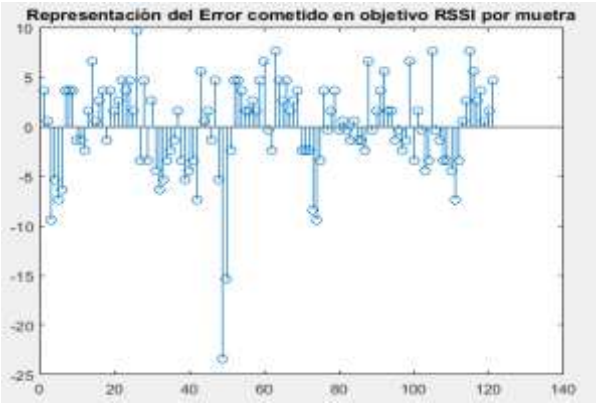

Figura 198: Representación error 3 capa oculta objetivo RSSI, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 3 capa oculta. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 22.17
    Resultado MEA(Promedio absoluto del error): 3.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

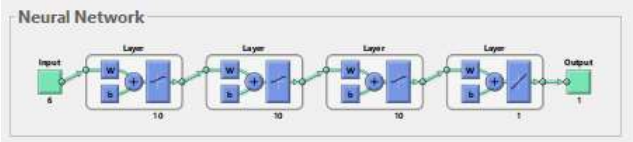

Figura 199: Red 3 capa oculta objetivo Ídem, función trainscg

Resultados de simulacion de red:

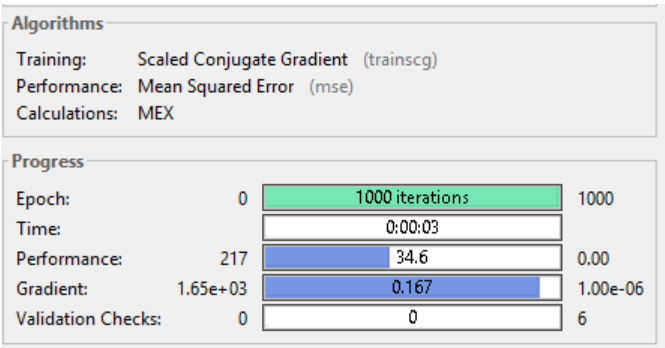

Figura 200: Resultados Red 3 capa oculta objetivo Ídem, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

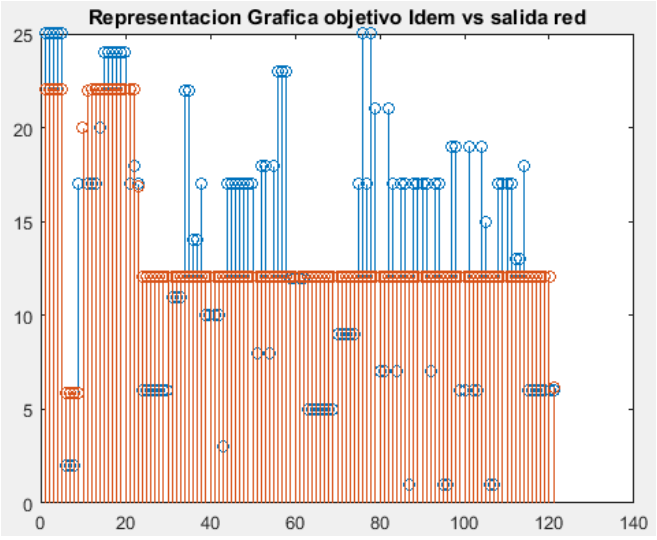

Figura 201: Representación salida 3 capa oculta objetivo Ídem, función trainscg

Representacion grafica del error cometido por dato:

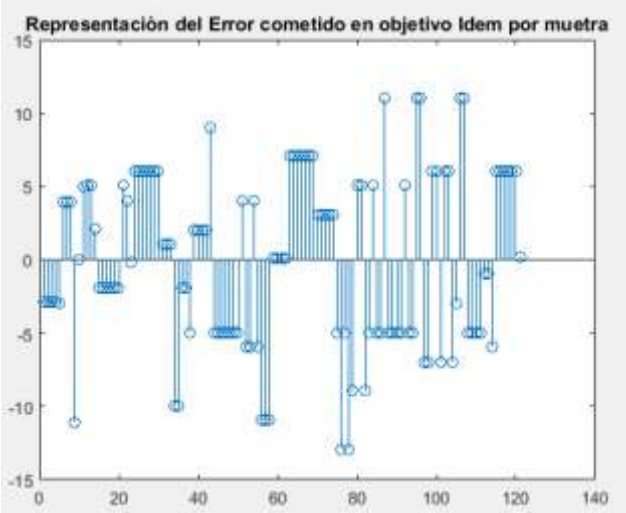

Figura 202: Representación error 3 capa oculta objetivo Ídem, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 3 capa oculta. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 34.64
    Resultado MEA(Promedio absoluto del error): 5.11
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.1.4 conclusiones y cuadro comparativo

Capas Ocultas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	RSSI	22.17	3.60
1	Idem	37.45	5.18
2	RSSI	22.17	3.60
2	Idem	28.19	4.14
3	RSSI	22.17	3.60
3	Idem	34.64	5.11

Tabla 8: Conclusiones según nº de capas ocultas y función Trainscg

Se observa que para este entrenamiento y con las caracteristicas conforme lo hemos definido se ajusta mejor la utilizacion de 2 capas ocultas, ya que produce menor pormedio cuadrado de error y menor promedio absoluto de error. Por lo que se decide continuar con las pruebas con 2 capas ocultas para las redes.

7.3.2 Caso 2 de entrenamiento variando nº de neuronas por capa

Como se acaba de comentar en el apartado anterior, se continúa con la realización de las pruebas con una red con 2 capa oculta con entrenamiento trainscg

En este apéndice se pretende variar el número de neuronas por capa oculta. Se van a realizar tres entrenamientos con 100 neuronas, 200 neuronas y 500 neuronas.

7.3.2.1 A continuación se muestran los resultados obtenidos con función trainscg con 2 capas ocultas, 100 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

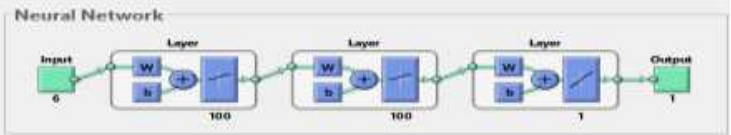

Figura 203: Red 2 capa oculta, 100 neuronas objetivo RSSI, función trainscg

Resultados de simulacion de red:

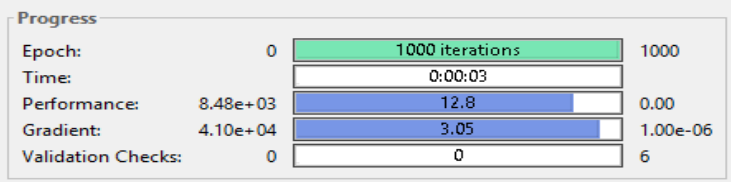

Figura 204: Resultados Red 2capa oculta, 100 neuronas objet. RSSI, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

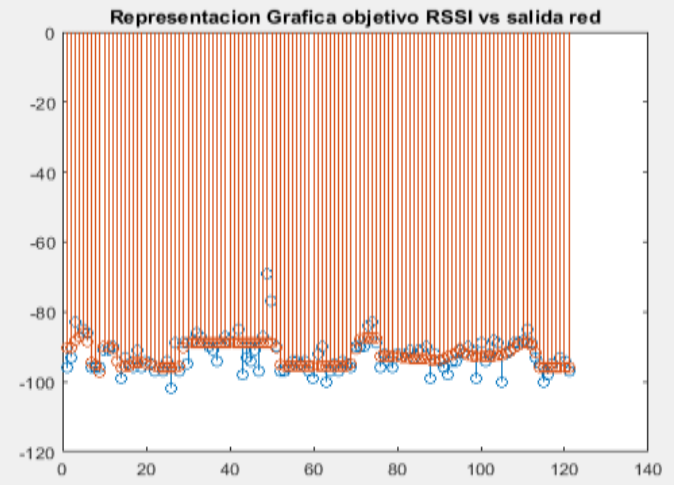

Figura 205: Representación salida 2ca.ocu, 100neuronas obj. RSSI, función trainscg

Representacion grafica del error cometido por dato:

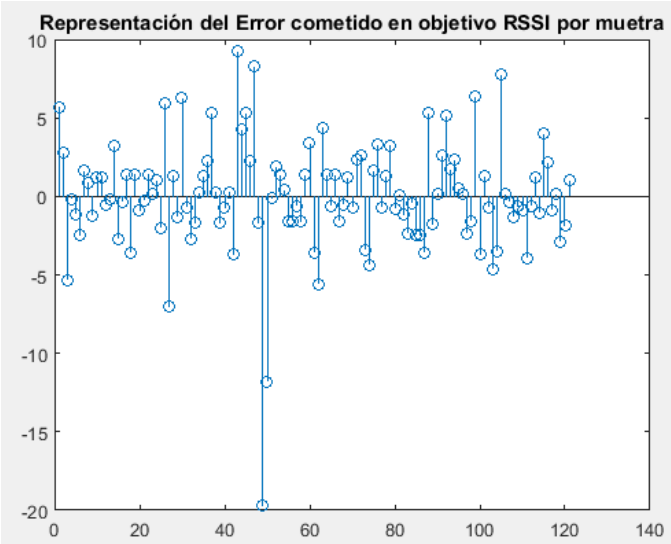

Figura 206: Representación error 2ca.ocu, 100neuronas obj. RSSI, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capas ocultas con 100 neuronas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 12.76
    Resultado MEA(Promedio absoluto del error): 2.43
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

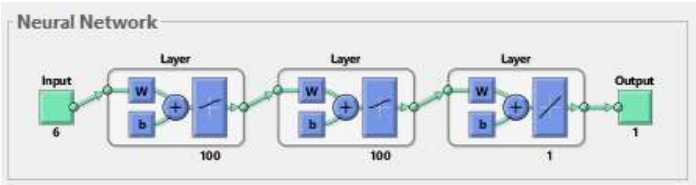

Figura 207: Red 2 capa oculta, 100 neuronas objetivo Ídem, función trainscg

Resultados de simulacion de red:

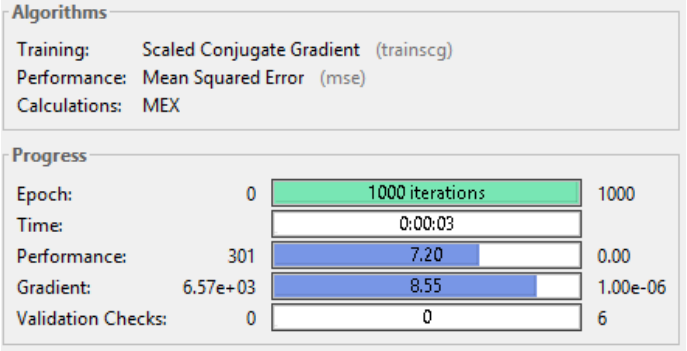

Figura 208: Resultados Red 2capa oculta, 100 neuronas objet. Ídem, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

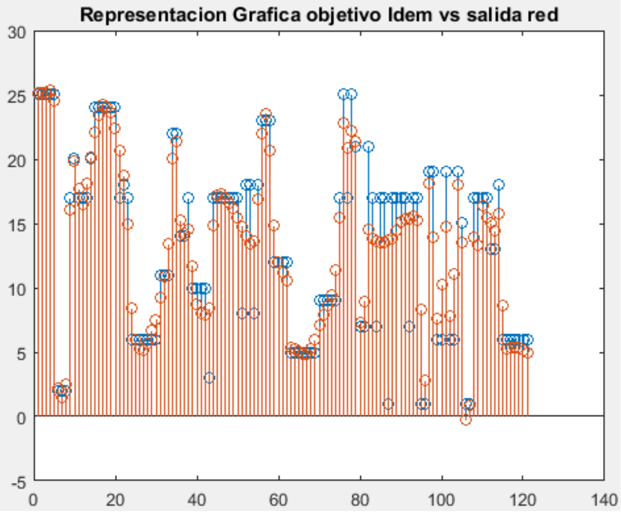

Figura 209: Representación salida 2ca.ocu, 100neuronas obj. Ídem, función trainscg

Representacion grafica del error cometido por dato:

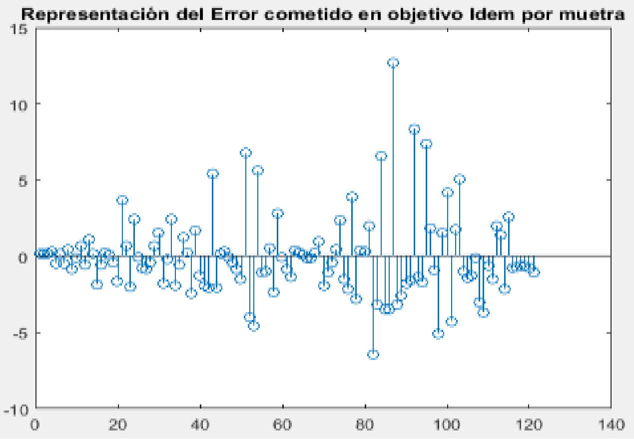

Figura 210: Representación error 2ca.ocu, 100neuronas obj. Ídem, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta, 100 neuroas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 7.20
    Resultado MEA(Promedio absoluto del error): 1.82
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.2.2 *A continuación se muestran los resultados obtenidos con función trainscg con 2 capas ocultas, 300 neuronas por capa y demás parámetros por defecto.*

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

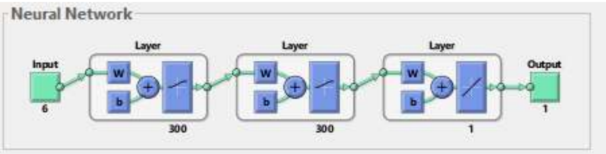

Figura 211: Red 2 capa oculta, 300 neuronas objetivo RSSI, función trainscg

Resultados de simulacion de red:

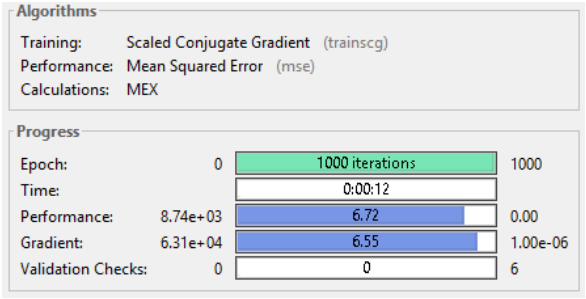

Figura 212: Resultados Red 2cap, 300 neuronas objetivo RSSI, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

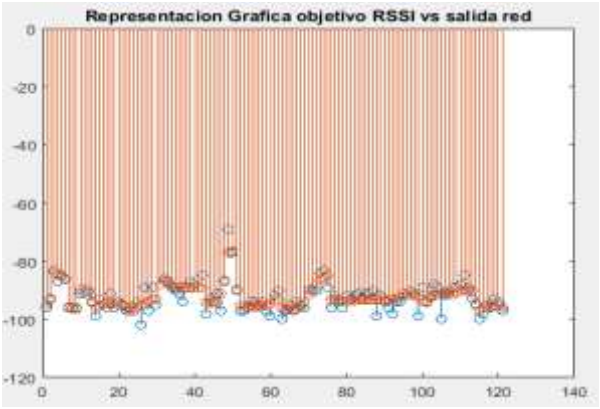

Figura 213: Representación salida Red 2cap, 300neuronas obj RSSI, función trainscg

Representacion grafica del error cometido por dato:

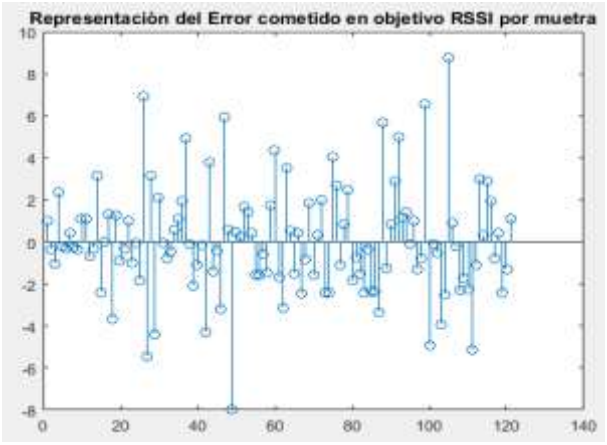

Figura 214: Representación error Red 2cap, 300neuronas obj RSSI, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capas ocultas con 300 neuronas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 6.72
    Resultado MEA(Promedio absoluto del error): 1.92
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

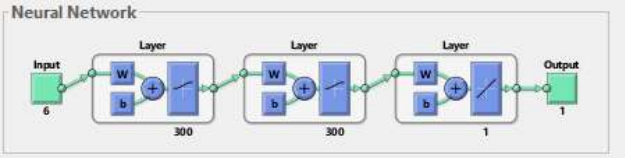

Figura 215: Red 2capa oculta, 300 neuronas objet. Ídem, función trainscg

Resultados de simulacion de red:

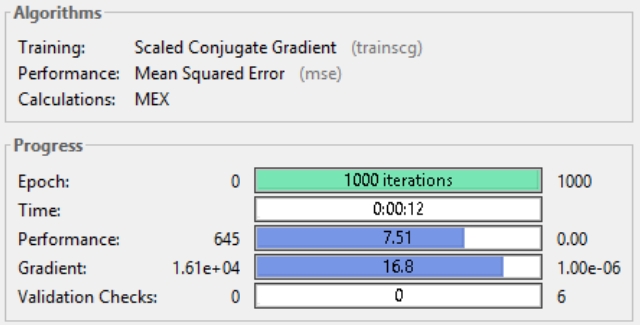

Figura 216: Resultados Red 2capa oculta, 300 neuronas objet. Ídem, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

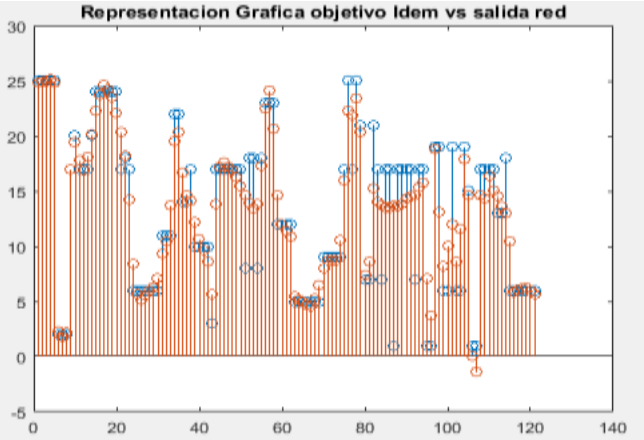

Figura 217: Representacion Salida 2cap 100 neuronas objet. Ídem, función trainscg

Representacion grafica del error cometido por dato:

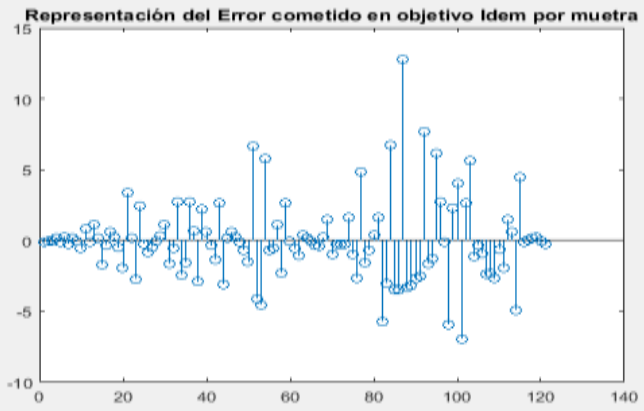

Figura 218: Representación error 2cap 100 neuronas objet. Ídem, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta, 300 neuroas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 7.51
    Resultado MEA(Promedio absoluto del error): 1.81
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.2.3 A continuación se muestran los resultados obtenidos con función trainscg con 2 capas ocultas, 500 neuronas por capa y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

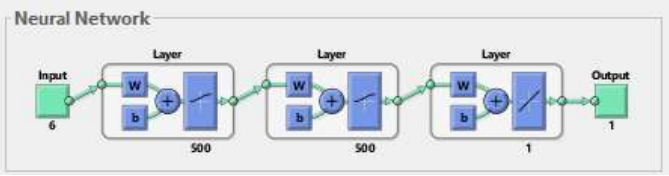

Figura 219: Red 2 capa oculta, 500 neuronas objetivo RSSI, función trainscg

Resultados de simulacion de red:

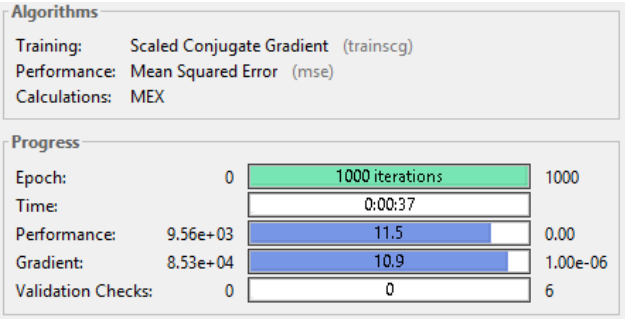

Figura 220: Resultados Red 2 cap 500 neuronas objetivo RSSI, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

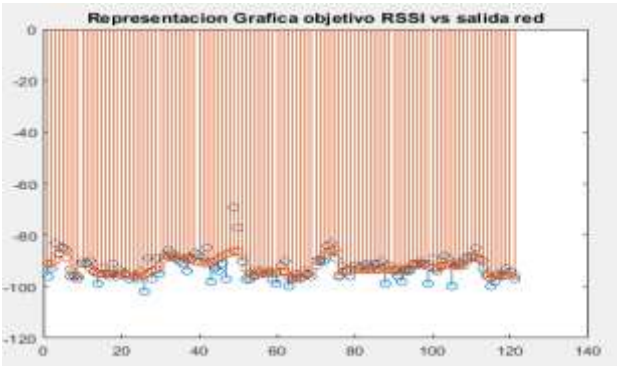

Figura 221: Representación salida 2 cap 500neuronas objetivo RSSI, función trainscg

Representacion grafica del error cometido por dato:

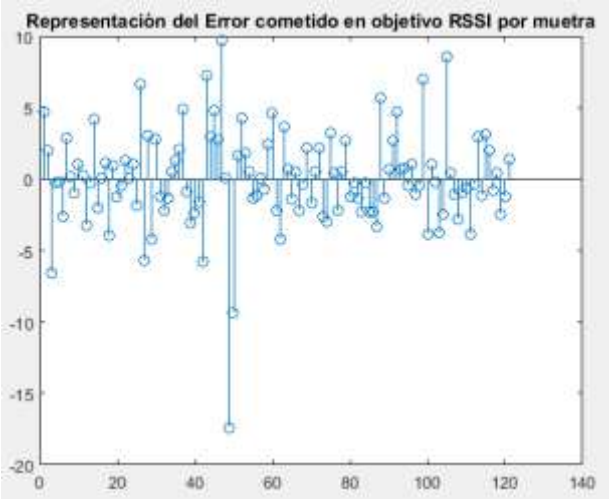

Figura 222: Representación error 2 cap 500neuronas objetivo RSSI, función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capas ocultas con 500 neuronas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 11.45
    Resultado MEA(Promedio absoluto del error): 2.37
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

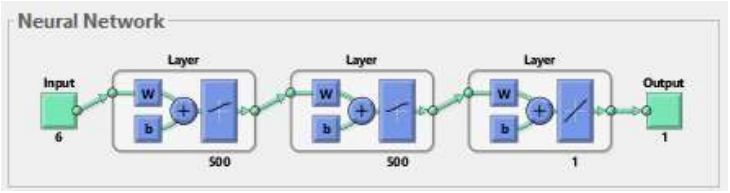

Figura 223: Red 2 capa oculta, 500 neuronas objetivo Ídem, función trainscg

Resultados de simulacion de red:

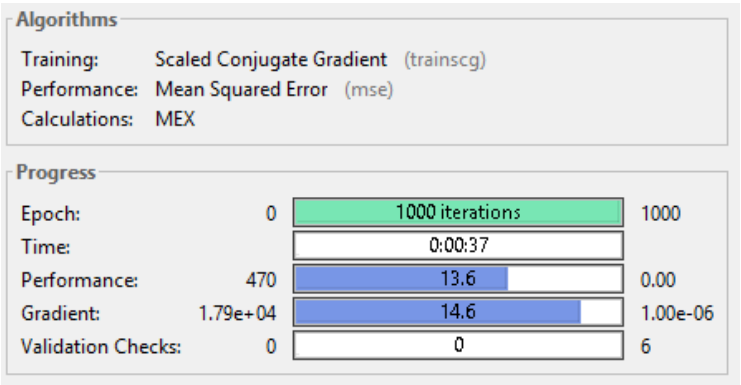

Figura 224: Resultados Red 2 cap 500 neuronas objetivo Ídem, función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

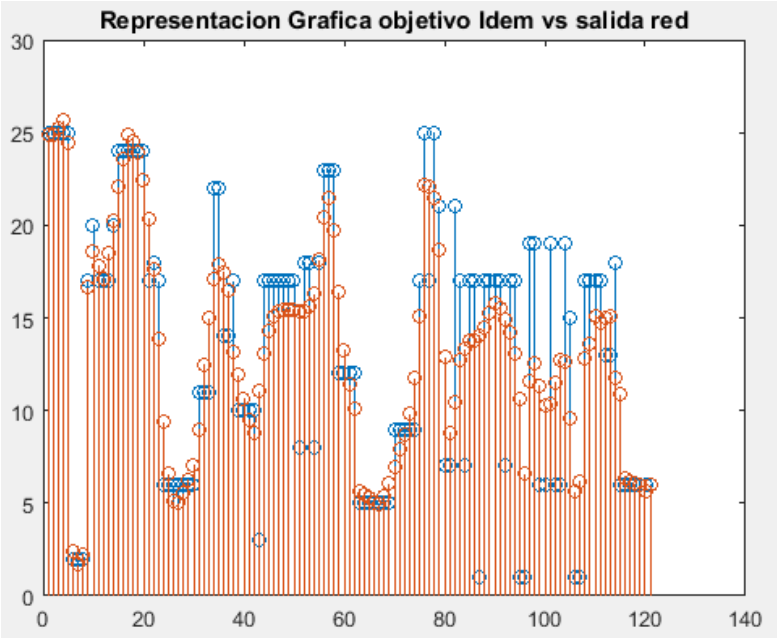

Figura 225: Representación salida 2 cap 500 neuronas objetivo Ídem función trainscg

Representacion grafica del error cometido por dato:

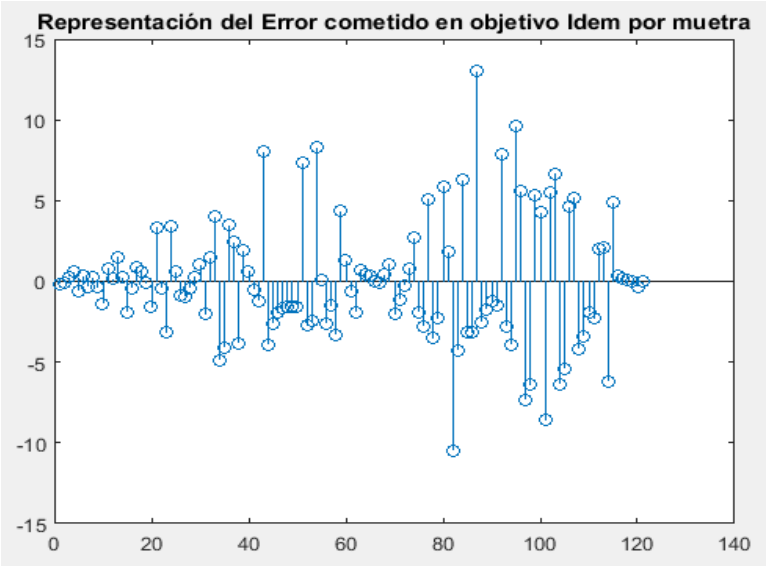

Figura 226: Representación error 2 cap 500 neuronas objetivo Ídem función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta, 500 neuroas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 13.58
    Resultado MEA(Promedio absoluto del error): 2.66
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.2.4 Conclusiones y cuadro comparativo

Capas Ocultas	Nº de neuronas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
2	100	RSSI	12.76	2.43
2	100	Idem	7.20	1.82
2	300	RSSI	6.72	1.92
2	300	Idem	7.51	1.81
2	500	RSSI	11.45	2.37
2	500	Idem	13.58	2.66

Tabla 9: Conclusiones según nº de neuronas y función Trainscg

Se observa que para este entrenamiento y con las caracteristicas conforme lo hemos definido se ajusta mejor la utilizacion de 2 capas ocultas con 300 neuronas por capa, ya que produce menor pormedio cuadrado de error y menor promedio absoluto de error. Por lo que se decide continuar con las pruebas con redes de 2 capas ocultas con 300 neuronas por capa.

7.3.3 Caso 3 de entrenamiento variando nº de épocas.

Como se acaba de comentar en el apartado anterior, se continúa con la realización de las pruebas con redes con 2 capa oculta y 300 neuronas.

En este apartado se pretende variar el número de épocas por capa. Se van a realizar tres entrenamientos con 500 épocas, 1500 épocas y 3000 épocas.

7.3.3.1 A continuación se muestran los resultados obtenidos con función trainscg con 2 capas ocultas, 300 neuronas por capa, 500 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

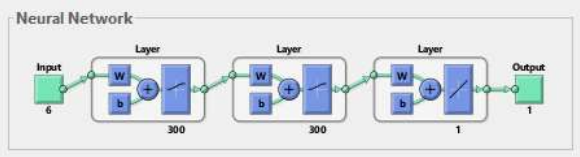

Figura 227: Red 2 capa oculta, 300 neuronas 500 epoc objetivo RSSI función trainscg

Resultados de simulacion de red:

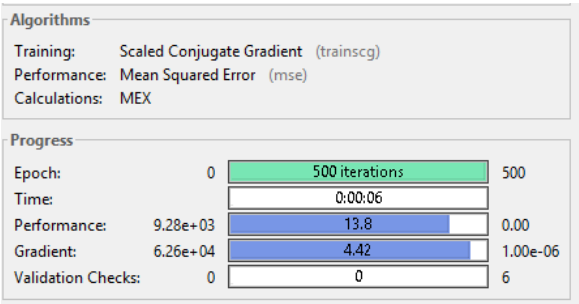

Figura 228: Resultados red 2 cap 300 neuronas 500 epoc obj RSSI función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

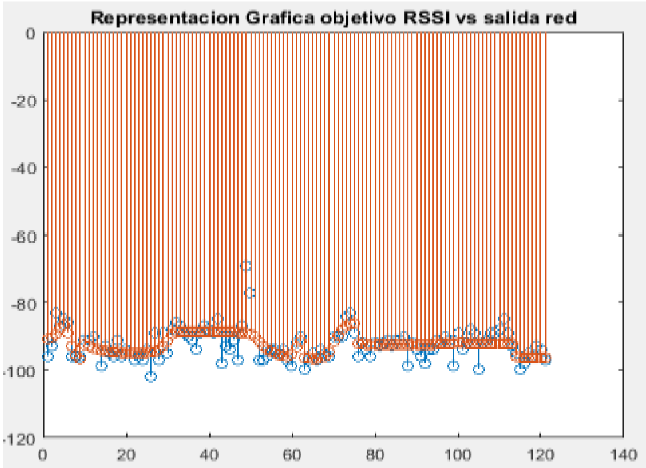

Figura 229: Representacion salida 2 cap 300 neur 500 epoc obj RSSI función trainscg

Representacion grafica del error cometido por dato:

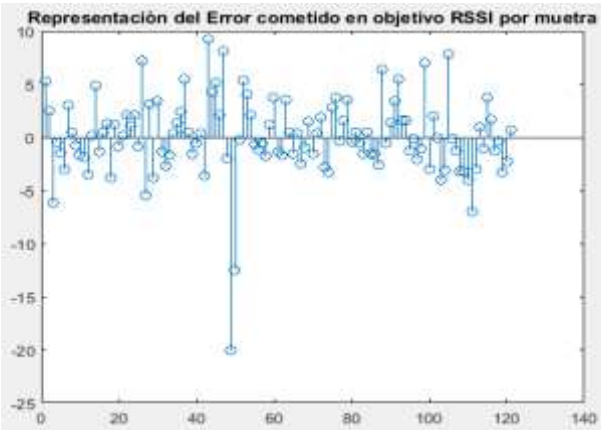

Figura 230: Representacion error 2 cap 300 neur 500 epoc obj RSSI función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capas ocultas con 300 neuronas y 500 epocas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 13.81
    Resultado MEA(Promedio absoluto del error): 2.60
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

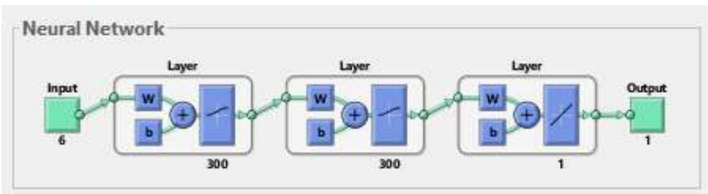

Figura 231: Red 2 capa oculta, 300 neuronas 500 epoc objetivo Ídem función trainscg

Resultados de simulacion de red:

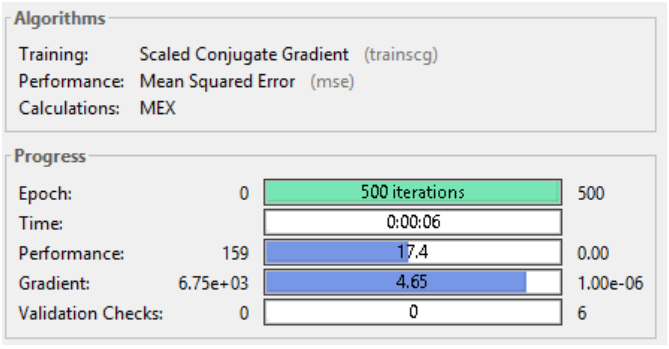

Figura 232: Resultados Red 2 cap, 300 neur 500 epoc objetivo Ídem función trainscg

Representación gráfica de salida de entrenamiento en color rojo y salida deseada en color azul:

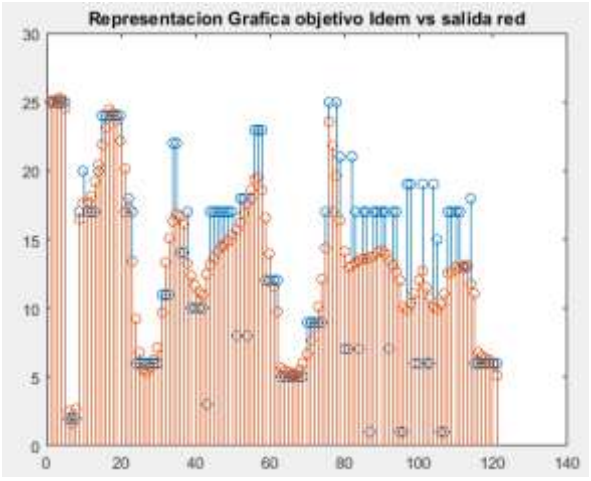

Figura 233: Representacion salida 2 cap, 300neur 500 epoc obj Ídem función trainscg

Representacion grafica del error cometido por dato:

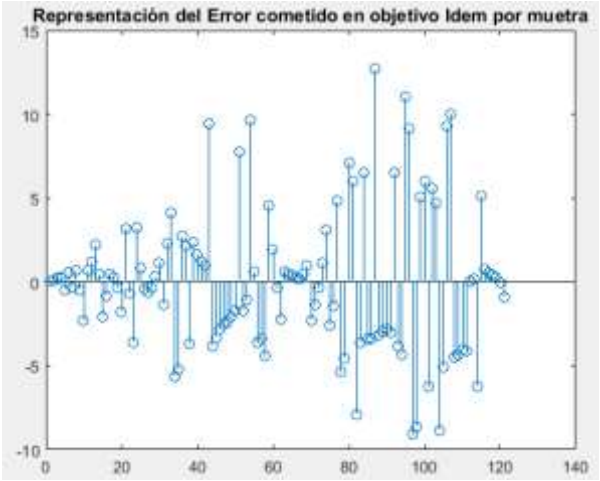

Figura 234: Representación error 2 cap, 300neur 500 epoc obj Ídem función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta, 300 neuroas y 500 epocas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 17.38
    Resultado MEA(Promedio absoluto del error): 3.06
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.3.2 *A continuación se muestran los resultados obtenidos con función trainscg con 2 capas ocultas, 300 neuronas por capa, 1500 épocas y demás parámetros por defecto.*

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

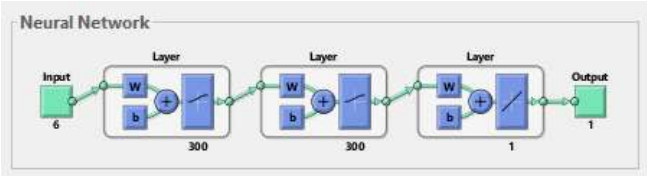

Figura 235: Red 2 capa oculta, 300 neuronas 1500epoc obj RSSI función trainscg

Resultados de simulacion de red:

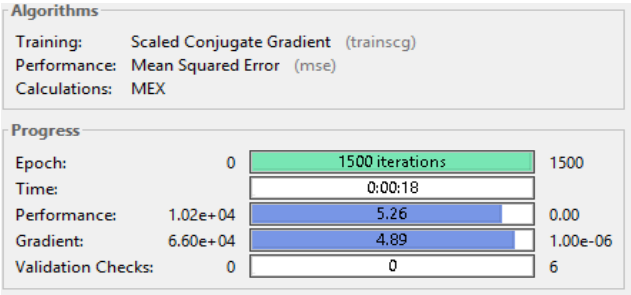

Figura 236: Resultados Red 2 cap, 300 neuronas 1500epoc obj RSSI función trainscg

Representación grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

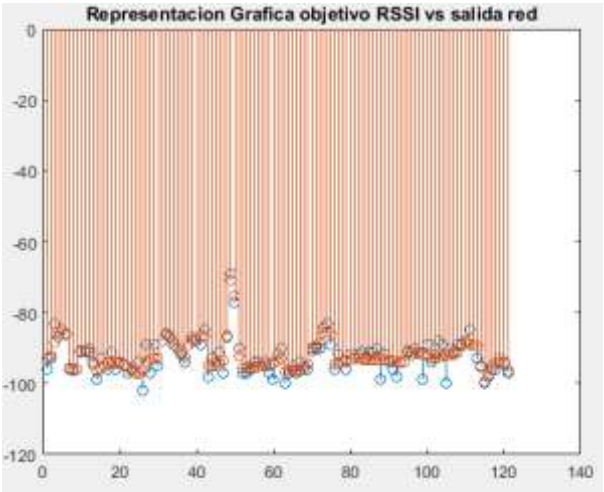

Figura 237: Repres. salida Red 2 cap, 300neur 1500epoc obj RSSI función trainscg

Representacion grafica del error cometido por dato:

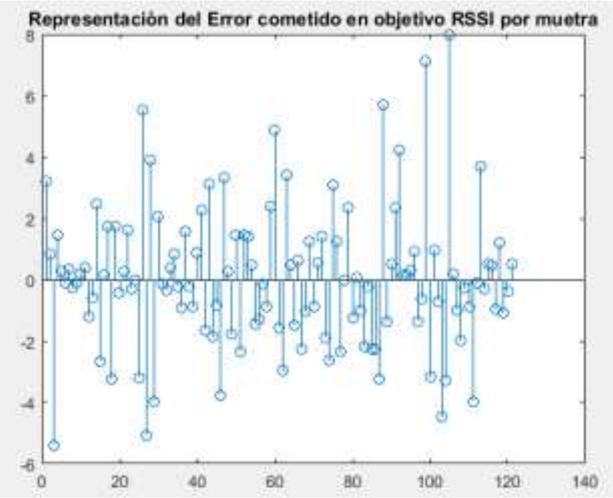

Figura 238: Repres. error Red 2 cap, 300neur 1500epoc obj RSSI función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capas ocultas con 300 neuronas y 1500 epocas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 5.26
    Resultado MEA(Promedio absoluto del error): 1.68
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

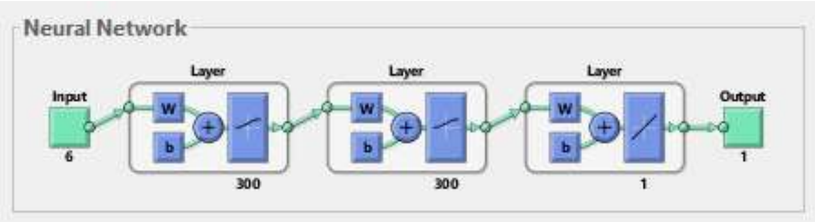

Figura 239: Red 2 capa oculta, 300 neuronas 1500epoc obj Ídem función trainscg

Resultados de simulacion de red:

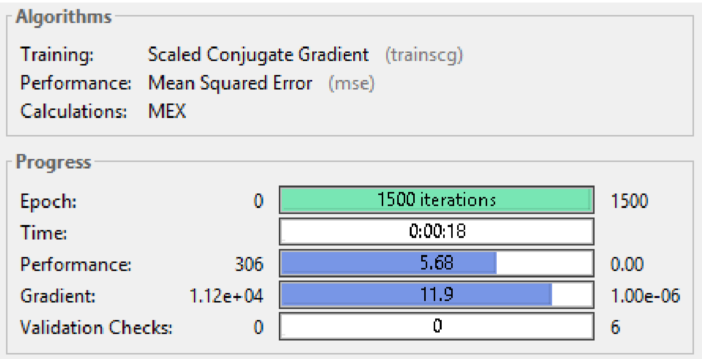

Figura 240: Resultados Red 2 cap, 300 neuronas 1500epoc obj Ídem función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

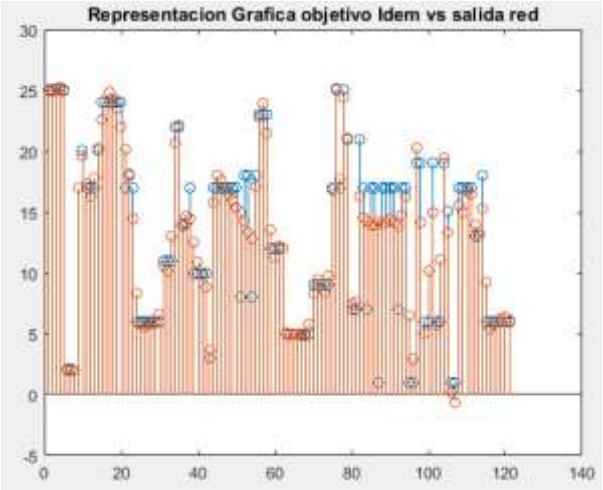

Figura 241: Repres. salida Red 2 cap, 300neur 1500epoc obj Ídem función trainscg

Representacion grafica del error cometido por dato:

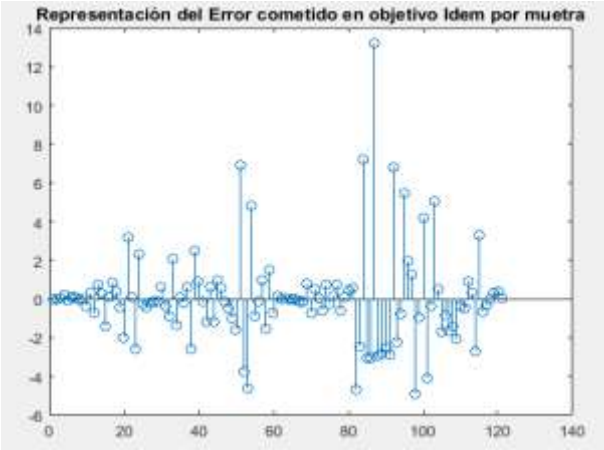

Figura 242: Repres. Error Red 2 cap, 300neur 1500epoc obj Ídem función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta, 300 neuronas y 1500 epocas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 5.68
    Resultado MEA(Promedio absoluto del error): 1.42
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

7.3.3.3 A continuación se muestran los resultados obtenidos con función trainscg con 2 capas ocultas, 300 neuronas por capa, 3000 épocas y demás parámetros por defecto.

Para entrenamiento con objetivo RSSI:

Representación gráfica de red:

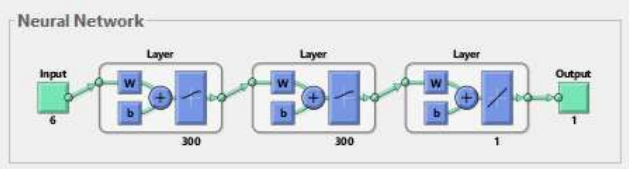

Figura 243: Red 2 capa oculta, 300 neuronas 3000epoc obj RSSI función trainscg

Resultados de simulacion de red:

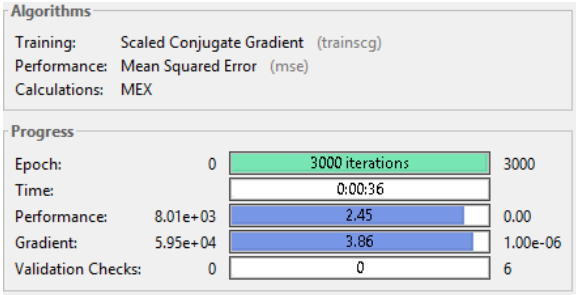

Figura 244: resultados Red 2 cap 300 neuronas 3000epoc obj RSSI función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

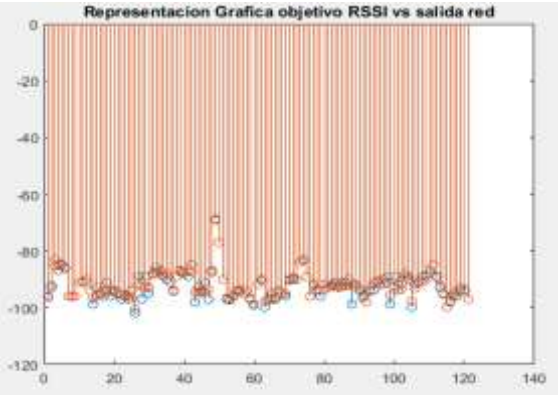

Figura 245: Represent salida Red 2 cap 300 neu 3000epoc obj RSSI función trainscg

Representacion grafica del error cometido por dato:

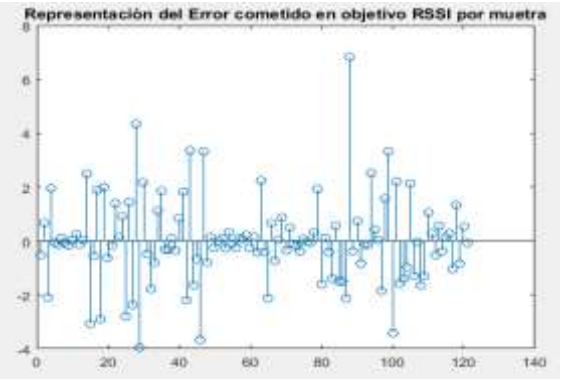

Figura 246: Represent error Red 2 cap 300 neu 3000epoc obj RSSI función trainscg

Resultados mostrados según programa implementado:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Red con funcion TRAINSCG y 2 capas ocultas con 300 neuronas y 3000 epocas. Salida RSSI
    Resultado MSE(Promedio cuadrado del error): 2.45
    Resultado MEA(Promedio absoluto del error): 1.07
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Para entrenamiento con objetivo Ídem:

Representación gráfica de red:

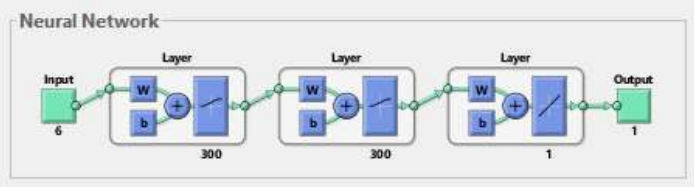

Figura 247: Red 2 capa oculta, 300 neuronas 3000epoc obj Ídem función trainscg

Resultados de simulacion de red:

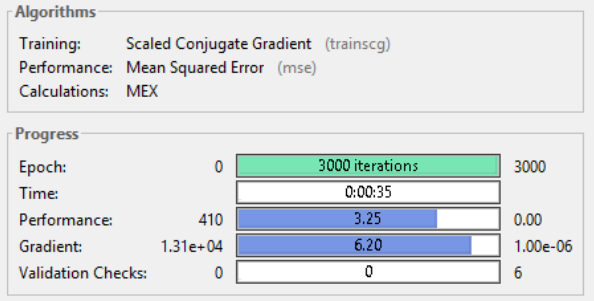

Figura 248: resultados Red 2 cap 300 neuronas 3000epoc obj Ídem función trainscg

Representacion grafica de salida de entrenamiento en color rojo y salida deseada en color azul:

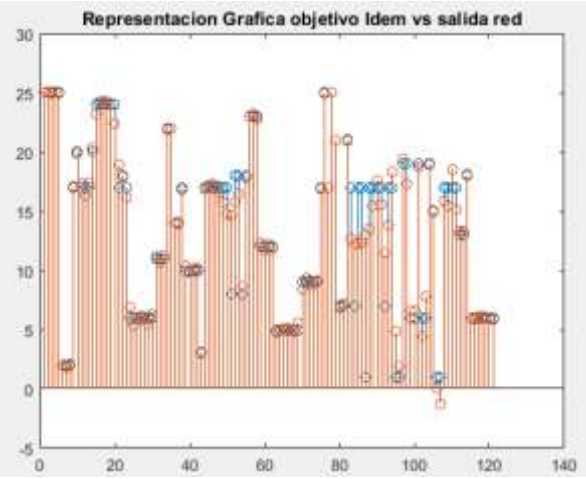

Figura 249: Represent salida Red 2 cap 300 neu 3000epoc obj Ídem función trainscg

Representacion grafica del error cometido por dato:

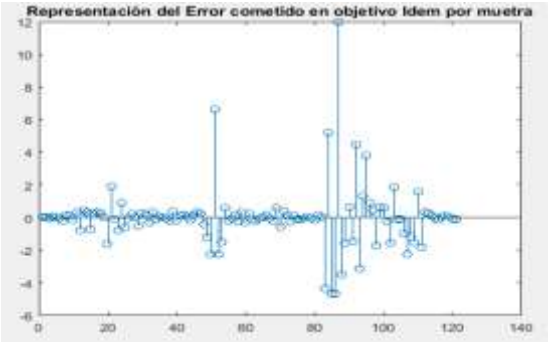

Figura 250: Represent error Red 2 cap 300 neu 3000epoc obj Ídem función trainscg

Resultados mostrados según programa implementado:

```
**************************************************************
    Resultados de Errores en Red con funcion TRAINSCG y 2 capa oculta, 300 neuronas y 3000 epocas. Salida Idem:
    Resultado MSE(Promedio cuadrado del error): 3.25
    Resultado MEA(Promedio absoluto del error): 0.85
**************************************************************
```

7.3.3.4 Conclusiones y cuadro comparativo

Capas Ocultas	Nº de neuronas	nº epocas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
2	300	500	RSSI	11.45	2.37
2	300	500	Idem	12.76	2.43
2	300	1500	RSSI	5.26	1.48
2	300	1500	Idem	5.68	1.43
2	300	3000	RSSI	2.45	1.07
2	300	3000	Idem	3.25	0.85

Tabla 10: Conclusiones según nº de épocas y función Trainscg

Se observa que con un numero mayor de epocas el entrenamiento de las redes ofrene menos promedio cuadrado de error y menos promedio absoluto de error.

7.4 Conclusiones de entrenamiento

Debido a la cantidad de resultados obtenidos en los entrenaminetos anteriores se realiza en este apartado un resumen con los resultados mas relevantes.

7.4.1 Mejores resultados y comentarios sobre el metodo de entrenamiento traingd:

Nº capas ocultas	Nº neuronas	Nº épocas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	100	500	RSSI	22.17	3.60
1	100	500	Idem	49.50	6.25
1	100	1500	RSSI	22.17	3.6
1	100	1500	Idem	49.50	6.25
1	100	3000	RSSI	22.17	3.6
1	100	3000	Idem	49.50	6.25

Tabla 11: Mejores resultados entrenamiento Traingd

Se observa que aún vairando el numero de capas ocultas, el numero de neuronas y el numero de epocas, se sigue produciendo demasiado promedio cuadrado de error en los entrenamientos.

Se observa que en los entrenamientos con variacion del nº capas ocultas se alcanza muy rapido el valor minimio de gradiente y por lo tanto la simualcion termina rápidamente.

Se observa que al incrementar el nº de neuroas por capa oculta empeora el entrenamiento probocando sobre entrenamiento y produciendo a la salida el siguiente resultado; NaN (not a Number).

Se observa que para este metodo de entrenamiento conforme se aumenta el numero de epocas se alcanza antes el minimo gradiente de simulacion, no permintiendo finalizar el entrenamiento con un minimo de error.

Por todas estas razones el metodo de entrenamiento traingd no es valido para la finalidad de este TFG ya que los resultados obtenidos no son concluentes.

7.4.2 Mejores resultados y comentarios sobre el metodo de entrenamiento trainrf:

Se aprecia que en los entrenamientos con variacion del nº de capas ocultas muestran una mejora en el rendimiento de la red, reduciendo el promedio cuadrado de error en más de la mitad. Por lo tanto se aprecia que al aumentar el numero de capas ocultas, los errores disminuyen y se producen mejores resultados. La siguiente tabla muestra los resultados de simulacion variando el numero de capas ocultas:

Nº capas ocultas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	RSSI	22.17	3.6
1	Idem	48.08	6.10
2	RSSI	12.46	2.48
2	Idem	14.63	6.68
3	RSSI	9.06	2.05
3	Idem	10.9	2.16

Tabla 12: resultados variando número de capas entrenamiento Trainrf

Se observa que en los entrenamientos con variacion del nº neuronas por capa oculta los resultados obtenidos son my buenos. Se muestra en la siguiente tabla que al aumentar el numero de neuronas el promedio de error cuadro practicamente desaparece.

Nº capas ocultas	Nº neuronas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
3	100	RSSI	1.56	0.79
3	100	Idem	0.02	0.05
3	300	RSSI	0.26	0.19
3	300	Idem	0.16	0.14
3	500	RSSI	0.07	0.13
3	500	Idem	0.03	0.05

Tabla 13: Mejores resultados entrenamiento Trainrf

Hay que hacer constancia de que al aumentar el nº de neuronas por capa oculta se produce un aumento en el tiempo de simulacion. En el entrenamiento con 100 neuronas tarda al rededor de 3 segundos mientras el entrenamiento con 500 neuronas tarda casi 50 segundos. Esta demora puede ocasinar un mayor problema cuando se aumenten los datos de entrada de entrenamiento.

Por ultimo, en los entrenamientos variando el numero de epocas se aprecia que al tener un elevado numero de epocas (mayor a 1500) el entrenamiento tarda demasiado tiempo en realizarse. Aunque dicho entrenamiento presenta mejores resultados con mayor numero de epocas, el tiempo empleado a razon de la mejora en los resultados no sale a cuenta.

Por lo tanto, este metodo de entrenamiento (trainrf) con la configuracion correspondiente (3 capas ocultas,500 neuronas por capa y 1500 epocas) si muestra resultados concluentes.

7.4.3 Mejores resultados y comentarios sobre el metodo de entrenamiento trainscg:

Se observa que la variacion en el numero de capas ocultas no afecta demasiado al error producido. Se muestra, en los entrenamientos realizados, que se alcanza demasiado rápido el minimo gradiente para la simulacion. Aún así, para una configuracion de este entrenamiento con 2 capas ocultas ofrece para una salida de la red con menor promedio cuadrado de error. Se muestra en la siguiente tabla dichos resultados:

Capas Ocultas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
1	RSSI	22.17	3.60
1	Idem	37.45	5.18
2	RSSI	22.17	3.60
2	Idem	28.19	4.14
3	RSSI	22.17	3.60
3	Idem	34.64	5.11

Tabla 14: Resultados entrenamiento Trainscg 2 capas

Se aprecia que al aumentar el numero de neuronas por capa oculta disminuye el promedio cuadrado de error producido. Tambien se aprecia que cuando se realiza el entrenamiento con un numero elvado de neuroas (500 o más) vuelve a aumentar el error producido, se cree que la causa por la cual se produce el aumento del error es por el sobre entrenamiento de la red. Como se muestra en la tabla inferior, el mejor resultado se ofrece con un valor intermedio de nº de neuronas.

Capas Ocultas	Nº de neuronas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
2	100	RSSI	12.76	2.43
2	100	Idem	7.20	1.82
2	300	RSSI	6.72	1.92
2	300	Idem	7.51	1.81
2	500	RSSI	11.45	2.37
2	500	Idem	13.58	2.66

Tabla 15: Resultados entrenamiento Trainscg con aumento de numero de neurona

Por último, se aprecia que con el aumento del número de épocas se produce un menor promedio cuadrado de error. Se muestra en la siguiente tabla los resultados obtenidos con la variación de número de épocas.

Capas Ocultas	Nº de neuronas	nº epocas	Objetivo	Promedio cuadrado del error	Promedio absoluto del error
2	300	500	RSSI	11.45	2.37
2	300	500	Idem	12.76	2.43
2	300	1500	RSSI	5.26	1.48
2	300	1500	Idem	5.68	1.43
2	300	3000	RSSI	2.45	1.07
2	300	3000	Idem	3.25	0.85

Tabla 16: Mejores resultados entrenamiento Trainscg

Por lo tanto, este metodo de entrenamiento (trainscg) con la configuración correspondiente (2 capas ocultas,300 neuronas por capa y 3000 epocas) si muestra resultados concluentes, anque se tendría que considerar la posivilidad de encontrar errores en los entrenamientos realizados, ya que el error ocasionado no se podria despreciar.

8 SIMULACIÓN Y CASOS PRÁCTICOS

Viendo las conclusiones y los resultados obtenidos en el apartado 7 se cree conveniente utilizar el mejor método de entrenamiento. Según estos resultados el mejor método de entrenamiento es el trainrp que aporta un error prácticamente insignificante siempre y cuando se entrene con su configuración correspondiente.

Por lo tanto se procede a crear programas en Matlab en el cual se pueda entrenar y posteriormente simular los tres casos diferentes de ubicaciones que encontramos. Para esto, se implementan por separado los entrenamientos de cada caso y a continuación se cargan las redes ya entrenadas en el programa genérico de simulación.

8.1 Ubicación en el centro de Seattle.

En este apartado se crea una red, se entrena y se simula con parámetros de entrada pertenecientes a valores de ubicaciones del centro de Seattle.

8.1.1 Entrenamiento

Se muestra a continuación las redes creadas y los resultados obtenidos al entrenarla para los objetivos de RSSI e Ídem:

El nombre de las redes es: netFinaltrainRSSIcentro.mat para RSSI y netFinaltrainIdemcentro.mat para Ídem.

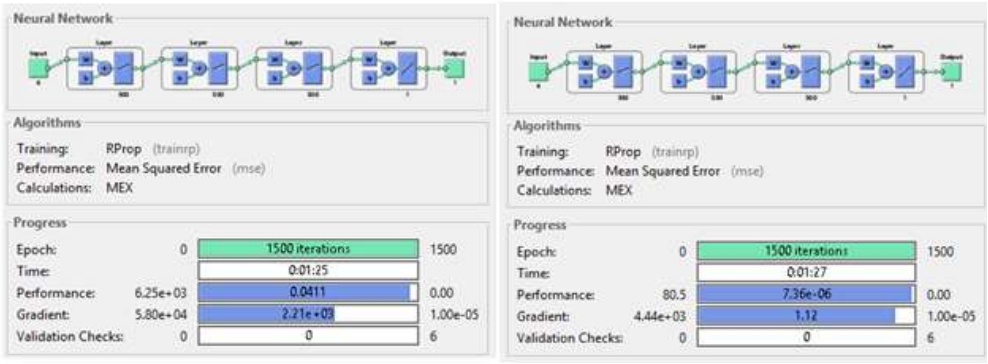

Figura 251: Redes implementadas para Centro Seattle

Resultados por muestra del error producido los entrenamientos:

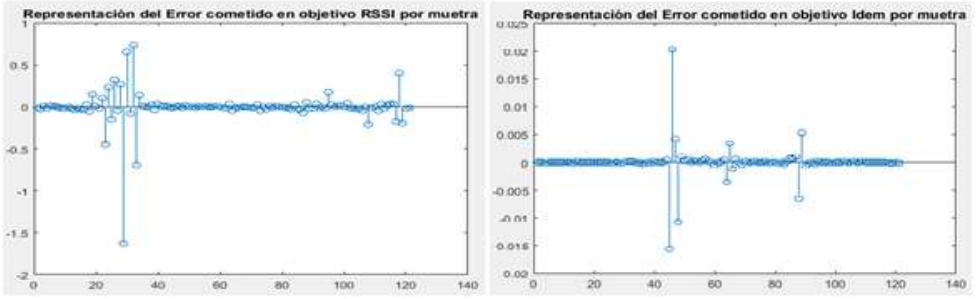

Figura 252: Representación del error en Redes Implementadas para centro Seattle

Se aprecia que el error producido es prácticamente 0 en ambas redes (exceptuado algunas muestras).

Se muestra ahora las salidas obtenidas del entrenamiento en color azul y las salidas reales en rojo.

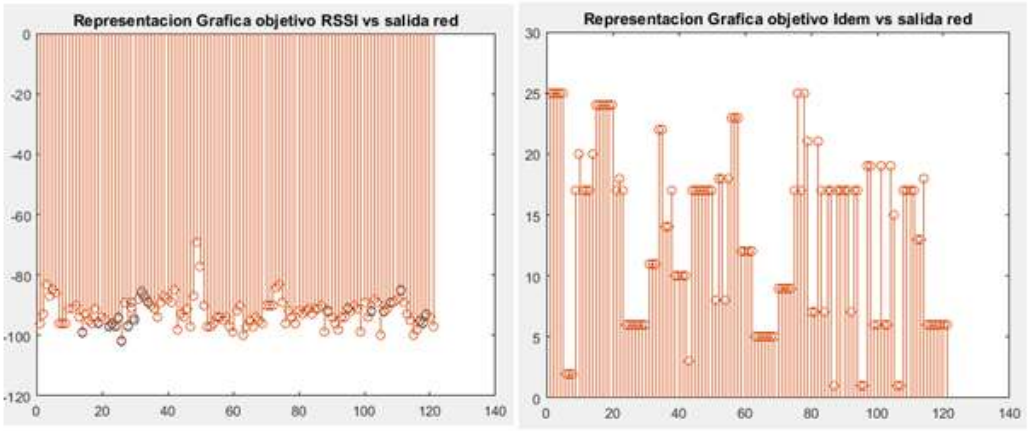

Figura 253: Representación de salidas en Redes Implementadas para centro Seattle

Por último, se muestra los resultados de error obtenidos por los entrenamientos empleados:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en RSSI para red con ubicaciones de Centro de Seattle
    Resultado MSE(Promedio cuadrado del error): 0.04
    Resultado MEA(Promedio absoluto del error): 0.07
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Idem para red con ubicaciones de Centro de Seattle
    Resultado MSE(Promedio cuadrado del error): 0.00
    Resultado MEA(Promedio absoluto del error): 0.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

8.1.2 Simulación

Al tener ya creadas las redes, se procede a cargar estas redes y simularlas. La simulación se realiza dentro del programa general como se comentó anteriormente.

Simulación 1:

Se realiza una simulación con los siguientes datos de ubicación:

```
>> GENERAL
Ingrese el valor de Latitud en Grados: 47
Ingrese el valor de Latitud en Minutos: 40
Ingrese el valor de Latitud en Segundos: 24
Ingrese el valor de Longitud en Grados: 122
Ingrese el valor de Longitud en Minutos: 9
Ingrese el valor de Longitud en Segundos: 49.9
Ingrese 1 si la ubicacion esta en zona centro, 2 si la ubicacion esta en zona revenna o 3 si la ubicacion esta en zona Kirkland:
1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Arpoximacion de valor RSSI: -87.43
    Arpoximacion de valor Idem: 10.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de

Lat_grados=47, Lat_min=40, Lat_seg=23.6, Lon_grados=122, Lon_min=9, Lon_seg=49.9 con resultados RSSI= -89 e Ídem=10 al modificar la ubicación se obtiene los resultados de RSSI=-87.43 e Ídem=10 por lo que los resultados obtenidos son correctos.

Simulación 2:

Se realiza una simulación con los siguientes datos de ubicación:

```
>> GENERAL
Ingrese el valor de Latitud en Grados: 47
Ingrese el valor de Latitud en Minutos: 40
Ingrese el valor de Latitud en Segundos: 5.1
Ingrese el valor de Longitud en Grados: 122
Ingrese el valor de Longitud en Minutos: 18
Ingrese el valor de Longitud en Segundos: 15.7
Ingrese 1 si la ubicacion esta en zona centro, 2 si la ubicacion esta en zona revenna o 3 si la ubicacion esta en zona Kirkland:
1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     Arpoximacion de valor RSSI: -81.08
     Arpoximacion de valor Idem: 25.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=40, Lat_seg=5.1, Lon_grados=122, Lon_min=18, Lon_seg=16.3 con resultados RSSI= -87 e Ídem=25 al modificar la ubicación se obtiene los resultados de RSSI=-81.01 e Ídem=25 por lo que los resultados obtenidos son correctos.

Simulación 3:

Se realiza una simulación con los siguientes datos de ubicación:

```
>> GENERAL
Ingrese el valor de Latitud en Grados: 47
Ingrese el valor de Latitud en Minutos: 40
Ingrese el valor de Latitud en Segundos: 45.9
Ingrese el valor de Longitud en Grados: 122
Ingrese el valor de Longitud en Minutos: 9
Ingrese el valor de Longitud en Segundos: 55.9
Ingrese 1 si la ubicacion esta en zona centro, 2 si la ubicacion esta en zona revenna o 3 si la ubicacion esta en zona Kirkland:
1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
     Arpoximacion de valor RSSI: -84.46
     Arpoximacion de valor Idem: 9.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=40, Lat_seg=45.9, Lon_grados=122, Lon_min=9, Lon_seg=55.7 con resultados RSSI= -90 e Ídem=9 al modificar la ubicación se obtiene los resultados de RSSI=-84.46 e Ídem=9 por lo que los resultados obtenidos son correctos.

8.2 Ubicación en Zona de Revenna en Seattle

En este apartado se crea una red, se entrena y se simula con parámetros de entrada pertenecientes a valores de ubicaciones de Revenna.

8.2.1 Entrenamiento

Se muestra a continuación las redes creadas y los resultados obtenidos al entrenarla para los objetivos de RSSI e Ídem:

El nombre de las redes es: netFinaltrainRSSIrevenna.mat para RSSI y netFinaltrainIderevenna.mat para Ídem.

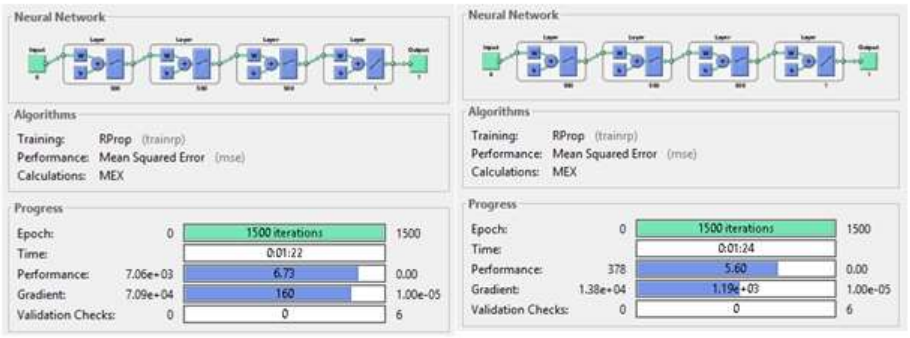

Figura 254: Redes implementadas para Revenna en Seattle

Resultados por muestra del error producido los entrenamientos:

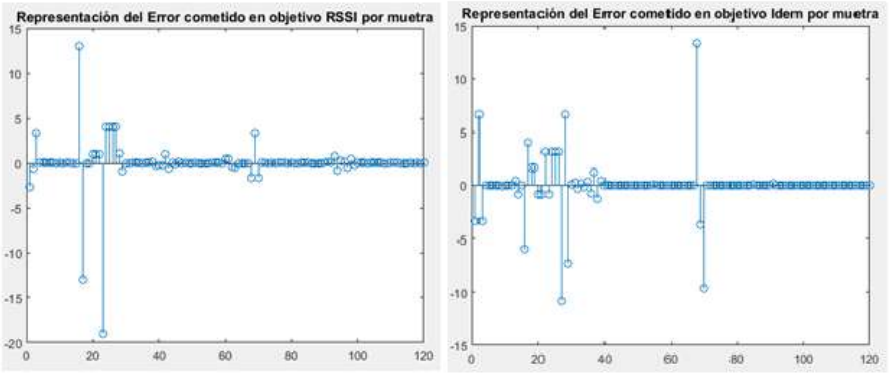

Figura 255: Representación del error en Redes Implementadas para Revenna Seattle

Se aprecia que el error producido para casi todas las muestras es prácticamente 0 en ambas redes, aunque en ciertas muestras suceden errores altos.

Se muestra ahora las salidas obtenidas del entrenamiento en color azul y las salidas reales en rojo.

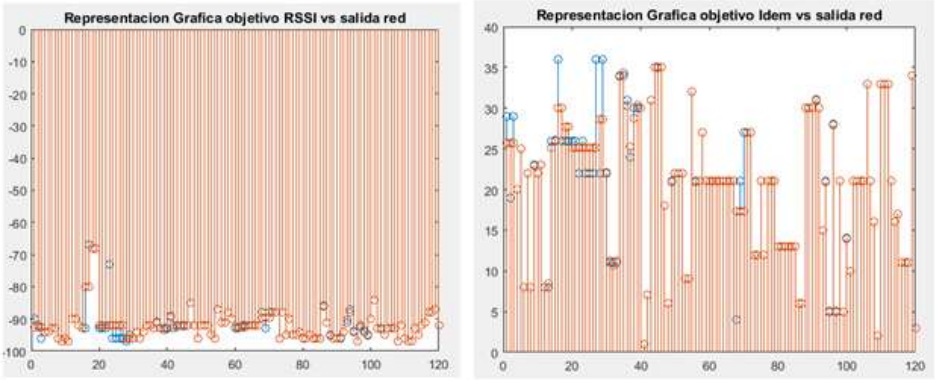

Figura 256: Representación de salidas en Redes Implementadas para Revenna Seattle

Por último, se muestra los resultados de error obtenidos por los entrenamientos empleados:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en RSSI para red con ubicaciones de Revenna en Seattle
    Resultado MSE(Promedio cuadrado del error): 6.73
    Resultado MEA(Promedio absoluto del error): 0.74
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Idem para red con ubicaciones de Revenna en Seattle
    Resultado MSE(Promedio cuadrado del error): 5.60
    Resultado MEA(Promedio absoluto del error): 0.84
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Se aprecia que para esta zona de Revenna existe mayor error tanto al intentar obtener el valor RSSI como el Ídem.

8.2.2 Simulación

Al tener ya creadas las redes, se procede a cargar estas redes y simularlas. La simulación se realiza dentro del programa general como se comentó anteriormente.

Simulación 1:

Se realiza una simulación con los siguientes datos de ubicación:

```
>> GENERAL
Ingrese el valor de Latitud en Grados: 47
Ingrese el valor de Latitud en Minutos: 39
Ingrese el valor de Latitud en Segundos: 49.9
Ingrese el valor de Longitud en Grados: 122
Ingrese el valor de Longitud en Minutos: 18
Ingrese el valor de Longitud en Segundos: 30.05
Ingrese 1 si la ubicacion esta en zona centro, 2 si la ubicacion esta en zona revenna o 3 si la ubicacion esta en zona Kirkland:
2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Arpoximacion de valor RSSI: -87.77
    Arpoximacion de valor Idem: 11.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=39, Lat_seg=49.9, Lon_grados=122, Lon_min=18, Lon_seg=30.1 con resultados RSSI= -88. e Ídem=11 al modificar la ubicación se obtiene los resultados de RSSI=-87.77 e Ídem=11 por lo que los resultados obtenidos son correctos.

Simulación 2:

Se realiza una simulación con los siguientes datos de ubicación:

```
>> GENERAL
Ingrese el valor de Latitud en Grados: 47
Ingrese el valor de Latitud en Minutos: 39
Ingrese el valor de Latitud en Segundos: 45.2
Ingrese el valor de Longitud en Grados: 122
Ingrese el valor de Longitud en Minutos: 18
Ingrese el valor de Longitud en Segundos: 22.8
Ingrese 1 si la ubicacion esta en zona centro, 2 si la ubicacion esta en zona revenna o 3 si la ubicacion esta en zona Kirkland:
2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Arpoximacion de valor RSSI: -91.80
    Arpoximacion de valor Idem: 22.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=39, Lat_seg=45.1, Lon_grados=122, Lon_min=18, Lon_seg=22.6 con resultados RSSI= -92 e Ídem=22 al modificar la ubicación se obtiene los resultados de RSSI=-91.8 e Ídem=22 por lo que los resultados obtenidos son correctos.

Simulación 3:

Se realiza una simulación con los siguientes datos de ubicación:

```
>> GENERAL
Ingrese el valor de Latitud en Grados: 47
Ingrese el valor de Latitud en Minutos: 39
Ingrese el valor de Latitud en Segundos: 47.4
Ingrese el valor de Longitud en Grados: 122
Ingrese el valor de Longitud en Minutos: 18
Ingrese el valor de Longitud en Segundos: 26.2
Ingrese 1 si la ubicacion esta en zona centro, 2 si la ubicacion esta en zona revenna o 3 si la ubicacion esta en zona Kirkland:
2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Arpoximacion de valor RSSI: -92.98
    Arpoximacion de valor Idem: 30.00
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=39, Lat_seg=47.4, Lon_grados=122, Lon_min=18, Lon_seg=25.9 con resultados RSSI= -93 e Ídem=30 al modificar la ubicación se obtiene los resultados de RSSI=-92.98 e Ídem=30 por lo que los resultados obtenidos son correctos.

8.3 Ubicación en Kirkland al este Seattle

En este apartado se crea una red, se entrena y se simula con parámetros de entrada pertenecientes a valores de ubicaciones de Kirkland en Seattle.

8.3.1 Entrenamiento

Se muestra a continuación las redes creadas y los resultados obtenidos al entrenarla para los objetivos de RSSI e Ídem:

El nombre de las redes es: netFinaltrainRSSIKirkland.mat para RSSI y netFinaltrainIdemKirkland.mat para Ídem.

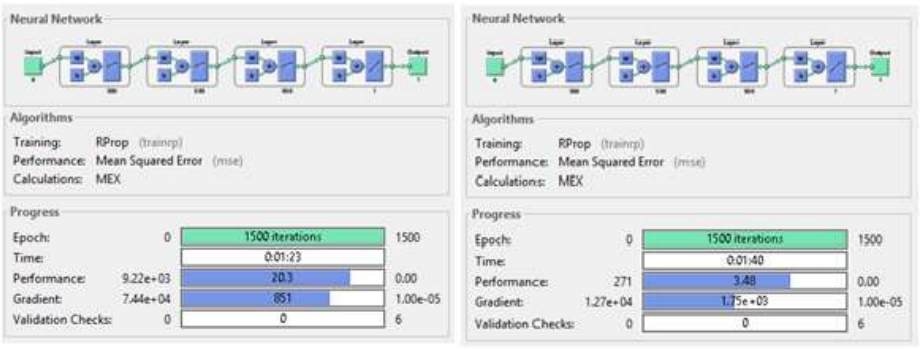

Figura 257: Redes implementadas para Kirkland en Seattle

Resultados por muestra del error producido los entrenamientos:

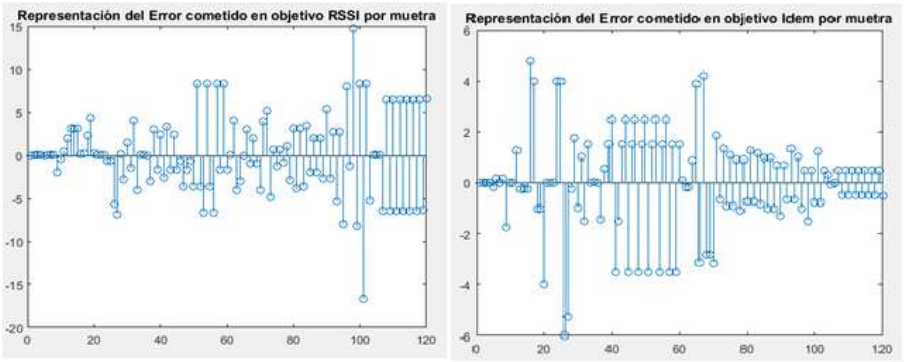

Figura 258: Representación del error en Redes Implementadas para Kirkland Seattle

Se aprecia que el error producido con este conjunto de datos es mayor.

Se muestra ahora las salidas obtenidas del entrenamiento en color azul y las salidas reales en rojo.

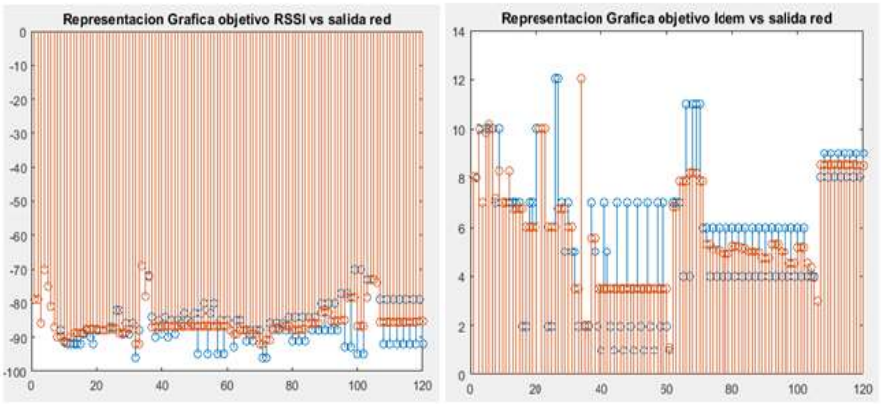

Figura 259: Representación de salidas en Redes Implementadas para Kirkland Seattle

Por último, se muestra los resultados de error obtenidos por los entrenamientos empleados:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en RSSI para red con ubicaciones de Kirkland en Seattle
    Resultado MSE(Promedio cuadrado del error): 20.32
    Resultado MEA(Promedio absoluto del error): 3.33
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    Resultados de Errores en Idem para red con ubicaciones de Kirkland en Seattle
    Resultado MSE(Promedio cuadrado del error): 3.48
    Resultado MEA(Promedio absoluto del error): 1.34
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Se aprecia que para este conjunto de datos se produce mayor error.

8.3.2 Simulación

Simulación 1:

Se realiza una simulación con los siguientes datos de ubicación:

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=39, Lat_seg=42.3, Lon_grados=122, Lon_min=18, Lon_seg=25.5 con resultados RSSI= -88 e Ídem=11 al modificar la ubicación se obtiene los resultados de RSSI=-90.13 e Ídem=8 por lo que los resultados obtenidos no son correctos.

Simulación 2:

Se realiza una simulación con los siguientes datos de ubicación:

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=39, Lat_seg=44, Lon_grados=122, Lon_min=18, Lon_seg=25.5 con resultados RSSI= -73e Ídem=4 al modificar la ubicación se obtiene los resultados de RSSI=-72.11 e Ídem=4 por lo que los resultados obtenidos son correctos.

Simulación 2:

Se realiza una simulación con los siguientes datos de ubicación:

Como se muestra en la imagen anterior, los resultados obtenidos son una aproximación de los resultados esperados. Es decir, para la ubicación entrenada de Lat_grados=47, Lat_min=39, Lat_seg=41,1, Lon_grados=122, Lon_min=18, Lon_seg=24 con resultados RSSI= -78 e Ídem=2 al modificar la ubicación se obtiene los resultados de RSSI=-71.55 e Ídem=1 por lo que los resultados obtenidos no son correctos.

9 CONCLUSIONES

A lo largo de todo el proyecto se han ido comentando los resultados y conclusiones obtenidos. En este capítulo se realiza un resumen de todas las conclusiones comentadas en capítulos anteriores y, también, se mencionan las conclusiones finales.

Haciendo referencia a los apartados 6.1, 6.2 y 6.3, en los que se realizan entrenamientos variando el nº de entradas, nº de salidas, los valores de entradas (coordenadas decimales o en grados) y los valores de salida (RSSI e Ídem), se llega a la conclusión de que, para realizar estimación de redes Wi-Fi, se produce menos promedio cuadrado de error (MSE) y menos error medio absoluto (MAE) utilizando los datos de entrada en coordenadas grados, minutos y segundos.

Al utilizar los datos de entrada en coordenadas grados, minutos y segundos se está utilizando un mayor número de entradas (serían 6) a diferencia de utilizar los datos de entrada en coordenadas decimales (serían 2). Teniendo más entradas a la red se pueden obtener más configuraciones internas.

Otro motivo, por el cual los resultados con entrada en coordenadas de grados, minutos y segundos son mejores, es la separación entre los valores de las muestras de entrada. Para esta entrada sí se puede apreciar la diferencia de las muestras a entrenar. Mientras que para la entrada en coordenadas decimales, los valores son muy parecidos y con variación en las muestras a partir de la quinta cifra significativa. Por lo que realizando entrenamientos con este tipo de datos de entrada, se produce un elevado nivel de promedio cuadrado de error y de error medio absoluto, conllevando a tener resultados de simulación incorrectos.

Otra conclusión a la que se llega es que, utilizando entrenamientos con una salida objetivo, se produce menor error que realizando éstos mismos entrenamientos con varias salidas objetivo. Dicho de otro modo, utilizando los mismos valores de entrada, variando la configuración de salida (teniendo 1 salida o 2) y el valor de salida (si es RSSI o Ídem) se obtiene valores menores de MSE e MAE y, por tanto, mejores resultados.

Al realizar el entrenamiento con la función traingd e ir modificando sus parámetros de configuración, se observa que no ocasiona ningún cambio en los errores evaluados la variación del nº de capas ocultas (en el apartado 7.1.1.4 se muestra comparativa) ni la variación del nº de neuronas (en el apartado 7.1.2.4 se muestra comparativa). Tampoco aporta ningún cambio la variación del nº de épocas (en el apartado 7.1.3.4 se muestra comparativa). Por lo que se determina que esta función, para la estimación de red WI-FI con los parámetros de entrada definidos, no es una opción aceptable.

Al utilizar el entrenamiento con la función trainrp e ir modificando los parámetros de configuración, se observa que, al aumentar hasta 3 el nº de capas ocultas el entrenamiento, se ajusta mejor. Esto se debe a que produce menor promedio cuadrado de error y menor promedio absoluto de error (se muestra en apartado 7.2.1.4). Siguiendo con las modificaciones, se aprecia que al variar el número de neuronas de las capas ocultas, el entrenamiento funciona mejor con un número elevado de neuronas (en torno a 500). Este análisis se muestra en el apartado 7.2.2.4. Como última modificación se realiza la variación del número de épocas de simulación y se aprecia que, a partir de un cierto numero de epocas de simulacion, los errores producidos son cercanos a 0 y lo unico que se incrementa es el tiempo de simulacion. Por lo que, en función de la cantidad de datos a entrenar, sería conveniente aumentar el número de épocas o no lo sería. Se concluye comentando que esta función sí aporta buenos resultados al realizar el entrenamiento.

Al realizar el entrenamiento con la función trainscg e ir variando sus parámetros de configuración, se observa que se ajusta mejor la utilización de 2 capas ocultas, ya que produce menor promedio cuadrado de error y menor promedio absoluto de error, como se muestra en el apartado 7.3.1.4. Siguiendo con las modificaciones realizadas, al variar el número de neuronas, se aprecia que los resultados son mejores para un número intermedio de neuronas. Es decir, se produce menos promedio cuadrado de error en entrenamientos con 300 neuronas. Cuando se entrena con un número de neuronas alto se corre el riesgo de sobreentrenar la red y que la red aprenda. Por último, se aprecia que con el aumento del número de épocas se produce un menor promedio cuadrado de error. Esta función también podría ser válida.

Como conclusión final, menciono las simulaciones de casos de uso en cada una de las zonas tratadas:

En el centro de Seattle, se aprecia que es dónde mejor resultados se obtienen a la hora de estimación de valores WI-FI. En el apartado 8.1.2 se muestran los resultados de las simulaciones para este caso.

Los resultados obtenidos para las simulaciones de estimación de valores WI-FI en la zona de Revenna son resultados correctos, aunque, como se aprecia en el apartado 8.2.1, existe un nivel de promedio cuadrado de error, por lo que en determinadas ubicaciones puede producirse un error considerable.

Para la estimación en ubicaciones de la zona de Kirkland no se obtienen resultados factibles. Como se aprecia en el apartado 8.3.1, aparece un alto nivel de promedio cuadrado de error, se piensa que la cercanía o la distancia entre las ubicaciones entrenadas tiene algo

que ver en la aparición de este error, aunque también cabe la posibilidad de que el promedio de error sea tan alto por la zona geográfica en la que se encuentran las ubicaciones.

10 BIBLIOGRAFÍA

[1] http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/peredo_a_s/capitulo1.pdf

[2] http://ocw.upm.es/teoria-de-la-senal-y-comunicaciones-1/radiocomunicacion/contenidos/presentaciones/propagacion-07.pdf

[3] http://www.tecnoWi-Fi.com/

[4] https://www.netspotapp.com/es/what-is-rssi-level.html

[5] https://www.lifewire.com/history-of-wireless-standard-802-11g-816556

[6] https://www.ecured.cu/Est%C3%A1ndar_inal%C3%A1mbrico_802.11b

[7] https://es.wikipedia.org/wiki/Comunicaci%C3%B3n_inal%C3%A1mbrica

[8] Anthony LaMarca, Yatin Chawathe, Jeffrey Hightower, Gaetano Borriello, CRAWDAD conjunto de datos intel / placelab (v. 2004-12-17),

[9] CRAWDAD conjunto de datos intel / placelab (v. 2004-12-17),

[10] Smith, K. y M. Palaniswani. Static and Dynamic Channel Assginment using Neural Networks. IEEE Journal on Selected Areas of Communications. vol 15. no 2. pp. 238-249. 1997.

[11] Hertz, J.; Krogh, A.; Palmer, R.G.; "Introduction to the theory of neural computation" Addison-Wesley Publishing Company. Capítulo 1, Apr 1995.

[12]Solin, Daniel. Qt Programming in 24 Hours. Sams. 435 pgs. USA. 2000.

[13]Freeman, J.; Skapura, D.; "Redes neuronales: algoritmos, aplicaciones y técnicas de programación" Addison-Wesley/ Diaz de Santos. Capítulo 4, 1993.

[14]Thuijsman, F.; Weijters, A.; "Artificial neural networks: an introduction to ANN theory and practice" Springer. Pág. 119 a 128, 1995.

[15]Garcia Martinez, Servente, & Pasquín, 2003