



Nombre del proyecto

Estimación del ángulo de fase de una señal RF de ultrasonido mediante una red de retropropagación.

Nombre:

Ing. Ricardo González Cruz

Institución:

Centro de Ingeniería y Desarrollo de Industrial

Programa:

Especialidad de tecnólogo en mecatrónica

Tutor:

Dr. Noé Amir Rodríguez Olivares

Cotutor:

Dr. Jorge Alberto Soto Cajiga

ÍNIDICE	
LISTA DE FIGURAS	2
LISTA DE TABLAS	3
Abstract.....	4
CAPÍTULO 1	5
1.2 Definición del proyecto	5
1.3 Justificación	6
1.4 Objetivos.....	6
1.4.1 Objetivo general.....	6
1.4.2 Objetivos específicos	6
1.5 Metodología propuesta	7
1.6 Alcance	7
1.7 Estado del arte.....	8
CAPITULO 2 MARCO TEORICO	9
2.1 Redes neuronales	9
2.2 La red de retropropagación	10
2.3 Estructura y Aprendizaje de la Red Retropropagación.....	11
2.4 Número de Capas Ocultas.....	11
2.5 El Perceptrón Multicapa	12
2.6 Arquitectura Típica	12
2.7 Pasos para aplicar el algoritmo de entrenamiento.....	13
2.8 Ultrasonido.....	15
2.9 Método pulso-eco	16
2.10 Características generales de las ondas ultrasónicas	17
2.11 Ondas longitudinales ultrasónicas (haz recto)	17
2.12 Propagación de las ondas	17
2.13 Equipo básico de ultrasonido	17
2.14 Técnicas de estimación de parámetros.....	18
CAPITULO 3 GENERACIÓN DE MATRIZ DE ENTRENAMIENTO	20
3.1 Análisis del modelo matemático del pulso ultrasónico.....	20
3.2 Generación de matriz de datos.....	22
3.3 Búsqueda del ínfimo “Y”, supremo y almacenamiento de la posición temporal	22
3.4 Eliminación de datos fuera de rango.....	23

3.5 Normalización de datos	25
3.6 Orden de los datos con base en su amplitud	26
3.7 Asignación de números en orden	27
3.8 Detección del pulso de máxima amplitud para realizar el normalizado	28
3.9 Generación de matriz de entrenamiento	29
CAPITULO 4 DISEÑO DE ARQUITECTURA DE LA RED NEURONAL.....	30
4.1 Asignación de entradas y salida de la red	30
4.2 Valores iniciales.....	30
4.3 Función	31
4.4 Capa oculta y capa de salida	32
4.5 Entrenamiento	32
4.6 Conexión de la capa oculta y de la capa de salida	32
4.7 Determinación del error, de la sensibilidad de la neurona y nuevos valores	32
4.8 Prueba de la neurona.....	33
CAPITULO 5 EXPERIMENTACION Y RESULTADOS	34
5.1 Entrenamiento de la red por época	34
5.2 Nuevos Valores.....	43
CAPITULO 6 RESULTADOS Y CONCLUSIONES	44
6.1 Resultado	44
6.2 Arquitectura final	45
6.3 Conclusiones del trabajo.....	46
6.4 Referencias y Bibliografía	46

LISTA DE FIGURAS

Figura 1.1 Metodología	7
Figura 2.1 Esquema de Red Neuronal	9
Figura 2.2 Neurona.	10
Figura 2.3 Capas de Neurona.	13
Figura 2.4 Reflexión de ultrasonido	16
Figura 2.5 esquema de ultrasonido	18
Figura 3.1 señal ultrasónica con ángulo de 11.25°	21
Figura 5.1 FA 0.001 VE 0.5	35
Figura 5.2 FA 0.01 VE 0.3	35
Figura 5.3 FA 0.01 VE 0.5	36
Figura 5.4 FA 0.1 VE 0.3	36

Figura 5.5 FA 0.1 VE 0.5	37
Figura 5.6 FA 0.005 VE 0.3	37
Figura 5.7 FA 0.005 VE 0.5	38
Figura 5.8 FA 0.05 VE 0.3	38
Figura 5.9 FA 0.06 VE 0.3	39
Figura 5.10 FA 0.06 VE 0.5	39
Figura 5.11 FA 0.06 VE 0.1	40
Figura 5.12 FA 0.06 VE 0.1	40
Figura 5.13 FA 0.06 VE 0.1	41
Figura 5.14 FA 0.06 VE 0.2	41
Figura 5.15 FA 0.08 VE 0.3	42
Figura 6.1 Resultado Señal ultrasonido.....	44
Figura 6.2 Arquitectura.....	45

LISTA DE TABLAS

Tabla 3.1 32 ángulos en grados y radianes	21
Tabla 3.2 puntos tomados en eje “X”, “Y” y signo.....	22
Tabla 3.3 Supremo e Ínfimo	23
Tabla 3.4 Datos de rango de 20dB	24
Tabla 3.5 Cambio de amplitud de los datos	25
Tabla 3.6 Asignación de número de acuerdo con el signo.....	26
Tabla 3.7 Matriz máx. y min. positivo.	27
Tabla 3.8 Matriz normalizada.....	28
Tabla 3.9 Matriz entrenamiento	29
Tabla 4.1 Y deseada.....	30
Tabla 4.2 Matriz ejemplo W1.....	30
Tabla 4.3 vector ejemplo W2	31
Tabla 4.4 vector ejemplo b1	31
Tabla 4.5 ejemplo b2	31
Tabla 5.1 Entrenamiento de la red.....	34
Tabla 5.2 nuevos pesos W1	43
Tabla 5.3 nuevos bias b1 Tabla 18 5.4 nuevos pesos W2	43
Tabla 5.5 nuevo bias b2.....	43
Tabla 5.6 Resultados “Y” deseada y “Y” estimada.....	44

Resumen

En este trabajo se desarrolla una red neuronal del tipo retropropagación, la cual es capaz de estimar el ángulo de fase de un pulso ultrasónico; para lograr esta estimación se hace un análisis del modelo matemático que describe a un pulso ultrasónico, además, se realiza un análisis sobre las técnicas de estimación de parámetros mediante la red de retropropagación. Posteriormente se mostrará el procedimiento para identificar la señal de ultrasonido, la selección de las variables utilizadas para el entrenamiento de la red, el desarrollo de la red de retropropagación y las pruebas realizadas a la red para interpretar los resultados.

Abstract

In this work we develop a backpropagation neural network type, which is able to estimate the phase angle of an ultrasonic pulse, to achieve this estimate we make an analysis of the mathematical model describing an ultrasonic pulse, also an analysis on the techniques of parameter estimation is performed by backpropagation network. This work also shows the procedure for identify the ultrasound signal, the selection of the variables used for training network, the development of the backpropagation network and testing to interpret the results.

CAPÍTULO 1

1.1 Introducción

Las redes neuronales artificiales se basan en las redes neuronales biológicas debido a las características similares entre estas, como son la capacidad de memorizar y de asociar hechos. Las personas son capaces de resolver situaciones acudiendo a la experiencia acumulada. Así, una buena forma de llevar a cabo la resolución de problemas consiste en la construcción de sistemas que sean capaces de reproducir esta característica humana. Una red neuronal es un sistema para el tratamiento de la información, cuya unidad básica de procesamiento está inspirada en la célula fundamental del sistema nervioso humano: la neurona.

Las redes neuronales son modelos que intentan reproducir el comportamiento del cerebro humano. Una red neuronal consiste en un conjunto de elementos de procesamiento, llamados neuronas, los cuales se conectan entre sí. La organización y disposición de las neuronas dentro de una red neuronal se denomina topología, y viene dada por el número de capas, la cantidad de neuronas por capa, el grado de conectividad, y el tipo de conexión entre neuronas.

Una vez determinada la topología de la red neuronal es necesario entrenarla. En la etapa de entrenamiento, la red es capaz de aprender relaciones complejas entre entradas y salidas, mediante el ajuste en los pesos de las conexiones entre neuronas.

1.2 Definición del proyecto

Una señal ultrasónica puede ser representada como una superposición de ecos Gaussianos. En los ensayos no destructivos mediante ultrasonido por la técnica de pulso-eco, el eco recibido por un reflector simple puede ser modelado como:

$$(1) \quad r(t) = \beta e^{-\alpha(t-\tau)^2} \cos(2\pi f_c(t - \tau) + \phi)$$

Los parámetros del modelo están directamente relacionados con las propiedades de la señal ultrasónica propagándose a través del material. Las variables son: el tiempo de arribo τ , la amplitud del eco β , la frecuencia central del pulso denominada como f_c , el ancho de banda establecido como α y el ángulo de fase del pulso ultrasónico establecido como ϕ .

Si se pretende expresar a la señal ultrasónica con base en los parámetros establecidos en el modelo establecido por Cardoso et al. [1], es necesario determinar todas las variables de la ecuación, entre ellas el ángulo del pulso ultrasónico, la determinación automática del ángulo de fase del pulso y de las demás variables. Esto permitiría que, en una aplicación industrial automatizada, pueda ser posible simplificar la información al tener la cualidad de poder estimar y adquirir solo los datos necesarios para reconstruir las señales.

1.3 Justificación

La estimación del ángulo en un pulso ultrasónico permitirá desarrollar técnicas de compresión que estén basadas en la estimación de los parámetros, además, al utilizar una red de retropropagación permitirá generar el entrenamiento fuera de línea, y una vez estimados los pesos y las bias, esta red podrá estimar el ángulo de un pulso ultrasónico de manera casi inmediata.

1.4 Objetivos

1.4.1 Objetivo general

- Desarrollar una red de retropropagación que sea capaz de estimar el ángulo de fase de una señal ultrasónica.

1.4.2 Objetivos específicos

- Analizar el modelo matemático del pulso ultrasónico.
- Estudiar las técnicas de estimación del ángulo de las señales RF de ultrasonido.
- Analizar el estado de arte sobre técnicas de estimación de parámetros mediante técnica de retropropagación.
- Desarrollar el procedimiento para identificar la señal ultrasónica.
- Seleccionar variables para el entrenamiento de la red.
- Proponer arquitectura de la red de retropropagación.
- Generar el espacio muestra para la red neuronal (época).
- Realizar el entrenamiento y las pruebas de la Red Neuronal.

1.5 Metodología propuesta

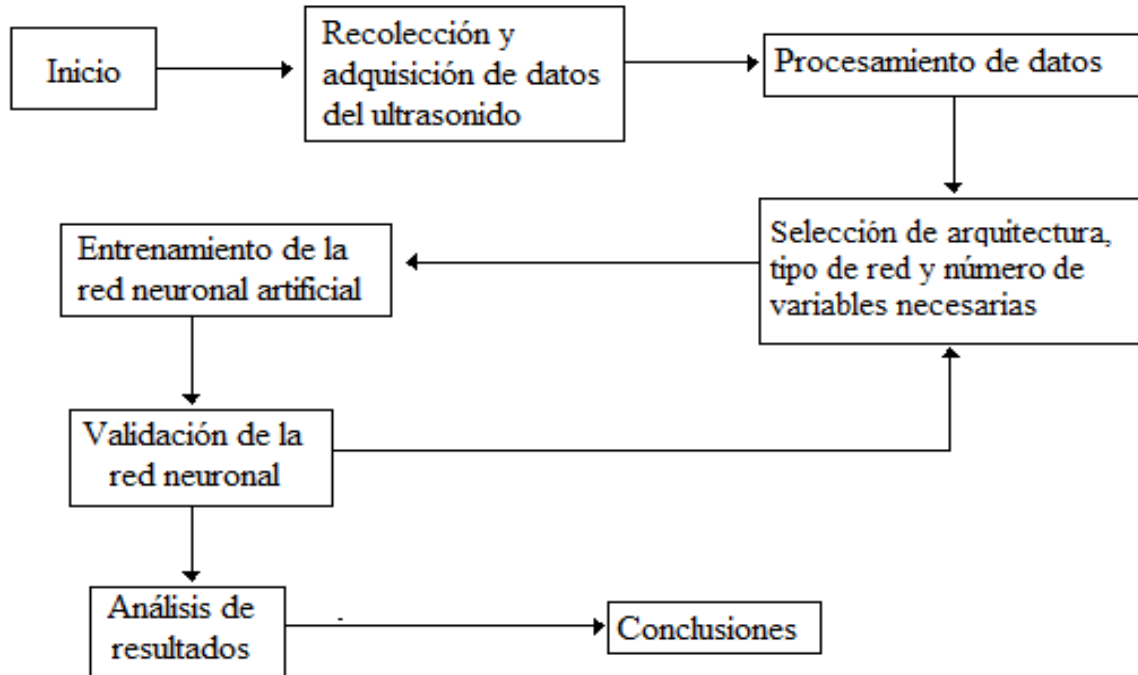


Figura 1.1 Metodología

1.6 Alcance

Determinar la arquitectura, los pesos y los bias de una red de retropropagación utilizada para la estimación del ángulo de fase de una señal de radiofrecuencia de ultrasonido.

1.7 Estado del arte

Las redes neuronales han sido utilizadas para resolver distintos problemas o facilitar el trabajo de análisis, se sabe que han sido utilizadas en reconocimientos de patrones como el reconocimiento de voz, reconocimiento de huella digital o de imágenes, se han utilizado en la identificación de origen de un vino, reconocimiento de código postal, reconocimiento de enfermedades al introducir los síntomas que afectan al paciente, predicen futuros cambios en el ambiente al introducir variables ambientales y climatológicas de una zona, etc..

El almacenamiento de las señales de ultrasonido a menudo requiere una gran capacidad de memoria, por lo que se busca reducir la información de la señal ultrasónica utilizando métodos de compresión, es deseable usar técnicas de compresión para reducir los datos y facilitar el análisis y el acceso de la información del ultrasonido. Cardoso et al. [1] establece un modelo para el pulso ultrasónico con el cual se facilita esta tarea, y por consecuencia, los datos que se tomarán para la red serán más exactos.

En el libro Neural Network Design se puede ver un ejemplo, caso 3 reconocimiento de patrones [2], se presenta un caso de estudio de reconocimiento de patrones utilizando redes neuronales para reconocer enfermedades del corazón por medio de la lectura de un electrocardiograma. Al observar este ejemplo y aplicando el modelo de la señal ultrasónica, se realiza una recopilación de datos y extracción de características tomando de estas 11 características con 32 ángulos distintos. Posteriormente se selecciona la arquitectura que se empleará en el desarrollo de la red neuronal, misma que una vez realizada será sometida a entrenamiento para lograr la correcta identificación de los ángulos deseados, donde por último se realizará la validación de los datos obtenidos.

CAPITULO 2 MARCO TEORICO

2.1 Redes neuronales

Las redes neuronales artificiales (RNA) son modelos matemáticos que intentan reproducir el funcionamiento del sistema nervioso. Como todo modelo, realizan una simplificación del sistema real que simulan y toman las características principales del mismo para la resolución de una tarea determinada.

Generalmente se pueden encontrar tres tipos de neuronas: Aquellas que reciben estímulos externos relacionados con el aparato sensorial, que tomarán la información de entrada. Dicha información es transmitida a las capas internas que se ocupan de su procesamiento. Es en las sinapsis y neuronas correspondientes a este segundo nivel donde se genera cualquier tipo de representación interna de información. Como no tienen relación directa con la información de entrada ni con la salida, estos elementos se denominan unidades ocultas. Una vez finalizado el período de procesamiento, la información llega a las unidades de salida, cuya misión es dar la respuesta al sistema. La red neuronal es entrenada para relacionar valores de entrada específicos con valores esperados de salida. En la red, los pesos (W) representan el conocimiento. Cada neurona posee un término bias (b), que constituye la predisposición de la neurona a activarse

En la figura 2.1 se puede ver el esquema de una red neuronal

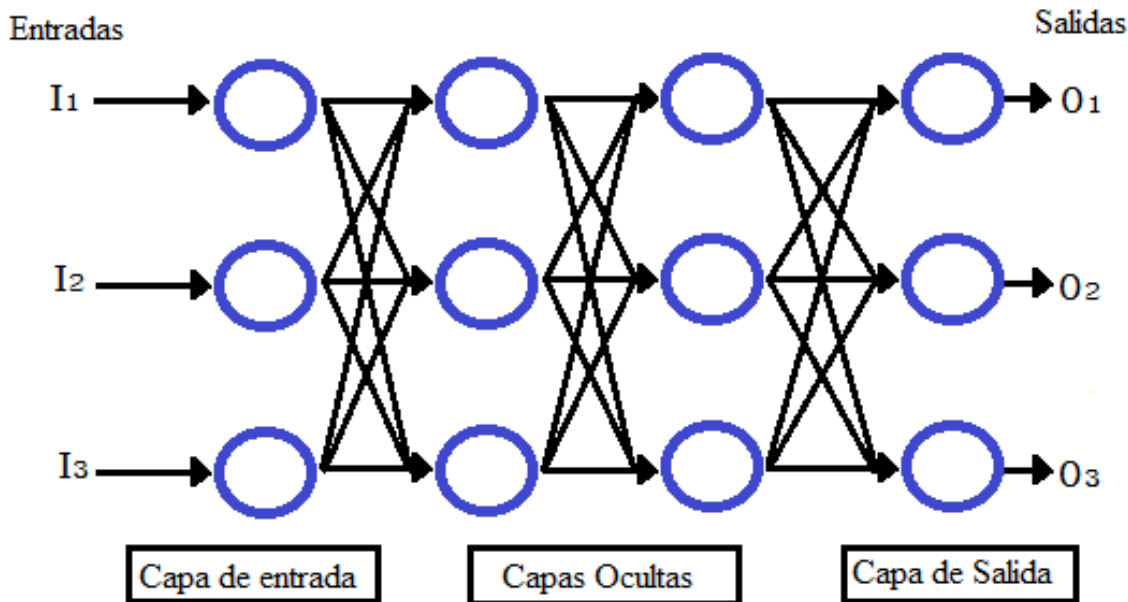


Figura 2.1 Esquema de Red Neuronal

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

La neurona artificial pretende imitar las características más importantes de las neuronas biológicas. Cada neurona i -ésima está caracterizada en cualquier instante por un valor numérico denominado valor o estado de activación $a_i(t)$; asociado a cada unidad, existe una función de salida, f_i , que transforma el estado actual de activación en una señal de salida. Dicha señal es enviada a través de los canales de comunicación unidireccionales a otras unidades de la red; en estos canales la señal se modifica de acuerdo con la sinapsis (el peso, w_{ij}) asociada a cada uno de ellos según determinada regla. Las señales moduladas que han llegado a la unidad j -ésima se combinan entre ellas, generando así la entrada total de la red como se muestra en la figura 2.2.

$$(2) \quad Y_i = \sum W_{ij} X_j$$

Una función de activación, F , determina el nuevo estado de activación $a_j(t+1)$ de la neurona, teniendo en cuenta la entrada total calculada y el anterior estado de activación $a_j(t)$.

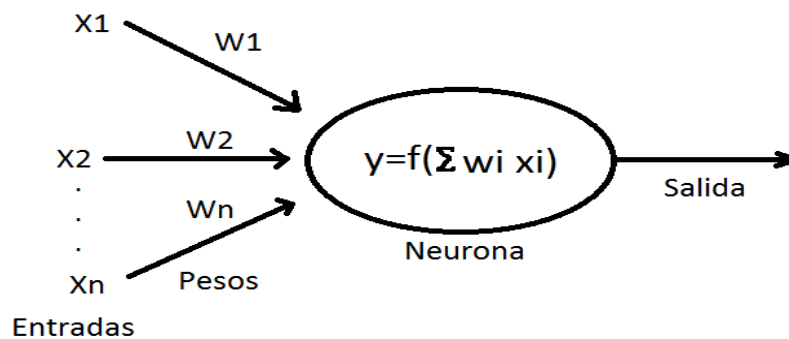


Figura 2.2 Neurona.

2.2 La red de retropropagación

El método retropropagación (propagación del error hacia atrás), basado en la generalización de la regla delta, a pesar de sus limitaciones, ha ampliado de forma considerable el rango de aplicaciones de las redes neuronales. El funcionamiento de la red retropropagación consiste en el aprendizaje de un conjunto predefinido de pares de entradas-salidas dados como ejemplo: primero se aplica un patrón de entrada como estímulo para la primera capa de las neuronas de la red, se propaga a través de todas las capas superiores hasta generar una salida, se compara el resultado en las neuronas de salida con la salida que se desea obtener y se calcula un valor de error para cada neurona de salida. A continuación, estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aportación

relativa al error total. Con base en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de tal manera que la siguiente ocasión que se presente el mismo patrón, la salida sea más cercana al valor deseado.

La importancia de la red de retropropagación consiste en su capacidad de auto adaptar los pesos de las neuronas de las capas intermedias, para aprender la relación que existe entre un conjunto de patrones de entrada y sus salidas correspondientes. Es importante la capacidad de generalización y facilidad de otorgar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento. La red debe ser capaz de encontrar una representación interna que le permita generar las salidas deseadas cuando se le dan entradas de entrenamiento.

2.3 Estructura y Aprendizaje de la Red Retropropagación

En una red retropropagación existe una capa de entrada con “n” neuronas y una capa de salida con “m” neuronas y al menos una capa oculta de neuronas internas. Cada neurona de una capa (excepto las de entrada) recibe entradas de todas las neuronas de la capa anterior y envía su salida a todas las neuronas de la capa posterior (excepto las de salida). No existe conexiones hacia atrás (feedback) ni laterales entre las neuronas de la misma capa.

La aplicación del algoritmo tiene dos fases, una hacia delante y otra hacia atrás. Durante la primera fase el patrón de entrada es presentado a la red y propagado a través de las capas hasta llegar a la capa de salida. Obtenidos los valores de salida de la red, se inicia la segunda fase, comparándose estos valores con la salida esperada para así obtener el error. Se ajustan los pesos de la última capa proporcionalmente al error. Se pasa a la capa anterior con una retropropagación del error, ajustando los pesos y continuando con este proceso hasta llegar a la primera capa. De esta manera es posible modificar los pesos de las conexiones de la red para cada patrón de aprendizaje del problema, del cual se conocía el valor de entrada y la salida deseada de la red ante dicho patrón.

La técnica retropropagación requiere el uso de neuronas cuya función de activación sea continua, por lo tanto, diferenciable. Generalmente, la función utilizada será del tipo sigmoideal esto para poder tener valores de entre -1 a 1.

2.4 Número de Capas Ocultas

En el diseño de las redes neuronales no existen reglas concretas para determinar el número de neuronas o número de capas de una red para resolver un problema concreto. Respecto al número de capas de la red, en general tres capas son suficientes (entrada –oculta-salida). Sin embargo, hay veces que un problema es fácil de resolver con más de una capa oculta. El tamaño de las capas, tanto de entrada como de salida, suele estar determinado por la naturaleza de la aplicación. En cambio, decidir cuántas neuronas debe tener una capa oculta no suele ser tan evidente. El número de neuronas ocultas interviene en la eficiencia de

aprendizaje y de generalización de la red. No hay ninguna regla que indique el número óptimo, para cada problema a resolver es necesario experimentar y observar el comportamiento de la red neuronal.

2.5 El Perceptrón Multicapa

Entre los diversos modelos que existen de RNA el más utilizado para aproximarse a las de funciones resulta el perceptrón multicapa asociado al algoritmo de aprendizaje backpropagation error (propagación del error hacia atrás), también denominado método de gradiente decreciente.

Este tipo de red exhibe la capacidad de “aprender” la relación entre un conjunto de entradas y salidas, a través del ajuste de los pesos de manera iterativa en la etapa de aprendizaje, y posteriormente aplica esta relación a nuevos vectores de entrada que el sistema no ha visto nunca en su etapa de entrenamiento, dando una salida activa si la nueva entrada es parecida a las presentadas durante el aprendizaje. Esta capacidad de generalización convierte a las redes perceptrón multicapa en herramientas de propósito general, flexibles y no lineales.

2.6 Arquitectura Típica

Un perceptrón multicapa está compuesto por una capa de entrada, una de salida y una o más capas ocultas; en la figura 2.3 podemos observar una red de este tipo. Para identificar la estructura de una red multicapa se utiliza la notación $r: s^1: s^2$ donde el número de entradas (r) va seguido por el número de neuronas en cada capa (s) y el exponente indica la capa a la cual la neurona corresponde.

En esta disposición, los elementos del vector de entrada “X” están conectados solo hacia delante sin que exista retroalimentación por lo que comúnmente se les clasifica en el grupo de redes con arquitectura feedforward. Las entradas se vinculan a través de la matriz de pesos (W) a cada una de las neuronas, donde se transforman en una suma ponderada con valores limitados por los bias (b) o umbrales de la neurona, el cual puede verse como un número que indica a partir de qué valor de potencial post-sináptico la neurona produce una salida significativa, este término es añadido a la suma ponderada para formar el vector modificado $[n]$ de S elementos (uno por cada neurona) de la red, los que finalmente son afectados por la función de activación para formar el vector columna de las salidas $[Y]$; si existe más de una capa estas salidas se pueden convertir en entradas de la próxima sobre las que se repite nuevamente el proceso.

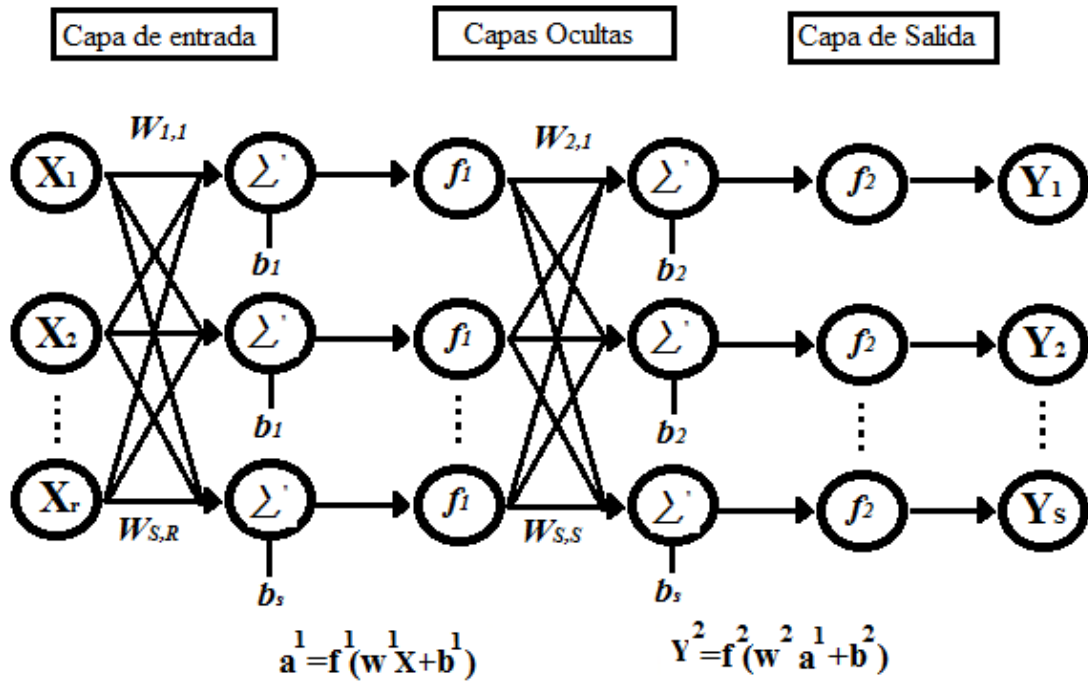


Figura 2.3 Capas de Neurona.

Figura 2.3 Representación de una RNA de una capa con “s” neuronas alimentadas por un vector de entrada “X” conformado por “r” elementos.

2.7 Pasos para aplicar el algoritmo de entrenamiento

Paso 1. Inicializar los pesos de la red con valores pequeños aleatorios.

Paso 2. Presentar un patrón de entrada y especificar la salida deseada que debe generar la red.

Paso 3. Calcular la salida actual de la red. Para ello se presentan las entradas a la red y se calcula la salida de cada capa hasta llegar a la capa de salida, ésta será la salida de la red.

Los pasos siguientes son calcular las entradas para las neuronas ocultas procedentes de las neuronas de entrada. Para una neurona j oculta:

$$(3) \quad net_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + b_j^h$$

En donde el índice “h” se refiere a magnitudes de la capa oculta; el subíndice “p”, al p-ésimo vector de entrenamiento, y “j” a la j-ésima neurona oculta. El bias “b” puede ser opcional, pues actúa como una entrada más.

Se calculan las salidas de las neuronas ocultas:

$$(4) \quad y_{pj} = f_j^h(\text{net}_{pj}^h).$$

Se realizan los mismos cálculos para obtener las salidas de las neuronas de salida:

$$(5) \quad \text{net}_{pk}^o = \sum_{j=1}^L w_{kj}^o y_{pj} + b_k^o$$

$$(6) \quad y_{pk} = f_k^o(\text{net}_{pk}^o)$$

Paso 4. Calcular los términos de error para todas las neuronas.

Si la neurona k es una neurona de la capa de salida, el valor de la delta es:

$$(7) \quad \delta_{pk}^o = (d_{pk} - y_{pk}) f_k^o(\text{net}_{pk}^o)$$

La función f debe ser derivable. En general disponemos de dos formas de función de salida:

La función lineal.

$$(8) \quad f_k(\text{net}_{jk}) = \text{net}_{jk}$$

La función sigmoideal.

$$(9) \quad f_k(\text{net}_{jk}) = \frac{1}{1 + e^{-\text{net}_{jk}}}$$

La selección de la función depende de la forma en que se desee representar la salida: si se desea que las neuronas de salida sean binarias, se utiliza la función sigmoideal, de otra forma, la lineal.

Para una función lineal, tenemos:

$$(10) \quad f_k^{o'} = 1,$$

Mientras que la derivada de una función sigmoideal es:

$$(11) \quad f_k^{o'} = f_k^o(1 - f_k^o) = y_{pk}(1 - y_{pk})$$

Por lo que los términos de error para las neuronas de salida quedan:

Para la salida lineal.

$$(12) \quad \delta_{pk}^o = (d_{pk} - y_{pk})$$

Para la salida sigmoïdal.

$$(13) \quad \delta_{pk}^o = (d_{pk} - y_{pk})y_{pk}(1 - y_{pk})$$

Si la neurona “j” no es de salida, entonces la derivada parcial del error no puede ser evaluada directamente, por lo tanto, se obtiene el desarrollo a partir de valores que son conocidos y otros que pueden ser evaluados.

La expresi3n obtenida en este caso es: $\delta_{pj}^h = f_j^{h'}(net_{pj}^h) \sum_k \delta_{pk}^o w_{kj}^o$ donde se observa que el error en las capas ocultas depende de todos los t3rminos de error de la capa de salida. De aqu3 surge el t3rmino propagaci3n hacia atr3s.

Paso 5. Actualizaci3n de los pesos: para ello se utiliza un algoritmo recursivo, comenzando por las neuronas de salida y procesando la informaci3n hacia atr3s hasta llegar a la capa de entrada, ajustando los pesos de la siguiente forma:

Para los pesos de las neuronas de la capa de salida:

$$(14) \quad w_{kj}^o(t + 1) = w_{kj}^o(t) + \Delta w_{kj}^o(t + 1)$$

$$(15) \quad \Delta w_{kj}^o(t + 1) = \alpha \delta_{pk}^o y_{pj}$$

Para los pesos de las neuronas de la capa oculta:

$$(16) \quad w_{ji}^h(t + 1) = w_{ji}^h(t) + \Delta w_{ji}^h(t + 1)$$

$$(17) \quad \Delta w_{ji}^h(t + 1) = \alpha \delta_{pj}^h x_{pi}$$

En ambos casos, para acelerar el proceso de aprendizaje se puede a3adir un t3rmino momento.

Paso 6. El proceso se repite hasta que el t3rmino de error $E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2$ resulta aceptablemente peque3o para cada uno de los patrones aprendidos.

2.8 Ultrasonido

La inspecci3n por ultrasonido es una prueba no destructiva, o un conjunto de ondas de alta frecuencia son introducidas en los materiales para la detecci3n de fallas en la superficie y sub-superficie.

Las ondas ultras3nicas viajan a trav3s del material perdiendo energ3a o potencia a trav3s del recorrido y son reflejadas a un palpador. El haz reflejado es mostrado y analizado para definir la presencia y localizaci3n de fallas y discontinuidades.

El grado de reflexi3n depende del estado f3sico de los materiales que forman la interfaz.

La mayoría de los instrumentos de inspección ultrasónica detectan fallas monitoreando uno o más de los siguientes puntos:

- La reflexión del sonido consistente en los límites del material o en discontinuidades dentro del material mismo.
- El tiempo de tránsito de la onda de sonido durante la prueba dentro de la pieza desde el punto de entrada del transductor hasta el punto de salida.
- La atenuación de las ondas de sonido en la pieza debido a la absorción y dispersión dentro de la pieza.

La mayoría de las inspecciones ultrasónicas son realizadas en frecuencias entre 0.1 y 25 MHz. Las ondas de ultrasonido son vibraciones mecánicas, las amplitudes de las vibraciones producen esfuerzos en las piezas por debajo de su límite elástico, de esta manera los materiales no producirán deformaciones plásticas.

2.9 Método pulso-eco

Es el más utilizado en inspecciones ultrasónicas, involucra la detección de ecos producidos cuando un pulso ultrasónico es reflejado por una discontinuidad en una pieza de trabajo. Este método es utilizado para detectar la localización de la falla y para medir espesores. La profundidad de la falla está determinada por el "tiempo de vuelo" entre el pulso inicial y el eco producido por la falla. La profundidad de la falla también se puede determinar por el tiempo relativo de tránsito entre el eco producido por una falla y el eco de la superficie de respaldo.

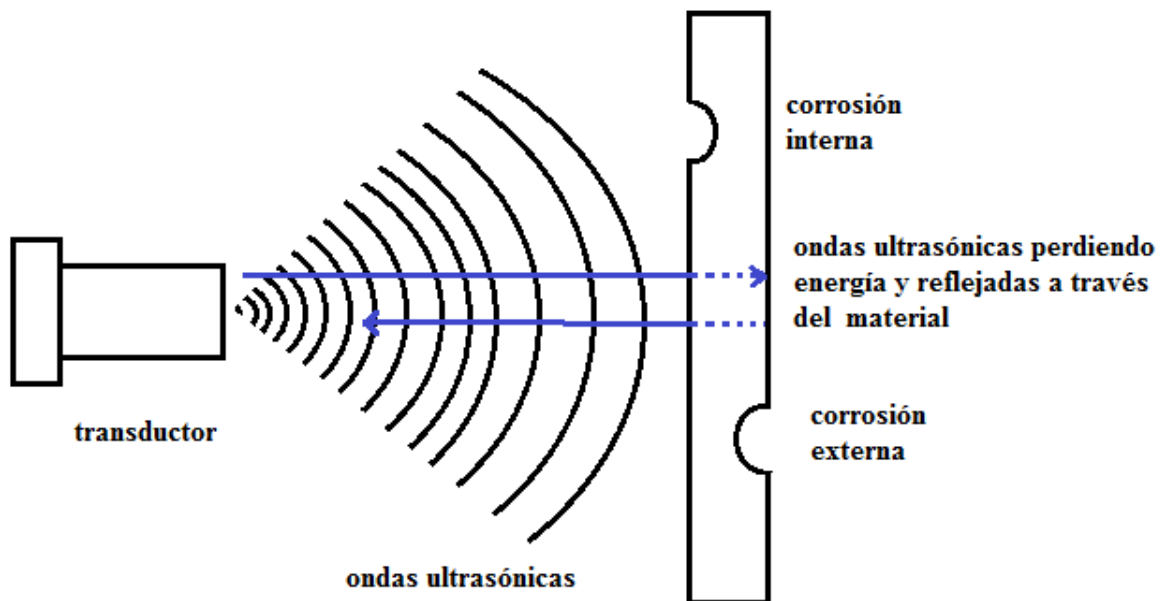


Figura 2.4 Reflexión de ultrasonido

2.10 Características generales de las ondas ultrasónicas

Las ondas ultrasónicas son ondas mecánicas que consisten en vibraciones oscilatorias de partículas atómicas o moleculares de una sustancia. Las ondas de ultrasonido se comportan igual que las ondas de sonido audible. Se pueden propagar a través de un medio elástico, ya sea sólido, líquido o gaseoso, pero no al vacío. En varios aspectos, un haz de ultrasonido es similar a un haz de luz, ambos son ondas y obedecen a la ecuación general de ondas. Cada onda viaja con características diferentes las cuales dependen del medio en el que se propaguen no de las características de la onda. Como un haz de luz, un haz de ultrasonido es reflejado de las superficies, refractado cuando cruza las fronteras entre dos sustancias que tienen diferentes características de velocidades y difractados en los bordes o alrededor de los obstáculos.

2.11 Ondas longitudinales ultrasónicas (haz recto)

Mayormente utilizadas en la inspección de materiales. Estas ondas viajan a través de los materiales como series alternadas de compresión y succión en las cuales las partículas transmiten las vibraciones de regreso y la dirección de viaje de las ondas. Las ondas longitudinales ultrasónicas se propagan rápidamente en líquidos y gases así también como en sólidos elásticos. La velocidad de una onda longitudinal ultrasónica es de 6000 m/s en aceros, 1500 m/s en agua y 330 m/s en aire.

2.12 Propagación de las ondas

Las ondas ultrasónicas se propagan en cierta medida en cualquier material elástico. Cuando las partículas atómicas o moleculares de un material elástico son desplazadas de sus posiciones de equilibrio por cualquier fuerza aplicada, esfuerzos internos actúan para restaurar o reacomodar las partículas a su posición original.

La relación de velocidad con frecuencia y longitud de onda está dada por:

$$(18) \quad V = f \cdot \lambda$$

Dónde V es velocidad (en metros por segundo), f es la frecuencia (en Hertz) y λ es la longitud de onda (en metros por ciclo).

2.13 Equipo básico de ultrasonido

La mayoría de los equipos de inspección por ultrasonido incluyen el siguiente equipo básico:

- Un generador electrónico de señal que produce ráfagas de voltaje alternadas.
- Un transductor que emite un haz de ondas ultrasónicas cuando las ráfagas de voltaje alternado son aplicadas.
- Un acoplador que transfiere la salida de las ondas de sonido (energía acústica) de la pieza al transductor.
- Un transductor (puede ser el mismo que el transductor que inicia las ondas ultrasónicas o puede ser otro diferente) para percibir las ondas ultrasónicas reflejadas

por el material inspeccionado y convertir dicha onda en valores analógicos de voltaje. En la mayoría de los sistemas un transductor simple actúa como emisor y receptor.

- Un dispositivo electrónico para amplificar y modificar las señales del transductor.
- Un dispositivo de salida que muestre la información resultante y la proyecte ya sea impresa o en pantalla.
- Un reloj electrónico o un cronómetro para controlar la operación de varios componentes del sistema.

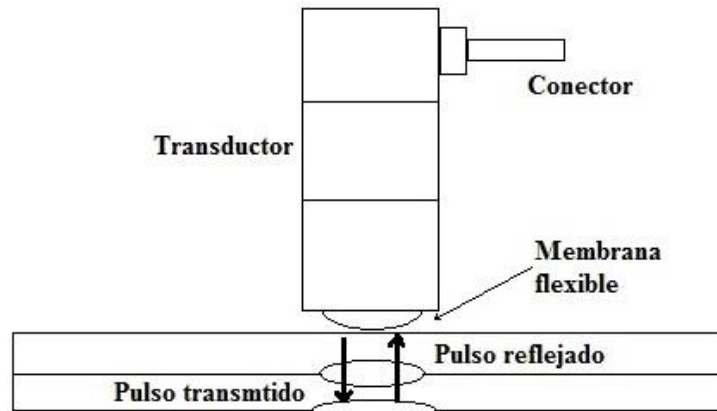


Figura 2.5 esquema de ultrasonido

2.14 Técnicas de estimación de parámetros

La red neuronal depende de los datos que son usados para entrenarla, por ello se requiere que los datos entren correctamente en la red para que puedan ser analizados, para ello se utilizan técnicas de estimación.

El principal propósito de la estimación de parámetros es facilitar el entrenamiento de la red. El procesamiento de datos consiste en pasos como normalización y extracción de características. La idea es realizar el procesamiento preliminar de los datos para que a la red le sea más sencillo extraer la información de interés.

Es una práctica estándar normalizar las entradas antes de introducirlos en la red. De esta manera, la inicialización de los pesos de la red con valores aleatorios pequeños garantiza que el producto de entradas con peso serán valores pequeños. También cuando los valores son normalizados, las magnitudes de los pesos tienen un significado coherente.

Existen métodos básicos de normalización.

El primer método normaliza los datos de manera que sus rangos queden entre valores de -1 a 1 esto se resuelve con la siguiente formula.

$$(19) \quad p^n = \frac{2(p-p^{min})}{(p^{max}-p^{min})-1}$$

Donde:

P^{min} Es el vector con el mínimo valor de las entradas

P^{max} Es el vector con el máximo valor de las entradas

P^n Es el resultado de normalizar el vector de entradas

Otro procedimiento de normalización es el de ajustar los datos para poder tener una media y varianza específica para obtener resultados de 0 a 1

$$(20) \quad P^n = \frac{(P - P^{media})}{p^{std}}$$

P^{media} Es el promedio de las entradas de vector

P^{std} Es la desviación estándar de cada elemento de entrada de vector

P^n Es el resultado de normalizar el vector de entradas

Generalmente la normalización es aplicada en las entradas del vector y los objetivos del vector. En la red usamos la siguiente formula:

$$P^n = \frac{(P - 127)}{p^{max}}$$

CAPITULO 3 GENERACIÓN DE MATRIZ DE ENTRENAMIENTO

En este capítulo se explicará el modelo matemático del pulso ultrasónico y como se obtienen los datos para generar una matriz la cual se usará para la red neuronal. Se seleccionarán datos del pulso ultrasónico que identifiquen los distintos ángulos y que sean características únicas. Se selecciona el valor más alto o valor supremo y el valor mínimo o valor ínfimo junto con su posición temporal, también por ser un modelo de un ultrasonido se excluirán los valores fuera de un rango de 20 dB. Posteriormente se realiza la normalización de los datos para que los valores se encuentren en un rango entre cero y uno, se ordenarán los datos según su signo y sus valores máximos positivos o mínimos negativos y de esa forma poder generar la matriz de entrenamiento.

3.1 Análisis del modelo matemático del pulso ultrasónico.

Para poder realizar la red neuronal que identifique patrones en el pulso ultrasónico es necesario entender primero cuales son los parámetros que tomarán del pulso con distintos ángulos, el presente proyecto comienza con la simulación del ultrasonido por medio del programa Matlab.

La siguiente ecuación describe el modelo, para expresar la señal ultrasónica usaremos los parámetros establecidos por Cardoso et al [1]

Modelo matemático de una señal de ultrasonido o de un pulso ultrasónico

$$(1) \quad r(t) = \beta e^{-\alpha(t-\tau)^2} \cos(2\pi f_c(t - \tau) + \phi)$$

Para analizar el modelo matemático del pulso ultrasonido se usan los siguientes parámetros:

Alfa (α): Ancho de banda de la señal de US.

Tau(τ): Retardo de la señal en el tiempo.

f_c : Frecuencia central del transductor.

beta(β): Amplitud de la señal de US.

ϕ : Ángulo de la señal.

$f_{muestreo}$: Frecuencia de muestreo del ADC (conversión analógica digital).

$t_{muestreado}$: Tiempo que se hace el muestreo del ADC.

$t = [0:1/f_{muestreo}: t_{muestreado}]$: Vector de tiempo para el primer eco.

Las señales de ultrasonido son señales continuas, por lo tanto, se debe discretizar dichas señales, realizando un muestreo a través del tiempo, obteniendo así diferentes patrones de la señal que dependen de la forma del eco ultrasónico. En la simulación de la señal de ultrasonido es necesario generar diferentes señales (32 señales) con distintos ángulos, y así poder tener una "Y" deseada la cual representa diferentes señales de ultrasonido dependiendo del ángulo. A continuación, se crea una matriz de 32x151 que es nombrada como Datos_US de la cual se pueden obtener los datos necesarios. Estas 32 señales equivalen a los 32 ángulos que se muestran en la tabla 3.1 y se obtienen al dividir 2π entre 32 que es igual a 0.19625 radianes los cuales equivalen a 11.25° de esta forma al usar Matlab se crean señales con los

32 ángulos y podemos generar la simulación del ultrasonido para después obtener datos de las señales simuladas.

Tabla 3.1 32 ángulos en grados y radianes

Ángulo	Grados	Radianes	Ángulo	Grados	Radianes
1	11.25°	0.19634954	17	191.25°	3.33794219
2	22.5°	0.39269908	18	202.5°	3.53429174
3	33.75°	0.58904862	19	213.75°	3.73064128
4	45°	0.78539816	20	225°	3.92699082
5	56.25°	0.9817477	21	236.25°	4.12334036
6	67.5°	1.17809725	22	247.5°	4.3196899
7	78.75°	1.37444679	23	258.75°	4.51603944
8	90°	1.57079633	24	270°	4.71238898
9	101.25°	1.76714587	25	281.25°	4.90873852
10	112.5°	1.96349541	26	292.5°	5.10508806
11	123.75°	2.15984495	27	303.75°	5.3014376
12	135°	2.35619449	28	315°	5.49778714
13	146.25°	2.55254403	29	326.25°	5.69413668
14	157.5°	2.74889357	30	337.5°	5.89048623
15	168.75°	2.94524311	31	348.75°	6.08683577
16	180°	3.14159265	32	360°	6.28318531

Gráfico de ultrasonido con un ángulo de 11.25°

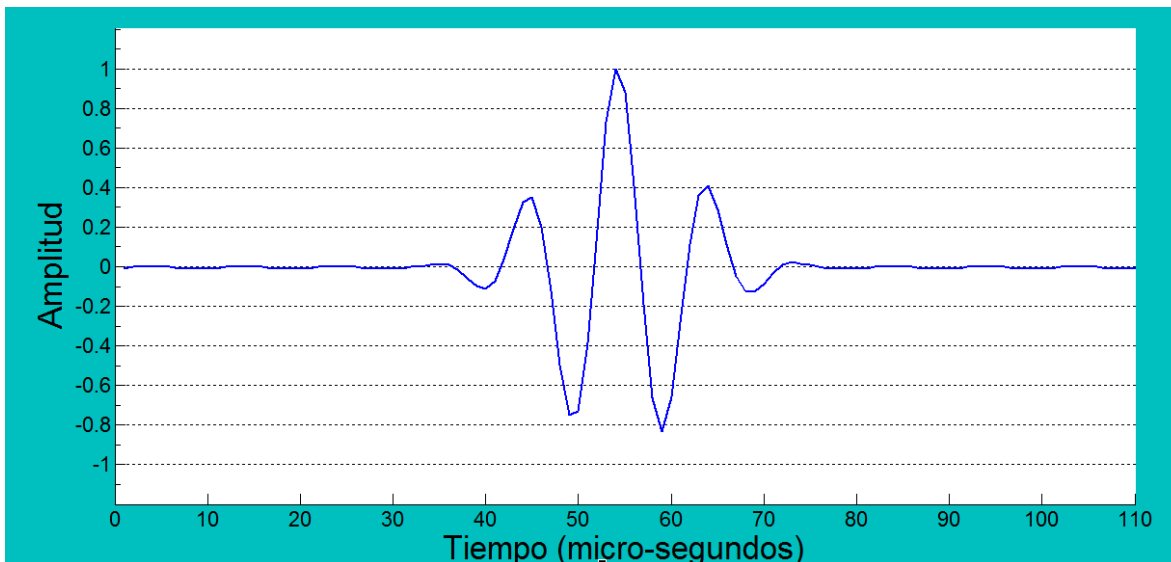


Figura 3.1 señal ultrasónica con ángulo de 11.25°

3.2 Generación de matriz de datos

Para generar la matriz de datos se debe capturar la amplitud del máximo o mínimo local Y, la posición en X del máximo o mínimo local y el signo de la amplitud de las variables que son introducidas en la red neuronal sólo se eligen: la amplitud máxima y mínima, la posición de la amplitud y el signo de la amplitud, el cual depende de si la amplitud se encuentra por debajo o es mayor a un umbral binario equivalente a 127. El valor de 127 es offset. Como se establece se establece tener un valor de offset igual a 127 se puede decir que a todo número que sea mayor a 127 se le asigna un 1 como signo, de lo contrario se le asigna un valor de 2 como signo

Tabla 3.2 puntos tomados en eje “X”, “Y” y signo

X								Y								Signo										
39	44	49	54	59	63	68	0	0	115	169	33	253	23	179	109	0	0	2	1	2	1	2	1	2	0	0
39	44	49	54	58	63	68	72	0	115	167	36	248	21	184	107	130	0	2	1	2	1	2	1	2	1	0
39	44	49	53	58	63	68	72	0	116	164	43	249	16	187	106	131	0	2	1	2	1	2	1	2	1	0
39	43	48	53	58	63	67	72	0	117	160	43	251	14	188	104	131	0	2	1	2	1	2	1	2	1	0
38	43	48	53	57	62	67	72	0	118	159	46	247	16	193	102	132	0	2	1	2	1	2	1	2	1	0
38	43	48	52	57	62	67	71	0	119	157	51	244	9	197	100	132	0	2	1	2	1	2	1	2	1	0
38	43	47	52	57	62	66	71	0	120	154	54	246	7	199	99	133	0	2	1	2	1	2	1	2	1	0
37	42	47	52	57	61	66	71	0	121	153	56	244	9	202	96	134	0	2	1	2	1	2	1	2	1	0
37	42	47	52	56	61	66	71	0	121	151	60	237	6	207	94	135	0	2	1	2	1	2	1	2	1	0
36	42	46	51	56	61	65	70	0	122	149	65	239	2	210	93	136	0	2	1	2	1	2	1	2	1	0
36	41	46	51	56	60	65	70	0	122	147	66	237	4	210	89	137	0	2	1	2	1	2	1	2	1	0
36	41	46	51	55	60	65	70	0	123	146	69	232	5	217	86	138	0	2	1	2	1	2	1	2	1	0
36	41	46	50	55	60	65	69	0	123	144	74	230	0	220	84	138	0	2	1	2	1	2	1	2	1	0
40	45	50	55	60	64	69	0	0	142	75	230	1	220	82	140	0	0	1	2	1	2	1	2	1	0	0
35	40	45	50	54	59	64	69	73	124	141	78	225	6	225	78	141	124	2	1	2	1	2	1	2	1	2
40	45	49	54	59	64	68	73	0	140	82	220	1	230	75	142	123	0	1	2	1	2	1	2	1	2	0
39	44	49	54	59	63	68	73	0	138	84	220	0	230	74	144	123	0	1	2	1	2	1	2	1	2	0
39	44	49	54	58	63	68	72	0	138	86	217	5	232	69	146	123	0	1	2	1	2	1	2	1	2	0
39	44	49	53	58	63	68	72	0	137	89	210	4	237	66	147	122	0	1	2	1	2	1	2	1	2	0
39	43	48	53	58	63	67	72	0	136	93	210	2	239	65	149	122	0	1	2	1	2	1	2	1	2	0
38	43	48	53	57	62	67	72	0	135	94	207	6	237	60	151	121	0	1	2	1	2	1	2	1	2	0
38	43	48	52	57	62	67	71	0	134	96	202	9	244	56	153	121	0	1	2	1	2	1	2	1	2	0
38	43	47	52	57	62	66	71	0	133	99	199	7	246	54	154	120	0	1	2	1	2	1	2	1	2	0
37	42	47	52	57	61	66	71	0	132	100	197	9	244	51	157	119	0	1	2	1	2	1	2	1	2	0
37	42	47	52	56	61	66	71	0	132	102	193	16	247	46	159	118	0	1	2	1	2	1	2	1	2	0
36	42	46	51	56	61	65	70	0	131	104	188	14	251	43	160	117	0	1	2	1	2	1	2	1	2	0
36	41	46	51	56	60	65	70	0	131	106	187	16	249	43	164	116	0	1	2	1	2	1	2	1	2	0
36	41	46	51	55	60	65	70	0	130	107	184	21	248	36	167	115	0	1	2	1	2	1	2	1	2	0
41	46	50	55	60	65	69	0	0	109	179	23	253	33	169	115	0	0	2	1	2	1	2	1	2	0	0
40	45	50	55	60	64	69	0	0	111	178	23	252	33	171	113	0	0	2	1	2	1	2	1	2	0	0
40	45	50	54	59	64	69	0	0	112	175	28	247	28	175	112	0	0	2	1	2	1	2	1	2	0	0

3.3 Búsqueda del ínfimo “Y”, supremo y almacenamiento de la posición temporal

La obtención del supremo y el ínfimo con sus respectivas posiciones temporales se encuentra al comparar cuales son los valores en Y más altos y más bajos y al mismo tiempo tomar su ubicación, el ínfimo y el supremo son los puntos mínimos y máximos respectivamente de la señal de ultrasonido.

Los datos obtenidos de las 32 señales se muestran en la tabla 3.3

Tabla 3.3 Supremo e Ínfimo

Supremo	Sup x	Ínfimo	Inf x	Supremo	Sup x	Ínfimo	Inf x
252	54	23	59	230	59	1	54
253	54	23	59	230	59	74	63
248	54	21	58	232	58	5	54
249	53	16	58	237	58	4	53
251	53	14	58	239	58	2	53
247	53	16	57	237	57	6	53
244	52	9	57	244	57	9	52
246	52	7	57	246	57	7	52
244	52	9	57	244	57	9	52
237	52	6	56	247	56	16	52
239	51	2	56	251	56	14	51
237	51	4	56	249	56	16	51
232	51	5	55	248	55	21	51
230	50	74	46	253	55	23	50
230	50	1	55	252	55	23	50
225	50	6	54	247	54	28	50

3.4 Eliminación de datos fuera de rango

Si el valor es mayor o igual a 127 y además es menor que el valor de la caída positiva, la cual es el rango máximo de 20 decibels, cambia su valor por cero, y si es menor a 127 y su valor es mayor que la caída negativa, su valor también cambia a cero.

Resultado de datos dentro rango

Tabla 3.4 Datos de rango de 20dB

	1° punto en Y	2° punto en Y	3° punto en Y	4° punto en Y	5° punto en Y	6° punto en Y	7° punto en Y	8° punto en Y
Ángulo 1	113	171	33	252	23	178	111	0
Ángulo 2	115	169	33	253	23	179	109	0
Ángulo 3	115	167	36	248	21	184	107	0
Ángulo 4	0	164	43	249	16	187	106	0
Ángulo 5	0	160	43	251	14	188	104	0
Ángulo 6	0	159	46	247	16	193	102	0
Ángulo 7	0	157	51	244	9	197	100	0
Ángulo 8	0	154	54	246	7	199	99	0
Ángulo 9	0	153	56	244	9	202	96	0
Ángulo 10	0	151	60	237	6	207	94	0
Ángulo 11	0	149	65	239	2	210	93	0
Ángulo 12	0	147	66	237	4	210	89	0
Ángulo 13	0	146	69	232	5	217	86	138
Ángulo 14	0	144	74	230	0	220	84	138
Ángulo 15	142	75	230	1	220	82	140	0
Ángulo 16	0	141	78	225	6	225	78	141
Ángulo 17	140	82	220	1	230	75	142	0
Ángulo 18	138	84	220	0	230	74	144	0
Ángulo 19	138	86	217	5	232	69	146	0
Ángulo 20	0	89	210	4	237	66	147	0
Ángulo 21	0	93	210	2	239	65	149	0
Ángulo 22	0	94	207	6	237	60	151	0
Ángulo 23	0	96	202	9	244	56	153	0
Ángulo 24	0	99	199	7	246	54	154	0
Ángulo 25	0	100	197	9	244	51	157	0
Ángulo 26	0	102	193	16	247	46	159	0
Ángulo 27	0	104	188	14	251	43	160	0
Ángulo 28	0	106	187	16	249	43	164	0
Ángulo 29	0	107	184	21	248	36	167	115
Ángulo 30	109	179	23	253	33	169	115	0
Ángulo 31	111	178	23	252	33	171	113	0
Ángulo 32	112	175	28	247	28	175	112	0

3.5 Normalización de datos

Para realizar la normalización de datos la Ynorm se resta 127, considerando la excepción del dato 0 ya que este permanecerá igual y no debe ser afectado por la resta.

Tabla 3.5 Cambio de amplitud de los datos

	1° punto en Y	2° punto en Y	3° punto en Y	4° punto en Y	5° punto en Y	6° punto en Y	7° punto en Y	8° punto en Y
Ángulo 1	-14	44	-94	125	-104	51	-16	0
Ángulo 2	-12	42	-94	126	-104	52	-18	0
Ángulo 3	-12	40	-91	121	-106	57	-20	0
Ángulo 4	0	37	-84	122	-111	60	-21	0
Ángulo 5	0	33	-84	124	-113	61	-23	0
Ángulo 6	0	32	-81	120	-111	66	-25	0
Ángulo 7	0	30	-76	117	-118	70	-27	0
Ángulo 8	0	27	-73	119	-120	72	-28	0
Ángulo 9	0	26	-71	117	-118	75	-31	0
Ángulo 10	0	24	-67	110	-121	80	-33	0
Ángulo 11	0	22	-62	112	-125	83	-34	0
Ángulo 12	0	20	-61	110	-123	83	-38	0
Ángulo 13	0	19	-58	105	-122	90	-41	11
Ángulo 14	0	17	-53	103	0	93	-43	11
Ángulo 15	15	-52	103	-126	93	-45	13	0
Ángulo 16	0	14	-49	98	-121	98	-49	14
Ángulo 17	13	-45	93	-126	103	-52	15	0
Ángulo 18	11	-43	93	0	103	-53	17	0
Ángulo 19	11	-41	90	-122	105	-58	19	0
Ángulo 20	0	-38	83	-123	110	-61	20	0
Ángulo 21	0	-34	83	-125	112	-62	22	0
Ángulo 22	0	-33	80	-121	110	-67	24	0
Ángulo 23	0	-31	75	-118	117	-71	26	0
Ángulo 24	0	-28	72	-120	119	-73	27	0
Ángulo 25	0	-27	70	-118	117	-76	30	0
Ángulo 26	0	-25	66	-111	120	-81	32	0
Ángulo 27	0	-23	61	-113	124	-84	33	0
Ángulo 28	0	-21	60	-111	122	-84	37	0
Ángulo 29	0	-20	57	-106	121	-91	40	-12
Ángulo 30	-18	52	-104	126	-94	42	-12	0
Ángulo 31	-16	51	-104	125	-94	44	-14	0
Ángulo 32	-15	48	-99	120	-99	48	-15	0

3.6 Orden de los datos con base en su amplitud

Para poder obtener las entradas es necesario ordenar los datos en el que se asignan valores al principio de la siguiente manera, representando signo positivo como un 1 si el valor de Y es mayor a 0 De lo contrario se asigna un 2 como signo negativo, el cual indica que el valor es menor que 0.

Tabla 3.6 Asignación de número de acuerdo con el signo

Signo	Orden de los datos según signo					
2	-14	-94	-104	44	125	51
2	-12	-94	-104	42	126	52
2	-12	-91	-106	40	121	57
1	37	122	60	-84	-111	-21
1	33	124	61	-84	-113	-23
1	32	120	66	-81	-111	-25
1	30	117	70	-76	-118	-27
1	27	119	72	-73	-120	-28
1	26	117	75	-71	-118	-31
1	24	110	80	-67	-121	-33
1	22	112	83	-62	-125	-34
1	20	110	83	-61	-123	-38
1	19	105	90	-58	-122	-41
1	17	103	93	-53	0	-43
1	15	103	93	-52	-126	-45
1	14	98	98	-49	-121	-49
1	13	93	103	-45	-126	-52
1	11	93	103	-43	0	-53
1	11	90	105	-41	-122	-58
2	-38	-123	-61	83	110	20
2	-34	-125	-62	83	112	22
2	-33	-121	-67	80	110	24
2	-31	-118	-71	75	117	26
2	-28	-120	-73	72	119	27
2	-27	-118	-76	70	117	30
2	-25	-111	-81	66	120	32
2	-23	-113	-84	61	124	33
2	-21	-111	-84	60	122	37
2	-20	-106	-91	57	121	40
2	-18	-104	-94	52	126	42
2	-16	-104	-94	51	125	44
2	-15	-99	-99	48	120	48

3.7 Asignación de números en orden

La asignación de números para formar la matriz máx. y min. negativos y positivos se realiza dependiendo del primer número de matriz entrenada, ya sea 1 o 2 se mandan a distintas opciones, si son valores mayores o menores a cero, con esto se obtiene el orden de los valores mínimo positivo, intermedio positivo, máximo positivo, mínimo negativo, intermedio negativo, máximo negativo.

1= mínimo positivo 2= intermedio positivo 3= máximo positivo
 4= mínimo negativo 5= intermedio negativo 6= máximo negativo

Tabla 3.7 Matriz máx. y min. positivo.

Signo	Asignación de números en orden máximo positiva a mínimo negativo					
2	4	5	6	1	3	2
2	4	5	6	1	3	2
2	4	5	6	1	3	2
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	6	4
1	1	3	2	5	4	6
1	1	3	2	5	6	4
1	1	2	3	5	6	4
1	1	2	3	4	6	5
1	1	2	3	6	4	5
1	1	2	3	4	6	5
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1
2	4	6	5	2	3	1

3.8 Detección del pulso de máxima amplitud para realizar el normalizado

Para normalizar los valores y crear una matriz normalizada, se realizará una operación usando el supremo y el ínfimo la cual compara cual es mayor en cada posición y este será el máximo que dividirá el valor de la matriz entrenada, se usará la siguiente ecuación.

Ejemplo:
$$\frac{\sqrt{(Matriz_entrena_{k,j})^2}}{\maximo_k} = \frac{\sqrt{(57_{3,5})^2}}{121_3} = 0.4711_{3,6}$$

Tabla 3.8 Matriz normalizada.

	1° punto Y	2° punto Y	3° punto Y	4° punto Y	5° punto Y	6° punto Y
Ángulo 1	0.1120	0.7520	0.8320	0.3520	1.0000	0.4080
Ángulo 2	0.0952	0.7460	0.8254	0.3333	1.0000	0.4127
Ángulo 3	0.0992	0.7521	0.8760	0.3306	1.0000	0.4711
Ángulo 4	0.3033	1.0000	0.4918	0.6885	0.9098	0.1721
Ángulo 5	0.2661	1.0000	0.4919	0.6774	0.9113	0.1855
Ángulo 6	0.2667	1.0000	0.5500	0.6750	0.9250	0.2083
Ángulo 7	0.2542	0.9915	0.5932	0.6441	1.0000	0.2288
Ángulo 8	0.2250	0.9917	0.6000	0.6083	1.0000	0.2333
Ángulo 9	0.2203	0.9915	0.6356	0.6017	1.0000	0.2627
Ángulo 10	0.1983	0.9091	0.6612	0.5537	1.0000	0.2727
Ángulo 11	0.1760	0.8960	0.6640	0.4960	1.0000	0.2720
Ángulo 12	0.1626	0.8943	0.6748	0.4959	1.0000	0.3089
Ángulo 13	0.1557	0.8607	0.7377	0.4754	1.0000	0.3361
Ángulo 14	0.1650	1.0000	0.9029	0.5146	0.0000	0.4175
Ángulo 15	0.1190	0.8175	0.7381	0.4127	1.0000	0.3571
Ángulo 16	0.1157	0.8099	0.8099	0.4050	1.0000	0.4050
Ángulo 17	0.1032	0.7381	0.8175	0.3571	1.0000	0.4127
Ángulo 18	0.1068	0.9029	1.0000	0.4175	0.0000	0.5146
Ángulo 19	0.0902	0.7377	0.8607	0.3361	1.0000	0.4754
Ángulo 20	0.3089	1.0000	0.4959	0.6748	0.8943	0.1626
Ángulo 21	0.2720	1.0000	0.4960	0.6640	0.8960	0.1760
Ángulo 22	0.2727	1.0000	0.5537	0.6612	0.9091	0.1983
Ángulo 23	0.2627	1.0000	0.6017	0.6356	0.9915	0.2203
Ángulo 24	0.2333	1.0000	0.6083	0.6000	0.9917	0.2250
Ángulo 25	0.2288	1.0000	0.6441	0.5932	0.9915	0.2542
Ángulo 26	0.2083	0.9250	0.6750	0.5500	1.0000	0.2667
Ángulo 27	0.1855	0.9113	0.6774	0.4919	1.0000	0.2661
Ángulo 28	0.1721	0.9098	0.6885	0.4918	1.0000	0.3033
Ángulo 29	0.1653	0.8760	0.7521	0.4711	1.0000	0.3306
Ángulo 30	0.1429	0.8254	0.7460	0.4127	1.0000	0.3333
Ángulo 31	0.1280	0.8320	0.7520	0.4080	1.0000	0.3520
Ángulo 32	0.1250	0.8250	0.8250	0.4000	1.0000	0.4000

3.9 Generación de matriz de entrenamiento

Los datos que se usan como entrada para la red neuronal serán los siguientes, de la matriz máx. y mín. se hace uso de la primera columna para saber si el signo es positivo o negativo, después se usan las columnas segunda y tercera para saber si tiene máximo, intermedio, o mínimo ya sea positivo o negativo. Las columnas cuarta y quinta son valores que vienen de la posición en el punto máximo y el mínimo de la señal de ultrasonido. Por último, se hace uso de los seis datos de la matriz normalizada, con ello finalmente se cuenta con los valores suficientes para introducir en la red neuronal quedando la matriz de la siguiente forma:

Tabla 3.9 Matriz entrenamiento

Signo	Max y Min		Posición X		Matriz normalizada de Y					
2	4	5	14	19	0.1120	0.7520	0.8320	0.3520	1.0000	0.4080
2	4	5	15	20	0.0952	0.7460	0.8254	0.3333	1.0000	0.4127
2	4	5	15	19	0.0992	0.7521	0.8760	0.3306	1.0000	0.4711
1	1	3	9	14	0.3033	1.0000	0.4918	0.6885	0.9098	0.1721
1	1	3	10	15	0.2661	1.0000	0.4919	0.6774	0.9113	0.1855
1	1	3	10	14	0.2667	1.0000	0.5500	0.6750	0.9250	0.2083
1	1	3	9	14	0.2542	0.9915	0.5932	0.6441	1.0000	0.2288
1	1	3	9	14	0.2250	0.9917	0.6000	0.6083	1.0000	0.2333
1	1	3	10	15	0.2203	0.9915	0.6356	0.6017	1.0000	0.2627
1	1	3	10	14	0.1983	0.9091	0.6612	0.5537	1.0000	0.2727
1	1	3	9	14	0.1760	0.8960	0.6640	0.4960	1.0000	0.2720
1	1	3	10	15	0.1626	0.8943	0.6748	0.4959	1.0000	0.3089
1	1	3	10	14	0.1557	0.8607	0.7377	0.4754	1.0000	0.3361
1	1	3	9	5	0.1650	1.0000	0.9029	0.5146	0.0000	0.4175
1	1	3	10	15	0.1190	0.8175	0.7381	0.4127	1.0000	0.3571
1	1	2	10	14	0.1157	0.8099	0.8099	0.4050	1.0000	0.4050
1	1	2	19	14	0.1032	0.7381	0.8175	0.3571	1.0000	0.4127
1	1	2	20	24	0.1068	0.9029	1.0000	0.4175	0.0000	0.5146
1	1	2	19	15	0.0902	0.7377	0.8607	0.3361	1.0000	0.4754
2	4	6	14	9	0.3089	1.0000	0.4959	0.6748	0.8943	0.1626
2	4	6	15	10	0.2720	1.0000	0.4960	0.6640	0.8960	0.1760
2	4	6	14	10	0.2727	1.0000	0.5537	0.6612	0.9091	0.1983
2	4	6	14	9	0.2627	1.0000	0.6017	0.6356	0.9915	0.2203
2	4	6	14	9	0.2333	1.0000	0.6083	0.6000	0.9917	0.2250
2	4	6	15	10	0.2288	1.0000	0.6441	0.5932	0.9915	0.2542
2	4	6	14	10	0.2083	0.9250	0.6750	0.5500	1.0000	0.2667
2	4	6	14	9	0.1855	0.9113	0.6774	0.4919	1.0000	0.2661
2	4	6	15	10	0.1721	0.9098	0.6885	0.4918	1.0000	0.3033
2	4	6	14	10	0.1653	0.8760	0.7521	0.4711	1.0000	0.3306
2	4	6	14	9	0.1429	0.8254	0.7460	0.4127	1.0000	0.3333
2	4	6	15	10	0.1280	0.8320	0.7520	0.4080	1.0000	0.3520

CAPITULO 4 DISEÑO DE ARQUITECTURA DE LA RED NEURONAL

En este capítulo se presenta el diseño de la arquitectura de la red. Utilizando la matriz de entrenamiento, se inicia con la asignación de las entradas y de las salidas de la red, después se forman los valores iniciales, y se verá la utilización de las funciones y las neuronas de la capa oculta y salida. Posteriormente se presenta el diseño de las épocas de error, se realiza el entrenamiento para determinar el error, la sensibilidad de la neurona y así poder obtener los nuevos valores que se utilizan en la prueba de la neurona.

4.1 Asignación de entradas y salida de la red

Para poder introducir los valores a la red se llama al archivo de la captura de los datos del ultrasonido. Al llamarlo, se puede tomar datos que se encuentren en este archivo, los cuales en el presente trabajo corresponden a los diferentes ángulos de la red neuronal. La Y deseada que será el resultado con el que se compara, será una línea de datos vertical ascendente que va de 0 a 1 con intervalos de 0.3125

Tabla 4.1 Y deseada

0.03125	0.06250	0.09375	0.12500	0.15625	0.18750	0.21875	0.25000
0.28125	0.31250	0.34375	0.37500	0.40625	0.43750	0.46875	0.50000
0.53125	0.56250	0.59375	0.62500	0.65625	0.68750	0.71875	0.75000
0.78125	0.81250	0.84375	0.87500	0.90625	0.93750	0.96875	1.00000

4.2 Valores iniciales

Los pesos ($W1$, $W2$) y los bias ($b1$, $b2$) se inicializan utilizando valores aleatorios del generador de MATLAB con un rango de $-1,1$. Para $W1$ como es la capa oculta de entrada se formará una matriz de 11×11 para la matriz $W2$ será un vector de 11×1 , para $b1$ será un vector de 11×1 y por último $b2$ un valor. También se pone un valor de aprendizaje el cual puede variar constantemente para encontrar los mejores resultados al estar entrenando la red.

Tabla 4.2 Matriz ejemplo W1

-0.629	-0.941	-0.698	0.908	0.626	0.003	-0.094	0.498	0.239	0.326	0.099
-0.812	-0.914	-0.868	0.806	0.020	-0.919	0.723	-0.232	-0.136	0.676	0.832
0.746	0.029	-0.357	-0.647	0.109	0.319	0.701	0.053	0.848	-0.589	0.542
-0.827	-0.601	-0.515	-0.390	-0.293	-0.171	0.485	0.297	0.892	0.378	-0.827
-0.265	0.716	-0.486	0.366	-0.419	0.552	-0.681	-0.662	-0.062	-0.057	0.695
0.805	0.156	0.216	-0.900	-0.509	-0.503	0.491	-0.171	-0.558	0.669	-0.652
0.443	-0.831	-0.311	0.931	0.448	0.490	-0.629	-0.099	-0.868	-0.204	-0.077
-0.094	-0.584	0.658	0.123	-0.359	-0.012	0.513	-0.834	0.740	0.474	-0.992
-0.915	-0.919	-0.412	0.237	-0.310	-0.398	-0.859	0.428	-0.138	-0.308	0.844
-0.930	-0.311	0.936	-0.531	0.675	-0.782	0.300	-0.514	0.061	-0.378	0.115
0.685	0.929	0.446	-0.590	0.762	-0.919	0.607	-0.507	0.976	-0.496	0.787

Tabla 4.3 vector ejemplo W2

0.924
-0.991
0.550
0.635
0.737
-0.831
-0.200
-0.480
0.600
-0.137
0.821

Tabla 4.4 vector ejemplo b1

-0.636
-0.472
-0.709
-0.728
0.739
0.159
0.100
-0.710
0.706
0.244
-0.298

Tabla 4.5 ejemplo b2

0.026

4.3 Función

Para desarrollar la red neuronal y por el tipo de datos utilizados que se necesitan de resultado se usó la función sigmoide la cual da como resultado valores de cero a uno. Para hacer una sigmoide como función de salida la operación es la siguiente:

$$(21) \quad \frac{1}{1+e^{-sumatoria}}$$

4.4 Capa oculta y capa de salida

Para obtener la sumatoria de la capa oculta y de salida se multiplican las entradas por los pesos (W) y se le suma los bias (b) para después utilizar el resultado en la función.

Las operaciones son:

$$(22) \quad suma = \sum(Xis * W) + bias$$

El resultado se introduce a f=Sigmoide (sumatoria) la cual es la forma de mandar los datos a la función de sigmoide.

4.5 Entrenamiento

El entrenamiento de la red neuronal consiste en realizar la operación de la capa oculta, la capa de salida, determinar el error, determinar la sensibilidad del error y asignar los nuevos valores. Cuando el error alcance el valor propuesto aleatoriamente dará los resultados. Aquí es donde se hacen pruebas cambiando el valor para encontrar el mejor resultado ya que junto con el factor de aprendizaje alcanzara este valor.

4.6 Conexión de la capa oculta y de la capa de salida

La conexión de la capa oculta y de salida se hará con los valores de entrada “x”, los pesos generados “W” y los bias “b” estos se introducen a una función dando como resultado la sumatoria de “x” por el peso “W1” más el bias “b1” igual a un valor que se llamara “s” y este se introduce a la función para obtener once resultados nombrados como “h1” hasta “h11” y la conexión de la capa de salida se hará al multiplicar los valores de las ”h” por los pesos “W2” más el bias “b2”, resultando un valor llamado “r” y después este valor se introduce a la función sigmoide para tener la salida llamada “Ok”

4.7 Determinación del error, de la sensibilidad de la neurona y nuevos valores

La determinación del error “e” se consigue con la Y deseada (Yd) menos el “Ok” y para la sensibilidad de la neurona se usa la resta de Y deseada menos Ok por Ok por uno menos Ok esto porque la salida es Sigmoide y así se obtiene “Sk”. Después utilizando “Sk”, haciendo una sumatoria de “Sk” por los once pesos para encontrar los nuevos valores

$$e = (Yd_n - Ok)$$

$$Sk = (Yd_n - Ok) * Ok * (1 - Ok)$$

$$Sumatoria = \sum (Sk * W2_n)$$

Al tener el resultado de la sumatoria esto se multiplica por las once “h” y se multiplica por uno menos “h” dando como resultado “Dj”.

$$Dj(n) = sumatoria * h(n) * (1 - h(n))$$

Los nuevos valores de los pesos y de las bias se forman sumando “W2” y “n” que es el factor de aprendizaje por “Sk” multiplicado por el vector de h1 a h11

W1 se obtiene con W1 más n factor de aprendizaje por x por Dj

Para el valor b2 es b2 más n por Sk y por último b1 es obtenido al tener b1 más n por Dj

Con este proceso se obtienen los nuevos valores y el ciclo se detendrá hasta que pase el error.

$$W2=W2+n*Sk*[h1; h2; h3; h4; h5; h6; h7; h8; h9; h10; h11]$$

$$W1=W1+n*x*Dj$$

$$b2=b2+n*Sk$$

$$b1=b1+n*Dj$$

4.8 Prueba de la neurona

La prueba de la neurona consiste en que al obtener los resultados de los nuevos pesos y bias en base al error estos entrarán a una red de propagación, dichas entradas pasarán por una capa oculta, y capa de salida y así obtener el valor de 32 Ok, que será la Y estimada, esta se podrá comparar con la Y deseada y se podrá ver que tan exacto es el resultado y seguir probando manipulando tanto el factor de aprendizaje como el error. Al obtener las Ok se grafica la Y estimada con la Y deseada y de esta forma poder ver el comportamiento.

CAPITULO 5 EXPERIMENTACION Y RESULTADOS

En este capítulo se realiza el entrenamiento de la red por época, se realiza la experimentación cambiando los factores para obtener distintas gráficas, al analizarlas se podrá observar la que mejor da resultados y saber que valores nuevos utilizar.

5.1 Entrenamiento de la red por época

El entrenamiento de la red se realizó cambiando el factor de aprendizaje y el valor del error, al correr nuestro programa y después de un número determinado de iteraciones se hace la prueba de la neurona, se obtiene y analiza una gráfica donde es posible apreciar si el factor de aprendizaje y el valor del error se encuentran en niveles aceptables, dicho procedimiento, es realizado con distintos valores, los resultados de estas gráficas son utilizados para observar cuales valores dan un resultado más exacto y conveniente. En la tabla 5.1 se ilustra cuáles fueron los valores que fueron utilizados, cuantas iteraciones se realizaron y el número de gráfico.

Tabla 5.1 Entrenamiento de la red

Factor de aprendizaje	de	Valor del Error	Iteraciones	Grafico
0.001		0.5	171338	Figura 5.1
0.01		0.3	147025	Figura 5.2
0.01		0.5	14687	Figura 5.3
0.1		0.3	13519	Figura 5.4
0.1		0.5	2317	Figura 5.5
0.005		0.3	123977	Figura 5.6
0.005		0.5	47012	Figura 5.7
0.05		0.3	31050	Figura 5.8
0.06		0.3	20894	Figura 5.9
0.06		0.5	74911	Figura 5.10
0.06		0.1	321641	Figura 5.11
0.06		0.1	455782	Figura 5.12
0.06		0.1	137484	Figura 5.13
0.06		0.2	14316	Figura 5.14
0.08		0.3	20101	Figura 5.15

En los resultados obtenidos se aprecia que la gráfica más exacta es la que usó un factor de aprendizaje de 0.06 y un error de 0.1 por ende es la que se va a tomar el error los nuevos pesos y bias.

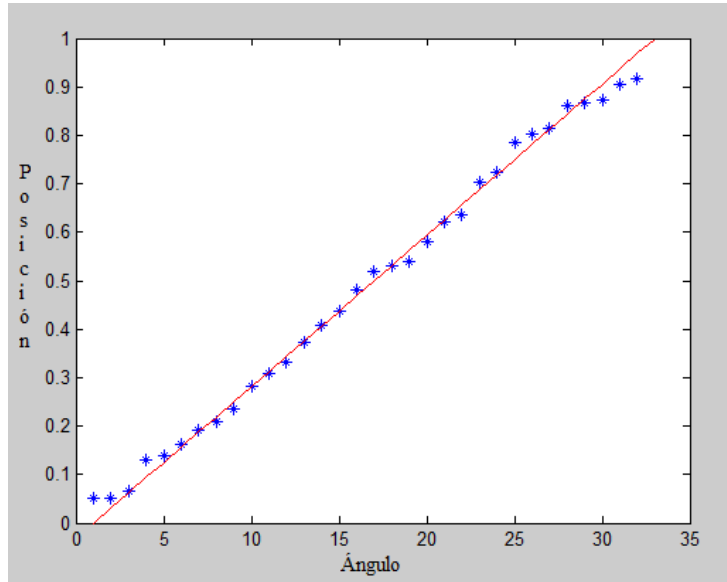


Figura 5.1 FA 0.001 VE 0.5

Factor de aprendizaje 0.001

Valor del error 0.5

Numero de iteraciones realizadas para resultado 171338

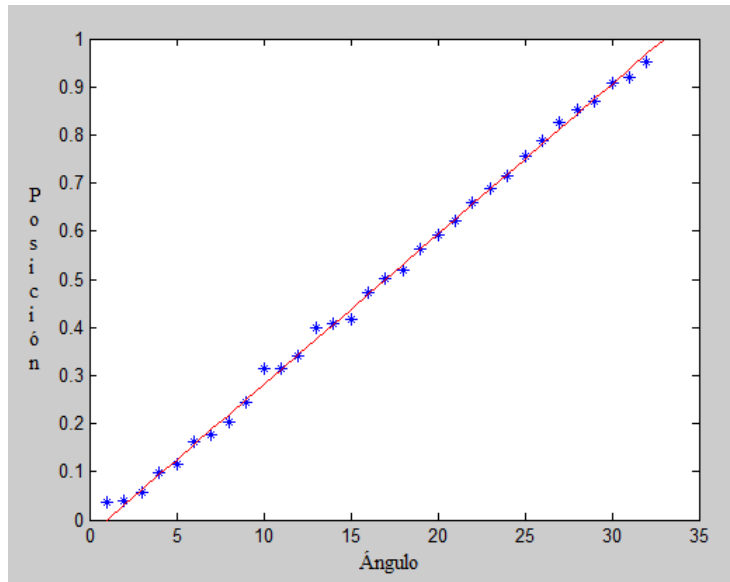


Figura 5.2 FA 0.01 VE 0.3

Factor de aprendizaje 0.01

Valor del error 0.3

Numero de iteraciones realizadas para resultado 147025

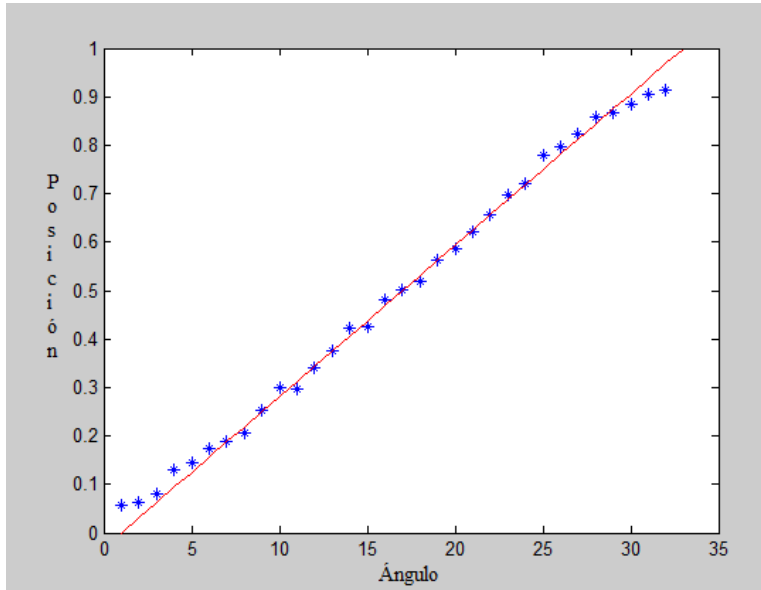


Figura 5.3 FA 0.01 VE 0.5

Factor de aprendizaje 0.01

Valor del error 0.5

Numero de iteraciones realizadas para resultado 14687

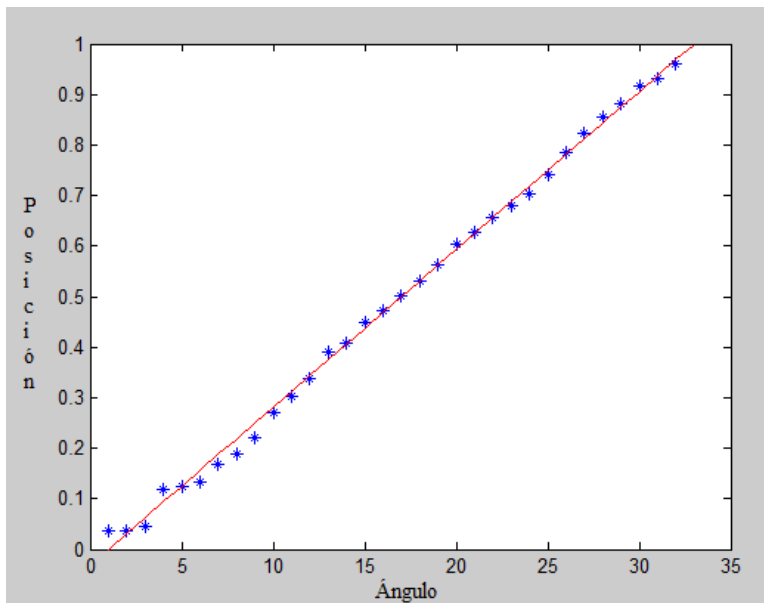


Figura 5.4 FA 0.1 VE 0.3

Factor de aprendizaje 0.1

Valor del error 0.3

Numero de iteraciones realizadas para resultado 13519

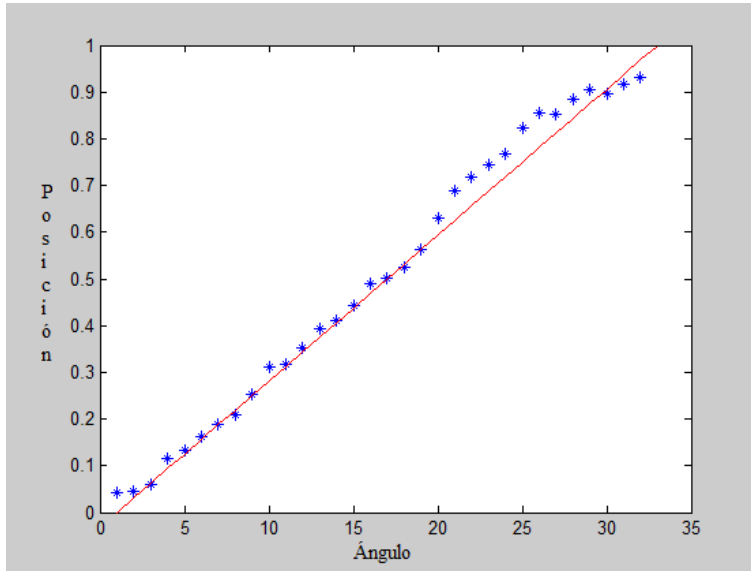


Figura 5.5 FA 0.1 VE 0.5

Factor de aprendizaje 0.1

Valor del error 0.5

Numero de iteraciones realizadas para resultado 2317

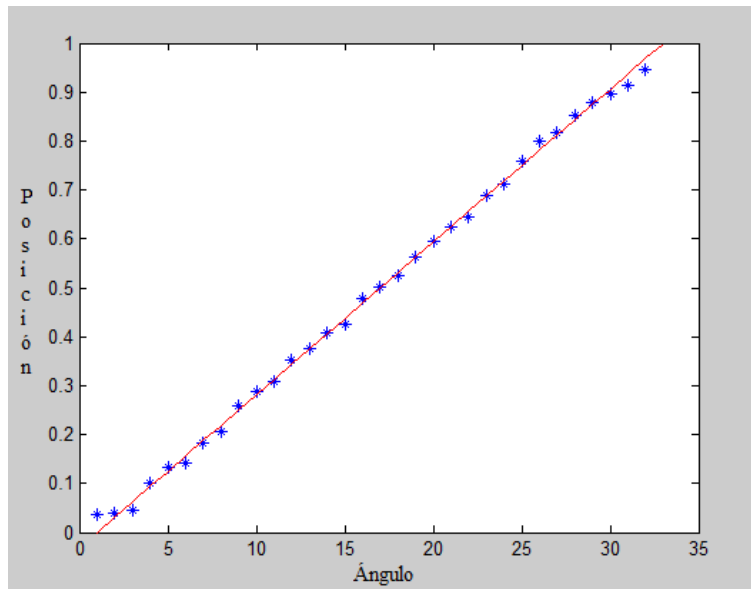


Figura 5.6 FA 0.005 VE 0.3

Factor de aprendizaje 0.005

Valor del error 0.3

Numero de iteraciones realizadas para resultado 123977

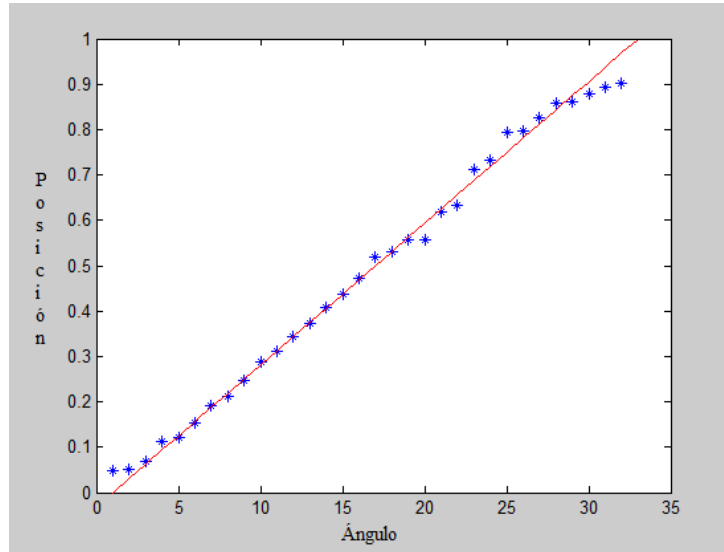


Figura 5.7 FA 0.005 VE 0.5

Factor de aprendizaje 0.005

Valor del error 0.5

Numero de iteraciones realizadas para resultado 47012

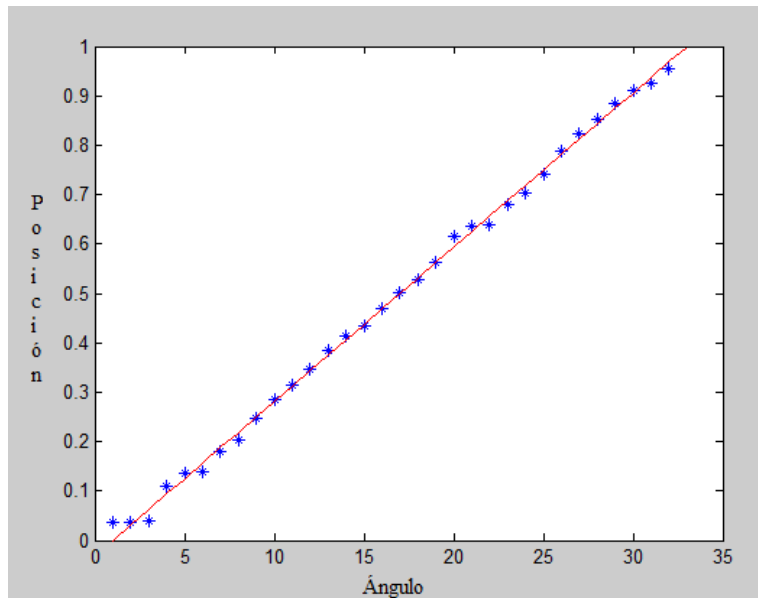


Figura 5.8 FA 0.05 VE 0.3

Factor de aprendizaje 0.05

Valor del error 0.3

Numero de iteraciones realizadas para resultado 31050

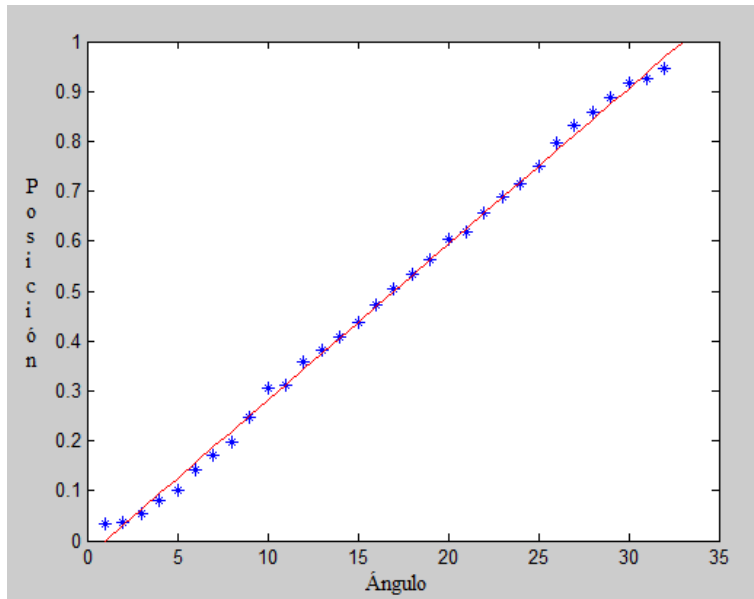


Figura 5.9 FA 0.06 VE 0.3

Factor de aprendizaje 0.06

Valor del error 0.3

Numero de iteraciones realizadas para resultado 20894

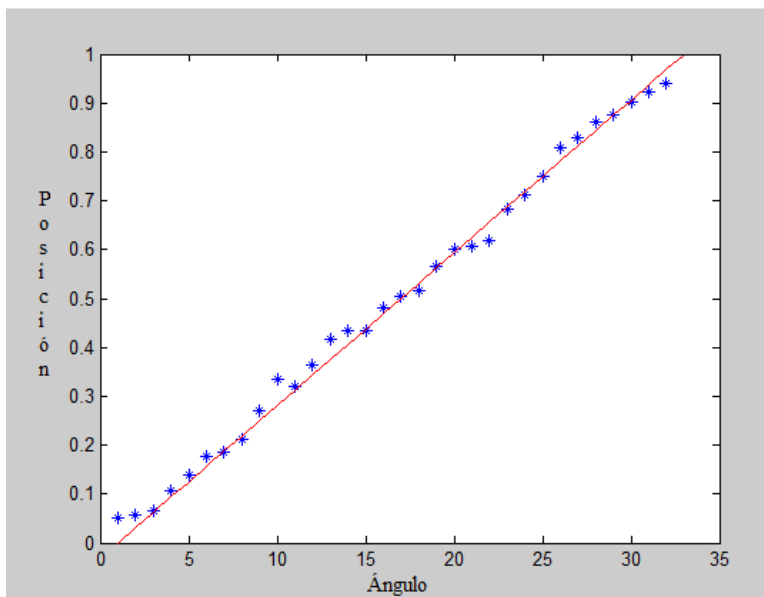


Figura 5.10 FA 0.06 VE 0.5

Factor de aprendizaje 0.06

Valor del error 0.5

Numero de iteraciones realizadas para resultado 7491

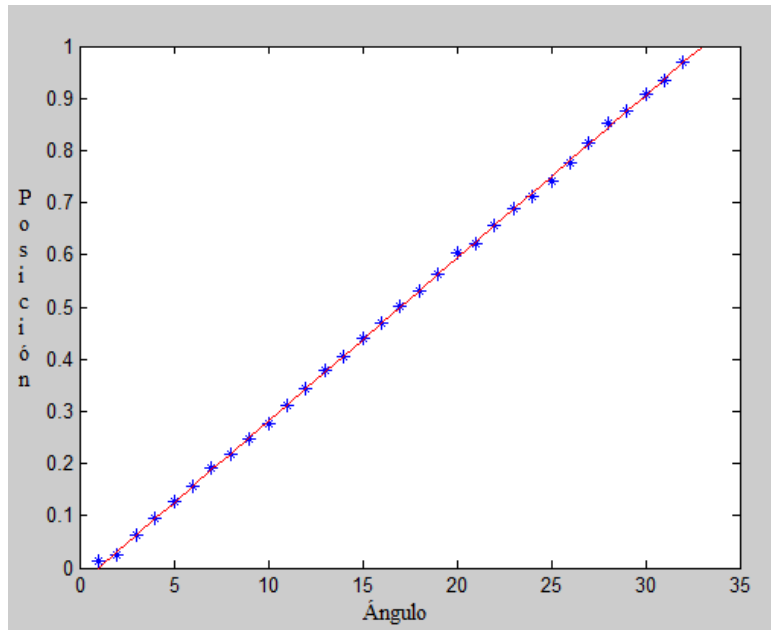


Figura 5.11 FA 0.06 VE 0.1

Factor de aprendizaje 0.06

Valor del error 0.1

Numero de iteraciones realizadas para resultado 321641

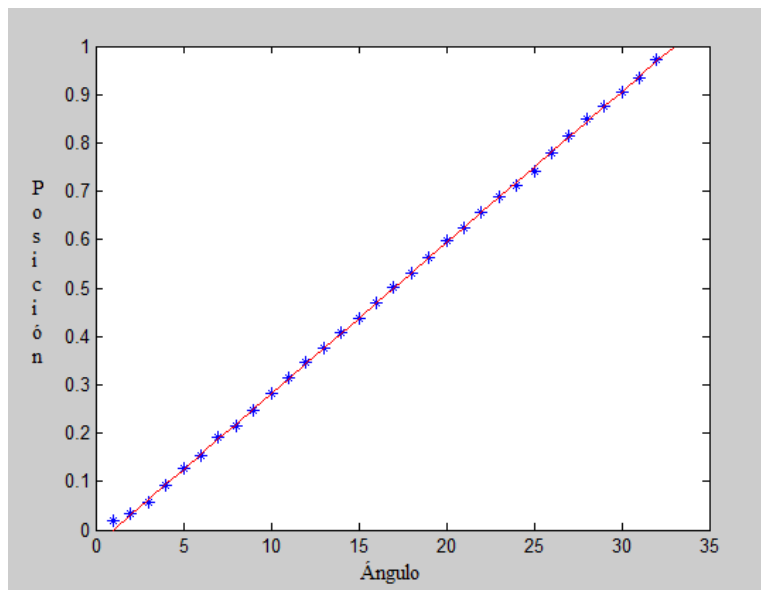


Figura 5.12 FA 0.06 VE 0.1

Factor de aprendizaje 0.06

Valor del error 0.1

Numero de iteraciones realizadas para resultado 455782

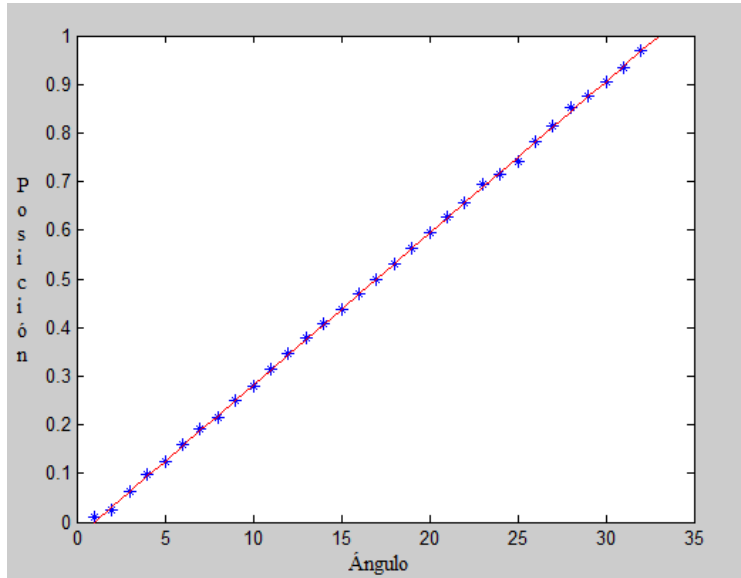


Figura 5.13 FA 0.06 VE 0.1

Factor de aprendizaje 0.06

Valor del error 0.1

Numero de iteraciones realizadas para resultado 137484

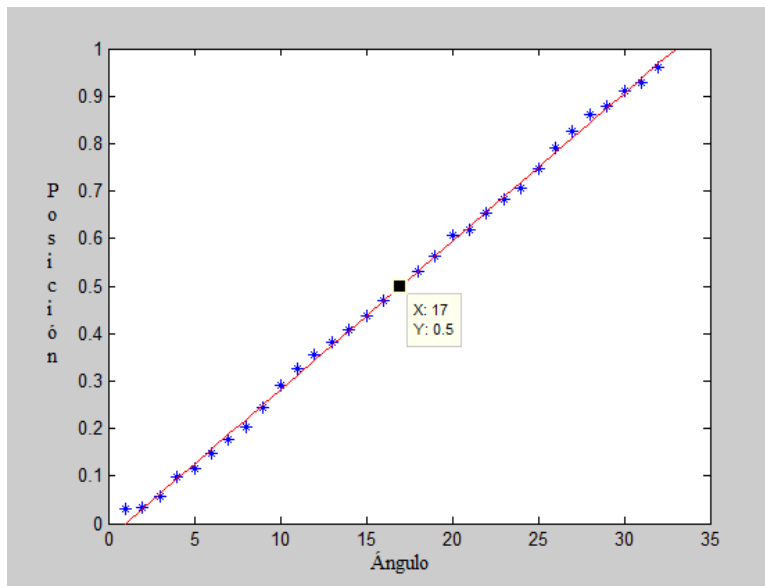


Figura 5.14 FA 0.06 VE 0.2

Factor de aprendizaje 0.06

Valor del error 0.2

Numero de iteraciones realizadas para resultado 14316

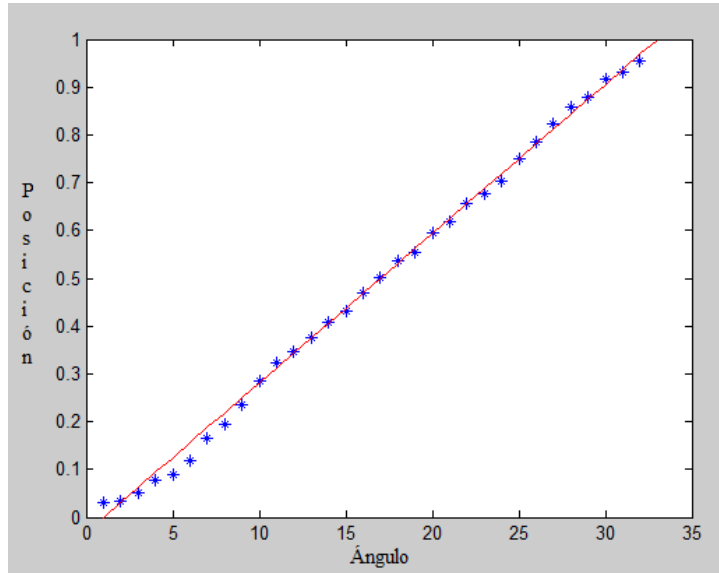


Figura 5.15 FA 0.08 VE 0.3

Factor de aprendizaje 0.08

Valor del error 0.3

Numero de iteraciones realizadas para resultado 20101

5.2 Nuevos Valores

Los nuevos valores $W1$, $b1$, $W2$ y $b2$ son los resultados que se obtienen cuando la red de retropropagación llega al error determinado en este caso 0.1 con un factor de aprendizaje de 0.06 y luego al error después de 321641 iteraciones.

Pesos $W1$ 11x11

Tabla 5.2 nuevos pesos $W1$

-0.633	-1.838	-0.699	0.854	1.113	-0.276	-0.504	-0.428	0.238	-0.220	-0.409
-0.815	-3.082	-0.869	0.710	3.407	-1.926	1.136	-1.803	-0.137	-2.785	1.643
0.735	-0.238	-0.360	-0.790	1.238	1.259	-0.876	0.275	0.845	-0.378	-0.327
-0.860	-2.105	-0.522	-0.810	-0.075	-3.453	0.252	2.680	0.883	0.710	-1.044
-0.283	1.858	-0.490	-0.034	-1.457	2.031	-0.820	-1.460	-0.073	-0.303	1.823
0.804	-1.760	0.215	-0.916	-0.949	-2.973	-2.831	-1.625	-0.559	-2.653	-2.747
0.439	-0.014	-0.312	0.896	-0.935	0.005	0.039	-1.753	-0.869	-0.593	-2.610
-0.097	1.545	0.657	0.106	-1.101	2.981	3.545	0.459	0.740	4.554	-0.786
-0.917	-2.240	-0.412	0.210	-1.263	-2.967	-2.094	-1.897	-0.138	-2.703	-1.910
-0.930	0.318	0.936	-0.534	2.341	1.537	4.453	-1.499	0.061	2.233	-2.342
0.683	2.568	0.446	-0.596	0.498	1.159	3.765	0.994	0.976	2.908	2.643

Bias $b1$ 11x1

Pesos $W2$ 11x1

Tabla 5.3 nuevos bias $b1$

Tabla 5.4 nuevos pesos $W2$

-0.63996824
-0.73414623
-0.70961239
-0.76073037
-0.22449908
0.243136924
-0.72105748
-1.31315168
0.70502433
1.155561257
-1.46509662

0.923198082
1.345722794
0.549714639
0.601509508
4.057066055
0.544695261
1.519259898
2.194180283
-2.7626884
2.900166738
0.955601012

Bias $b2$ 1x1

Tabla 5.5 nuevo bias $b2$

-3.33645396

CAPITULO 6 RESULTADOS Y CONCLUSIONES

En el capítulo se muestra los resultados y la arquitectura final de la red de retropropagación que es utilizada, se continuara con la conclusión sobre el proyecto, y se finaliza con la sección de referencias y las biografías utilizadas.

6.1 Resultado

En la figura 6.1 se puede ver como al introducir la señal de ultrasonido a través de la red neuronal con los nuevos valores, el resultado será que al adquirir los datos la “Y” estimada tendrá una coincidencia con la “Y” deseada

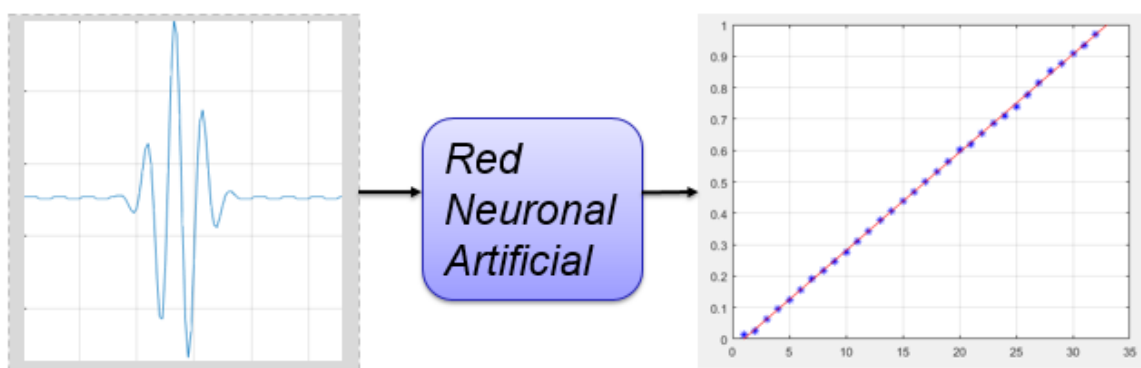


Figura 6.1 Resultado Señal ultrasonido

Resultados “Y” deseada y “Y” estimada con factor de aprendizaje 0.06 y un valor del error 0.1

Tabla 5.6 Resultados “Y” deseada y “Y” estimada

	Y deseada	Y estimada	% de error
Ángulo 1	0	0.0123	1%
Ángulo 2	0.03125	0.02583	1%
Ángulo 3	0.0625	0.0625	0%
Ángulo 4	0.09375	0.09351	0%
Ángulo 5	0.125	0.125	0%
Ángulo 6	0.15625	0.15541	0%
Ángulo 7	0.1875	0.19176	0%
Ángulo 8	0.21875	0.21742	0%
Ángulo 9	0.25	0.25	0%
Ángulo 10	0.28125	0.27484	1%
Ángulo 11	0.3125	0.3094	0%

Ángulo 12	0.34375	0.34267	0%
Ángulo 13	0.375	0.378	0%
Ángulo 14	0.40625	0.40643	0%
Ángulo 15	0.4375	0.4384	0%
Ángulo 16	0.46875	0.46677	0%
Ángulo 17	0.5	0.5	0%
Ángulo 18	0.53125	0.53099	0%
Ángulo 19	0.5625	0.5630	0%
Ángulo 20	0.59375	0.60304	1%
Ángulo 21	0.625	0.6201	0%
Ángulo 22	0.65625	0.65597	0%
Ángulo 23	0.6875	0.6880	0%
Ángulo 24	0.71875	0.71074	1%
Ángulo 25	0.75	0.74	1%
Ángulo 26	0.78125	0.77730	0%
Ángulo 27	0.8125	0.8151	0%
Ángulo 28	0.84375	0.85256	1%
Ángulo 29	0.875	0.876	0%
Ángulo 30	0.90625	0.90704	0%
Ángulo 31	0.9375	0.9344	0%
Ángulo 32	0.96875	0.96882	0%
Total de porcentaje de error			10%

6.2 Arquitectura final

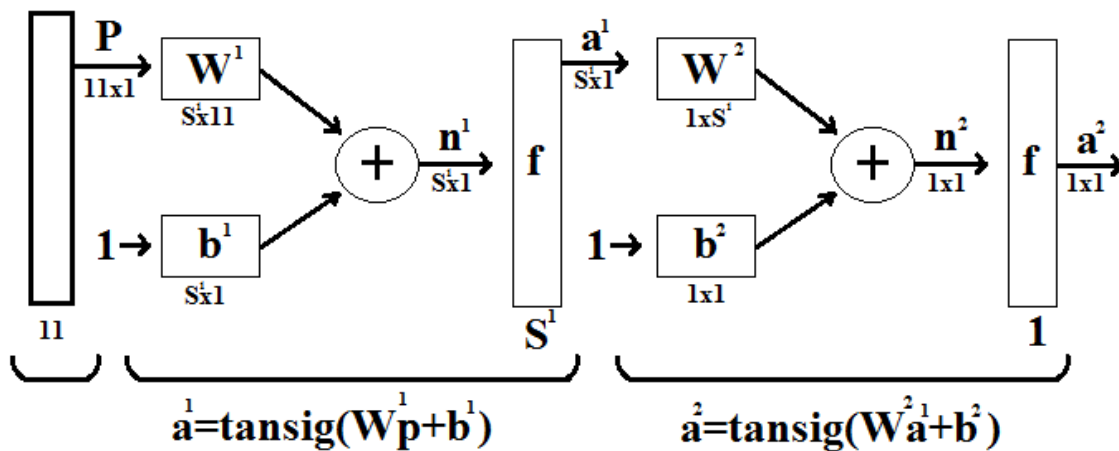


Figura 6.2 Arquitectura

6.3 Conclusiones del trabajo

Al desarrollar una arquitectura para la red neuronal de retropropagación con la utilización de las herramientas de Matlab y la normalización de los parámetros se realizó un procedimiento para identificar la señal ultrasónica con lo que se logró hacer una matriz de entrada para la red. Con el entrenamiento de la red se pudo obtener gráficas que sirven para observar cuáles serán los nuevos valores que den un resultado más exacto a la “Y” deseada, esto da como resultados que con un factor de aprendizaje de 0.1, con un error determinado de 0.06 y después de 321641 iteraciones se pudo conseguir los valores nuevos y de esta forma poder concluir que podemos identificar los ángulos del ultrasonido por medio de los valores que se consiguieron con la red de retropropagación.

6.4 Referencias y Bibliografía

- [1] Guilherme Cardoso and Jafar Saniie, Senior Member, IEEE. Ultrasonic Data Compression via Parameter Estimation
- [2] Case study 3, Pattern Recognition, Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, Orlando De Jesus Neural Network Design, 2nd Edition, eBook
- [3] Martin T. Hagan, Howard B. Demuth, Mark Hudson Beale, Orlando De Jesus Neural Network Design, 2nd Edition, eBook
- [4] Carlos Rubio, Obdulio Marrero. Fabricación de Transductores Ultrasónicos para Equipos automatizados de inspección de líneas de Tuberías