ESCUELA POLITÉCNICA DEL EJÉRCITO

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

PROYECTO DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERÍA

RECONOCIMIENTO DE CARACTERES DE UNA PLACA DE AUTOMÓVIL MEDIANTE REDES NEURONALES ARTIFICIALES UTILIZANDO MATLAB.

JONATHAN LEONARDO OCAMPO CARRIÓN

SANGOLQUÍ – ECUADOR

2011

CERTIFICACIÓN

Certificamos que el presente proyecto de grado titulado “Reconocimiento de Caracteres de una Placa de Automóvil Mediante Redes Neuronales Artificiales Utilizando Matlab “, ha sido desarrollado en su totalidad por el Sr. Jonathan Leonardo Ocampo Carrión, como requisito previo para la obtención del título de Ingeniero Electrónico.

Sangolquí, diciembre del 2011.

Ing. Víctor Proaño	Ing. Paul Bernal
DIRECTOR	CODIRECTOR

RESUMEN

Este proyecto presenta un estudio y diseño de algoritmos necesarios para reconocer los caracteres alfanuméricos presentes en una placa vehicular, estos algoritmos son diseñados tanto en el campo de procesamiento digital de imágenes como en el de redes neuronales artificiales.

Se realiza una breve descripción de los modelos de redes neuronales tipo perceptrón multicapa, base radial y Hopfield, estableciendo los fundamentos teóricos necesarios para la construcción y entrenamiento de dichas redes. Además, se establecen parámetros para el diseño de la arquitectura de cada tipo de red que nos permiten una solución satisfactoria al problema de clasificación de caracteres.

El programa realizado extrae de la imagen la zona perteneciente a la placa basándose en la morfología de la misma, posteriormente se extrae cada uno de los objetos pertenecientes a los caracteres de la placa, para luego ser clasificados por medio de redes neuronales.

Finalmente se realiza un análisis de resultados obtenidos evaluando con dos conjuntos de pruebas cada tipo de red neuronal estableciendo el tipo de red que mejor rendimiento presenta para la solución de este problema. Se diseña también una aplicación de control de acceso vehicular que permite visualizar de mejor manera los resultados obtenidos.

DEDICATORIA

Dedicado a mis padres, hermanos y abuelitas ya que sin su apoyo y paciencia no hubiera sido posible terminar mi carrera.

AGRADECIMIENTO

A Dios, por bendecirme todos los días de mi vida, por darme salud, fortaleza y paciencia para terminar este proyecto. Por obsequiarme la presencia de las personas que día a día dan felicidad a mi vida.

A mi madrecita del cielo, la virgencita del Cisne, por cuidarme todo este tiempo lejos de mi hogar.

A mis padres Paúl y Wilma, por darme la vida y estar siempre brindándome su apoyo incondicionalmente, es especial a mi madre por ser una mujer excepcional, que me ha enseñado a luchar y salir adelante en todo lo que me he propuesto.

A mis hermanos Yomayra, Angelette y Paulo, por ser mi razón para superarme, teniendo como objetivo ser un buen ejemplo para ellos.

A mis abuelitas Mariana y Mercedes, por su cariño brindado sin pedir nada a cambio.

A los ingenieros Paul Bernal y Víctor Proaño por su tiempo y conocimientos aportados para la culminación de este proyecto.

A mis amigos y compañeros a lo largo de estos años de estudios, en especial a mis amigos de la residencia politécnica fueron como mi familia durante todo este tiempo.

PROLOGO

En la actualidad en nuestro país no se cuenta con sistemas de reconocimiento de placas vehiculares, siendo esta una de las principales necesidades existentes para la realización de un control vehicular más rápido y eficaz.

En este proyecto se presenta un conjunto de herramientas de software que permitirán obtener de una forma más rápida y menos complicada el número de placa del vehículo en varias aplicaciones de control vehicular como son: control de accesos, control de parqueaderos, control de tele peaje, etc. Estas herramientas incluyen algoritmos de procesamiento digital de señales y redes neuronales artificiales realizándose una comparación entre tres diferentes tipos de redes neuronales.

Al final se realiza un análisis de resultados obtenidos aplicando cada uno de los tipos de redes al reconocimiento de caracteres, determinando cual tipo de red neuronal presenta un mejor rendimiento para esta problemática.

INDICE DE CONTENIDO Pags.

INDICE DE FIGURAS

1. CAPÍTULO 1

INTRODUCCIÓN

1.1. ANTECEDENTES

En la actualidad con el incremento abrupto del parque automotor en nuestro país, se vuelve de vital importancia el desarrollo de aplicaciones que permitan controlar el flujo de tráfico de manera rápida y eficiente. Esto puede ser realizado por un operario humano, o por un equipo especial inteligente el cual sea capaz de reconocer vehículos por su número de placa. Estos sistemas son denominados de Reconocimiento Automático de Números de Placas (ANPR), por sus siglas en inglés "Automatic Number Plate Recognition"; los mismos que forman parte de varios campos de investigación: procesamiento de imágenes, reconocimiento de patrones y redes neuronales.

Varias técnicas de reconocimiento han sido desarrolladas y los sistemas de reconocimientos de placas son hoy en día usados en varias aplicaciones para control de tráfico y seguridad, tales como parqueamientos, control de acceso, en fronteras, control de tele peajes, rastreo de autos robados. Actualmente en algunos países existen sistemas ANPR instalados en fronteras que automáticamente detectan y monitorean los autos que cruzan por ellas. Cada vehículo puede ser registrado en una base de datos central y comparada con una lista de vehículos robados o que tengan problemas con la ley.

Otra de sus principales aplicaciones se da en los parqueaderos, los números de placa son usados para calcular el tiempo que el automóvil se mantuvo parqueado. Cuando el vehículo ingresa por la puerta de acceso, el número de placa es automáticamente reconocido y es almacenado en una base de datos. Cuando el vehículo más tarde sale del área de parqueo a través de la puerta de salida, el número de placa es reconocido

nuevamente y es emparejado con el anteriormente registrado, permitiendo así el cálculo de la tarifa de parqueamiento una vez obtenida la diferencia de tiempo entre capturas de las imágenes. Los sistemas de reconocimiento automático de placas pueden ser usados también en control de accesos. Por ejemplo en compañías o urbanizaciones privadas para garantizar el acceso de vehículos autorizados.

En el presente proyecto se realizará el estudio de Reconocimiento de Caracteres de la Placa mediante el uso varios modelos de redes neuronales apoyándose además en herramientas de adquisición y procesamiento digital de imágenes presentes en el software MATLAB®. Además se realizara una interfaz de usuario para el uso de la aplicación. Una vez desarrollado el proceso con cada uno de los modelos se procederá a realizar un análisis de eficiencia de cada uno de los modelos planteados. Posteriormente se realizará el almacenamiento de la placa en una base de datos para su posterior tratamiento.

Se debe recalcar que para el presente proyecto se utilizó imágenes capturadas mediante varias cámaras de diferente marca, manteniendo la misma resolución de las imágenes en cada una de ellas.

1.2. JUSTIFICACIÓN

En la actualidad en nuestro país no se cuenta con sistemas de reconocimiento de placas vehiculares, siendo esta una de las principales necesidades existentes para la realización de un control vehicular más rápido y eficaz.

Esta investigación permitirá obtener de una forma más rápida y menos complicada el número de placa del vehículo en varias aplicaciones como son: control de accesos, control de parqueaderos, control de tele peajes con optimización de recursos para este fin.

1.3. OBJETIVOS

El objetivo principal de esta investigación es el diseño de una aplicación en Matlab la cual permita el reconocimiento de caracteres de una placa vehicular utilizando redes

neuronales artificiales y la posterior conexión a una base de datos para el almacenamiento y obtención de datos afines al número de placa procesado, la cual también posea una alta usabilidad para el usuario que la manipule. Con el propósito de alcanzar estos objetivos se procederá a simular el comportamiento de varios modelos de redes neuronales y se evaluará su eficiencia en el reconocimiento de caracteres.

1.4. ESTRUCTURA DEL PROYECTO

En el Capítulo 2 se encuentra una descripción de los conceptos teóricos utilizados para el reconocimiento de patrones mediante el uso de redes neuronales en Matlab®, se incluye conceptos importantes de procesamiento digital de imágenes y redes neuronales utilizados en este proyecto.

En el Capítulo 3 se incluye procesamiento digital de la imagen capturada necesario para la obtención de cada uno de los caracteres de la placa; además se realiza el diseño de los diferentes modelos de redes neuronales utilizados como clasificadores en el proyecto como son: perceptrón multicapa, de base radial y Hopfield. En este capítulo también se incluye el diseño de una pequeña aplicación de un sistema de control de acceso y su conexión a una base de datos central.

El Capítulo 4 presenta un análisis de las pruebas realizadas con cada modelo de red neuronal, estableciendo su eficiencia de acuerdo a los resultados obtenidos.

En el Capítulo 5 se encuentra las conclusiones con respecto a cada modelo de red neuronal y se definen posibles líneas de mejora para trabajos futuros.

En la Figura 1.1 se muestra un diagrama general del trabajo a realizarse a lo largo de este proyecto:

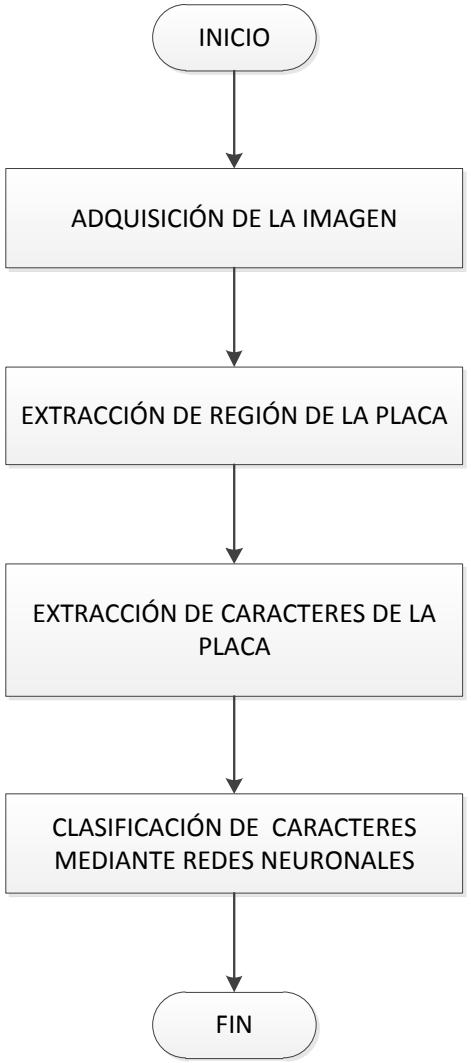


Figura.1. 1. Diagrama General del Proyecto

2. CAPITULO 2

RECONOCIMIENTOS DE PATRONES MEDIANTE REDES NEURONALES EN MATLAB®

2.1. PROCESAMIENTO DIGITAL DE IMÁGENES EN MATLAB®

El procesamiento digital de imágenes (PDI) se refiere a la manipulación de imágenes del mundo real de manera digital por medio de un computador. Su principal objetivo es el mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertas características relevantes para la aplicación que se desee desarrollar. La *Visión Artificial* se define como el conjunto de procesos para la adquisición, caracterización e interpretación de imágenes[1]. Estos procesos pueden ser subdivididos en cinco áreas:

La captura o adquisición.- Es el proceso a través del cual se obtiene una imagen digital utilizando un dispositivo de captura como una cámara digital, video cámara, escáner, etc.

El preprocesamiento.- Incluye técnicas que permiten mejorar la calidad de la imagen tales como la reducción del ruido, realce del contraste, realce de ciertos detalles, o características de la imagen.

La segmentación.- Es el proceso mediante el cual se divide la imagen en regiones importantes para los intereses de la aplicación.

---

[1] Santillan, Ivan Danilo Garcia. Vision Artificial y Procesamiento Digital de Imagenes usando matlab. Ibarra : Pontificia Universidad Catolica del Ecuador, 2008.

La descripción.- Es el proceso que obtiene características de los objetos dentro de una imagen, las cuales permitirán diferenciar un tipo de objeto de otro, como: la forma, el tamaño, área, etc.

El reconocimiento e interpretación.- Es la etapa en la que se identifica y se asocia un significado a los diferentes objetos previamente segmentados en la imagen.

No todas las aplicaciones requieren de todos los procesos y depende básicamente de la complejidad del problema que se va a resolver. Los resultados obtenidos en este tipo de aplicaciones dependen de la calidad de la imagen original, por lo que se deben tomar todas las precauciones necesarias como tener una iluminación adecuada y uniforme en el momento de su adquisición. Algunos campos de aplicación son: la medicina, astronomía, industria, control vehicular, etc.

El software MATLAB® es un entorno de desarrollo de aplicaciones que integra análisis numérico, cálculo matricial, procesamiento de señales y visualización gráfica en un entorno completo.

Además, MATLAB® dispone de una amplia gama de colecciones de funciones especializadas, denominados *Toolbox*, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos *Toolboxes* cubren casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos: procesamiento de imágenes, procesamiento de señales, control robusto, estadística, análisis financiero, matemática simbólica, redes neuronales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, Simulink, etc.

2.1.1. TOOLBOX DE PROCESAMIENTO DE IMAGENES

Este Toolbox proporciona a MATLAB® un conjunto de funciones que amplían las capacidades del software para el desarrollo de aplicaciones y de nuevos algoritmos en el campo del procesamiento y análisis de imágenes. Entre las operaciones más importantes que MATLAB® soporta están:

- Transformaciones espaciales de imágenes
- Diseño de filtros y recuperación de imágenes.
- Mejora de imágenes.
- Operaciones morfológicas.
- Definición de mapas de colores y modificación gráfica.
- Operaciones morfológicas.

El toolbox de Procesamiento de Imágenes maneja cuatro tipos de imágenes básicos: imágenes indexadas, imágenes con intensidad (niveles de grises), imágenes binarias e imágenes RGB; además soporta los siguientes formatos de imagen: JPEG, TIFF, GIF, BMP, PNG, HDF, PCX, XWD, ICO y CUR.[2]

MATLAB® almacena la mayoría de las imágenes como arreglos bidimensionales (matrices) en los cuales cada elemento de la matriz corresponde a la intensidad de un píxel de la imagen.

2.1.2. DESCRIPCIÓN DE OPERACIONES UTILIZADAS PARA EL PROCESAMIENTO DIGITAL DE LA IMAGEN EN MATLAB®

2.1.2.1.OPERACIONES MORFOLÓGICAS

La Morfología matemática es una técnica de procesado no lineal de imágenes, interesada en la geometría de los objetos. Las operaciones morfológicas son operaciones sobre imágenes basadas en formas o estructuras. Estas operaciones toman como entrada una imagen regresando como resultado una imagen del mismo tipo modificada, el valor de cada píxel de la imagen resultado es basado en el valor del correspondiente píxel de la imagen original y de sus vecinos. Entre sus principales aplicaciones están la eliminación de ruido y segmentación de objetos.

[2] MathWorks. Image Processing Toolbox. [Citado el: 15 de Agosto de 2011.] http://www.mathworks.com/help/toolbox/images/.

Existe tres tipos de morfología: Morfología binaria, Morfología de niveles de gris y Morfología de imágenes poli cromáticas. En esta sección se muestra las herramientas utilizadas en Morfología binaria y de niveles de gris.

Elementos del procesamiento morfológico. Los fundamentos del análisis y procesado morfológico se basan en el álgebra de conjuntos. Existen tres elementos en el proceso:

a) Conjuntos (Imágenes).-En una imagen binaria los conjuntos están formados por puntos en un espacio 2D, donde cada elemento es un punto de coordenadas (x,y) con un valor de cero (fondo) o uno (primer plano).

En una imagen de niveles de gris los elementos que forman los conjuntos se encuentran en un espacio 3D, donde los dos primeros componentes de cada elemento se refieren a las coordenadas de cada pixel y el tercer componente está relacionado con la intensidad del mismo.

b) Elemento Estructural.- El elemento estructural (EE) es un conjunto de puntos que forman una pequeña imagen que determinará la estructura de la imagen sobre la que se aplicará la operación morfológica[3]. Este será el responsable de la forma y el tamaño de los objetos que forman la imagen. El EE puede tener cualquier tamaño y forma (horizontal, vertical, cuadrado, circular, etc.). En Matlab® se lo implementa a través de la función **strel,** el cual crea un elemento de estructura morfológica.

c) Operaciones Morfológicas

Dilatación

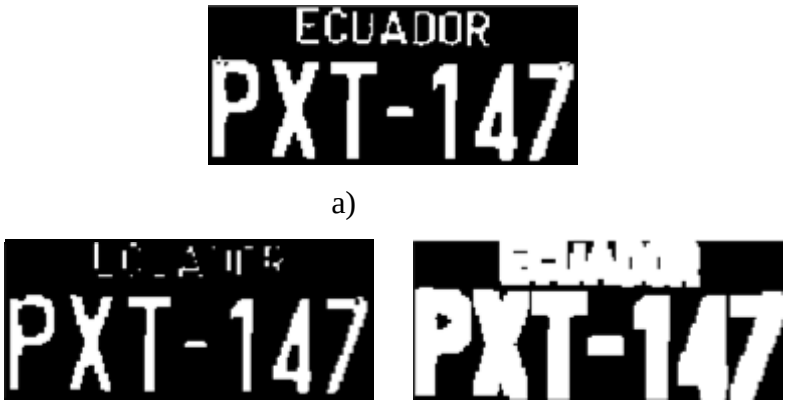
La dilatación expande los píxeles de los objetos presentes en la imagen sobre la que se aplica. En Matlab® se lo implementa mediante la función ***imdilate***.

[3]Elementos del Procesado Morfologico.[Citado el: 10 de Octubre de 2011.] http://www.tsc.uc3m.es/imagine/Curso_ProcesadoMorfologico/Contenido/Elementos/Elementos.html#operadoresMorfologicos\.

Erosión

La erosión es la operación opuesta a la dilatación, produce un efecto de encogimiento, reducción o contracción de los objetos de la imagen.
Su implementación en Matlab® se la realiza mediante la función ***imerode***.

En la Figura 2.1 se observa la aplicación de las operaciones morfológicas de e*rosión* (b) y *dilatación* (c) aplicadas a la imagen de una placa vehicular.



a)

b) c)

Figura.2. 1. Operaciones Morfológicas: a) Imagen original; b) Operación de Erosión; c) Operación de Dilatación.

Apertura (Opening)

Realiza una erosión seguida de una dilatación, utilizando el mismo elemento estructural en ambas operaciones. Este método se aplica cuando se desea eliminar los pequeños objetos y mantener el tamaño en los grandes (eliminar ruido). En Matlab® se lo implementa mediante la función ***imopen***.

Cierre (Closing)

Dilatación seguida de una erosión. Este proceso se caracteriza por rellenar huecos y conectar objetos que están próximos entre sí. Su implementación en Matlab® se la realiza mediante la función ***imclose***.
En la Figura 2.2 se muestra el resultado de realizar operaciones de cierre (b) y apertura (c).

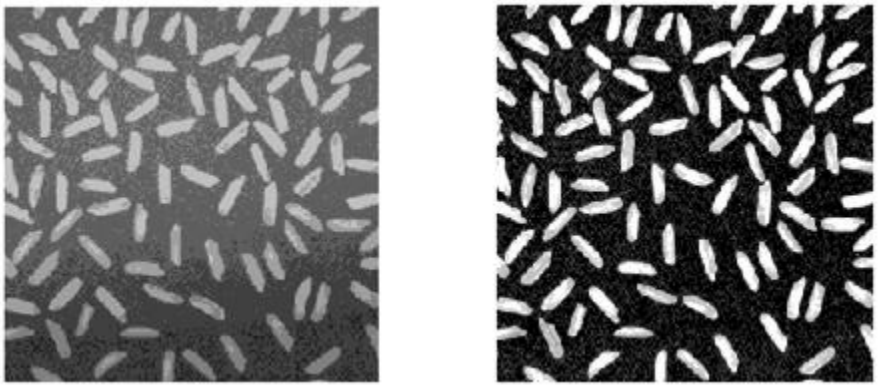
a) b) c)

Figura.2. 2. Operaciones morfológicas: a) Imagen Original; b) Operación de cierre; c) Operación de apertura.

2.1.2.2.FILTROS MORFOLÓGICOS

Los filtros morfológicos se realizan sobre una imagen en escala de grises o binaria usando un elemento estructural *EE* y resaltan objetos de color contrario al fondo. Existen dos tipos de filtros:

- *Positivo (white top-hat):* Esta operación es útil para resaltar detalles en la presencia de sombras. En la Figura 2.3 se muestra como se resaltan los pixeles más claros sobre los oscuros.



a) b)

Figura.2. 3. Filtro morfológico: a) Imagen Original; b) Filtro Top-Hat

- *Negativo (black top-hat o bottom-hat):* Su aplicación es fundamentalmente para que resalte detalles oscuros sobre un fondo local blanco. El resultado de este filtro se puede apreciar en la Figura 2.4.

a) b)

Figura.2. 4. Filtro morfológico: a) Imagen Original; b) Filtro Bottom-Hat

2.1.2.3.FILTRADO ESPACIAL

El filtrado espacial es una técnica para modificar o mejorar a una imagen, resaltando o atenuando características de la misma. El filtrado es una operación de vecindario, en la cual el valor de un píxel dado en la imagen procesada se calcula mediante algún algoritmo que toma en cuenta los valores de los píxeles de la vecindad de la imagen original. De esta forma es necesario configurar una matriz (mascara, kernel, ventana) que considere cuales vecinos y en qué forma influirán en la determinación del nuevo pixel. Existen de dos clases: Filtros de paso bajo y filtros de paso alto.[4]

- **Filtros espaciales de paso bajo (Suavizantes)**

Los filtros suavizantes se emplean para hacer que la imagen aparezca algo borrosa y también para reducir el ruido. Se tiene dos subtipos principales:

- **Filtro de Promedio**.- Se basa en el promediado de los pixeles adyacentes al pixel que se evalúa tomando en cuenta máscaras de vecindad cuadráticas. Su implementación en Matlab® se la realiza mediante las funciones ***fspecial*** e ***imfilter***. La Figura 2.5 (c) muestra la imagen con el filtro de promedio.[5]

[4] **Santillan, Ivan Danilo Garcia.** *Vision Artificial y Procesamiento Digital de Imagenes usando matlab.* Ibarra : Pontificia Universidad Catolica del Ecuador, 2008.

[5] Blanchet, Gérard y Charbit, Maurice . *Digital Signal and Image Processing using MATLAB.* Londres : ISTE Ltd., 2006.

- **Filtro de la mediana**.- Éste filtro se basa en sustituir el valor de un píxel por el de la mediana del conjunto formado por el mismo y sus ocho vecinos, se suele utilizar en el procesamiento de imágenes para reducir el ruido "sal y pimienta". En Matlab® este filtro se encuentra implementado en la función ***medfilt2***. La Figura 2.5 (b) muestra la imagen con el filtro de mediana.

a) b) c)

Figura.2. 5. Filtros Pasa Bajo; a) Imagen con ruido, b) Resultado de aplicar filtro de Mediana, c) Resultado de aplicar filtro de promedio

- **Filtros espaciales de paso alto**

Los filtros espaciales de paso alto se corresponden con las altas frecuencias, suelen corresponder a los bordes de los objetos presentes en las imágenes. Se pueden utilizar para el realce de bordes o para la detección de contornos.

- **Realce de bordes.** consiste en resaltar aquellos píxeles que tienen un valor de gris diferente al de sus vecinos. Si la imagen contiene ruido, su efecto se multiplicará, por lo que primero se debe eliminar el ruido. Su implementación en Matlab® se consigue con el uso de la función ***filter2.*** En la Figura 2.6 se muestra un ejemplo del efecto causado por esta operación.

a) b)

Figura.2. 6. Filtro Pasa Alto; a) Imagen original, b) Imagen con los bordes realzados

- **Detección de contornos** es un paso intermedio en el reconocimiento de patrones en imágenes digitales. En una imagen, los contornos corresponden a los límites de los objetos presentes en la imagen. Para hallar los contornos se buscan los lugares en la imagen en los que la intensidad del píxel cambia rápidamente. Su implementación en Matlab® se realiza usando la función ***edge*** y se aplica a imágenes de niveles de gris. Se pueden usar varios métodos para la detección de contornos, entre ellos están: Sobel, Prewitt, Robert o Canny. En la Figura 2.7 se muestra un ejemplo de detección de contornos.

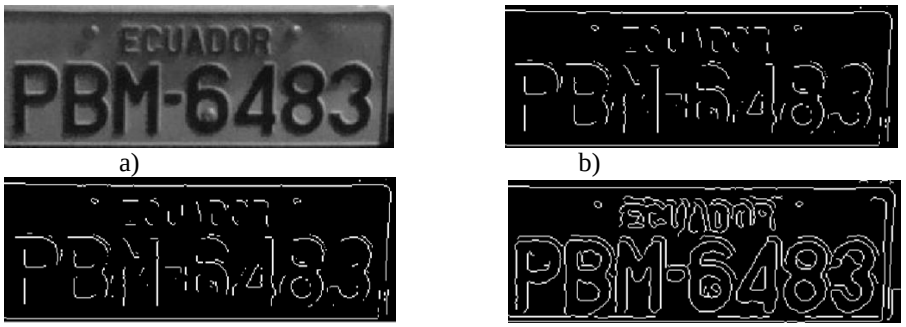


Figura.2. 7. Detección de Contornos; a)Imagen original, b) Método de Prewitt, c) Método de Sobel, d) Método de Canny.

2.1.2.4.DETECCIÓN Y CORRECCIÓN DE ÁNGULO

Algunas ocasiones las imágenes obtenidas pueden estar rotadas de varias formas debido a la posición del objeto a ser capturado respecto a la cámara. Un método ampliamente utilizado para la detección del ángulo de rotación es la transformada de Hough.

La transformada de Hough es una técnica de extracción de características usada en el procesamiento de imágenes digitales. La transformada clásica identifica líneas en la imagen, pero se ha extendido para identificar la posición de imágenes arbitrarias. El éxito de esta transformada (en su forma más general) se debe a que permite encontrar una forma

aunque esta haya sido ampliada, reducida y/o rotada en un tiempo computacionalmente aceptable.[6]

La transformada clásica transforma cada punto en pares de coordenadas $[\theta, \rho]$ que representan líneas (la cantidad de líneas a las que se transforma depende de las necesidades particulares) que pasan por ese punto.

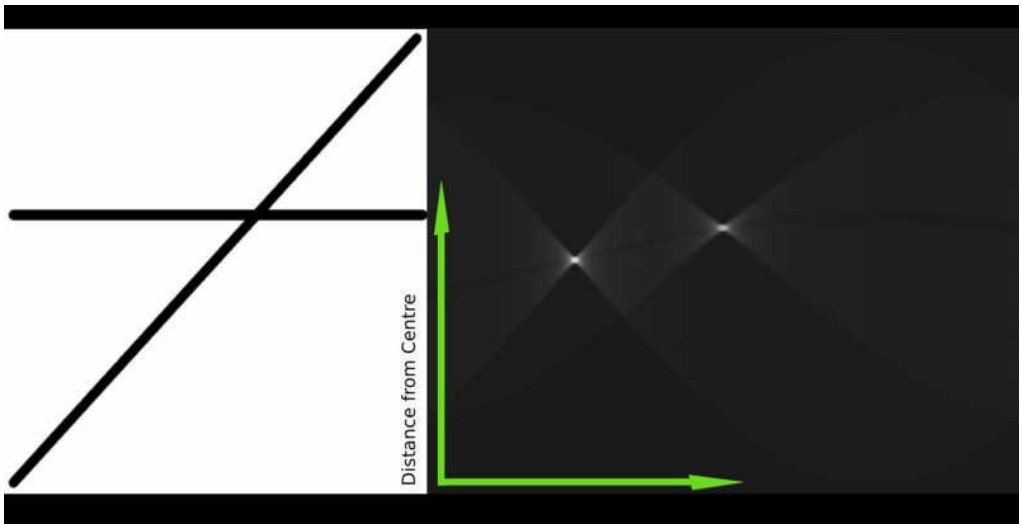


Figura.2. 8. Transformada de Hough; izquierda: Espacio de la Imagen; Derecha: Espacio de Transformada de Hough

La Figura 2.8 muestra el resultado de aplicar la transformada de Hough a una imagen compuesta por dos líneas gruesas. Los ejes de la imagen de la derecha representan "distancia desde el origen" en el eje vertical y "ángulo con respecto al eje X positivo" en el horizontal.

Cada punto de la imagen de la izquierda fue transformado en varias líneas, cada una de estas líneas es un punto en la imagen de la derecha, las líneas que pasan por pocos puntos quedan de un color gris oscuro, mientras que las que pasan por más puntos son más

[6] Gonzales, Rafael, Woods, Richard y Eddins, Steven. *Digital Image Processing Using MATLAB.* New Jersey : Prentice Hall, 2009.

brillantes. Los dos puntos más brillantes que aparecen en la imagen transformada representan a las dos líneas de la imagen original.[7]

En resumen la transformada de Hough transforma una línea en el espacio de la imagen en un punto $[\rho, \theta]$, denominado pico; en el espacio de la transformada.

2.1.2.5.SEGMENTACIÓN

La segmentación es uno de los procesos más importantes y complejos en el reconocimiento de placas; si la segmentación falla, un carácter puede ser inapropiadamente dividido en dos piezas, o dos caracteres pueden ser unidos por error. La segmentación consiste en dividir la imagen digital en regiones homogéneas o similares con respecto a una o más características (brillo, color, tamaño, longitud, área, forma, etc.) con el fin de facilitar su posterior análisis y reconocimiento.

No existe un método universal de segmentación. Su proceso está ligado al tipo de tarea que se desea resolver y termina cuando satisface los objetivos del observador, es decir, cuando se hayan detectado todos los objetos de interés para una aplicación específica. Existen 2 tipos de segmentación que son:

- **Segmentación basada en cálculo del Umbral**

  Segmentación basada en cálculo del Umbral es un proceso que permite convertir una imagen de niveles de gris o color en una en imagen binaria, de tal forma que los objetos de interés se etiqueten con un valor distinto de los píxeles del fondo. Es una técnica rápida y su costo computacional es bajo. El método de mayor confiabilidad es el método de Otsu[8] el cual permite encontrar el umbral óptimo para la segmentación de la imagen a segmentar. Matlab® posee la función ***graythresh***, que calcula el umbral de la imagen global utilizando el método de Otsu.

---

[7] Procesamiento Global empleando la Transformada de Hough. Universidad nacional de Quilmes, 2005. [Citado el: 3 de Septiembre de 2011.] http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/TransformadadeHough.pdf.

[8] Universidad Nacional de Quilmes**.** Segmentacion Por Umbralizacion. Octubre de 2005. [Citado el: 8 de Agosto de 2011.] http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Segmentación por umbralización - Método de Otsu.pdf.

- **Segmentación basada en Regiones**

La segmentación por regiones es utilizada para separar los objetos de interés. En este caso, la imagen es particionada en diferentes regiones, quedándose cada una relacionada en ciertas características y conjuntos de píxeles conectados.[9] Así, a partir de la segmentación de la imagen en regiones, pueden ser realizadas las medidas sobre cada región y las relaciones entre las regiones adyacentes pueden ser establecidas. La segmentación por regiones está implementada en Matlab® a través de los siguientes comandos:

- ✓ ***Bwlabel:*** Etiqueta los componentes conectados en una imagen binaria.

- ✓ ***Regionprops*****:** Mide las propiedades de las regiones etiquetadas previamente en una imagen. Permite obtener valores de propiedades como: área, caja que cubre la región, centroide, excentricidad, etc.

2.2. TEORÍA DE REDES NEURONALES ARTIFICIALES

El cerebro humano es el sistema de cálculo más complejo conocido por el hombre. El ordenador y el hombre realizan bien diferentes clases de tareas; así la operación de reconocer la placa de un vehículo resulta una tarea relativamente sencilla para el hombre y difícil para el ordenador, mientras que la contabilidad de una empresa es tarea costosa para un experto contable y una sencilla rutina para un ordenador básico[10]. La capacidad del cerebro humano de pensar, recordar y resolver problemas ha inspirado a muchos científicos intentar o procurar modelar en el ordenador el funcionamiento del cerebro humano.

---

[9] Martinez, E. *Procesamiento Digital de Imagenes.* Mexico D. F. : Universidad Nacional Autonoma de Mexico, 2005.

[10] Moreno, Gustavo Adolfo. *Diseño e Implementacion de un Sistema Basado en una Red Neuronal no Supervisada para el Control de Moviminetos de un Robot Movil.* Quito : Escuela Politecnica del Ejercito, 2005.

En términos más generales las redes neuronales son herramientas para el modelado de datos, no lineales y estadísticas; también pueden ser usadas para modelar relaciones complejas entre entradas y salidas o para buscar patrones.

2.2.1. REDES NEURONALES BIOLÓGICAS.

El cerebro biológico está formado de muchas neuronas conectadas entre sí utilizando la información recibida para dar una respuesta a cada situación. Desde un punto de vista funcional las neuronas constituyen procesadores de información sencillos altamente interconectados funcionando paralelamente.[11] En la Figura 2.9 se puede observar un esquema de una neurona biológica con todas sus partes.

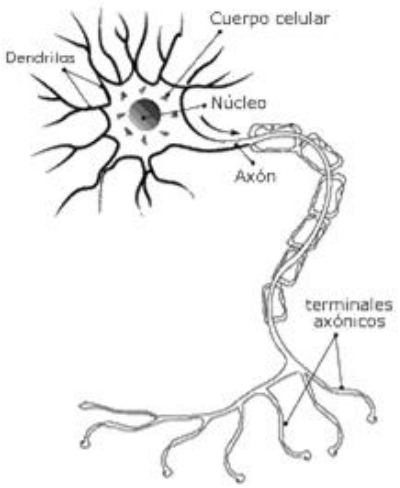


Figura.2. 9. Neurona biológica

En el esquema se observa que la neurona consta de un cuerpo celular y un núcleo, idéntico al resto de células existentes en el organismo, sin embargo cuenta con varios elementos específicos de ella. En primer lugar se observa el Axón, el cual es una ramificación de salida de las neuronas, la cual a su vez ramifica en un gran número de terminales axónicos, encargados de emitir impulsos hacia las demás neuronas. Desde el cuerpo celular se extienden ramificaciones cortas denominadas dendritas que son estructuras de entrada que permiten recibir las señales de otras neuronas y propagarlas al interior de la misma.

[11] Sanz Molina, Alfredo y del Brio, Bonifacio Martin. *Redes Neuronales y Sistemas Difusos.* Mexico, DF : Alfaomega, 2002.

La conexión entre el axón de una neurona y las dendritas de otra es conocida como sinapsis. A través de la sinapsis la neurona recoge información electro-química procedente de las células vecinas a las que la neurona está conectada; esta información llega al núcleo en donde es procesada hasta generar una respuesta que es propagada a través del axón.

2.2.2. DEFINICIÓN DE RED NEURONAL ARTIFICIAL

"Una red neuronal artificial es un procesador masivo paralelo formado por unidades simples de procesamiento que tienen una propensión natural para almacenar conocimiento experimental, haciéndolo viable para su uso. La red neuronal se asemeja al cerebro en dos aspectos:

1. *El conocimiento es adquirido por la red desde su entorno a través de un proceso de aprendizaje.*
2. *Las fuerzas de las conexiones entre neuronas, conocidas como pesos sinápticos, son usadas para almacenar el conocimiento."*[12]

Las redes neuronales artificiales (RNA) son un conjunto unidades de procesamiento elementales con una arquitectura definida que tratan de asemejar el funcionamiento de las redes neuronales biológicas. En la Figura 2.10 se muestra un diagrama general de una red neuronal artificial.

[12] Haykin, Simon. *Neural Networks and Learning Machines.* New Jersey : Prentice Hall, 2009.

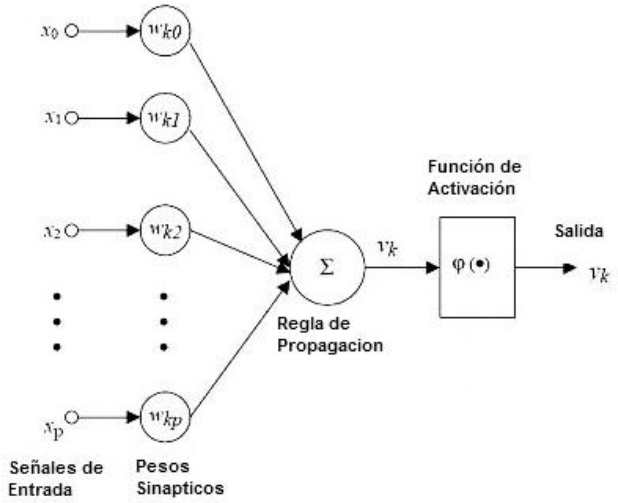


Figura.2. 10. Red neuronal artificial

Su unidad de procesamiento es la neurona artificial, la cual está constituida por varios elementos como son:

- Conjunto de elementos de entrada.- Señales de entrada, pueden ser discretas o continuas.

- Pesos sinápticos.- Representan la intensidad de interacción en una sinapsis. Estos pesos pueden ser variables o fijos.

- Regla de propagación.- Permite obtener a partir de las entradas y los pesos sinápticos el valor del potencial post-sináptico.

- Función de activación.- Llamada también función de trasferencia, proporciona el estado actual de activación en función de su estado anterior y el valor post-sináptico para posteriormente entregar la salida global de la neurona. Las funciones de activación pueden ser de distintas formas como se puede apreciar en la Figura 2.11

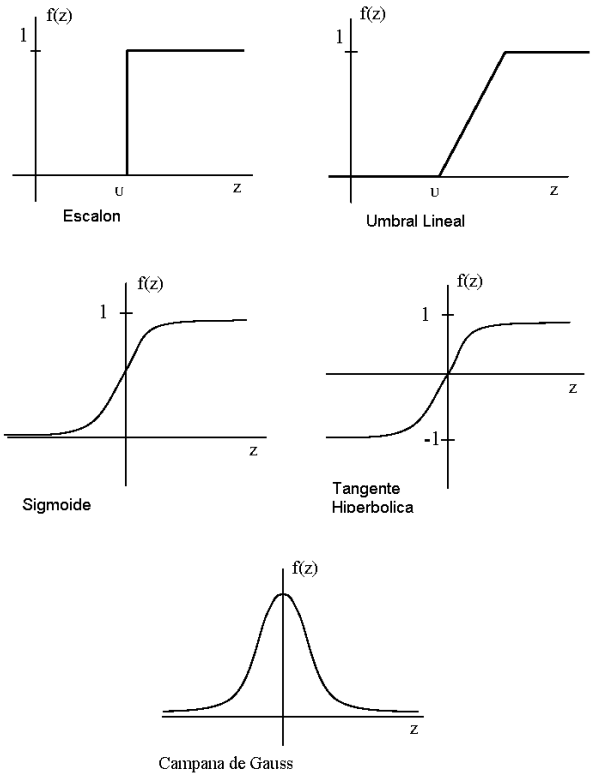


Figura.2. 11. Distintas formas de funciones de activación

2.2.3. CARACTERÍSTICAS DE LAS REDES NEURONALES ARTIFICIALES.

Existen cuatro aspectos importantes que caracterizan una red neuronal: su arquitectura, el mecanismo de aprendizaje, tipo de asociación entre la información de entrada y de salida, y la forma de representación de estas informaciones.[13]

2.2.3.1.Arquitectura de Red

Consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas. Los parámetros fundamentales son: número de capas, número de neuronas por capa, grado de conectividad y tipo de conexión entre neuronas. La clasificación de las RNAs según su arquitectura es la siguiente:

---

[13] Villanueva Espinoza, María del Rosario. Las Redes Neuronales Artificiales y su Importancia como Herramienta en la Toma de Desiciones.2002. http://sisbib.unmsm.edu.pe/Bibvirtual/tesis/Basic/Villanueva_EM/Contenido.htm.

1. **Red monocapa**: se establecen conexiones laterales entre las neuronas que pertenecen a la única capa que constituye la red. Las redes monocapa se utilizan típicamente en tareas relacionadas con lo que se conoce como auto-asociación; por ejemplo, para regenerar informaciones de entrada que se presenta como incompleta o distorsionada.

2. **Redes multicapa**: Las redes multicapa se forman con un grupo de capas simples en cascada. La salida de una capa es la entrada de la siguiente capa. Se ha demostrado que las redes multicapa presentan cualidades y aspectos por encima de las redes de una capa simple. Dado que este tipo de redes disponen de varias capas, las conexiones entre neuronas pueden ser del tipo *feedforward* (conexión hacia adelante) o del tipo *feedback* (conexión hacia atrás). Las redes *feedforward* son especialmente útiles en aplicaciones de reconocimiento o clasificación de patrones. En la Figura 2.12 se muestra la estructura general de una red de este tipo.

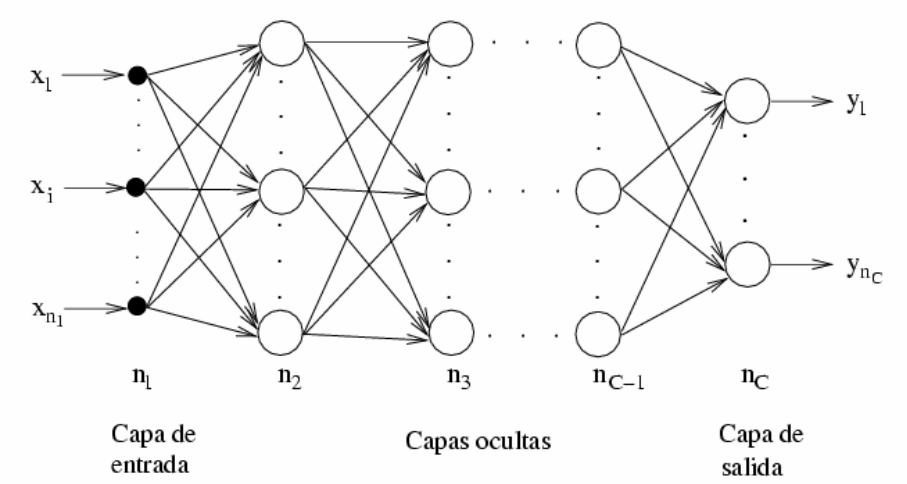


Figura.2. 12. Estructura general de red multicapa

La mayoría de redes multicapa son bicapa. Este tipo de estructura es particularmente adecuada para realizar una asociación de una información o patrón de entrada con otra información de salida en la segunda capa.

2.2.3.2.Mecanismo de aprendizaje

Los modelos neuronales utilizan varios algoritmos de estimación, aprendizaje o entrenamiento para encontrar los valores de los pesos sinápticos.

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante la etapa de aprendizaje se reducen a la destrucción (el peso de la conexión toma el valor 0), modificación y creación (el peso de la conexión toma un valor distinto de 0) de conexiones entre las neuronas. La finalización del periodo de aprendizaje se puede determinar mediante:

- Numero de ciclos predeterminado. Se decide con antelación cuantas veces será introducido todo el conjunto, una vez superado dicho número se detiene el proceso de aprendizaje y se da por acertada la red resultante.

- El error es menor a una cantidad preestablecida. Se debe definir primeramente una función de error. Se decide un valor aceptable para dicho error, y solo se detiene el proceso de aprendizaje cuando la red produzca un valor de error por debajo del prefijado.

- La modificación de los pesos sea irrelevante. Cuando en el proceso de aprendizaje se llegue a un punto en que ya no se producen variaciones de los pesos de ninguna conexión.

El entrenamiento se realiza mediante patrones ejemplos, siendo dos los tipos de aprendizaje: supervisado y no supervisado.

1. **Aprendizaje Supervisado.** Este tipo de aprendizaje se caracteriza porque el proceso se realiza con el control de un agente externo que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor comprueba la salida generada por el sistema y en caso de que no coincida con la esperada, procederá a

modificar los pesos de sinápticos de la red. En la Figura 2.13 se muestra un diagrama de aprendizaje supervisado.

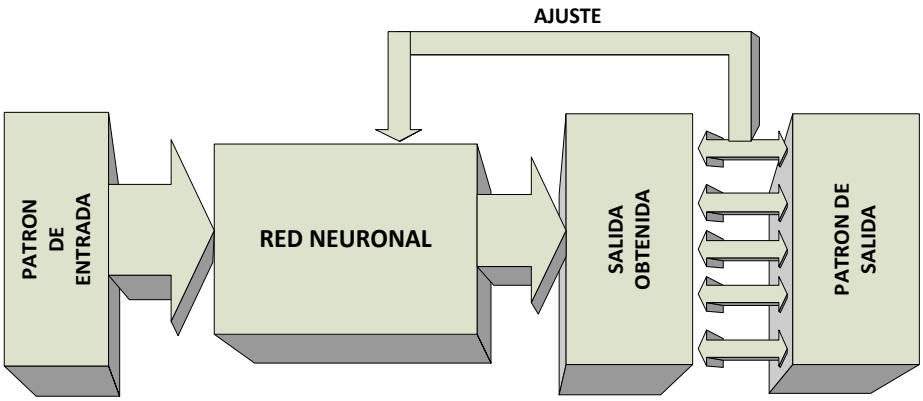


Figura.2. 13. Diagrama de aprendizaje supervisado

El conjunto de aprendizaje debe poseer las siguientes características:

- **Ser Significativo.** Debe haber un número suficiente de patrones ejemplos (del 70 al 80 % del número de muestras total). Si el conjunto de aprendizaje es reducido, la red no será capaz de adaptar sus pesos de forma eficaz.

- **Ser Representativo.** Los componentes del conjunto de aprendizaje deberá ser diversos. Si un conjunto de aprendizaje tiene muchos más ejemplos de un tipo que del resto, la red se especializara en dicho subconjunto de datos y no será de aplicación general.

  En este tipo de aprendizaje se suelen considerar tres formas: Aprendizaje por corrección de error, Aprendizaje por refuerzo y aprendizaje estocástico.

a) Aprendizaje por corrección de error.- Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida, es decir en función del error cometido.

b) Aprendizaje por refuerzo.- Es más lento que el aprendizaje anterior, que se basa en no disponer de un ejemplo completo de comportamiento deseado; es decir, no indicar exactamente la salida que se desea que proporcione la red ante determinada entrada. La función del supervisor se reduce a indicar mediante una señal de refuerzo si la salida obtenida se ajusta a la deseada (éxito= +1 o fracaso=-1), y en función de ello se ajustan los pesos.

c) Aprendizaje estocástico.- Este aprendizaje consiste en realizar cambios aleatorios a los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

2. Aprendizaje no supervisado. Este tipo de aprendizaje no requiere la influencia externa para ajustar los pesos de las conexiones entre las neuronas. La salida no recibe ninguna información del entorno que le indique que la salida generada en respuesta a una determinada entrada es o no correcta; son capaces de auto-organizarse.

Estas redes deben encontrar las características, regularidades, correlaciones o categorías que se pueden establecer entre los datos de entrada. Existen varias posibilidades en cuanto a la interpretación de las salidas, que dependen de su estructura y del algoritmo de aprendizaje empleado.

En algunos casos, la salida representa el grado de familiaridad o similitud entre las entradas y las informaciones que se le han mostrado hasta entonces. En otros casos podría realizar un establecimiento de categorías, indicando a la salida a que categoría pertenece la información presentada en la entrada, siendo la propia red quien debe encontrar las categorías apropiadas a partir de correlaciones entre las informaciones presentadas. En la Figura 2.14 se muestra el diagrama de aprendizaje no supervisado.

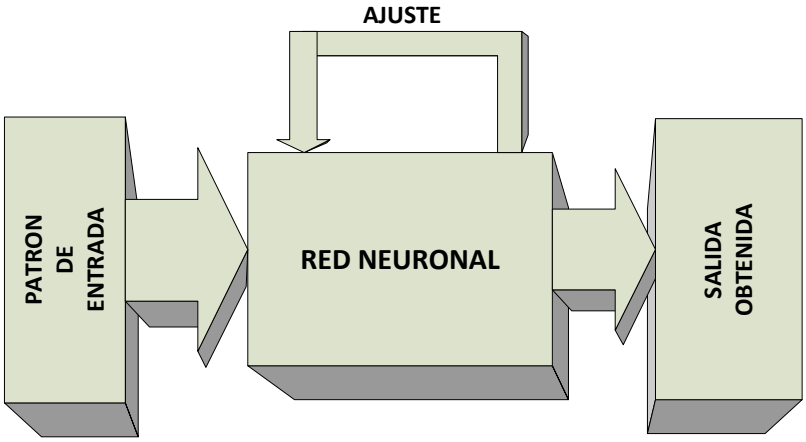


Figura.2. 14. Diagrama de aprendizaje no supervisado

2.2.3.3.Tipo de asociación entre las informaciones de entrada y salida

Las redes neuronales son sistemas que almacenan cierta información aprendida; esta se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas. Hay que establecer cierta relación o asociación entre la información presentada a la red y la salida ofrecida por esta. Es lo que se conoce como memoria asociativa. Existen dos formas de realizar esta asociación entrada/salida y que generan dos tipos de redes:

1. ***Redes hetero asociativas***: La red aprende parejas de datos de tal forma que cuando se le presente determinada información de entrada responda con la salida correspondiente. Al asociar informaciones de entrada con diferentes informaciones de salida, precisan al menos de 2 capas, una para captar y retener la información de entrada y otra para mantener la salida con la información asociada. El aprendizaje de este tipo de redes puede ser con supervisión.

2. ***Redes auto-asociativas***: La red aprende ciertas informaciones de forma que cuando se le presenta una información de entrada realizará una autocorrelación, respondiendo con uno de los datos almacenados, el más parecido al de entrada. Este tipo de redes pueden implementarse con una sola capa de neuronas. El tipo de aprendizaje utilizado habitualmente es el no supervisado y suelen utilizarse en tareas de filtrado de información para la reconstrucción de datos, eliminando distorsiones o ruido, explorar relaciones entre informaciones similares para facilitar

la búsqueda por contenido en bases de datos y para resolver problemas de optimización

2.2.3.4.Representación de la información de entrada y salida

Las RNA pueden representar la información de entrada y de salida de manera diferentes:

Redes continuas: Tanto los datos de entrada como de salida son de naturaleza analógica. En este caso las funciones de activación de las neuronas serán también continuas, del tipo lineal o sigmoidea.

Redes discretas: Redes que sólo admiten valores discretos [0,1] a la entrada, generando también en la salida respuestas de tipo binario. La función de activación en este caso es del tipo escalón.

Redes híbridas: La información de entrada es continua pero a la salida ofrecen información binaria.

2.2.4. TIPOS DE REDES NEURONALES UTILIZADAS EN EL PROYECTO.

Existen dos tipos básicos de redes neuronales: Redes Neuronales *Feedforward* y Redes Neuronales Recurrentes.

2.2.4.1.Redes Neuronales Feedforward

Son redes neuronales no cíclicas, en las cuales los datos fluyen en una sola dirección; desde la entrada hasta la salida, sin posibilidad de retroalimentación. Una vez que la red neuronal es entrenada, su estado es fijo y no se altera cuando se le presentan nuevos datos de entrada.

- ✓ **Red Perceptrón Multicapa**

Es aquella red neuronal que contiene varias capas, una de entrada, una o más capas intermedias ocultas y una capa de salida; en esta red cada neurona de una capa proporciona una entrada para cada una de las neuronas que existen en la siguiente capa.

Es una red de tipo de aprendizaje supervisado ya que se requiere de una salida esperada para poder determinar el error y realizar el aprendizaje; una de sus principales limitaciones en este tipo de red es el largo proceso de aprendizaje para problemas complejos dependientes de un amplio número de variables. El aprendizaje de este modelo de red se lo suele realizar con el algoritmo de retro propagación de errores (*back propagation*).

La habilidad del perceptrón multicapa para aprender a partir de un conjunto de ejemplos, aproximar relaciones no lineales, filtrar ruido en los datos, etc. hace que sea un modelo adecuado para abordar problemas reales, sin que esto indique que son los mejores aproximadores universales. Cada una de las clases de aproximadores tiene sus propias características; se conocen ciertas condiciones bajo las cuales un método es preferible a otro. Serán las consideraciones prácticas de cada problema las que determinen la elección de un aproximador u otro.

Arquitectura del Perceptrón Multicapa

Su arquitectura se caracteriza porque tiene sus neuronas agrupadas en capas de diferentes niveles. Cada capa está formada por un conjunto de neuronas.

Las neuronas de la capa de entrada se encargan únicamente de recibir las señales o patrones que provienen del exterior y propagar dichas señales a todas las neuronas de la siguiente capa. La última capa actúa como salida de la red, proporcionando la respuesta para cada uno de los patrones de entrada. Las neuronas de las capas ocultas realizan un procesamiento no lineal de los patrones recibidos.

Las conexiones del perceptrón multicapa siempre están dirigidas hacia adelante, es decir las neuronas de una capa se conectan con las neuronas de la siguiente capa. Las conexiones entre neuronas llevan asociado un número real, llamado peso de conexión; además todas las neuronas de la red llevan asociado un umbral, la cual es una conexión extra a la neurona, cuya entrada es constante e igual a 1. En la Figura 2.15 se muestra la arquitectura general de una red perceptrón multicapa.

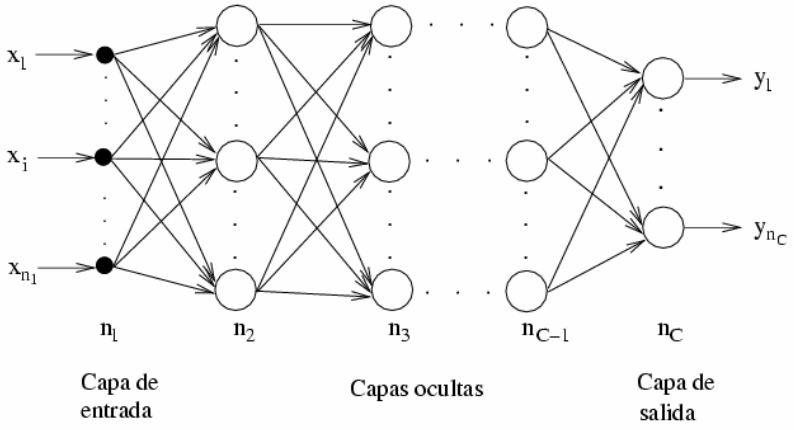


Figura.2. 15. Arquitectura de Red Perceptrón Multicapa

Diseño de la arquitectura del perceptrón multicapa

El diseño de la arquitectura de la red implica la determinación de la función de activación a emplear, el número de neuronas y el número de capas en la red.

La elección de la función de activación se suela hacer basándose en el recorrido deseado, y el hecho de elegir una u otra, generalmente, no influye en la capacidad de la red para resolver el problema.

En cuanto al número de neuronas y capas, algunos de estos parámetros vienen dados por el problema y otros deben ser elegidos por el diseñador. El número de neuronas en la capa de entrada, como en el de salida, vienen dadas por el número de variables de entrada y salida que definen el problema. El número de capas ocultas y el número de neuronas en estas capas deben ser elegidos por el diseñador. No existe un método que determine el número óptimo de capas y neuronas ocultas para resolver un problema dado. En la mayor

parte de las aplicaciones, estos parámetros se determinan por prueba y error. Partiendo de una arquitectura ya entrenada, se realizan cambios variando estos parámetros hasta conseguir una arquitectura adecuada para la resolución del problema, que pudiera no ser la óptima, pero que proporciona una solución.

Es lógico pensar que la solución idónea seria implementar una red con muchas capas ocultas y una gran cantidad de neuronas en cada una de ellas; sin embargo, esto tiene una serie de inconvenientes:

- Aumento de la carga computacional. Implicando una mayor dificultad de la implementación en tiempo real y un crecimiento en el tiempo de aprendizaje por parte de la red.

- Perdida en la capacidad de generalización.

Aprendizaje

El método de retro propagación de errores consiste en presentar a la red un patrón de entrada con el cual es calculada la salida de la red, una vez que se obtiene la salida de la red esta se compara con la salida deseada para dicho patrón y se obtiene el valor de error, luego se transmite los errores hacia atrás, partiendo de la capa de salida hacia las neuronas de las capas intermedias u ocultas y luego de capa en capa hasta que todas las neuronas reciben la parte relativa del error que aportan al error total, con este valor de error se alteran los pesos de las neuronas de acuerdo al error presente en cada una, de forma que la siguiente vez que se presente el mismo patrón de entrada se tenga un error menor. La facilidad en el proceso de aprendizaje, radica en la capacidad de la red de propagación de errores de adaptar los pesos de las capas intermedias aprendiendo la relación entre los patrones de entrada y las salidas de la red neuronal, de forma que al presentarse nuevos patrones, luego de la etapa de aprendizaje, esta pueda presentar una salida correcta[14].

[14] Isasi Viñuela, Pedro y Galvan Leon, Ines M. *Redes Neuronales Artificiales. Un Enfoque Practico.* Madrid : Prentice Hall, 2004.

Capacidad de Generalización

Uno de los aspectos fundamentales de las redes neuronales artificiales es su capacidad de generalizar a partir de ejemplos, lo que constituye el problema de memorización frente a generalización. Por generalización se entiende la capacidad de la red de dar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento.

Cuando se realiza el proceso de aprendizaje de la red es muy importante evaluar la capacidad de generalización. Para ello es necesario disponer de dos conjuntos de muestra o patrones, uno para entrenar la red (conjunto de entrenamiento) y otro para medir la capacidad de la red para responder correctamente ante patrones que no han sido utilizados en su entrenamiento (conjunto de validación).

Dependiendo de las características de los conjuntos de entrenamiento y validación, un entrenamiento riguroso podría anular la capacidad de generalización de la red. Por tanto en ocasiones puede ser conveniente exigir un menor aprendizaje con el objetivo de obtener mejores propiedades de generalización. Con este propósito es necesario evaluar el error que comete la red sobre el conjunto de patrones de validación, durante el proceso de aprendizaje. Cada cierto número de ciclos, se debe presentar a la red el conjunto de patrones de validación y calcular el error cometido por la red sobre dicho conjunto, para analizar el error del conjunto de entrenamiento conjuntamente con el de validación. Se pueden dar las siguientes situaciones.

- Ambos errores, de entrenamiento y validación, permanecen estables después de un cierto número de ciclos; este comportamiento se muestra en la Figura 2.16 (a). En este caso, se puede decir que el aprendizaje ha concluido con éxito, pues la red ha sido capaz de extraer las características del problema, alcanzando un buen nivel de generalización.

- A partir de cierto número de ciclos, el error de validación comienza a aumentar como se puede observar en la Figura 2.16 (b). En este caso se puede decir que el número de ciclos realizado es adecuado para encontrar un mínimo del error de

entrenamiento, a costa de perder propiedades de generalización. En esta situación se suele decir que se ha producido sobre aprendizaje o memorización en la red. Para corregir esta situación se debe detener el proceso de entrenamiento en el momento que el error de validación comienza a crecer, para poder disponer así de una red con mejor capacidad de generalización.

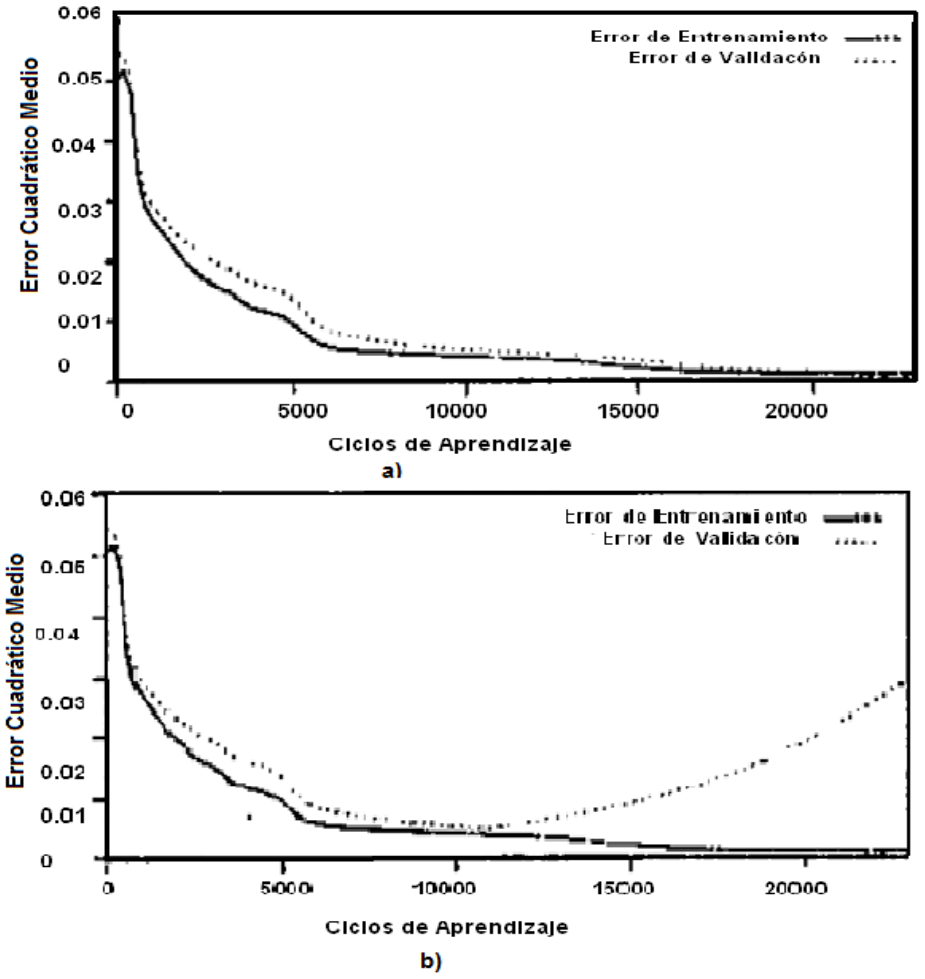


Figura.2. 16. Evolución de errores de entrenamiento y validación a lo largo del proceso de aprendizaje[15]

[15] Figura tomada del libro de Isasi Viñuela, Pedro y Galvan Leon, Ines M. *Redes Neuronales Artificiales. Un Enfoque Practico.* Madrid : Prentice Hall, 2004.

- ✓ **Redes de Base Radial**

Las redes neuronales de base radial son redes multicapa con conexiones hacia adelante, las cuales se caracterizan porque están formadas por una única capa oculta y cada neurona de esta capa posee un carácter local, cada neurona oculta de la red se activa en una región diferente del espacio de los patrones de entrada (dividiendo un problema complejo en varios problemas de menor complejidad).[16] Su carácter local viene dado por las funciones de base radial, por lo general la función gaussiana, como función de activación. Las neuronas de la capa de salida realizan una combinación lineal de la activación de las neuronas ocultas. Al igual que el Perceptrón Multicapa, las redes de base radial son aproximadores universales.

Arquitectura de la Red de Base Radial

Las redes neuronales de base radial están formadas por tres capas de neuronas: la capa de entrada, una única capa oculta y la capa de salida.

En la Figura 2.17 se muestra la arquitectura general de una red de base radial con sus diferentes capas.

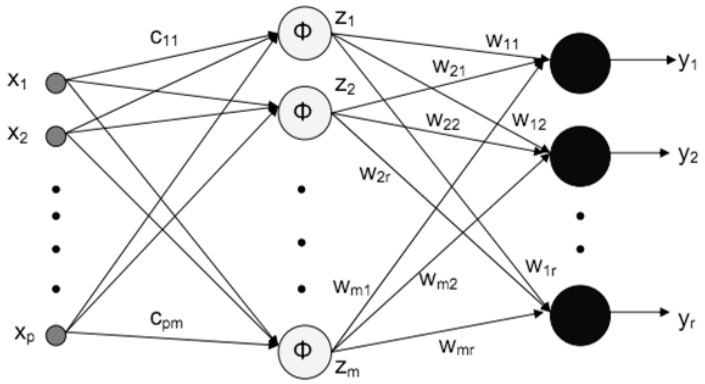


Figura.2. 17. Arquitectura de red neuronal de base radial

[16] Salazar, Jorge Enrique. *Diseño e Implementacion de un Sistema de Entranamiento en Redes Neuronales Utilizando el Software Neusosystems de Siemens.* Sangolqui : Escuela Politecnica del Ejercito, 2009.

La **capa de entrada** la forman un conjunto de neuronas que reciben las señales del exterior transmitiendo a la siguiente capa sin modificarlas.

La **capa oculta** recibe las señales de la capa de entrada y realizan una transformación local y no lineal sobre dichas señales (esta es la única capa que contiene elementos no lineales), las conexiones de la capa de entrada a la capa oculta no llevan asociado ningún peso, sin embargo las conexiones de la capa oculta a la capa de salida si llevan asociado un peso de conexión; cada neurona tiene asociada una función de base radial de tal manera que representa una clase o categoría, la misma que viene dada por (Ci, di) donde Ci representa los centros de la función de base radial y di son números reales que representan la desviación de la función de base radial. La salida de cada elemento de la capa oculta $z_i(n)$se calcula como la distancia que existe entre el patrón de entrada $X(n)$ al centro de la función de base radial ponderada inversamente por la desviación di y aplicando después este valor a una función de base radial φ.

$$Z_i(n) = \varphi\left(\frac{\left(\sum_{j=1}^{p}\left(x_j(n)-c_{ji}\right)^2\right)^{1/2}}{d_i}\right) \tag{2.1}$$

Las funciones de base radial poseen un carácter local, pues son funciones que alcanzan un nivel cercano al máximo cuando el patrón de entrada está próximo al centro de la neurona; a medida que el patrón se aleja del centro, el valor de la función va tendiendo al mínimo de su recorrido.[17] La función de base radial puede ser de diferentes formas:

- Función Gaussiana:

$$\varphi(r) = e^{\left(-\frac{r^2}{2}\right)} \tag{2.2}$$

17 Isasi Viñuela, Pedro y Galvan Leon, Ines M. *Redes Neuronales Artificiales. Un Enfoque Practico.* Madrid : Prentice Hall, 2004.

- Función Inversa Cuadrática:

$$\varphi(r) = \frac{1}{1+r^2} \tag{2.3}$$

- Función Inversa Multi-cuadrática:

$$\varphi(r) = \frac{1}{\sqrt{1+r^2}} \tag{2.4}$$

En la Figura 2.18 se muestra gráficas de las funciones de base radial antes mencionadas.

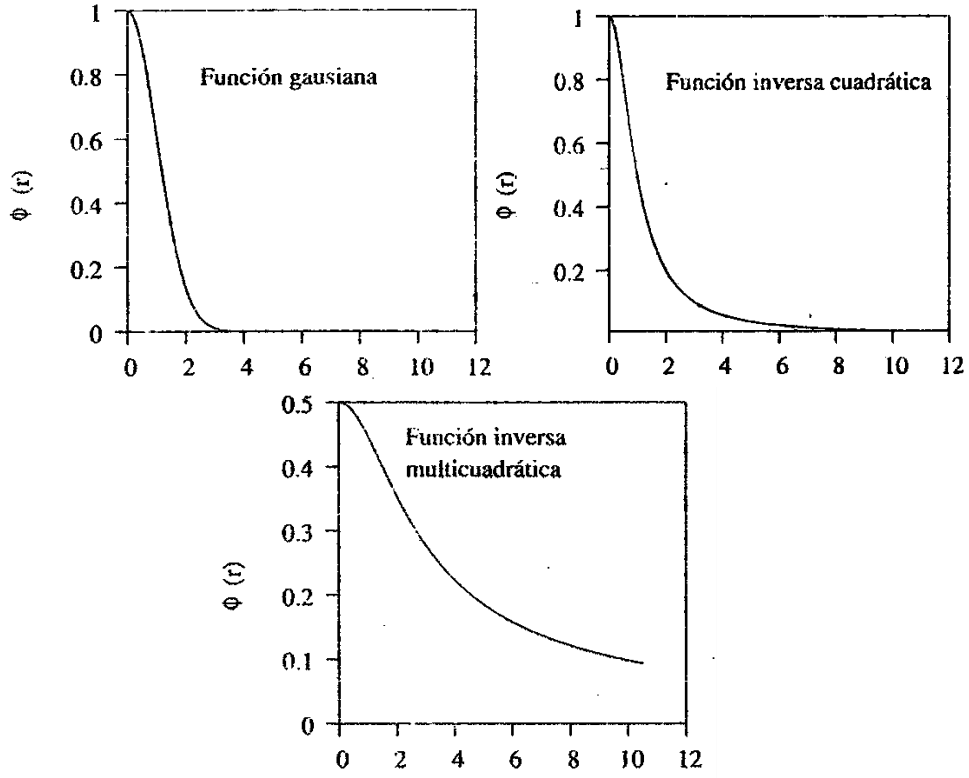


Figura.2. 18. Formas de funciones de base radial[18]

La **capa de salida** realiza una combinación lineal de las activaciones de las neuronas ocultas, actuando también como salida de la red. Esta capa es la única en la que las

[18]Figura tomada del libro de Isasi Viñuela, Pedro y Galvan Leon, Ines M. *Redes Neuronales Artificiales. Un Enfoque Practico.* Madrid : Prentice Hall, 2004.

neuronas poseen un umbral, la cual se suele considerar como una conexión más de la neurona cuyo valor es constante e igual a 1.

En la Figura 2.19 se muestra la respuesta de las funciones de base radial de la capa oculta; en donde los puntos representan patrones en el espacio de entrada que se agrupan alrededor de los centros.

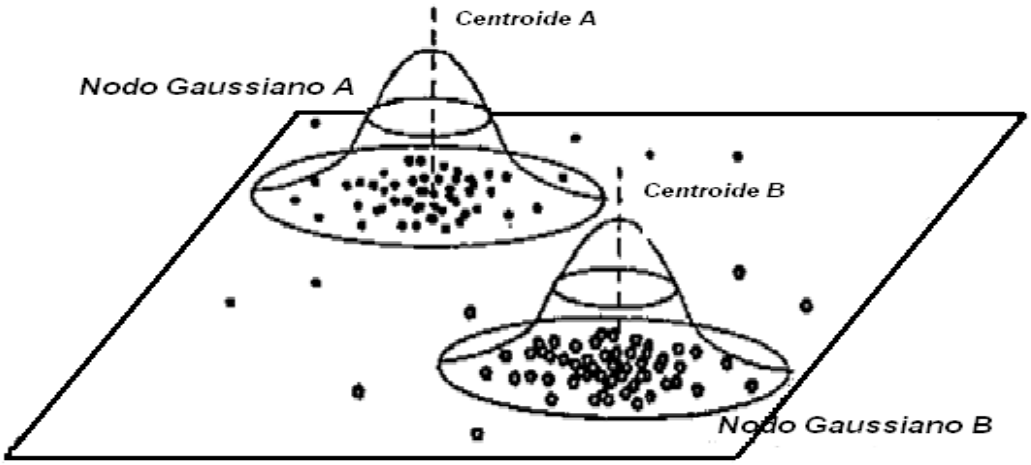


Figura.2. 19. Respuesta localizada de las neuronas ocultas de RBF.[19]

Diseño de arquitectura de redes de base radial

El número de entradas y salidas en una red de base radial viene dado por el número de variables que definen el problema.

Generalmente el número de neuronas ocultas en la red se determina por prueba y error, variando el número de neuronas hasta obtener una red capaz de resolver el problema. Se debe tener cuidado en la variación del número de neuronas ocultas ya que pequeñas variaciones pueden influir significativamente en los resultados obtenidos por la red, debido a que cada neurona representa una determinada región del espacio de entrada, pudiendo no

[19] Figura tomada del libro de Sanz Molina, Alfredo y del Brio, Bonifacio Martin. *Redes Neuronales y Sistemas Difusos.* Mexico, DF : Alfaomega, 2002.

estar bien representado dicho espacio, ya sea por la presencia de demasiadas o pocas neuronas ocultas en una misma zona.

Aprendizaje

Para este tipo de redes el aprendizaje consiste en determinar todos los parámetros de la red.

- Capa de Salida: Pesos; se realiza la optimización en base a las salidas que se desea obtener.

- Capa Oculta: Centros y desviaciones; se los determina mediante la optimización en el espacio de entrada, ya que cada neurona va a representar una zona de dicho espacio.

Existen dos tipos de aprendizaje que se pueden aplicar a las redes neuronales de base radial y son:

- **Aprendizaje Hibrido.** Este aprendizaje está compuesto por dos fases:

    - Fase no Supervisada, en la que se determina los parámetros de la capa oculta, la determinación de centros se realiza mediante algoritmos de k-medias o mapas de Kohonen[20]; mientras que, las desviaciones se deben calcular de manera que cada neurona de la capa oculta se active en una región del espacio de entradas de manera que el solapamiento de las zonas de activación sea lo más ligero posible, para el cálculo de estas desviaciones existen una serie de aproximaciones que se pueden considerar. Uno de estos criterios viene dado por la ecuación 2.5.

[20] **Zubizarreta, Asier.** *Aplicación de las técnicas de redes neuronales para el diagnóstico on-line del proceso de electroerosión por hilo.* 11 de Octubre de 2006. [Citado el: 28 de Noviembre de 2011.] http://www.disa.bi.ehu.es/spanish/profesores-etsi-bilbo/~jtpcaaxi/PFC/wwwANN/informacion_de_contacto.htm.

$$d = \frac{D_{max}}{\sqrt{2 * m_1}} \tag{2.5}$$

Donde D_{max} es la máxima distancia entre los centros m_1 es el número de centros

- Fase Supervisada, en la que se determinan los pesos de la capa de salida, usando una técnica de minimización de la función de Error

- **Aprendizaje Totalmente Supervisado.** Este tipo de aprendizaje no conserva las propiedades o características locales de las redes de base radial. En este caso todos los parámetros de la red se determinan de manera completamente supervisada con el objetivo de minimizar el error cuadrático medio. En este tipo de aprendizaje no se tiene en cuenta que las desviaciones tomen valores tales que provoquen solapamiento de las activaciones de las neuronas de la capa oculta. Para el cálculo de los parámetros se usa las ecuaciones descritas en la Tabla 2.1.

Tabla.2. 1. Ecuaciones para el cálculo de parámetros de una red de base radial.

Parámetro	Ecuación
Pesos	$w_{ik}(n) = w_{ik}(n-1) - \alpha_1 \frac{\delta e(n)}{\delta w_{ki}}$ (2.6)
Centros	$c_{ij}(n) = c_{ij}(n-1) - \alpha_2 \frac{\delta e(n)}{\delta c_{ij}}$ (2.7)
Desviaciones	$d_i(n) = d_i(n-1) - \alpha_3 \frac{\delta e(n)}{\delta d_i}$ (2.8)
Umbrales	$u_k(n) = u_k(n-1) - \alpha_1 \frac{\delta e(n)}{\delta u_k}$ (2.9)
Donde $\alpha_1, \alpha_2, y\ \alpha_3$ *son las tasas de aprendizaje para los pesos, centros y amplitudes respectivamente*	

Similitudes y diferencias entre redes de base radial y perceptrón multicapa

- **Similitudes**

Son redes agrupadas en capas y las conexiones son dirigidas hacia adelante.

Ambas son aproximadores universales, pueden aproximar cualquier función continua.

Las conexiones de la capa de oculta hacia la de salida llevan pesos asociados.

Ambas redes son capaces de resolver un mismo problema.

- **Diferencias**

La función de activación de las neuronas ocultas de la red, en las redes de base radial se usa funciones de base radial, las cuales hacen que las neuronas posean un carácter local, activándose cada neurona en una determinada región del espacio de entrada. Por otro lado el perceptrón emplea funciones de activación sigmoidales, que permiten que las neuronas ocultas posean una activación global (se activan en todo el espacio de entrada). En la Figura 2.20 se muestra el carácter local de las relaciones hechas por las redes base radial y las relaciones globales que construye el perceptrón multicapa.

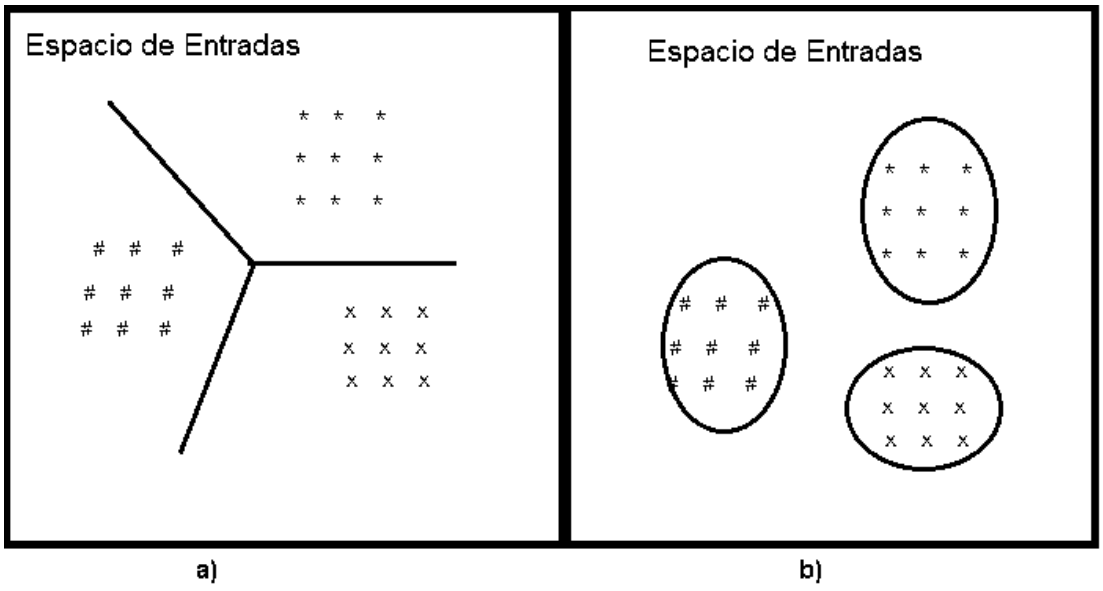


Figura.2. 20 Relaciones construidas por las redes neuronales. a) Red perceptrón multicapa; b) Redes de base radial

El argumento de la función de activación en las neuronas de base radial es la distancia entre el vector de entrada y el centro de esa unidad. Mientras que en las neuronas del perceptrón es el producto entre el vector de entrada y el vector de pesos de cada unidad.

En el perceptrón pueden tener tantas capas ocultas como se desee y en las redes de base radial existe una única capa oculta.

El aprendizaje de la red perceptrón multicapa es lento ya que al construir relaciones globales el cambio en uno solo de los pesos provoca cambios en la salida para todos los patrones de entrada, reduciendo así el efecto de previos ciclos de aprendizaje y retrasando la convergencia del algoritmo de aprendizaje. En las redes de base radial el aprendizaje es más rápido debido a la construcción de relaciones locales, ya que el cambio en un solo peso de la red afecta únicamente a la neurona oculta asociada a dicho peso y por lo tanto a un determinado grupo de patrones.

En las redes de base radial las conexiones de la capa de entrada a la capa oculta no llevan pesos asociados. Mientras que en el perceptrón si los llevan.

La función de activación de la capa de salida en las redes de base radial es lineal, mientras que en el perceptrón esta condición de linealidad no es imprescindible.

En casos en los que el número de variables de entrada es alto, el número de neuronas ocultas aumenta exponencialmente, lo cual podría influir negativamente en la capacidad de generalización de la redes de base radial. En cambio, en las redes perceptrón multicapa no se necesita un gran número de neuronas ocultas para abordar el problema ya que construye relaciones globales.

2.2.4.2.Redes Neuronales Recurrentes

Estas redes pueden tener conexiones que van hacia atrás desde los nodos de salida a los nodos de entrada, o también conexiones arbitrarias entre cualquiera de sus nodos. De esta forma, el estado interno de la red se modificará a medida que se le presenten datos, simulando una memoria.

Este tipo de redes son útiles para resolver problemas donde la salida no depende solamente de la entrada actual, sino también de las entradas anteriores. Algunos ejemplos podrían ser la predicción del clima en base al de los días anteriores o la predicción de la caída o ascensión del precio de un producto en base a valores anteriores.

- ✓ **Red Neuronal de Hopfield**

La red neuronal de Hopfield es conocida como un modelo de memoria auto-asociativa de patrones, ya que es capaz de aprender a reconstruir los patrones de entrada que memorizan durante su entrenamiento a partir de información incompleta sobre los patrones.[21]

La arquitectura de la red de Hopfield consiste en una única capa formada por n elementos de procesamiento interconectados que cambian sus valores de activación independientemente, cada una conectada a todas las demás, por lo tanto todos los elementos son a su vez entradas y salidas. Las neuronas de la red de Hopfield poseen dos estados, generalmente {-1, 1} o {0,1} (según convenga), que vienen determinados por el nivel de activación que recibe la neurona. Además los pesos asociados a pares de neuronas son simétricos, es decir existe la igualdad $W_{ij} = W_{ji}$.

Un diagrama de la arquitectura de este tipo de redes se puede observar en la Figura 2.21.

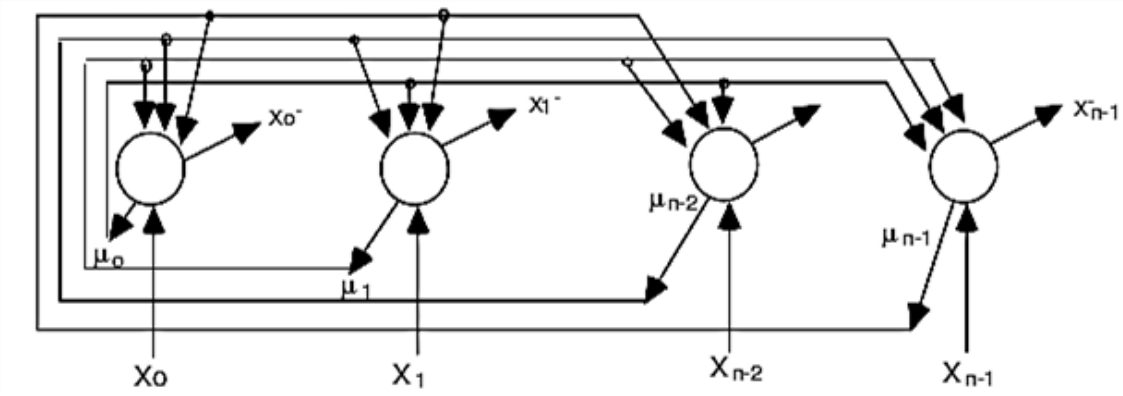


Figura.2. 21. Arquitectura de una red tipo Hopfield

[21] Basogain Olabe, Xabier. Redes Neuronales Artificiales y sus Aplicaciones. Bilbao : Escuela Superior de Ingenieria de Bilbao, 2008.

Funcionamiento de la Red

El vector de entrada x debe tener tantas componetes como neuronas tenga la red. La entrada es aplicada en t=0 a la unica capa que tiene la red, determinandose asi las salidas. Debido a las realimentaciones, estas salidas se convierten en las nuevas entradas de la red.

Se tiene una relación dinámica que responde a la siguiente ecuación:

$$x_j(t+1) = f\left(\sum_{i=1}^{N} w_{ji} \cdot x_i(t) - \theta_j\right) \tag{2.10}$$

Donde f es la función de activación signo y corresponde al umbral asignado a cada neurona.

Ya que la red de Hopfield funciona como una memoria dinámica que asocia la entrada de la red con unos determinados patrones, y que como se trata de una red recurrente las salidas de la red se convierten en las nuevas entradas; consecuentemente la condición de parada sera aquella que lleve a la red a una situación de equililibrio, que se producirá cuando se cumpla la siguiente condición:

$$x(t+1) = x(t) \tag{2.11}$$

La idea de Hopfield es localizar cada patrón que se quiere almacenar en la red en el fondo de un "valle" de la funcion de energía. El modo de funcionamiento de esta memoria dinámica se basa en partir de un determinado estado inicial tras lo cual se dejará evolucionar el sistema hasta llegar a un estado estable. Este estado estable sera el patrón que corresponde a nuestro estado inicial.

Al presentar una nueva entrada a la red esta tomará una configuración inicial parecida a alguno de los estados de un estímulo almacenado, luego comenzará a cambiar hasta llegar al estado que representa a la señal almacenada en la etapa de aprendizaje. Si el nuevo estímulo es muy diferente a cualquier de los que ya están almacenados se alcanzará un punto que no representa ningún recuerdo. Otro caso que podría darse es que la red

comience a presentar estados que se repiten periódicamente y no llegue a estabilizarse nunca.

Aprendizaje

Existen dos fases: Fase de almacenamiento y fase de recuperación. Durante la fase de almacenamiento se van a determinar los valores que deben tomar los pesos de la red para almacenar un conjunto de patrones, y la fase de recuperación describe el mecanismo para recuperar la información almacenada a partir de información incompleta.

Un estado estable de la red es un mínimo local de la función de energía por lo tanto los recuerdos almacenados por la red son mínimos locales de la red, el objetivo del aprendizaje es el que la red almacene como estados estables un conjunto de recuerdos o memorias dado, con este fin se debe determinar el conjunto de pesos sinápticos que hacen que la función de energía tenga esos patrones como mínimos locales. Cada mínimo local forma una cuenca de atracción en la *Hipersuperficie de Energía* de la red. En la Figura 2.22 se observa la evolución del estado de la red hacia un mínimo local donde cada uno representa una memoria o recuerdo.

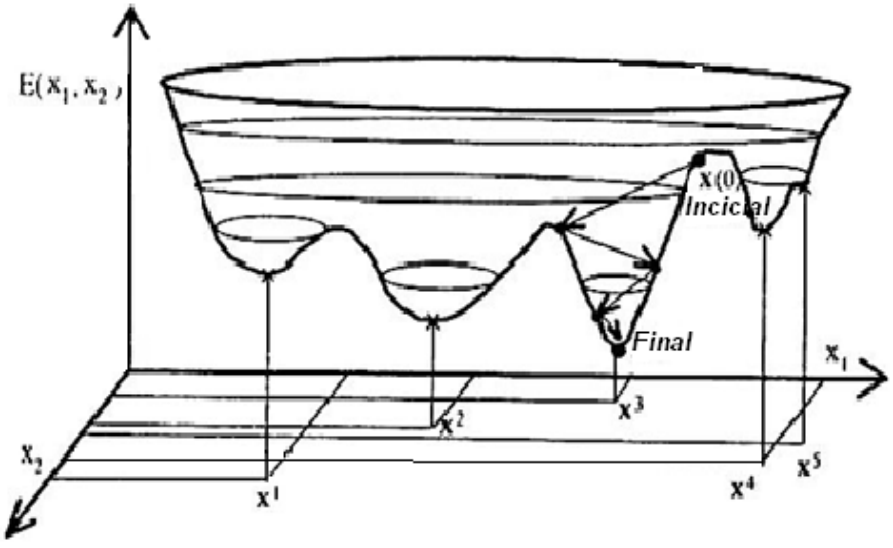


Figura.2. 22. Hipersuperficie de funciones de Energía de la red[22]

[22] Figura tomada del libro de Figura tomada del libro de Isasi Viñuela, Pedro y Galvan Leon, Ines M. *Redes Neuronales Artificiales. Un Enfoque Practico.* Madrid : Prentice Hall, 2004.

En este tipo de redes es corriente emplear conjuntos de entrenamiento donde solo se presenten entradas, y asumir como salidas deseadas los propios patrones de entrada. Así, se presentan más tarde en funcionamiento patrones incompletos o parcialmente erróneos para su reconstrucción.

- **Regla de la Pseudoinversa.** Esta regla asegura el almacenamiento exacto de los patrones y permite almacenar un número de patrones igual al número de neuronas de la red, no obstante si se almacenan un número de patrones mayor a la mitad del número de neuronas de la red, las cuencas de atracción resultan despreciables.

  El objetivo es encontrar una matriz W que cumpla la siguiente condición:

$$W = \Sigma . \Sigma^{+} \tag{2.11}$$

  Donde W es la matriz de pesos sinápticos, $\sum$ es la matriz de patrones de aprendizaje colocados como columnas y $\sum^{+}$ es la matriz Pseudoinversa de $\sum$. La Matriz Pseudoinversa se puede calcular por el método de Greville.[23]

- **Capacidad de almacenamiento.** Esta capacidad se refiere a la cantidad de información que puede ser guardada sin error, esta puede medirse como: C = (No de patrones guardados)/(No de Neuronas). La capacidad depende de los pesos de la conexión, los patrones almacenados y la diferencia entre los patrones de entrada y los guardados. El número máximo de patrones no correlacionados que puede almacenar es igual al 15% del número de neuronas de la red.

2.2.5. APLICACIÓN DE LAS REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales tienen un amplio campo de aplicaciones en varias áreas, resolviendo problemas concretos de la vida real. A continuación se destaca aplicaciones en varias áreas como son:

---

[23] Duarte, Fernando. Control de Robots Redundantes. [Citado el: 11 de Noviembre de 2011.] http://www.ipv.pt/millenium/Millenium24/8.pdf.

- Reconocimiento de Patrones: Reconocimiento de caracteres impresos y manuscritos, reconocimiento del habla, reconocimiento de firmas.

- Telecomunicaciones: Construcción de ecualizadores lineales y canceladores de ecos, anulación de ruido y vibraciones en instalaciones industriales.

- Banca y Finanzas: Asesoría de préstamos, concesión de créditos, evolución de precios, identificación de falsificaciones, criptografía, códigos de seguridad adaptativos, análisis de mercados, etc.

- Medicina.- Análisis de electroencefalograma y electrocardiograma, diseño de prótesis, reconocimiento de células portadoras de cáncer, diagnóstico y tratamiento de enfermedades a partir de síntomas, síntesis de nuevos medicamentos, etc.

- Control y Optimización: Optimización y control de procesos, sistemas de control de calidad, modelamiento de procesos, etc.

- Aplicaciones Militares: Guiado automático de misiles, análisis de situación de vuelo en combate aéreo.

3. CAPITULO 3

DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS

3.1. IMPLEMENTACIÓN DE ALGORITMOS PARA EL PROCESAMIENTO DE LA IMAGEN CAPTURADA.

Las imágenes utilizadas para la realización del proyecto tienen el tamaño de 1600 x 1200 pixeles, y fueron obtenidas mediante varios tipos de cámaras de diferentes marcas.

3.1.1. EXTRACCIÓN DE LA PLACA

Para la extracción de la placa se sigue el siguiente procedimiento:

a. **Preprocesamiento de la Imagen.** Una vez capturada la imagen se procede a analizar si es una imagen policromática o una imagen en niveles de gris, en caso de que ésta sea policromática se procede a convertirla en una imagen en niveles de gris para luego proceder a recortar márgenes innecesarios de la imagen, enfocándonos en un área central de la imagen en donde se supone se debe encontrar la placa del vehículo, el resultado de ésta operación se muestra en la Figura 3.1.

a) b)

Figura.3. 1. Preprocesamiento de imagen. a) Imagen original b) Imagen con márgenes recortados

b. Realce de Contraste. Enseguida, se procede a dar ganancia en contraste a la imagen con el fin de resaltar objetos como la placa, sobre todo en imágenes que carecen de buena iluminación, se lo consigue aplicando a la imagen un filtro morfológico tipo *Bottom-Hat* como se puede observar en la Figura 3.2.

Figura.3. 2. Realce de contraste mediante la aplicación de filtro *Bottom-Hat*.

Posteriormente se calcula el umbral óptimo de la imagen por medio del método de Otsu para convertirla en imagen binarizada (monocromática) para posteriormente aplicar operaciones morfológicas con el fin de resaltar las secciones de la imagen donde podría estar ubicada la placa, el resultado de esta operación se muestra en la Figura 3.3.

Figura.3. 3. Imagen monocromática con el umbral óptimo obtenido por el método de Otsu.

c. Filtrado Morfológico. Se inicia aplicando una operación de cierre, utilizando un elemento estructural tipo línea horizontal, cuya longitud es dada en base a la separación existente entre los caracteres de la placa; paso seguido se aplica una dilatación con elemento estructural lineal horizontal y otro vertical para cerrar pequeños agujeros remanentes de la anterior operación. En la Figura 3.4 se puede observar el resultado obtenido.

Figura.3. 4. Operación de cierre con elemento estructural lineal horizontal

El siguiente paso es aplicar una operación de apertura con un elemento lineal vertical fundamentándose en la altura de los caracteres. Además se elimina objetos muy pequeños presentes en la imagen. En la Figura 3.5 se muestra la imagen resultante luego de realizar esta operación.

Figura.3. 5. Operación de apertura con elemento estructural vertical

Posteriormente se efectúan dilataciones horizontales y verticales con el propósito de expandir la zona de la placa permaneciendo luego de este punto solamente regiones a las que puede pertenecer la placa como se observa en la Figura 3.6.

a) b)

Figura.3. 6. Operaciones morfológicas: a) Dilatación horizontal; b) Dilatación vertical

d. Etiquetado y Discriminación. A pesar de que con las operaciones morfológicas se elimina una gran cantidad de ruido, en ocasiones se siguen presentando en la imagen objetos adicionales a la placa, esto se debe básicamente a la composición física del automóvil. Por esta razón se procede a etiquetar cada uno de los objetos y se los discrimina en primer lugar en relación a su área, la cual debe ser de un tamaño mayor a 6000 pixeles. Posteriormente se discrimina los objetos por su relación ancho/alto la cual debe estar en un rango entre 1.15 y 5.8, este rango es obtenido mediante pruebas empíricas de diferentes imágenes en donde la placa se encuentra con un diferente ángulo de inclinación. En caso de que persista la existencia de más de un objeto que cumpla con estas características de discriminación se procede a tomar como región de la placa la que se encuentre

posicionada a una altura menor que el resto de regiones, procediendo a realizar el recorte del área de la región de la placa en la imagen original. En la Figura 3.7 se puede observar la región correspondiente a la placa obtenida al finalizar el proceso de extracción.

Figura.3. 7. Imagen de región de la placa localizada

En la Figura 3.8 se presenta un diagrama de flujo indicando las operaciones de manera detallada el proceso de extracción y limpieza de ruido de la placa.

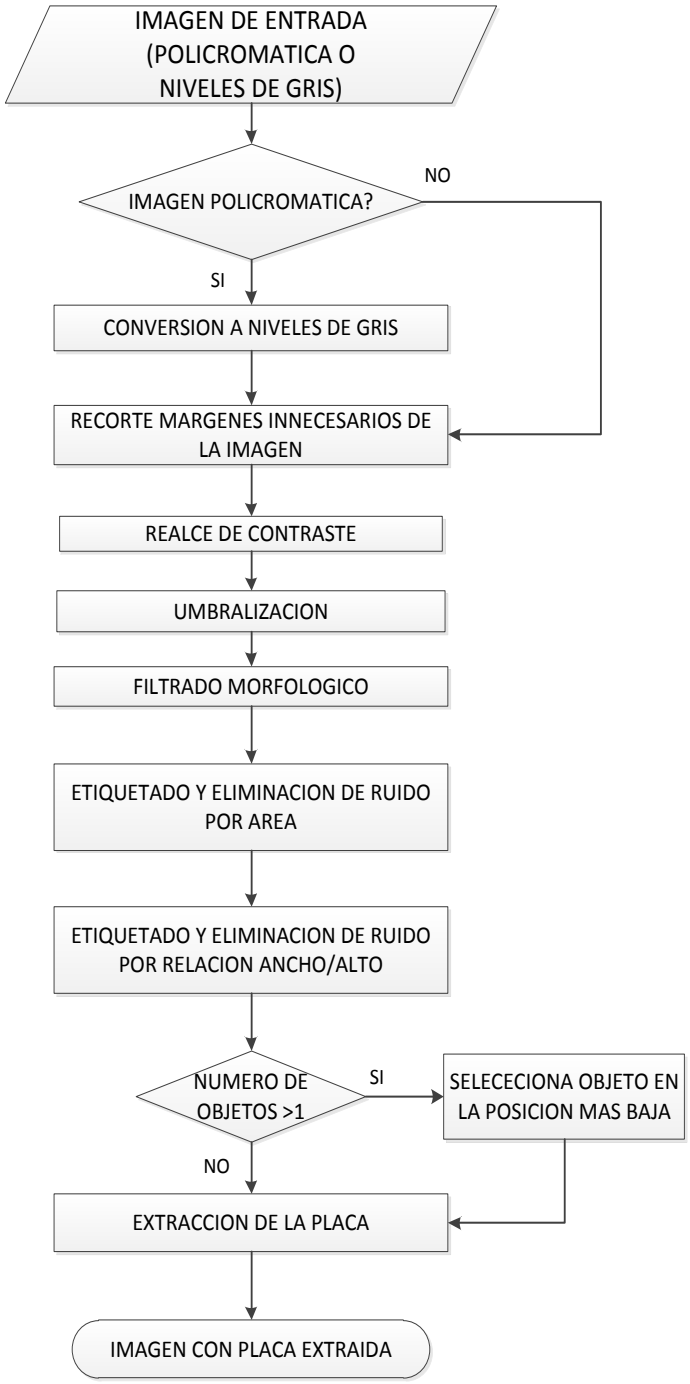


Figura.3. 8. Diagrama de flujo para la extracción de la placa

3.1.2. DETECCIÓN Y CORRECCIÓN DEL ÁNGULO DE LA PLACA

Una vez extraída la región que contiene la placa se procede al cálculo aproximado del ángulo de inclinación de la placa, esto se realiza mediante el siguiente procedimiento:

a. **Determinación de umbral Óptimo.** se calcula el umbral óptimo general de la imagen de la región de la placa localizada, para posteriormente proceder a la binarización de la imagen.

b. **Determinación de ángulo de inclinación de la placa.** se procede a detectar los bordes existentes en la imagen a través de un filtro pasa alto tipo *Sobel*, obteniendo el resultado de la Figura 3.9 a). Posteriormente se aplica la transformada de Hough para la detección de líneas rectas, de las rectas obtenidas se escoge la línea recta más larga en base a la geometría rectangular de la placa, significando esta línea la base del rectángulo que forma la placa, en la Figura 3.9 b) se observa que se resalta esta línea con un color azul.

a) b)

Figura.3. 9. Determinación de ángulo de la placa: a) Detección de bordes; b) Determinación de línea que define el ángulo.

c. **Corrección de ángulo de inclinación.** Una vez determinada la línea que caracteriza la placa, se obtiene el valor del ángulo de la misma y se rota la imagen corrigiendo el ángulo de inclinación, en el caso que sea necesario. El resultado se observa en la Figura 3.10.

Figura.3. 10. Imagen de la placa con ángulo de inclinación corregido.

3.1.3. SEGMENTACIÓN

Una vez obtenida la imagen de la placa con el ángulo de inclinación corregido se debe extraer cada uno de los caracteres existentes en la placa, para su posterior clasificación y reconocimiento. Para lograr este cometido en este proyecto se realizó el siguiente procedimiento.

a. Se determina el área que corresponde al rectángulo que contiene la placa, eliminando áreas en donde no existen caracteres obteniéndose como resultado la Figura 3.11.

Figura.3. 11. Determinación del área rectangular de la placa.

b. El siguiente paso es normalizar las dimensiones de la imagen obtenida, la cual será de 325 pixeles de ancho por 100 pixeles de alto. Posteriormente se etiqueta cada uno de los objetos de la imagen y se elimina objetos cuya área sea menor a 300 pixeles, eliminando de esta manera los objetos como tornillos y los

caracteres de la palabra “ECUADOR”. Al final los objetos restantes son discriminados por varios filtros basándose en características que solamente los caracteres posean; el primero de ellos es que cumplan con una relación alto/ancho dentro de un rango de 1.05 a 10 (valores obtenidos de manera empírica), este rango se toma debido a existencia de caracteres inclinados, y caracteres cuya relación alto/ancho es muy grande (caso: número 1 y letra *I*); el siguiente filtro se basa en que la atura de cada uno no debe exceder el +/- 20% de la altura promedio de todos los elementos. En la Figura 3.12 se observa la región de la placa en donde los objetos de la imagen cumplen con los filtros aplicados.

Figura.3. 12. Objetos que cumplen con filtros de discriminación.

c. El último paso es la extracción de los caracteres de la placa, a los cuales se los redimensiona a un tamaño de 24x42 pixeles, y posteriormente se los almacena a cada uno de ellos dentro de una estructura de seis o siete elementos según la placa para su posterior clasificación y reconocimiento. En la Figura 3.13 se puede observar cada uno de los caracteres extraídos.

Figura.3. 13. Caracteres de la placa extraídos.

En la Figura 3.14 se muestra un diagrama general de la detección y corrección del ángulo de la placa así como también la segmentación de los caracteres de la placa.

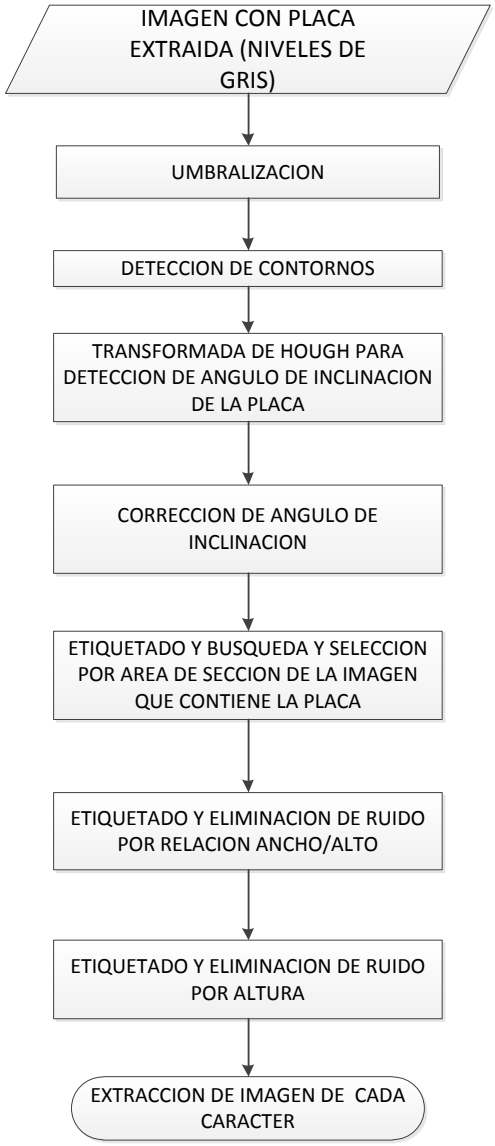


Figura.3. 14. Diagrama de detección de ángulo de inclinación de la placa y extracción de caracteres.

3.2. DISEÑO E IMPLEMENTACIÓN DE LAS REDES NEURONALES

3.2.1. PERCEPTRÓN MULTICAPA

Para la creación de este tipo de red neuronal se tomó un conjunto de 300 muestras, las cuales se las divide en dos grupos, el 75 % de las muestras son usadas como conjunto de aprendizaje y el 25% restante de las muestras se usa como conjunto de validación o test.

Previo el ingreso de las imágenes de los caracteres a la red neuronal es necesario convertir su representación en matriz a una representación de un simple vector, de esta manera los conjuntos de entrenamiento y validación son representados como una matriz formada por vectores que representan cada carácter. Esta operación se realiza mediante el uso de la función n=**N**(:); donde **N** es la matriz que representa cada carácter y n es el vector que representa a dicha matriz.

3.2.1.1.Diseño de la arquitectura del perceptrón multicapa

El diseño de la arquitectura de la red implica la determinación de la función de activación a emplear, el número de neuronas y el número de capas en la red. Esta red posee tres capas, capa de entrada, capa de salida y una única capa oculta.

El número de neuronas en la capa de entrada viene dado por el problema, en este caso ya que los caracteres extraídos de la placa tienen un tamaño de 24x42 pixeles, el número de neuronas de la capa de entrada es igual a 1008 neuronas.

En la capa de salida existe una sola neurona ya que se obtiene una sola salida por cada patrón ingresado a la red.

La capa oculta está formada por 26 neuronas para la red clasificadora de letras y 13 para la red de números, ya que se hizo varias pruebas y se encontró este número de neuronas como el más adecuado para la resolución del problema, ya que con un número menor de neuronas el sistema emplea demasiados ciclos de entrenamiento para lograr cumplir con el criterio de error; mientras, con un número mayor de neuronas no se obtiene una gran mejora en cuanto a reducir el tiempo de entrenamiento, además la red empieza a tener dificultades en la generalización, ya que empieza a hacerse específica para el conjunto de patrones de entrenamiento[24].

[24] Las pruebas realizadas para la determinación de la arquitectura adecuada para el perptron multicapa se encuentran en el Anexo1. Informe de Pruebas seccion: Pruebas de redes perceptron multicapa.

En cuanto a las funciones de activación las neuronas de la capa oculta utilizan una función tipo sigmoidea ‘logsig’ mientras que para la capa de salida se usa una función de tipo lineal ‘purelin’.

3.2.1.2.Creación de la red

Para crear una red neuronal perceptrón multicapa se utiliza la función *newff* del toolbox de redes neuronales de Matlab®. Su sintaxis es *red = newff (**P**,[S_1 S_2...],{TF_1 TF_2...}, Tipo de entrenamiento)*. Esta función recibe varios argumentos para poder crear la red neuronal los cuales son:

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

- [S_1, S_2,…] es un vector que describe el número de neuronas que tiene cada capa oculta de la red y el número de neuronas de salida. El tamaño de dicho vector permite conocer al programa el número de capas que debe tener la red que se va a crear.

- {TF_1 TF_2…} es un vector en el que se señala las funciones de activación que van a poseer todas las neuronas de una capa. Este vector debe tener tantas funciones como capas vaya a tener nuestra red.

- Tipo de entrenamiento: Nombre del tipo de entrenamiento que va a seguir la red.

En código de programa para la creación de la red con la arquitectura diseñada anteriormente es el siguiente:

```
net=newff(minmax(Patron),[26,1],{'logsig','purelin'},'trainscg');
```

3.2.1.3.Simulación de la Red

Para obtener las salidas de una determinada red ante ciertas entradas se utiliza la función *sim*. Esta función devuelve un vector cuyas componentes son las salidas que se obtienen de cada neurona de salida de la red. A dicha función se le introducen como argumentos el nombre del objeto que representa a la red en Matlab®, y el vector de entradas a ser introducido en la red. Se puede obtener varias simulaciones a la vez introduciendo una matriz compuesta por cada uno de los vectores de entrada de los que se quiere obtener su salida.

3.2.1.4.Entrenamiento

El entrenamiento de la red consiste en presentarle unas entradas y sus correspondientes salidas para que la red vaya reajustando su salida mediante la modificación de sus pesos y umbrales, de manera que la función de evaluación error de la red se minimice. La función de evaluación de error utilizada por defecto en Matlab® es el error cuadrático medio, si se desea utilizar alguna otra función se debe modificar la variable *net.perform.fcn*. Existen dos formas de actualizar los pesos, se puede realizar en el modo conocido como *incremental*, bajo el cual los pesos son actualizados después de que cada patrón se presente, o utilizar el modo *Batch* en el cual los pesos no son actualizados hasta que todos los patrones no se hayan introducido en la red. Es decir, solo se actualizan cuando se termina una época de entrenamiento. Para ejecutar el entrenamiento en modo *Batch* es necesario introducir los patrones en forma de matriz.

Para entrenar una red en Matlab® se utiliza la función *train*. Dicha función utiliza como argumentos el nombre de la red que se quiere entrenar y los patrones (a) y sus correspondientes salidas (p). Su sintaxis en Matlab® es: *[net,pr] = train (net,a,p).*

La variable *pr* contiene información sobre el proceso de entrenamiento y la variable *net* contiene a la red ya entrenada, es decir con sus pesos y umbrales ajustados. Por otro lado, esta función utiliza una serie de variables para definir el entrenamiento y que pueden definirse con anterioridad. Estas variables varían según el tipo de entrenamiento utilizado. Entre las principales variables están:

- *epochs:* Define el máximo número de épocas de entrenamiento que puede tener el proceso de aprendizaje.

- *show*: Indica la forma de visualización que deseamos tener durante el entrenamiento de la red. Si su valor es '*Nan'* quiere decir que no se quiere ningún tipo de visualización.

- *goal*: Esta variable indica el valor mínimo que debe alcanzar la función error de la red. Si ésta alcanza dicho valor el entrenamiento se detendrá automáticamente.

- *time*: Este parámetro indica el tiempo máximo en segundos que durará el entrenamiento de la red. Una vez que el tiempo del proceso alcance dicho valor el entrenamiento se detendrá.

- *min_grad*: Determina el valor mínimo de la gradiente del error, una vez que la gradiente tome un valor menor a este parámetro el aprendizaje se detendrá.

- *max_fail*: Es el máximo número de iteraciones que puede incrementarse el error de validación antes de detenerse el entrenamiento.

- *lr*: Es la tasa de aprendizaje

A continuación se describirán algunos de estos tipos de entrenamiento.

- **Regla delta generalizada.** Este algoritmo de entrenamiento se basa en actualizar los pesos y valores umbrales en la dirección de la gradiente negativa de la función del error. Para utilizar este entrenamiento se introduce en el argumento correspondiente al entrenamiento la cadena *'traingd'*. Uno de los parámetros asociados con este tipo de entrenamiento es la tasa se aprendizaje *lr*. Este tipo de entrenamiento es a menudo muy lento para problemas prácticos. A continuación se describe algunos algoritmos de alto rendimiento los cuales convergen entre 10 a 100 veces más rápido que el algoritmo anterior. Todos estos algoritmos trabajan en modo *batch.*

- **Tasa de aprendizaje variable.** En los algoritmos anteriores la tasa de aprendizaje se mantiene constante durante el entrenamiento En este algoritmo se va variando la tasa de aprendizaje para que el orden de convergencia sea mayor. Si al evaluar la salida de la red el nuevo error excede el error anterior con una diferencia mayor a una tasa de cambio predefinida, los nuevos pesos son descartados y la tasa de aprendizaje es disminuida. Caso contrario se mantienen los nuevos pesos. Si el error es menor que el anterior, la tasa de aprendizaje es incrementada. La cadena que hay que introducir para implementar este tipo de entrenamiento es *traingda*.

- **Algoritmos de gradiente conjugada.** El algoritmo básico *backpropagation* ajusta los pesos en la dirección negativa de la gradiente, sin embargo esto no necesariamente produce una convergencia muy rápida. En los algoritmos de gradiente conjugada[25] se realiza una búsqueda a lo largo de las direcciones conjugadas, lo que produce generalmente una convergencia más rápida.

  - ✓ **Gradiente conjugada escalada.** Los algoritmos anteriores son computacionalmente costosos, ya que requieren que la red responda a todos los patrones de entrenamiento. La cadena de caracteres que se debe introducir para implementar este entrenamiento es *trainscg*.

Es muy difícil el conocer que algoritmo de entrenamiento será el más rápido para un problema dado. Esto dependerá de muchos factores, incluyendo la complejidad del problema, el número de patrones del conjunto de entrenamiento, el número de pesos y umbrales de la red y el error mínimo que se quiere obtener al finalizar el entrenamiento. En general en redes que contienen unos pocos cientos de pesos el algoritmo de Levenberg-Marquardt[26] convergerá más rápidamente. El método *quasi-Newton* [27] es el siguiente algoritmo más rápido en converger con redes de tamaño moderado.

[25] Moller, A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning
[26] MathWorks, *Neural Networks Toolbox User Guide*, Quinta ed., 1998. pag 167
[27] MathWorks, *Neural Networks Toolbox User Guide*, Quinta ed., 1998. pag 165

En la mayoría de situaciones se recomienda usar en primer lugar el algoritmo *Levenberg-Marquardt*. Si este algoritmo requiere mucha memoria se recomienda usar uno de los métodos de gradiente conjugada. Siguiendo esta recomendación en este proyecto se inició usando el algoritmo *Levenberg-Marquardt* para el entrenamiento de las redes neuronales tipo perceptrón multicapa; sin embargo debido a las dimensiones de los patrones de entrenamiento, este algoritmo usa una gran cantidad de memoria y el tiempo necesario para minimizar la función error es muy extenso. Finalmente se optó por usar el algoritmo de gradiente conjugada escalada que permite la minimización de la función de error en un tiempo mucho menor al del resto de tipos entrenamientos y reduce el consumo de memoria.

Para lograr un adecuado entrenamiento, es necesario almacenar una cantidad considerable de patrones que cumplan con las características de los caracteres segmentados en la imagen; se debe aclarar que los caracteres que se extraen de la imagen segmentada no siempre presentan las mismas características debido a los elementos ajenos en la placa, como los tornillos, las calcomanías, etc. Se puede apreciar un ejemplo de estas variedades de formas para un mismo carácter en la Figura 3.15.

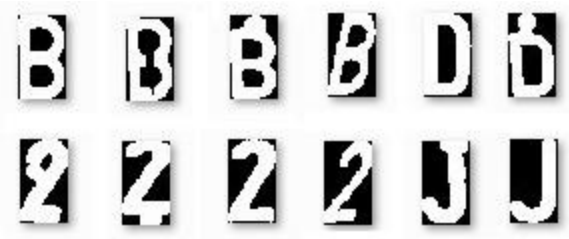


Figura.3. 15. Caracteres con diversas características.

3.2.1.5.Métodos que permiten mejorar la capacidad de generalización de la red implementados en Matlab®.

Uno de los problemas que ocurre durante el entrenamiento de la red es el sobre-aprendizaje el cual inhibe la capacidad de generalización de la red.

Uno de los métodos para evitar el sobre-aprendizaje es diseñar un entrenamiento con la extensión justa, pero es imposible saber de antemano cual debe ser el tamaño de la

muestra para una aplicación específica. Por ello existen dos métodos que se han implementado en Matlab® para mejorar la capacidad de generalización de la red. Así tenemos el método de *regularización*, y el de *parada temprana*.

- **Regularización.** Este método implica modificar la definición de la función error utilizada en el entrenamiento. Normalmente esta función se define como la media de los cuadrados de los errores en el conjunto de entrenamiento.

$$F = MSE = \frac{1}{N} * \sum_{i=1}^{N} {e_i}^2 \tag{3.1}$$

  Si a esta expresión se le añade un término que consiste en la media de los cuadrados de los pesos y umbrales de la red se mejora la generalización de la red, esta función de error toma el nombre de *"Error Medio Cuadrático Regularizado"*.

$$MSEREG = \gamma * MSE + (1 - \gamma) msw \tag{3.2}$$

$$msw = \frac{1}{n} * \sum_{j=1}^{n} w_j^2 \tag{3.3}$$

  Donde γ es un parámetro de optimización.

  Esta definición de la función error provoca que los pesos y umbrales de la red obtenidos sean pequeños, lo que fuerza una respuesta más suave de la red y hace menos probable el fenómeno de sobre-aprendizaje.

  En Matlab® se lo consigue asignando la cadena 'msereg' a la variable *net.perforFcn*. El problema de este método es conocer el valor optimo del parámetro γ , si es muy grande puedes provocar sobre-aprendizaje, si es demasiado pequeño, la red no se adecua bien a los datos de aprendizaje. Este valor se modifica mediante la siguiente línea de código: *net.performParam.ratio =0.5;*

- **Parada temprana.** Esta técnica divide los datos disponibles en tres subconjuntos. El primer subconjunto es el de entrenamiento que es usado para calcular el

gradiente y la actualización de los pesos y umbrales. El segundo subconjunto es el de validación. Normalmente el error de validación es monitorizado durante el proceso de entrenamiento. Usualmente este error va decreciendo a medida que transcurre el entrenamiento, pero en el caso de que se produzca sobre-aprendizaje, este error comienza a crecer. Cuando el error de validación comienza a crecer en un especificado número de iteraciones, se detiene el entrenamiento y se utilizan los pesos y umbrales de la iteración de menor error de validación.

El tercer subconjunto es el de prueba. No es usado durante el entrenamiento pero es útil para comparar distintos modelos. Puede ser útil también graficar este error durante el entrenamiento. Si este error muestra un mínimo en un número de iteraciones significativamente diferente al de error de validación, puede mostrar una pobre división de los datos.

Este método se utiliza conjuntamente con cualquiera de los otros métodos de entrenamiento descritos anteriormente.

3.2.1.6.Análisis de la red entrenada

A menudo es útil investigar la respuesta de la red con más detalle. Una manera es realizar un análisis de regresión entre la respuesta de la red y sus salidas patrón. Esto se realiza con ayuda de la función *postreg*, se debe introducir como parámetros la salida de la red y la salida esperada de la misma. Esta devuelve tres valores, los dos primeros corresponden a la pendiente y a la ordenada en el origen de la recta de regresión, y el tercer valor indica el grado de correlación de los datos. Si el grado de correlación de datos es igual a la unidad quiere decir que la respuesta de la red ante esas entradas es perfecta.

Resumiendo, después de una serie de pruebas realizadas variando los parámetros de diseño y entrenamiento de las redes perceptrón multicapa para las letras y los números se logró establecer como parámetros adecuados para cada una de las redes los siguientes:

- **Red de Letras**

Para la red de letras se establecieron los parámetros mostrados en la Tabla 3.1.

Tabla.3. 1. Parámetros de la red perceptrón multicapa clasificadora de letras

Parámetro	Valor
Número de capas ocultas	1
Número de neuronas en la capa oculta	26
Función de activación de la capa oculta	Sigmoidea Logarítmica
Función de error	Error Cuadrático Medio Regularizado
Valor de función de error a alcanzar	0.003
Parámetro de optimización γ	0.8
Número máximo de épocas	2000
Tipo de entrenamiento	Gradiente Conjugada Escalada.

En la Figura 3.16 se muestra la herramienta *Neural Network Training* en donde se puede apreciar los parámetros anteriormente descritos:

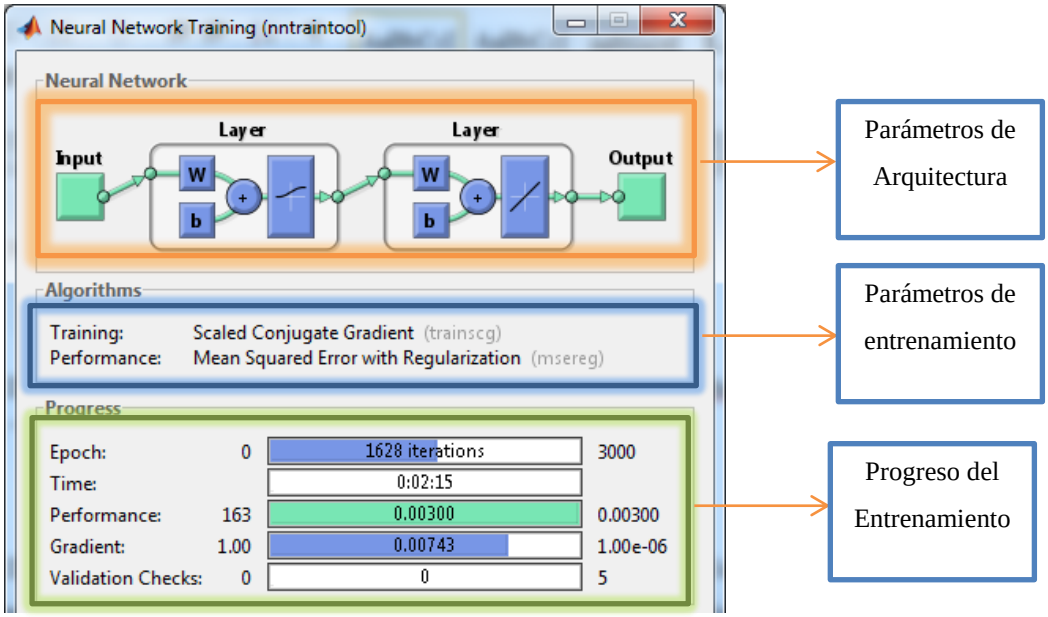


Figura.3. 16. Entrenamiento de red perceptrón multicapa (PM) para clasificación de letras.

En la Figura 3.17 se muestra la evolución de la función de error durante el entrenamiento hasta alcanzar el máximo error predeterminado.

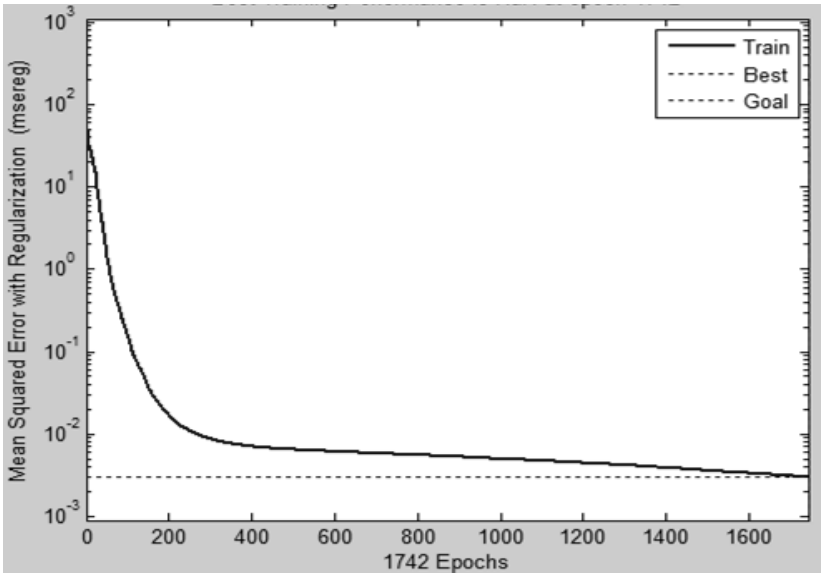


Figura.3. 17. Evolución de la función de error durante el entrenamiento de red de letras PM.

En la Figura 3.18 se observa el nivel de correlación que existe entre la red y los patrones de entrenamiento, indicando que la respuesta de la red ante esas entradas es perfecta.

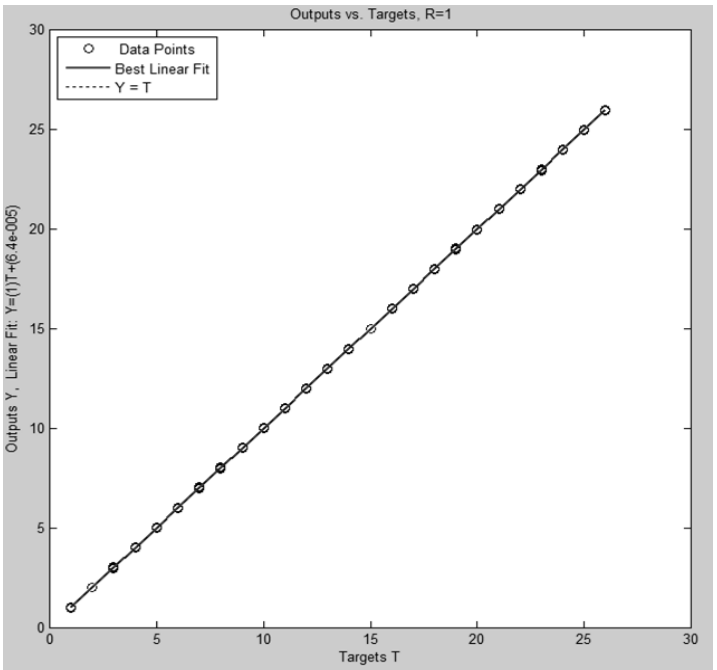


Figura.3. 18. Correlación entre la red creada y salidas patrón de red de letras PM.

- **Red de Números**

Para la red de números se establecieron los parámetros mostrados en la Tabla 3.2.

Tabla.3. 2. Parámetros de la red multicapa clasificadora de números

Parámetro	Valor
Número de capas ocultas	1
Número de neuronas en la capa oculta	13
Función de activación de la capa oculta	Sigmoidea Logarítmica
Función de error	Error Cuadrático Medio Regularizado
Valor de función de error a alcanzar	0.001
Parámetro de optimización γ	0.8
Número máximo de épocas	2000
Tipo de entrenamiento	Gradiente Conjugada Escalada.

En la Figura 3.19 se muestra la herramienta *Neural Network Training* el entrenamiento para esta red con los parámetros anteriormente establecidos:

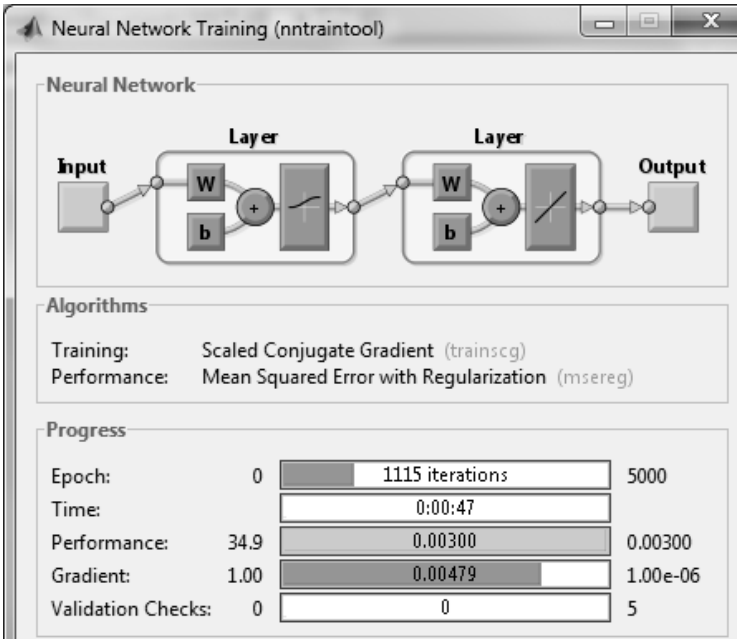


Figura.3. 19. Entrenamiento de red perceptrón multicapa para clasificación de números.

En la Figura 3.20 se muestra la evolución de la función de error durante el entrenamiento hasta alcanzar el máximo error predeterminado.

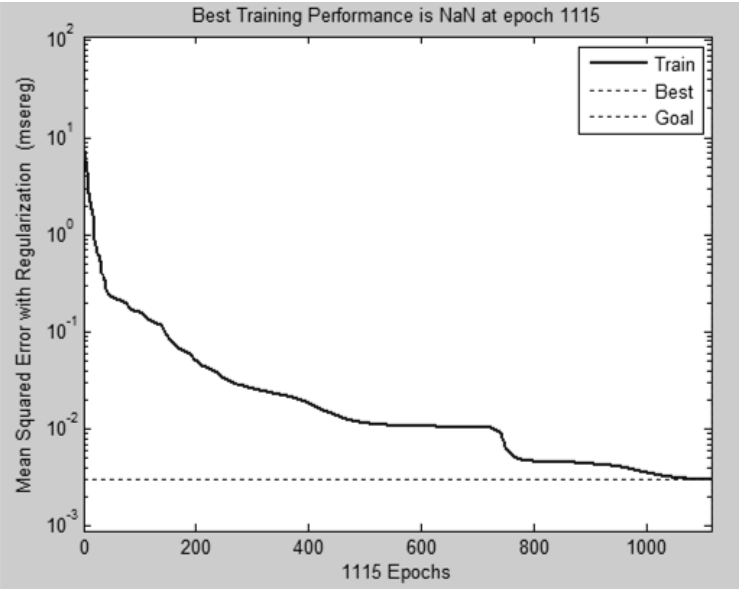


Figura.3. 20. Evolución de la función de error durante entrenamiento red de números PM.

En la Figura 3.21 se observa el nivel de correlación que existe entre la red y los patrones de entrenamiento para la red de números.

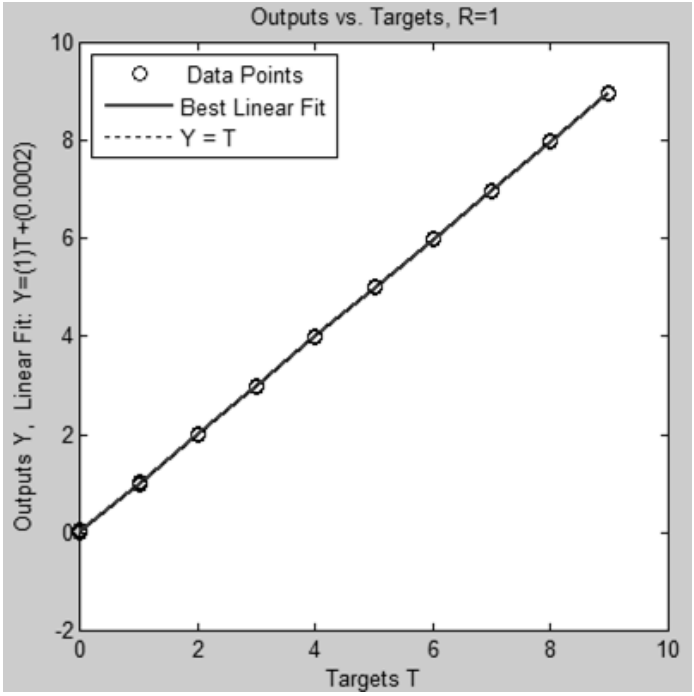


Figura.3. 21. Correlación entre la red creada y salidas patrón de red de números PM.

3.2.2. REDES NEURONALES DE BASE RADIAL

Las redes neuronales de base radial podrían requerir más neuronas que las redes neuronales perceptrón multicapa. Pero su entrenamiento es similar a estas. Trabajan mejor cuando hay muchos datos de entrenamiento disponibles. En el presente proyecto se creó dos redes neuronales de este tipo; una para la clasificación de las letras y otra para los números.

3.2.2.1.Creación de red

Las redes neuronales de base radial pueden ser implementadas por medio de dos funciones: *newrbe* y *newrb*.

- **Función newrbe.** Esta función puede producir una red con un error de valor cero sobre los patrones de entrenamiento. Su sintaxis es: *Net=newrbe(P,T,SPREAD)*

  Esta función toma como parámetros una matriz de vectores de entrada (P), sus correspondientes salidas patrón (T) y una constante *SPREAD*, que señala el área de actuación de cada neurona, y devuelve una red cuyos pesos y valores umbrales son tales que la red devuelve exactamente las salidas esperadas cuando las entradas son los patrones de entrenamiento.

  Esta función crea una red cuyo número de neuronas en la capa oculta es igual al número de entradas diferentes que se le proporciona a la red durante su entrenamiento. De esta manera cada neurona de esta capa actúa como un detector de un tipo de entrada en concreto.

  La constante SPREAD es muy importante ya que define el campo de actuación de cada neurona. Debe ser lo suficientemente grande para que las neuronas actúen correctamente en regiones de solapamiento provocando una respuesta suave de la red que resulte en una mejor generalización. Sin embargo si esta constante es muy grande todas las neuronas de la red responderían en el mismo campo del espacio de entradas. El problema con esta función es que si se necesitan demasiados vectores

de entrada para caracterizar correctamente la red, esta tendrá en su capa oculta demasiadas neuronas.

- **Función newrb.** Esta función plantea una forma más eficiente para crear la red de base radial. En este método, la función crea la red de forma iterativa, creando una neurona más en cada iteración. Las neuronas son añadidas hasta que el error medio cuadrático se ubique por debajo del parámetro *GOAL*, o se haya alcanzado un determinado número de neuronas máximo. La sintaxis de dicha función en Matlab® es: *net = newrb(P,T,GOAL,SPREAD).*

En el presente proyecto; luego de varias pruebas[28], las redes de base radial que brindaron una mejor solución a la clasificación de las letras y números se realizaron mediante la función *newrbe* con los siguientes parámetros:

- **Red de Letras**

Para la red de base radial clasificadora de las letras se usa los parámetros descritos en la Tabla 3.3.

Tabla.3. 3. Parámetros de la red de base radial clasificadora de letras

Parámetro	Valor
GOAL	0.001
SPREAD	10.5

En la Figura 3.22 se observa la evolución de la función de error para la red de base radial clasificadora de letras, se muestra que fueron necesarias 835 neuronas en la capa oculta para poder alcanzar una respuesta adecuada por parte de la red.

[28] Refiérase al Anexo 1. Informe de Pruebas sección pruebas redes de base radial.

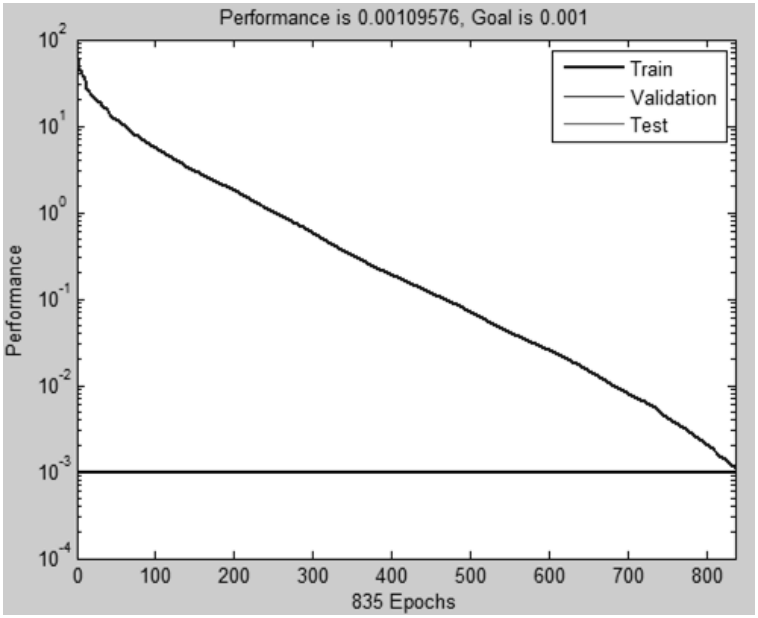


Figura.3. 22. Evolución de la función de error de la red de base radial de letras.

En la Figura 3.23 se observa el nivel de correlación que existe entre la red de base radial creada y los patrones de entrenamiento para la red de letras.

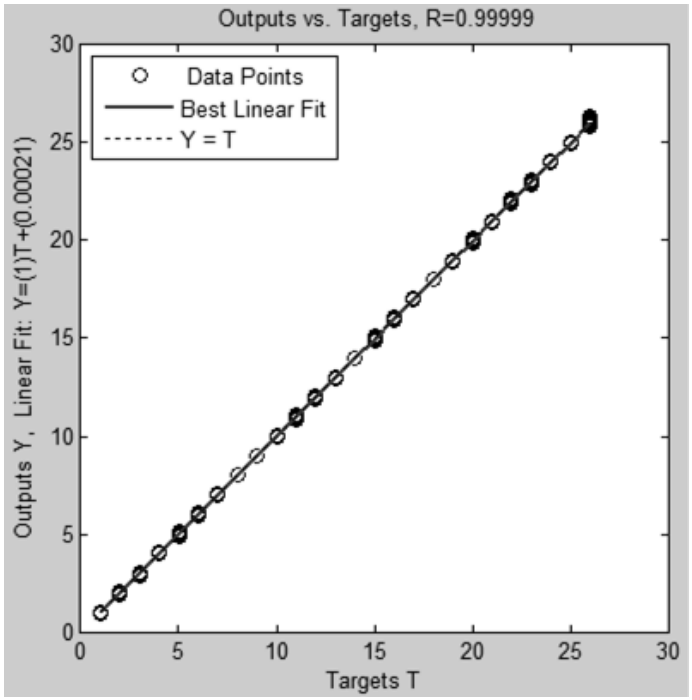


Figura.3. 23. Correlación entre la red de base radial creada y los patrones de salida de red de letras.

- **Red de Números**

Para la red de base radial clasificadora de números se toman los parámetros de la Tabla 3.4.

Tabla.3. 4. Parámetros de la red de base radial clasificadora de números

Parámetro	Valor
GOAL	0.001
SPREAD	20

En la Figura 3.24 se puede observar la evolución de la función de error para la red de base radial que clasificara a los números, ahí se observa que fueron necesarias 550 neuronas en la capa oculta para poder alcanzar una respuesta adecuada por parte de la red.

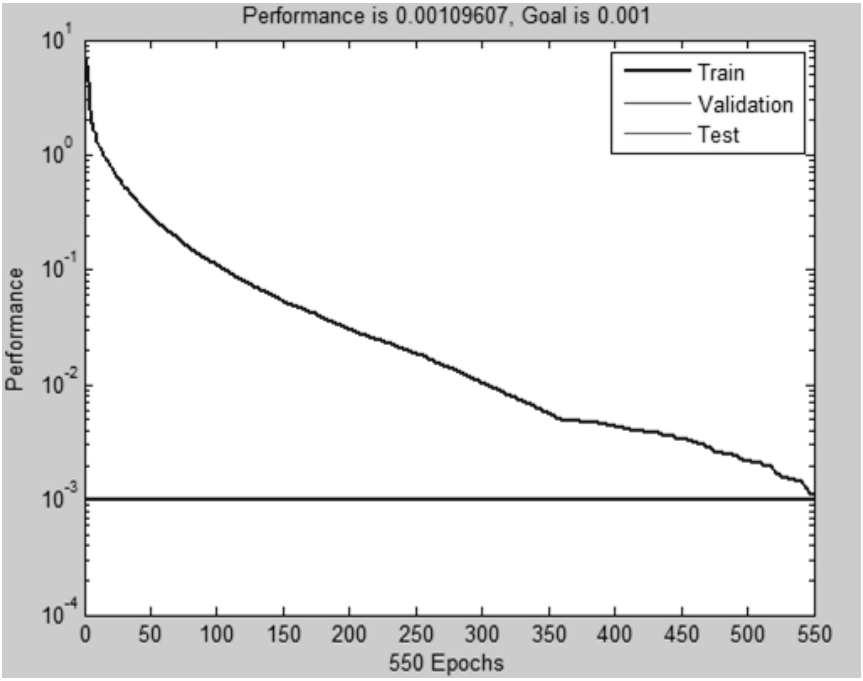


Figura.3. 24 Evolución de la función de error de la red de base radial de números.

En la Figura 3.25 se observa el nivel de correlación que existe entre la red de base radial creada y los patrones de entrenamiento para la red de letras.

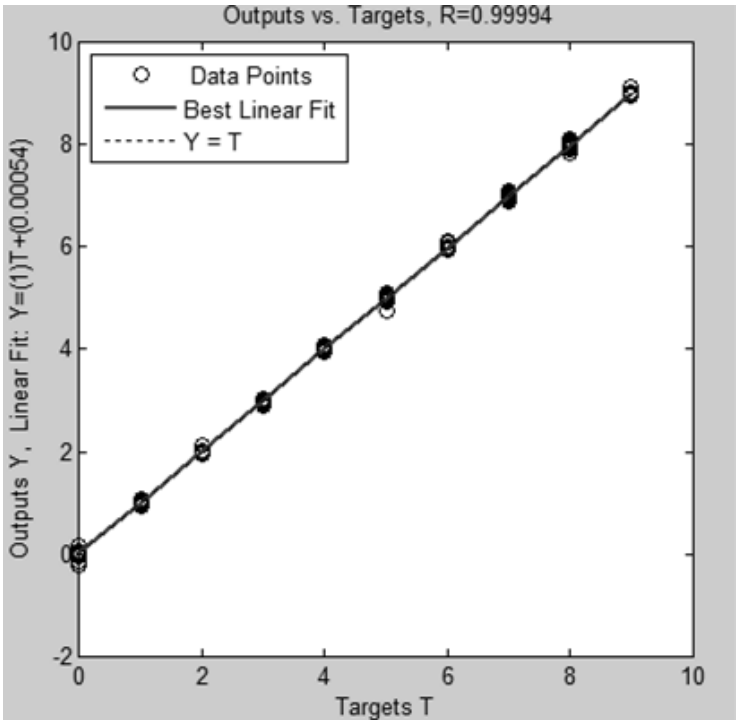


Figura.3. 25 Correlación entre la red de base radial creada y los patrones de entrenamiento de red de números.

3.2.3. REDES DE HOPFIELD

El objetivo es diseñar una red que almacene un conjunto de puntos de equilibrio tales que, cuando las condiciones iniciales sean presentadas, la red eventualmente alcance un punto diseñado.

La función de Matlab® que permite implementar este tipo de redes neuronales es *newhop*. Su sintaxis es: *Net=newhop(T);*

Donde T son los puntos de equilibrio deseados representados como una matriz. Esta función devuelve los pesos y los umbrales de la red. Una vez que la red ha sido diseñada, esta puede ser probada con uno o más vectores de entrada mediante la función *sim*. Cabe recalcar que el número de neuronas en la única capa de la red es igual a la longitud de los patrones de entrada.

Para el proyecto se implementó dos redes neuronales de este tipo una para las letras y otro para los números existentes en la placa. El entrenamiento para este tipo de red se lo realizó con 26 patrones diferentes para las letras y 10 patrones para la red de números. Se debe acotar que los patrones o puntos de equilibrio a los que se pretende llegar en esta red son una imagen que corresponde al promedio de todas las muestras presentes para cada carácter, así por ejemplo, el estado de equilibrio para la letra A es una imagen promedio de las 30 diferentes muestras presentes para este carácter.

3.3. DISEÑO DE APLICACIÓN DE CONTROL DE ACCESO

En el proyecto se creó una pequeña aplicación de control de acceso para vehículos, cuya finalidad es solamente demostrar el reconocimiento de las placas usando como ente clasificador las redes neuronales; para cumplir con este objetivo se diseñó una interfaz a través de la herramienta de creación de interfaces de usuario de Matlab® denominado GUIDE. Esta interfaz permite desarrollar ventanas, gráficos, botones, menús, etc. que tienen distintas funcionalidades y además permiten visualizar programas desarrollados en este entorno.

Figura.3. 26. Interfaz de aplicación de control de acceso.

En la Figura 3.26 se muestra la interfaz diseñada la cual tiene las siguientes funcionalidades:

El botón *Cargar Imagen* permite buscar en el directorio del computador y cargar una imagen del vehículo del cual se quiere reconocer la placa, automáticamente al escoger la imagen se inicia el proceso de reconocimiento de la placa, devolviendo los caracteres de la misma en la celda denominada *placa reconocida*. Luego de obtener los caracteres de la placa el programa automáticamente realiza la búsqueda en una base de datos para tener información del vehículo relacionado con ese número de placa, en caso de que la placa se encuentre registrada en la base de datos mostrara la información del dueño del vehículo y mostrara un cuadro de dialogo indicando que ese vehículo puede acceder; caso contrario muestra un cuadro indicando que el acceso del vehículo a esa zona es restringido. En el caso de que se tenga la necesidad de dar acceso a un vehículo nuevo el operador podrá ingresar sus datos en el campo *usuario* dentro del panel *Usuario*, posteriormente se deberá presionar el botón *Ingreso a base de datos*.

3.3.1. CONEXIÓN A LA BASE DE DATOS

Se realizó una base de datos desarrollada en MySql con los siguientes campos: *Número de Placa, Nombre de usuario.* Para la conexión hacia esta base de datos se utilizó el siguiente código:

```
conexión=database('mysql','root','admin');
```

Donde *mysql* es el nombre del conector ODBC asignado a la base de datos, *root* es el nombre de usuario y *admin* la contraseña de acceso a la base de datos. Para la inserción de datos se utiliza el siguiente código:

```
datosenv={placa,nombre,marca};
n_columnas={'Placa','Nombre','Marca'};
insert(conexión,'Informacion',n_columnas,datos_env);
```

Donde *conexión* es el nombre de la conexión a la base realizada anteriormente, *n_columnas* es el nombre de los campos a modificar en la base de datos, *datos_env* son los

valores a ser asignados a cada uno de los campos e *Información* es el nombre de la tabla en la cual serán almacenados los datos. Para la realización de consultas se utiliza el siguiente código:

```
Datos= fetch(conexión,[‘select Nombre  from Informacion where Placa=’’’’’ placa ‘’’’]);
```

4. CAPITULO 4

PRUEBAS Y RESULTADOS

4.1. DESCRIPCIÓN DE LAS PRUEBAS REALIZADAS

En el presente capítulo se ha dividido el proceso de reconocimiento de caracteres en tres etapas, que son:

- Extracción de la placa.
- Segmentación de caracteres de la placa.
- Clasificación de caracteres de la placa mediante redes neuronales.

Estableciéndose la eficiencia de cada etapa a lo largo de las pruebas realizadas. Para las tres etapas se utilizó dos conjuntos de pruebas, el primer conjunto lo conforman 350 imágenes, entre las que se hallan las 250 utilizadas para realizar el entrenamiento de las redes; el segundo conjunto lo forman 100 imágenes indiferentes al proceso de entrenamiento, las cuales nos permitirán determinar el nivel de generalización alcanzado por las redes. Las imágenes utilizadas para realizar la evaluación del rendimiento de cada una de las etapas fueron tomadas a una distancia entre 1 y 3 metros del vehículo, tanto de la parte frontal como trasera, y a una altura entre 0.5 y 1.5 metros.

4.2. EXTRACCIÓN DE LA PLACA.

Las pruebas para la extracción se las realiza por medio de la función *Extraccion(I)*, en la cual se halla implementado el proceso para extraer la región a la que debe pertenecer la placa, el parámetro *I* corresponde a la imagen del auto previamente adquirida.

Sobre el primer conjunto presentado de 350 imágenes, en 348 casos la región correspondiente a la placa fue extraída correctamente obteniéndose un 99.42% de eficiencia sobre este conjunto. Al aplicar el segundo conjunto de imágenes en 99 de ellas se realizó una correcta extracción de la zona de la placa, obteniéndose un 99% de eficiencia sobre este conjunto. A continuación se calcula la eficiencia de este proceso tomando el promedio de los resultados de las dos pruebas. Esto se puede observar con más detalle en la Tabla 4.1.

Tabla.4. 1. Resultados de Pruebas realizadas para la etapa de extracción de la placa.

Conjunto de Prueba	Extracción Correcta	Extracción Incorrecta	% Eficiencia
350 imágenes	348	2	99.42
100 imágenes	99	1	99
TOTALES	**447**	**3**	**99,33**

En la Figura 4.1 se muestra una representación gráfica del porcentaje de placas extraídas correctamente durante las dos pruebas realizadas.

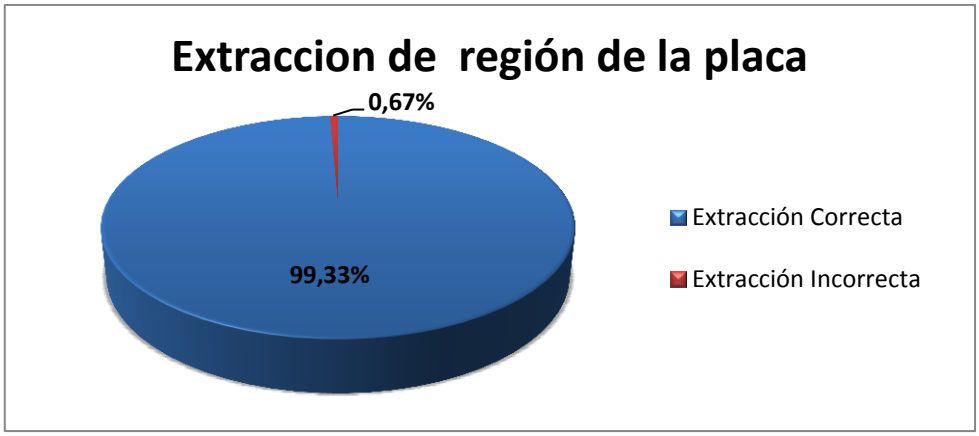


Figura.4. 1. Eficiencia total de la etapa de extracción de la placa.

Con el fin de mejorar la eficiencia de esta etapa en futuros proyectos se debería implementar un método que permita determinar si la placa se encuentra de frente con cierto ángulo de inclinación o si esta girada sobre un eje referencial ubicado en la primera línea vertical característica del rectángulo formado por la placa, logrando establecer cada una de las dos situaciones para su posterior corrección.

4.2.1. Casos Erróneos

Los casos en los cuales no se pudo realizar una correcta extracción de la placa se deben a presencia dos factores importantes:

1. Un alto nivel de luminosidad sobre el vehículo provocando un realce de contraste en ciertos objetos del vehículo. Como se puede observar en la Figura 4.2 entre las regiones candidatas se encuentra la zona perteneciente a la placa, sin embargo esta no es escogida ya que existe un elemento que presenta un alto contraste en una posición inferior a esta.

Figura.4. 2. Extracción incorrecta de la placa debido a un alto nivel de luminosidad.

2. Presencia de pintura de colores claros sobre la calzada, este material provoca que exista un alto contraste entre el asfalto y la pintura, tornándose en una zona candidata que podría contener la placa. Como se puede observar en la Figura 4.3 aunque la zona de la placa esta entre las regiones candidatas, esta no s escogida debido a que la región con pintura se encuentra más baja.

Figura.4. 3. Extracción incorrecta debida a la presencia de pintura en la calzada.

4.3. SEGMENTACIÓN DE LOS CARACTERES DE LA PLACA

Para evaluar la eficiencia de esta etapa se utiliza la función *segmentacion(Iplaca),* la cual toma como parámetro la región de la placa extraída previamente por la función *extracción*.

De la misma manera que en la etapa anterior se tomó el total de placas extraídas correctamente en la etapa de extracción, y se hizo una evaluación con el fin de determinar qué tan eficiente es el proceso de segmentación de caracteres. Del primer conjunto de prueba formado por 348 imágenes, se realizó una correcta segmentación de los caracteres de 327 casos, teniendo 21 casos en donde la extracción es errónea. Del segundo conjunto de 99 imágenes se obtuvo 93 casos de segmentación correcta y 6 de segmentación incorrecta.

Al igual que para la etapa anterior se calcula la eficiencia del proceso de segmentación, en la Tabla 4.2 se pueden observar los resultados de las pruebas más claramente.

Tabla.4. 2. Resultados de las pruebas de la etapa de segmentación

Conjunto de Prueba	Segmentación Correcta	Segmentación Incorrecta	% Eficiencia
348 imágenes	327	21	93.97
99 imágenes	93	6	93.94
TOTALES	**420**	**27**	**93,96**

En la Figura 4.4 se observa el porcentaje de eficiencia total para la etapa de segmentación de caracteres.

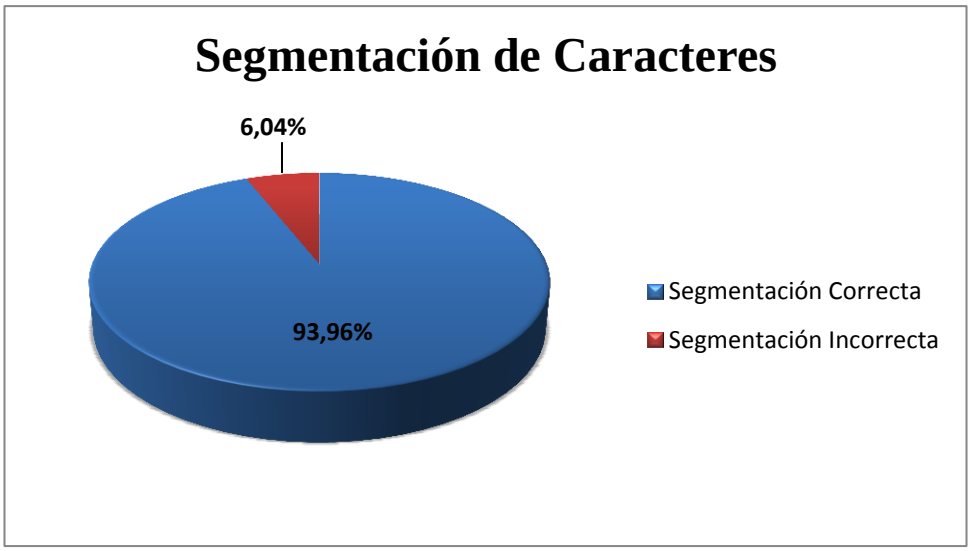


Figura.4. 4. Eficiencia total de la etapa de segmentación de caracteres.

4.3.1. Casos Erróneos

Los casos en los cuales se realizó una mala segmentación se deben a varios factores entre los cuales se encuentran los siguientes:

1. Placa en mal estado. En los casos en que la placa se encuentra deteriorada o sucia, no se logra determinar objetos que pueden pertenecer a los caracteres. En la Figura 4.5 se observa un ejemplo para estos casos mostrando que el resultado al aplicar el proceso de segmentación es erróneo ya que toma todo el conjunto de la placa como un objeto sin poder separar cada uno de los caracteres.

Figura.4. 5. Segmentación incorrecta por placa en mal estado.

2. Mascarillas mal colocadas sobre las placas. Existen casos en los cuales la mascarilla cubre una sección de los caracteres, provocando la unión entre varios de estos, a los cuales el programa toma como un solo objeto que no cumple con los criterios de discriminación. En la Figura 4.6 se observa un ejemplo de estos casos, mostrando el resultado en el proceso de segmentación, el cual une varios objetos como uno solo evitando de esta manera que el programa escoja este objeto como un caracter ya que este nuevo objeto no cumple con las características preestablecidas.

Figura.4. 6. Segmentación incorrecta de caracteres por presencia de mascarilla mal colocada.

3. Bordes de la placa mal procesados que cumplen con características similares a la de los caracteres. Este problema se presenta en los casos en que la imagen del vehículo no es adquirida de manera frontal, al momento de la corrección del ángulo existen elementos que cumplen con las características discriminatorias de este proceso. En la Figura 4.7 se muestra un ejemplo de estos casos en el que se puede observar que se toma un objeto extra como parte de los caracteres de la placa.

Figura.4. 7. Segmentación incorrecta de caracteres por bordes mal procesados.

4.4. CLASIFICACIÓN DE CARACTERES MEDIANTE REDES NEURONALES

Cada una de las pruebas realizadas sigue una secuencia preestablecida, la cual se detalla a continuación.

1. Se realiza el entrenamiento con los parámetros establecidos en el capítulo anterior para cada tipo de red.

2. Se evalua su desempeño mediante la ejecución del programa completo de reconocimiento de caracteres de placas sobre cada uno de los dos conjuntos de datos de pruebas, que superaron las dos etapas anteriores, es decir, 327 para el

primer conjunto de prueba y 93 para el segundo y su resultado es almacenado dentro de una base de datos.

3. Se exporta los datos desde la base de datos hacia el programa Microsoft Excel, aquí se realiza la determinación de su eficiencia manualmente. Para cumplir con este objetivo una vez realizada la prueba se evalua las redes creadas determinando cuantos números y letras son clasificados correctamente, obteniendose de esta manera el porcentaje de caracteres reconocidos al usar determinada configuración de la red neuronal. A continuación se muestra un ejemplo de como se realiza este proceso:

Tabla.4. 3. Muestra de Evaluación de pruebas de redes neuronales

PLACA ORIGINAL	PLACA RECONOCIDA	# AUTO	EFICIENCIA	C.CORRECTOS	# CARACTERES
GLT219	GLT219	1	100	6	6
PBL1824	PBL1824	2	100	7	7
PBN7300	PBN7300	3	100	7	7
PQW210	PQW210	4	100	6	6
PWL271	PVL777	5	50	3	6
PBL8003	PBL8003	6	100	7	7
PVF830	PVF830	7	100	6	6
PBO9087	PBQ9087	8	85,71	6	7
PVF830	PVF830	9	100	6	6
PXZ937	PXZ937	10	100	6	6

Una vez realizado este proceso sobre todas las muestras presentes se calcula el promedio de la eficiencia de cada una de las redes, estableciendo de esta manera la eficiencia para las redes creadas.

4.4.1. REDES PERCEPTRÓN MULTICAPA

Al culminar el proceso de evaluación sobre el primer conjunto de pruebas se obtuvo una eficiencia total en la clasificación de caracteres de 96.94%. De la misma manera sobre el segundo conjunto de prueba se obtuvo una eficiencia total del 92.11%. En la Tabla 4.4 se puede observar la eficiencia de cada una de estas pruebas.

Tabla.4. 4. Eficiencia de las redes perceptrón multicapa en la clasificación de caracteres

Conjunto de Prueba	Eficiencia
327 Imágenes	96.94%
93 Imágenes	92.11%
TOTAL	94.53%

En la Figura 4.8 se puede observar la eficiencia total sobre este conjunto de pruebas al usar redes neuronales perceptrón multicapa.

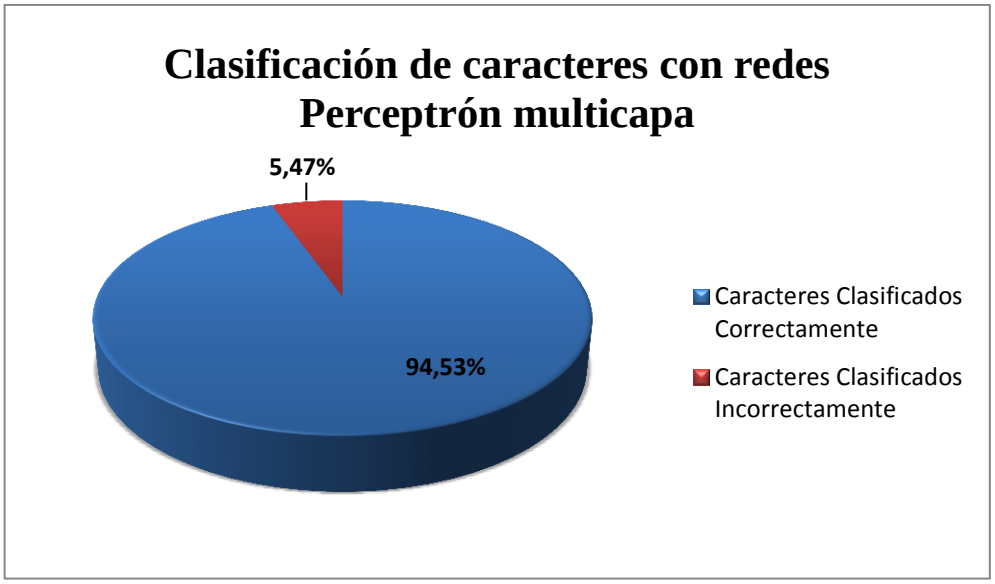


Figura.4. 8. Eficiencia de la etapa de clasificación de caracteres utilizando redes perceptrón multicapa.

4.4.2. REDES DE BASE RADIAL

De la misma manera que se realizó para las redes multicapa se procede a evaluar los dos conjuntos de pruebas para las redes de base radial diseñadas en el capítulo 3, para el primer conjunto se obtuvo una eficiencia total en la clasificación de caracteres de 81%. De la misma manera sobre el segundo conjunto de prueba se obtuvo una eficiencia total del 77.24%. En la Tabla 4.5 se puede observar la eficiencia de cada una de estas pruebas.

Tabla.4. 5. Eficiencia de las redes de base radial en la clasificación de caracteres

Conjunto de Prueba	Eficiencia
327 Imágenes	81%
93 Imágenes	77.24%
TOTAL	79,12%

En la Figura 4.9 se puede observar la eficiencia total sobre este conjunto de pruebas al usar redes neuronales de base radial.

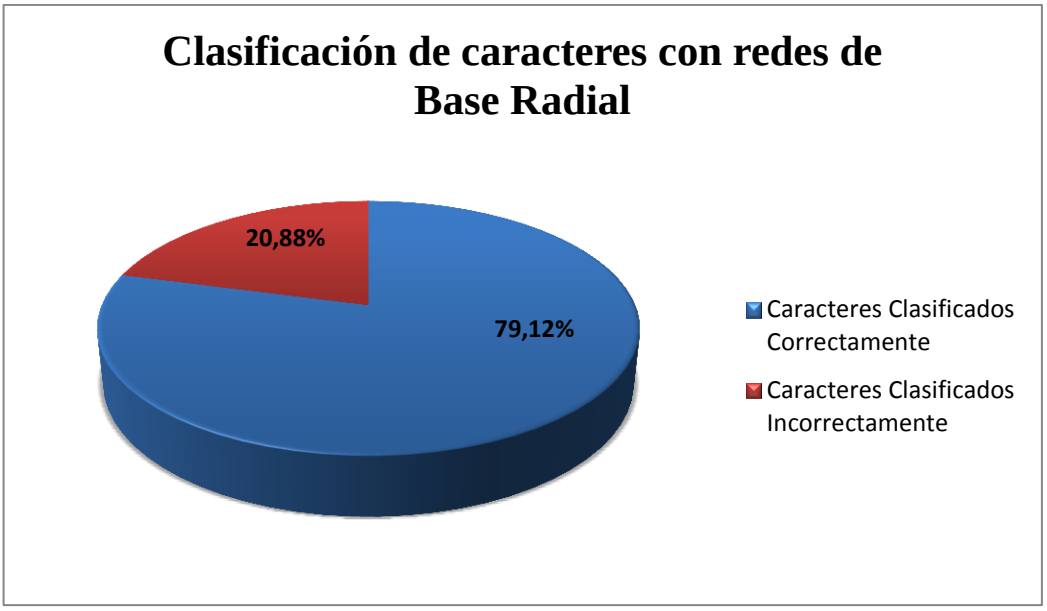


Figura.4. 9. Eficiencia de la etapa de clasificación de caracteres utilizando redes de base radial.

4.4.3. REDES TIPO HOPFIELD

De la misma manera se procede a evaluar los dos conjuntos de pruebas para las redes tipo Hopfield diseñadas en el capítulo anterior. Para el primer conjunto se obtuvo una eficiencia total en la clasificación de caracteres de 96.42%. De la misma manera sobre el segundo conjunto de prueba se obtuvo una eficiencia total del 94,92%. En la Tabla 4.6 se puede observar la eficiencia de cada una de estas pruebas y la eficiencia promedio sobre las dos pruebas.

Tabla.4. 6. Eficiencia de las redes perceptrón multicapa en la clasificación de caracteres

Conjunto de Prueba	Eficiencia
327 Imágenes	96.42%
93 Imágenes	94,92%
TOTAL	95,67%

En la Figura 4.10 se puede observar la eficiencia total sobre este conjunto de pruebas al usar redes neuronales Hopfield.

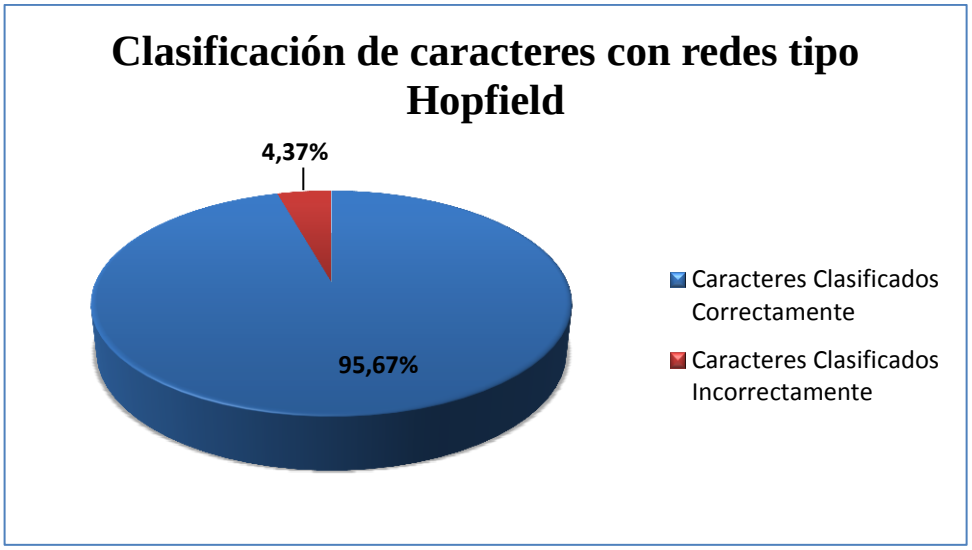


Figura.4. 10. Eficiencia de la etapa de clasificación de caracteres utilizando redes tipo Hopfield.

4.4.4. ANÁLISIS DE LAS REDES CREADAS

Para el desarrollo de la aplicación de control de acceso se toma como redes clasificadoras de caracteres las que presenten un mayor rendimiento.

La redes perceptrón multicapa presentan un alto rendimiento sobre los conjuntos de pruebas a los que fueron sometidas, probando de esta manera que poseen una buena capacidad de generalización. Además de que su costo computacional es bajo, ya que solo se necesitan 26 y 10 neuronas ocultas para las redes de letras y números respectivamente.

Las redes de base radial no presentan muy buenos resultados, especialmente en la red de letras, esto debido a que no se pudo definir un parámetro SPREAD óptimo para esta red durante las pruebas realizadas para determinar la arquitectura de esta red. Además poseen un alto costo computacional al poseer 835 y 550 neuronas ocultas para las redes de letras y números respectivamente. Además el tiempo que ocupa para realizar el entrenamiento de las redes es alto, manteniéndose en un rango de 20 a 30 minutos.

Las redes tipo Hopfield presentan un alto porcentaje de eficiencia en las pruebas realizadas sin embargo, en algunos casos en que los caracteres presentan cierta inclinación la red no responde adecuadamente. Sin embargo poseen un alto costo computacional ya que están formadas por 1008 neuronas en la única capa que poseen y para clasificar los caracteres es necesario realizar una correlación entre la imagen del carácter obtenida como salida y la imagen almacenada como recuerdo.

En la Tabla 4.7 se muestra la eficiencia de cada uno de los tipos de redes utilizados para la clasificación de caracteres.

Tabla.4. 7. Eficiencia en la clasificación de caracteres por cada una de las redes.

Tipo de Red	Eficiencia
Perceptrón Multicapa	94.53%
Base Radial	79%
Hopfield	95.67%

En la Figura 4.11 se muestra una representación gráfica de los valores mostrados en la Tabla 4.7.

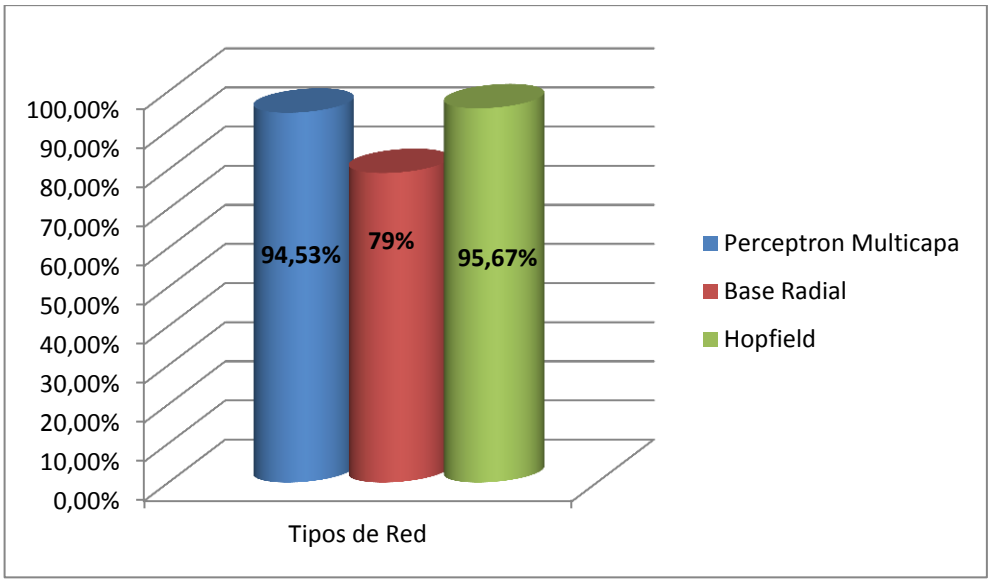


Figura.4. 11. Grafica comparativa de eficiencia de tipos de red utilizados en el proyecto.

Del análisis realizado anteriormente se desprende que las redes que mejores resultados presentan son las redes perceptrón multicapa, tomando estas redes como adecuadas para la aplicación de control de accesos.

5. CAPITULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

- Se logró diseñar una aplicación de control de acceso para vehículos mediante el uso de herramientas de procesamiento digital de imágenes, redes neuronales y base de datos, logrando resultados satisfactorios en la detección de la placa vehicular. Esta aplicación posee un alto nivel de usabilidad por parte del operador debido a su sencillez.

- El procesamiento de imágenes basado en operaciones morfológicas resultó eficiente ya que se pudo discriminar la zona de la placa basándose en las características morfológicas de la misma. Además las funciones incluidas en el toolbox de Matlab® para procesamiento digital de imágenes fueron de mucha ayuda, ya que facilitan el trabajo a realizar.

- Del análisis de pruebas se concluye que la etapa de segmentación de caracteres es la que presenta un mayor porcentaje de error, ya que la eficiencia de esta etapa es directamente proporcional al estado en que se encuentra la matrícula del vehículo, ya que muchas de estas poseen objetos ajenos a las características de la placa; adicionalmente, es importante resaltar que el grado de inclinación de la cámara también afecta la eficiencia del sistema.

- Las redes neuronales que mejores resultados presentaron fueron las tipo Hopfield y las perceptrón multicapa, a pesar de poseer una arquitectura y modo de funcionamiento diferentes, los dos tipos presentaron una buena capacidad de generalización ya que con 250 imágenes patrones usadas para el entrenamiento se obtuvo un buen rendimiento para las 450 imágenes de prueba, demostrando que el conjunto de aprendizaje cumple los requerimientos de ser significativo y representativo.

- En el caso de las redes neuronales de función de base radial, no se pudo crear una red cuyos resultados sean muy satisfactorios ya que es muy difícil determinar el valor de la constante de desviación para las funciones de activación.

- Para la aplicación realizada se tomó como agentes de clasificación las redes neuronales perceptrón multicapa, debido a un factor muy importante que es el no necesitar de un número excesivo de neuronas para resolver un problema como en el caso de las redes de base radial y tipo Hopfield, esto es debido a que entre mayor sea el número de neuronas mayor es el número de pesos sinápticos de la red.

- Debido a la no existencia de un método matemático que permita determinar la arquitectura óptima para los diferentes tipos de redes neuronales, la determinación de la misma se la realiza de manera heurística, obteniéndose resultados adecuados para solucionar un problema, sin ser esta necesariamente la solución más óptima. Siendo el proceso de diseño de la red altamente costoso en cuanto a tiempo se refiere.

- Para obtener un mayor rendimiento por parte de las redes neuronales, se debe incrementar el número de patrones de entrenamiento, debido a que entre mayor sea éste número, la red de adaptará sus pesos de forma más eficaz para la solución del problema.

- A pesar de que el tamaño y tipo de letra utilizados en las placas vehiculares son normados por la Comisión Nacional de Transito, se encontró casos en los que estas poseían otro tipo de letra lo que dificulta el proceso de clasificación.

5.2. RECOMENDACIONES

- Para una mayor eficiencia del sistema, se recomienda que las imágenes capturadas se encuentren dentro de la distancia máxima descrita en el proyecto ya que si el vehículo se encontrara a una mayor distancia la zona de la placa ocuparía un menor espacio dentro de la imagen y seria eliminada por parte del programa. Además, las imágenes adquiridas deben tener un tamaño de 1200 x 1600 pixeles.

- Otro punto importante para mejorar la eficiencia es la posición de la cámara, si está ubicada de manera frontal a la placa del vehículo se tendrá un mejor rendimiento. Además el lugar de la captura de la imagen debe tener condiciones de iluminación lo más homogéneas posibles.

- Una posible mejora al presente proyecto se la puede realizar en la etapa de segmentación de caracteres, realizando una correcta corrección del ángulo de inclinación de los caracteres. De esta forma solo se tendrá caracteres en posición totalmente vertical lo cual facilitara a su vez la clasificación de los mismos por parte de las redes neuronales.

- Para obtener un mayor rendimiento por parte de las redes de base radial se puede trabajar con caracteres cuyo tamaño sea menor al tomado para este proyecto, ya que este tipo de redes responden de mejor manera cuando el tamaño de los patrones de entrada es pequeño.

BIBLIOGRAFÍA

1. **Zubizarreta, Asier.** *Aplicación de las técnicas de redes neuronales para el diagnóstico on-line del proceso de electroerosión por hilo.* 11 de Octubre de 2006. [Citado el: 28 de Noviembre de 2011.] http://www.disa.bi.ehu.es/spanish/profesores-etsi-bilbo/~jtpcaaxi/PFC/wwwANN/informacion_de_contacto.htm.

2. **www.quercus.biz.** The Automatic Number Plate Recognition Tutorial. 15 de Agosto de 2006. [Citado el: 20 de Julio de 2001.] http://www.anpr-tutorial.com/.

3. **Vinuela, Pedro Isasi y Leon, Ines M. Galvan.** *Redes Neuronales Artificiales. Un enfoque practico.* s.l. : Prentice Hall.

4. **Villanueva Espinoza, María del Rosario.** Las Redes Neuronales Artificiales y su Importancia como Herramienta en la Toma de Desiciones. 2002. [Citado el: 26 de Octubre de 2011.]

http://sisbib.unmsm.edu.pe/Bibvirtual/tesis/Basic/Villanueva_EM/Contenido.htm.

5. **Vera, Alexander, y otros, y otros.** Diseño de un Sistema de Seguridad Basado en Procesamiento de Imágenes para el Acceso Vehicular a un Campus. *INGENIUM.* Bogota : Facultad de Ingeniería de la Universidad de San Buenaventura, 2010, Vol. 11.

6. **Velappa , Ganapathy y Wen Lik , Dennis LUI.** A Malaysian Vehicle License Plate Localization and Recognition System. 2007. [Citado el: 18 de Agosto de 2011.] http://www.freewebs.com/dennislui/JSCI%20Paper.pdf.

7. **Tasiguano Pozo, Cristian.** *Desarrollo de Algoritmos de Reconocimiento de Placas de Vehiculos.* Quito : Escuela Politecnica Nacional, 2011.

8. **Sanz Molina, Alfredo y del Brio, Bonifacio Martin.** *Redes Neuronales y Sistemas Difusos.* Mexico, DF : Alfaomega, 2002.

9. **Santillan, Ivan Danilo Garcia.** *Vision Artificial y Procesamiento Digital de Imagenes usando matlab.* Ibarra : Pontificia Universidad Catolica del Ecuador, 2008.

10. **Salazar, Jorge Enrique.** *Diseño e Implementacion de un Sistema de Entranamiento en Redes Neuronales Utilizando el Software Neusosystems de Siemens.* Sangolqui : Escuela Politecnica del Ejercito, 2009.

11. **S. Adebayo, Daramola.** Automatic Vehicle Identification System using License Plate. Febrero de 2011. [Citado el: 19 de Agosto de 2011.]

http://www.ijest.info/docs/IJEST11-03-02-240.pdf.

12. **Romano, Pablo David.** *Reconocimiemto de Patentes de Automovil.* Buenos Aires : Universidad de Buenos Aires, 2007.

13. **Rodríguez, Fernando Martín y Fernández Hermida, Xulio.** RAMA: Reconocedor Automático de Matrículas de Automóviles. [Citado el: 10 de Julio de 2011.] http://webs.uvigo.es/gpi-rv/ficheros/pub/papers/tiarp01.pdf.

14. **Phalgun Pandya, Mandeep Singh.** Morphology Based Approach To Recognize Number Plates in India. Julio de 2011. [Citado el: 8 de Septiembre de 2011.] http://www.ijsce.org/attachments/File/Vol-1_Issue-3/C072071311.pdf.

15. **Palacios, Rafael y Amar, Gupta.** *Sistema de reconocimiento de caraceres para lectura automatica de cheques.* 2003.

16. **Ozbay, Serkan y Ercelebi, Ergun.** Automatic Vehicle Identification by Plate Recognition. 2005. [Citado el: 12 de Julio de 2011.]

http://www.waset.org/journals/waset/v9/v9-41.pdf.

17. **Ni, Dong Xiao.** Application of Neural Networks to Character Recognition. 4 de Mayo de 2007. [Citado el: 20 de Julio de 2011.]

http://csis.pace.edu/~ctappert/srd2007/c4.pdf.

18. **Moreno, Gustavo Adolfo.** *Diseño e Implementacion de un Sistema Basado en una Red Neuronal no Supervisada para el Control de Moviminetos de un Robot Movil.* Quito : Escuela Politecnica del Ejercito, 2005.

19. **MathWorks.** *Neural Network ToolBox, User Guide.* 1998.

20. Neural Network Toolbox, User Guide. 1998.

21. Image Processing Toolbox. [Citado el: 15 de Agosto de 2011.] http://www.mathworks.com/help/toolbox/images/.

22. **Martinsky, Ondrej.** *Recognition of Vehicle Number Plates.* Berno : Universidad de Tecnologia de Berno, 2007.

23. **Martinez, E.** *Procesamiento Digital de Imagenes.* Mexico D. F. : Universidad Nacional Autonoma de Mexico, 2005.

24. **Mamedov, Fakhraddin y Jamal Fathi , Abu Hasna.** Character Recognition using Neural Networks. 2006. [Citado el: 2 de Agosto de 2011.] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.86.2207.

25. **Isasi Viñuela, Pedro y Galvan Leon, Ines M.** *Redes Neuronales Artificiales. Un Enfoque Practico.* Madrid : Prentice Hall, 2004.

26. **Haykin, Simon.** *Neural Networks and Learning Machines.* New Jersey : Prentice Hall, 2009.

27. **Gonzales, Rafael, Woods, Richard y Eddins, Steven.** *Digital Image Processing Using MATLAB.* New Jersey : Prentice Hall, 2009.

28. **Fernández Hermida, Xulio, Rodríguez, Martin y Fernández Lijó, José Luis.** An O.C.R. for V.L.P.'s (Vehicle License Plate). [Citado el: 12 de Septiembre de 2011.] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.141.6690.

29. **Fakhraddin, Mamedov y Hasna, Jamal Fathi Abu.** *CHARACTER RECOGNITION USING NEURAL NETWORKS.*

30. **Faaborg, Alexander J.** *Using Neural Networks to Create an Adaptive Character.* 2002.

31. **Duarte, Fernando.** Control de Robots Redundantes. [Citado el: 11 de Noviembre de 2011.] http://www.ipv.pt/millenium/Millenium24/8.pdf.

32. **Cuevas Jimenez, Erik y Zaldivar Navarro, Daniel .** Visión por Computador utilizando MatLAB y el Toolbox de Procesamiento Digital de Imágenes. 2007. [Citado el: 25 de Julio de 2011.]

http://www.fileden.com/files/2007/10/21/1529625/MATLAB_PSOC/vision%20artificial.pdf.

33. **Constenla, Gabriel Roberto.** *Reconocimiento Óptico de Dígitos con Redes.* Buenos Aires : Universidad de Belgrano, 2010.

34. **Cardenas, Paul, y otros, y otros.** *Diseño de Sistema de Reconocimiento de Placas Utilizando Matlab.* Mexico, D. F. : Instituto Politecnico Nacional, 2009.

35. **Blanchet, Gérard y Charbit, Maurice .** *Digital Signal and Image Processing using MATLAB.* Londres : ISTE Ltd., 2006.

36. **BINTI MUSTAFA, RABI'ATUL ADAWIYAH.** *Automatic Car License Plate Recognition System (CLPR).* RABI'ATUL ADAWIYAH BINTI MUSTAFA : Universidad Malaysia Pahang, 2005.

37. **Basogain Olabe, Xabier.** *Redes Neuronales Artificiales y sus Aplicaciones.* Bilbao : Escuela Superior de Ingenieria de Bilbao, 2008.

38. **Barragán Guerrero, Diego Orlando.** Manual de Interfaz Gráfica de Usuario en Matlab. 25 de Mayo de 2008. [Citado el: 15 de Agosto de 2011.] http://www.dspace.espol.edu.ec/bitstream/123456789/10740/19/%255Bmatlab%255D_MATLAB_GUIDE.pdf.

39. **Arnau Prieto, Raúl .** *Sistema de Control de Tiempo Real Basado en Reconocimiento de Imagenes.* Cantabria : Universidad de Cantabria, 2005.

40. **Araokar, Shashank.** *Visual Character Recognition Using Artificial Neural Network.*

41. **Anish Lazrus, Siddhartha Choubey.** A Robust Method of License Plate Recognition using ANN. 2011. [Citado el: 2 de Agosto de 2011.] http://www.ijcsit.com/docs/Volume%202/vol2issue4/ijcsit2011020426.pdf.

42. **Anil K. Jain, Jianchang Mao, K. M.Mohiuddin.** *Artificial Neural Networks: A Tutorial.* 1996.

43. **Alfonso, Ballesteros.** Tutorial introduccion a las Redes Neuronales. http://www.redes-neuronales.netfirms.com/tutorial-redes-neuronales/tutorial-redes.htm.

44. **Aldabas-Rubira, Emiliano.** Introducción al Reconocimiento de Patrones Mediante Redes Neuronales. 2002. [Citado el: 14 de Octubre de 2011.] http://www.jcee.upc.es/JCEE2002/Aldabas.pdf.

45. **Universidad Nacional de Quilmes.** Segmentacion Por Umbralizacion. Octubre de 2005. [Citado el: 8 de Agosto de 2011.]

http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Segmentación por umbralización - Método de Otsu.pdf.

46. Redes de Neuronas. 2011. [Citado el: 1 de Noviembre de 2011.]

http://www.redesdeneuronas.es/.

47. **Universidad Don Bosco.** Reconocimiento de caracteres. [Citado el: 16 de Julio de 2011.]

http://www.udb.edu.sv/Academia/Laboratorios/electronica/Redes%20neuronales/guia7RN.pdf.

48. Procesamiento Global empleando la Transformada de Hough. Universidad nacional de Quilmes, 2005. [Citado el: 3 de Septiembre de 2011.] http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Transformada%20de%20Hough.pdf.

49. **MathWorks.** *Neural Networks Toolbox User Guide.* Quinta. 1998.

50. Elementos del Procesado Morfologico. [Citado el: 10 de Octubre de 2011.] http://www.tsc.uc3m.es/imagine/Curso_ProcesadoMorfologico/Contenido/Elementos/Elementos.html#operadoresMorfologicos\.

51. **Moller, Martin F.** A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. Noviembre de 1990. [Citado el: 3 de Octubre de 2011.] https://copilosk.fbk.eu/images/d/db/Scaled-conjugate-gradient.pdf.