ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Proyecto Fin de Carrera

Mejora en la detección de microcalcificaciones en mamografías digitalizadas mediante la aplicación de arquitecturas neuronales

AUTOR: Pedro José García Laencina
DIRECTOR: José Luis Sancho Gómez

Septiembre / 2004

Autor	Pedro José García Laencina
E-mail del Autor	pjglaencina@wanadoo.es
Director	José Luís Sancho Gómez
E-mail del Director	josel.sancho@upct.es
Título del PFC	Mejora en la detección de microcalcificaciones en mamografía digitalizada mediante la aplicación de arquitecturas neuronales
Descriptores	redes neuronales, MLP, RBF; SVM, mamografía, microcalcificaciones

Resúmen

En este proyecto fin de carrera se pretende desarrollar un sistema que detecte microcalcificaciones mediante técnicas de análisis de imagen y reconocimiento de patrones sobre mamografías digitalizadas. En concreto, se analiza la pertenencia a una microcalcificación de cada uno de los píxeles de la imagen y no su posible pertenencia a un estado de malignidad o benignidad.

En concreto, amplia la etapa de clasificación de un sistema CAD de detección de microcalcificaciones desarrollado en la Universidad Politécnica de Madrid, para lo cual se han desarrollado las siguientes arquitecturas neuronales:

- Perceptrón Multicapa (MLP, *MultiLayer Perceptron*)
- Redes de Funciones de Base Radial (RBF, *Radial Basis Functions*)
- Maquinas de vectores de soporte (SVM, *Support Vector Machines*)

analizando comparativamente los resultados obtenidos para los distintos de clasificadores, teniendo como último objetivo conseguir mejores resultados en términos de sensibilidad y especificidad

Titulación	Ingeniero de Telecomunicación
Intensificación	Planificación y Gestión de Telecomunicaciones
Departamento	Tecnologías de la Información y Comunicaciones
Fecha de Presentación	Septiembre - 2004

A mis padres, a toda mi familia
y a mi novia R*aquel.*

Agradecimientos

En primer lugar agradezco a José Luis, director de mi proyecto, el apoyo que me ha dado desde el comienzo, además de la libertad que me ha brindado para poder desempeñar mi investigación, junto con la visión que tuvo para dirigir mi trabajo, ya que sin todo lo anterior me habría resultado muy complicado la finalización del proyecto fin de carrera.

Tengo que agradecer mucho a Antonio Vega, por cederme desde un principio desinteresadamente el trabajo realizado durante la elaboración de su tesis, el cual supone la gran base sobre la que se ha sustenta este proyecto, y por solventar la mayoría de las dudas que me han ido surgiendo durante estos meses.

En tercer lugar, quiero recordar a todos los compañeros de carrera con los que he compartido muchas alegrías y penas durante estos cinco años.

Por ultimo, un agradecimiento especial a mis padres y hermanas por el apoyo que me han dado, ya que han hecho que la realización de este proyecto haya sido placentera. Junto con ellos, gracias Raquel por tu apoyo, comprensión y cariño sin los cuales hubiese sido imposible acabar.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Capítulo 1
Introducción

1.1. Introducción

En este proyecto fin de carrera se pretende desarrollar un sistema que detecte microcalcificaciones mediante técnicas de análisis de imagen y reconocimiento de patrones sobre mamografías digitalizadas. En concreto, se analiza la pertenencia a una microcalcificación de cada uno de los píxeles de la imagen y no su posible pertenencia a un estado de malignidad o benignidad.

Esta introducción analiza brevemente los conceptos básicos del procesado de imágenes, particularizando posteriormente al caso de las imágenes digitales médicas, abarcando desde el sistema de digitalización hasta los distintos tipos de imágenes que están presentes en la medicina. Tras ello, se procede a analizar los diferentes motivos que llevan al desarrollo de este proyecto, resaltando el problema al que se enfrenta el médico Radiólogo en el proceso de diagnosis y ampliando el contexto de la aplicación a los sistemas asistidos por ordenador, en la detección del cáncer de mama como un sistema de apoyo (segunda opinión) en este proceso de diagnosis. Dada la naturaleza de este trabajo, también se presentarán los términos médicos, ya que no son comunes en el ámbito de la ingeniería (completándose la definición de estos conceptos en el apéndice A).

A continuación se analizan las distintas investigaciones realizadas hasta el momento en la detección de microcalcificaciones, desarrollando el trabajo realizado por Vega Corona, en el cual están basadas las investigaciones realizadas en este proyecto. Para concluir, se exponen las bases y objetivos del proyecto fin de carrera.

1.2. Tratamiento de imágenes médicas

El tratamiento o ***procesamiento digital de imágenes,*** incluye un conjunto de técnicas que operan sobre la representación digital de una imagen, a objeto de destacar algunos de los elementos que conforman la escena, de modo que se facilite su posterior análisis, bien sea por parte de un usuario (humano) o un sistema de visión artificial. En general, las técnicas de procesamiento de imágenes son aplicadas cuando resulta necesario realzar o modificar una imagen para mejorar su apariencia o para destacar algún aspecto de la información contenida en la misma, o cuando se requiere, medir, contrastar o clasificar algún elemento contenido en la misma. También se utilizan técnicas de procesamiento, cuando se requiere combinar imágenes o porciones de las mismas o reorganizar su contenido.

El tratamiento digital de imágenes está siendo aplicado en diferentes ámbitos de la medicina. Hay dos principales ramas:

1. La asistencia al médico en la elaboración de los diagnósticos sobre los pacientes.
2. La aceleración de las investigaciones en las Ciencias de la Salud.

Cuando es empleado para la diagnosis de los pacientes, suelen utilizarse técnicas comunes de procesado de imagen. La aplicación más simple sería un sistema que realzase las imágenes provenientes de un scanner o de un microscopio, de forma que permitiese facilitar la interpretación física de los procesos biológicos. Estructuras más complejas contienen sistemas expertos junto con algoritmos de visión para la generación de diagnósticos automáticos. En cuanto a las áreas de investigación, las aplicaciones van desde el realce de imágenes hasta la reconstrucción tridimensional de objetos biológicos.

En concreto, la imaginología médica, por su parte, considera un conjunto de modalidades de adquisición de imágenes médicas, las cuales se diferencian entre ellas en cuanto a la naturaleza de los principios físicos involucrados en el proceso de adquisición. Adicionalmente existen también diferencias en cuanto a la aplicación médica. Las modalidades más comunes de imaginología médica son los rayos X, la tomografía computada, la resonancia magnética nuclear, la imaginología nuclear y la imaginología por ultrasonidos.

En los apartados siguientes se introducen brevemente los conceptos genéricos de adquisición y procesado de imágenes, así como también se describe cada uno de los tipos de imágenes presentes en el ámbito de la medicina.

Para una mayor información acerca del tratamiento de imágenes digitales ver [0], en el cual está basado este apartado.

1.2.1. Adquisición y almacenamiento de imágenes digitales

Las imágenes digitales representan información visual asociada con una escena ambiental real que correspondería a lo que observamos con el sentido de la vista o bien información no visible pero que puede ser medida utilizando sensores apropiados tales como radiación infrarroja, ultravioleta, rayos X ultrasonidos, etc.

El proceso de adquisición de la imagen involucra un sensor apropiado para detectar el tipo de fuente de información visual o emisión y convertirla en una señal eléctrica. Posteriormente esta señal eléctrica se convierte en un arreglo de cantidades binarias las cuales se pueden almacenar o procesar utilizando una computadora. La imagen digital corresponde a una estructura de dos dimensiones (2D) que se podría denotar como $f(x,y)$ en donde cada punto se denomina ***píxel*** y tiene asociadas las coordenadas espaciales definidas por x e y. La imagen tiene un tamaño de NxM píxeles, en donde N corresponde al ancho de la imagen y M corresponde al largo de la imagen. Cada píxel corresponde a un valor de intensidad representativa de la información visual o emisión que se ha adquirido. Tal valor binario requiere un determinado número de bits para representar la información y lo más usual es 8 bits que corresponde a un byte o bien, 16 bits o 32 bits que corresponden a 2 bytes y 4 bytes respectivamente. Las imágenes tri-dimensionales (3D) se denotan como $f(x, y, z)$ en donde cada punto se denomina ***vóxel*** y tiene asociadas tres coordenadas espaciales definidas por x, y y z. En este caso el tamaño total sería NxMxP vóxeles y es equivalente a manejar P imágenes bidimensionales cada una de tamaño NxM píxeles. Una vez adquirida la imagen se puede procesar y/o almacenar en disco duro, cintas magnéticas, discos compactos (CD), etc.

1.2.2. Técnicas de Procesamiento basadas en Puntos de la Imagen

Estas técnicas consisten en algoritmos que modifican el valor de un píxel basados únicamente en el valor previo de tal píxel o en su localización. Ningún otro valor de píxel se involucra en la transformación. El procesamiento se realiza desarrollando un barrido píxel por píxel dentro de la imagen a procesar. Si la transformación a aplicar, depende solo del valor original del píxel, en su implantación, puede resultar de utilidad el uso de tablas de búsqueda (Look-Up Table). Si por el contrario, se considera además del valor previo del píxel, la posición del mismo, puede resultar necesario utilizar fórmulas o una combinación de las mismas con tablas de búsqueda. De manera general estas técnicas no modifican las relaciones espaciales dentro de la imagen y en consecuencia no pueden modificar el grado de detalle contenido en las mismas, son simples y pueden resultar útiles solas o en conjunto con otras técnicas más complejas.

1.2.2.1. Histograma de una imagen

El histograma de una imagen es ampliamente utilizado como herramienta tanto cualitativa como cuantitativa. Este corresponde a un gráfico de la distribución de valores de intensidad de los píxeles de una imagen (niveles de gris) o de una porción de la misma. Podemos denotar como *h(i)*, el número de píxeles que dentro de la región de interés tiene el valor de intensidad *i*, donde *i = 0, 1, 2, ..., L-1* es el número posible de niveles de gris para la imagen. Los valores *h(i)*, corresponderán entonces a los valores del histograma. El gráfico del histograma es bidimensional y en él se representa *h(i)* en función de *i*. Tal gráfico, puede proporcionar importante información acerca del brillo y contraste de una imagen así como de su rango dinámico. En la Figura 1.1 (b) se muestra el histograma de niveles de gris para la mamografía digitalizada de la Figura 1.1(a).

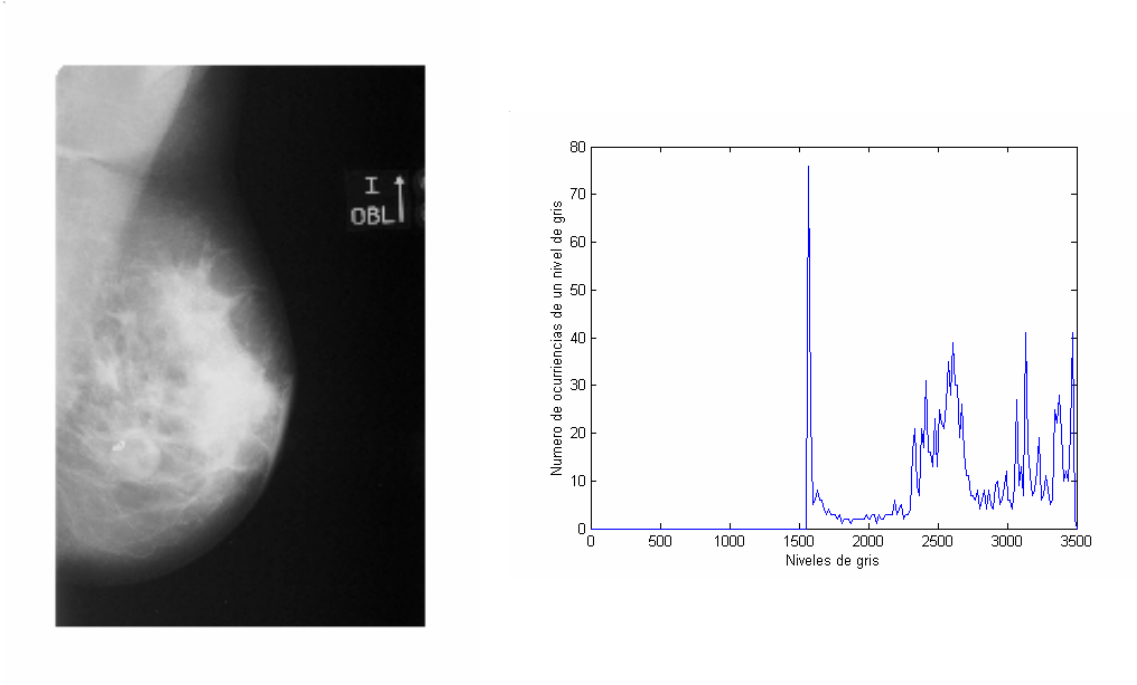


Figura 1.1: (a) Mamografía digitalizada; (b) Histograma de niveles de gris

1.2.2.2. Realce de imágenes por modificación del contraste

Una de las imperfecciones más comunes de las imágenes digitales, es el pobre contraste resultante de un rango de intensidad reducido en comparación al rango disponible de niveles de gris (por ejemplo de 0 a 255 niveles). El ***contraste de una imagen***, puede mejorarse mediante *el re-escalamiento de la intensidad de cada píxel*. Según este método, el nivel de gris correspondiente a un píxel en la imagen de entrada y que denotaremos por *i*, se modifica de acuerdo a una transformación específica. Tal transformación *g=T(i)*, relaciona la intensidad de entrada *i*, con la intensidad de salida *g* y usualmente se representa mediante un dibujo o una tabla. A modo de ejemplo, la Figura 1.2(a) muestra una imagen de 4 x 4 píxeles, donde cada píxel se ha representado con 3 bits, de modo que en total sería posible representar 8 niveles de gris. La transformación que relaciona la intensidad de entrada con la intensidad de salida, se muestra en la Figura 1.2 (b). De acuerdo a tal transformación, para cada píxel de la imagen de entrada, se obtiene la correspondiente intensidad en la imagen de salida. El resultado obtenido en este caso particular se muestra en la Figura 1.2 (c), en donde podemos observar que el contraste entre las zonas obscuras y claras dentro de la imagen, se incrementa apreciablemente. Eligiendo apropiadamente la transformación específica, puede modificarse de manera casi arbitraria el contraste y rango dinámico de la imagen.

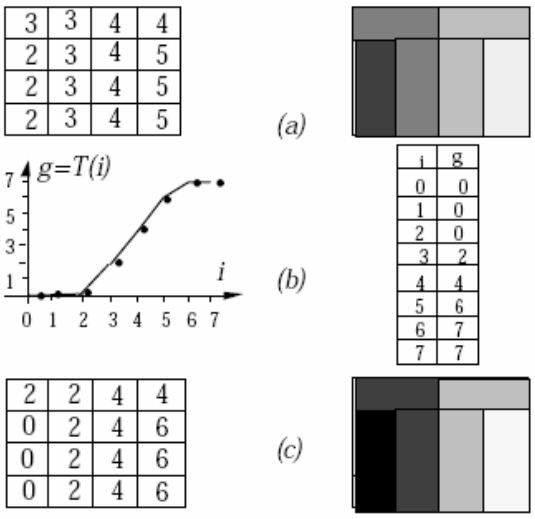


Figura 1. 2: Ejemplo de la modificación de la escala de grises. (a) Imagen de 4 x 4 píxeles, con cada píxel representado por 3 bits; (b) Función de transformación de los niveles de gris; (c) Resultado de modificar la imagen en (a), usando la transformación de niveles de gris especificada en (b).

1.2.2.3. Técnicas de colores falsos y seudocolor

Es bien conocido que el ojo humano, es bastante sensible al color, así el número de niveles de gris que puede discriminarse como tal, es bastante más pequeño que el número de colores. Por otra parte, las imágenes a color, son más agradables a la vista que las imágenes en blanco y negro.

La *técnica de colores falsos*, usualmente se emplea cuando se desea asociar a un conjunto de datos, un conjunto de colores para distinguir en los mismos, ciertos atributos, tal como ocurre cuando un sensor remoto adquiere información en la banda de

infrarrojo (la cual no es visible), en tal caso, lo que se hace es asociar el color a este conjunto de datos, para apreciar mejor los detalles.

La *técnica de seudocolor* consiste en transformar una imagen monocromática (en niveles de gris) en una imagen a color, al asignar a cada píxel un color basado por ejemplo en su intensidad. Un método pudiera ser el siguiente: se procesa la imagen monocromática con tres filtros, un paso-bajo, uno pasa-banda y uno paso-alto. La imagen procesada con el filtro paso-bajo se asigna al color azul, la imagen procesada con el filtro pasa-banda se asocia al verde y la procesada con el filtro paso-alto se asocia al rojo; luego estas tres imágenes se combinan para producir una imagen a color. Usando técnicas de seudocolor, más complejas que la mencionada previamente y mediante mucha intervención humana, ha sido posible dar color películas filmadas originalmente en blanco y negro.

1.2.3. Procesamiento basado en una región de la imagen

Las técnicas de procesamiento basadas en una región tienen muchas aplicaciones en la obtención de primitivas características de la imagen como por ejemplo la extracción de contornos, para realzar los contornos, para suavizar una imagen, para introducir borrosidad dentro de la misma y para atenuar el ruido aleatorio. Usan un *grupo de píxeles* dentro de la imagen a procesar, con el propósito de extraer información acerca de la misma. El grupo de píxeles que se estudia en este caso, se denomina usualmente ***ventana***. Por lo general una ***ventana de píxeles*** es una matriz bidimensional de valores de píxeles con un número impar de filas y columnas. El píxel de interés que normalmente es reemplazado por un nuevo valor, producto de la aplicación de un algoritmo, se ubica por lo general, en el centro de la ventana.

Al utilizar una ventana en el procesamiento, se puede aprovechar la información acerca del comportamiento regional de la imagen en cuestión, mejor conocida como *frecuencia espacial*, la cual podría definirse como la tasa de cambio de la intensidad de los píxeles dividido por la distancia sobre la cual ocurre el cambio. La frecuencia espacial tiene componentes en las direcciones horizontal y vertical dentro de la imagen. Por ejemplo, la imagen de un patrón tipo tablero de ajedrez presentará un alto contenido de frecuencia espacial, el cual aumentará en la medida que el tamaño de los cuadros disminuya. Por su parte una imagen con un bajo contenido de frecuencia espacial por lo general tiene amplias áreas con valores casi constantes de los píxeles.

Muchas de las técnicas de procesamiento basadas en una región de la imagen, al tener acceso a la información referente a la frecuencia espacial, pueden actuar como filtros que atenúan o realzan ciertas componentes de la frecuencia espacial contenidas dentro de la imagen. En la implantación de estas técnicas, se utilizan métodos lineales tales como la convolución o no lineales como el filtraje de mediana. En todo caso, el procedimiento que se sigue es el siguiente:

(a) Se realiza una sola pasada sobre la imagen de entrada realizando un barrido píxel por píxel, según las filas y columnas.
(b) Cada píxel de la imagen de entrada es procesado, considerando una ventana del mismo y utilizando un algoritmo apropiado.
(c) El nuevo valor del píxel, obtenido de acuerdo a lo especificado en (b), es ubicado en la imagen de salida, ocupando la misma posición que ocupaba en la imagen de entrada.

El hecho de considerar los píxeles de una ventana, hace que las técnicas de procesamiento basadas en una región tengan un mayor coste de cálculo numérico que las técnicas basadas en un solo punto. Este coste dependerá del tamaño de la ventana a considerar, así como del tipo de representación numérica utilizada. Sin embargo, para la mayoría de las aplicaciones y con las computadoras disponibles actualmente, se pueden obtener muy buenos resultados en términos de tiempo de cálculo, al procesar imágenes de un tamaño mediano (256 x 256 ó 512 x 512).

1.2.3.1. Convolución

Si consideramos una imagen como un estructura bidimensional denotado por *x(i,j)* y el filtro (núcleo o máscara de convolución) con respuesta impulsiva *h(i,j)*, su convolución produce una imagen de salida *y(i,j)*, de acuerdo a la siguiente ecuación, en donde *m, n* definen la ventana a considerar de acuerdo al tamaño del núcleo de convolución *h(i,j)*:

$$y(i,j) = \sum_m \sum_n h(m,n)\, x(i-m, j-n) \qquad (1.1)$$

La implantación de esta ecuación de convolución se hace de manera directa cuando el tamaño del filtro o máscara de convolución es pequeño (usualmente menor a 9 x 9 píxeles), pues en tales casos el coste computacional no es exagerado, sin embargo, cuando se tienen filtros de mayor tamaño, lo más recomendable es implantar esta ecuación de convolución mediante la utilización de la transformada rápida de Fourier.

1.2.3.2. Filtrado Espacial Paso-Bajo

Los filtros espaciales paso-bajo, dejan el contenido de baja frecuencia inalterado mientras que atenúan los contenidos de alta frecuencia, este tipo de filtros resulta adecuado para atenuar ruido aditivo aleatorio presente en la imagen. En la Figura 1.3 se muestran tres máscaras de convolución frecuentemente utilizadas para realizar el filtrado paso-bajo, una de las propiedades de tales máscaras es que la suma de todos sus valores debe ser igual a la unidad.

$$\frac{1}{9}\cdot\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix} \quad \frac{1}{10}\cdot\begin{bmatrix}1&1&1\\1&2&1\\1&1&1\end{bmatrix} \quad \frac{1}{16}\cdot\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

Figura 1.3: Tres mascaras que permiten el filtrado paso-bajo de una imagen

1.2.3.3. Filtrado Espacial Paso-Alto

Los ***filtros espaciales paso-alto***, tienen la propiedad de acentuar los detalles de alta frecuencia de una imagen, normalmente los filtros paso-alto se utilizan cuando se quiere examinar objetos con alto contenido de frecuencia espacial, como consecuencia de tal procesamiento, las porciones de una imagen que presentan componentes de alta frecuencia, serán resaltadas mediante la utilización de niveles de gris más claros, mientras que aquellas con componentes de baja frecuencia serán más oscuras, en este sentido, este tipo de filtro puede ser utilizado para reforzar los bordes presentes en la imagen o también, como en nuestro caso, para el etiquetado de microcalcificaciones, como ya se analizara en apartados posteriores. Uno de los efectos indeseados de estos

filtros es que pueden acentuar el ruido de la imagen. En la Figura 1.4 se muestran tres máscaras de convolución para implantar el filtro pasa alto.

-1	-1	-1
-1	9	-1
-1	-1	-1

0	-1	0
-1	5	-1
0	-1	0

1	-2	1
-2	5	-2
1	-2	1

Figura 1. 4: Tres mascaras que permiten el filtrado paso-alto de una imagen

1.2.3.4. Detección de contornos

Las ***técnicas de detección de contornos*** son útiles en diferentes contextos, en particular la detección de contornos es una de las etapas del proceso de segmentación cuyo objeto es dividir la imagen en regiones asociadas a los diferentes elementos que componen la escena, y que puede ser utilizada posteriormente para el análisis automático de los mismos mediante algoritmos de reconocimiento de patrones.

Un *borde* en una imagen, es un límite o contorno en el cual ocurren cambios significativos en algún parámetro físico de la imagen, tal como la reflectancia superficial, la iluminación o la distancia de la superficie visible al observador. Los cambios en los parámetros físicos de la imagen se manifiestan de diversas formas, incluyendo cambios en intensidad, color y textura; sin embargo, en este trabajo, nos limitaremos a únicamente los cambios en la intensidad de la imagen.

En la Figura 1.5, se muestran los diagramas esquemáticos de bordes unidimensionales y bidimensionales, modelados usualmente como un incremento en rampa, en los niveles de gris de la imagen, en este caso de un nivel de gris bajo a un nivel de gris alto, siendo valida la definición para el caso contrario. En el caso bidimensional, cada punto (*x*,*y*) define la posición del píxel y la coordenada *z* define la amplitud del nivel de gris.

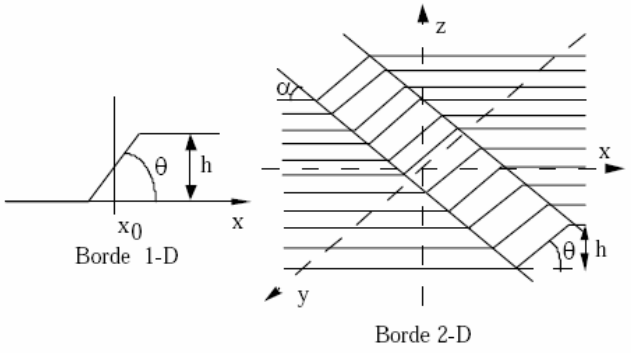


Figura 1.5: Modelo para el borde unidimensional y bidimensional

Existen diversas técnicas para la detección de contornos, entre las que se destacan las siguientes:

- *Detección de contornos basado en gradientes.* Esta técnica se basa en la derivada de la señal para el caso unidimensional, mientras que en caso de tener imágenes (bidimensional) se emplea el gradiente.
- *Detección de contorno basado en la Laplaciana.* Otra posibilidad para la acentuación de bordes, consiste en usar la segunda derivada, en tal caso se usan los puntos de cruce por cero para realizar la detección de borde. En el caso de la imagen por ser una señal bidimensional, la extensión de la segunda derivada

unidimensional corresponderá en este caso a la Laplaciana. La Laplaciana puede ser utilizada para realizar el reforzamiento de bordes o bien para detectar contornes en base a los cruces por ceros

1.2.3.5. Segmentación de imágenes

Para realizar la identificación de estructuras anatómicas presentes en la imagen, se utilizan las técnicas de ***segmentación***, las cuales permiten dividir la imagen en un conjunto no solapado de regiones, cuya unión es la imagen completa. Uno de los métodos que usualmente se sigue para implantar la segmentación consiste en primero determinar los bordes del objeto, utilizando las técnicas estudiadas en las secciones anteriores, seguidamente resulta necesario determinar el interior del objeto y clasificar los píxeles incluidos en tal borde como pertenecientes al objeto. Otra de las técnicas comúnmente utilizadas en segmentación, es la segmentación basada en el uso de un umbral y la segmentación por crecimiento de regiones.

- *Segmentación basada en uso de umbral*. Este tipo de segmentación, permite separar un objeto dentro de la imagen del fondo que lo circunda, la técnica se basa en comparar alguna propiedad de una imagen con un umbral fijo o variable, realizando tal comparación para cada uno de los píxeles que conforman la imagen, si el valor de la propiedad de un píxel supera el valor del umbral, entonces el píxel pertenece al objeto, en caso contrario, el píxel pertenece al fondo. Cuando la segmentación se realiza basada en el nivel de gris de la imagen, el valor del nivel de gris de cada píxel debe ser comparado con el umbral, para decidir si tal píxel pertenece al objeto o al fondo. La imagen de salida, es una imagen binaria en la cual aquellos píxeles cuyo valor es 1, pertenecen al objeto y los píxeles cuyo valor es cero, pertenecen al fondo. La selección del valor del umbral, se realiza generalmente a partir del histograma de la imagen.

- *Segmentación por crecimiento de regiones.* Se buscan píxeles que tengan características similares (por ejemplo niveles de gris similares) y que adicionalmente sean vecinos. El método comienza con un píxel, el cual es seleccionado automáticamente o proporcionado por el usuario y a continuación examina los píxeles vecinos para decidir si tienen características similares. De ser así, el píxel vecino que cumpla con tal condición de similaridad, es agrupado junto con los anteriores para conformar así una región.

1.2.4. Tipos de imágenes médicas

Si bien el 70% de las imágenes médicas son todavía radiografías, cada vez hay un mayor incremento en todos los ámbitos de las Ciencias de la Salud del uso de imágenes digitales para un posterior tratamiento y almacenamiento: *tomografías, resonancias magnéticas, ecografía, medicina nuclear*, entre otras, que ocupan el 30% restante.

Las fuentes de la imaginología médica se diferencian fundamentalmente por el tipo de información que es detectada. Cada modalidad da una representación diferente de los órganos o de su funcionamiento. En la siguiente sección se dará una breve descripción de las diferentes modalidades de la imaginología médica.

1.2.4.1. La ecografía

La ***ecografía*** es una técnica de *exploración por ultrasonidos*. El dispositivo transductor (barra piezoeléctrica que funciona como emisor y receptor) emite una onda ultrasónica de una frecuencia de varios MHz, en una orientación dada. La señal detectada corresponde a la superposición de ecos o reflexiones que se producen debido a los cambios de impedancia acústica en las diferentes fronteras de los órganos.

La imagen 2D se obtiene por el barrido de un haz según un plano cualquiera. Una de las dimensiones está dada por el barrido del haz, la otra dimensión corresponde al tiempo de retorno del eco. Si bien la resolución geométrica es inferior en relación con otras modalidades (tomografía computada y resonancia magnética nuclear), sus ventajas se fundamentan en el hecho que la velocidad de adquisición es elevada (15 a 30 imágenes por segundo), lo cual permite la exploración de órganos en movimiento, la naturaleza no ionizante del proceso de adquisición y el precio razonable de los equipos.

Una ecografía que muestra el riñón se presenta en la Figura 1.6.

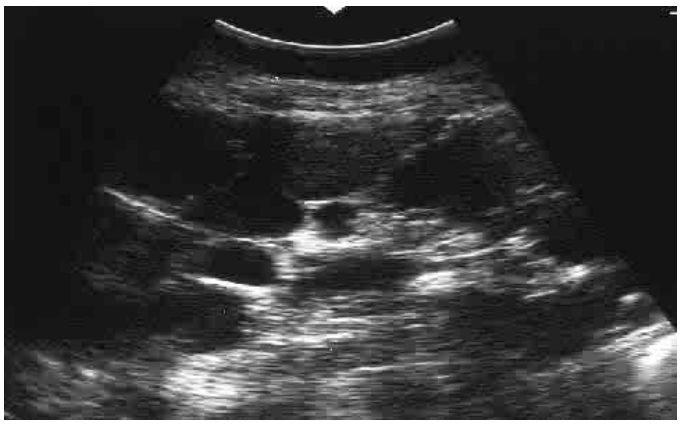


Figura 1.6: Ecografía de un riñón

1.2.4.2. La Tomografía Axial Computerizada

Una importante modalidad de la imaginología médica es la ***tomografía axial por computadora*** (***TAC***) en donde las imágenes del interior del cuerpo se reconstruyen a partir de un conjunto de medidas de proyección. Tales proyecciones son obtenidas mediante la exposición de un objeto a un tipo de radiación, según diversos ángulos y midiendo, para cada posición angular, la intensidad del haz de radiación que atraviesa el objeto.

En el caso de la tomografía computada, las proyecciones se obtienen mediante la exposición a los rayos X. La medición de la intensidad de rayos X que atraviesa un plano o capa del cuerpo humano según un ángulo dado da lugar a un perfil de proyección. La reconstrucción tomográfica de una capa del cuerpo humano se obtiene a partir de los diferentes perfiles de proyección obtenidos utilizando técnicas de reconstrucción de tipo analíticas o algebraicas. La calidad de la imagen depende del número de perfiles considerados. Para obtener imágenes útiles desde el punto de vista del diagnóstico médico, es necesario considerar un número importante de perfiles de proyección.

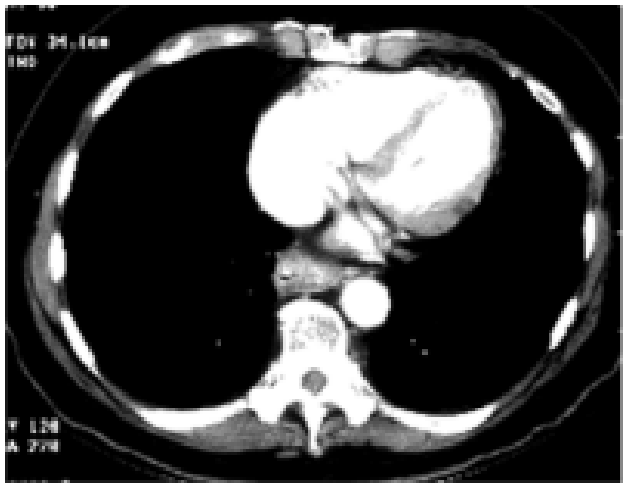


Figura 1.7: Tomografía del tórax

1.2.4.3. La resonancia magnética

La ***imagen por resonancia magnética*** (***IRM***), consiste en medir la concentración y el tiempo de relajación de ciertos núcleos atómicos (generalmente núcleos de Hidrógeno) excitados bajo la acción de un campo magnético fijo y de un pulso de radio-frecuencia. Cuando se ubica un objeto en un gradiente de campo magnético, la frecuencia de las señales de resonancia magnética producidas por los núcleos atómicos es dependiente del campo magnético local aplicado y de la interacción a nivel molecular. Un núcleo sometido a un campo magnético H_0 se comporta como vector dipolo magnético, el cual gira alrededor de la dirección de H_0 a una frecuencia conocida como *frecuencia de Larmor*, la cual es característica del elemento químico considerado y de la intensidad del campo H_0. En presencia de un campo magnético H_1 capaz de orientar al dipolo magnético total en sentido perpendicular a H_0 corresponde a un estado excitado de los núcleos atómicos. Al apagar el campo de radiofrecuencia H_1 el estado de excitación va decayendo progresivamente en función de la densidad de los tejidos. El retorno al equilibrio está caracterizado por dos constantes de tiempo denominadas T_1 y T_2.

Cada punto de la imagen es función de la posición en el espacio tridimensional, de la concentración p de núcleos atómicos y de las constantes de tiempo de relajación, es decir: $I(x,y,z) = f(x,y,z,p,T_1,T_2)$

El interés de la IRM se debe al hecho de que el equipo utiliza una radiación no ionizante que permite una buena discriminación de los tejidos y la adquisición tridimensional de una zona del cuerpo. El contraste de los tejidos, puede mejorarse adicionalmente mediante la utilización de materiales de contraste paramagnéticos inyectados en el cuerpo. El campo de aplicación de esta modalidad es muy amplio y la limitación principal son los costos de los equipos. Gracias a la utilización de protocolos de adquisición sincronizados con la señal electrocardiográfica (ECG), es posible obtener imágenes 3D del corazón. La utilización de la modalidad de IRM, permite también la generación de imágenes 3D de vasos sanguineos.

Figura 1.8: Resonancia magnética de la vesícula biliar

1.2.4.4. Imaginología nuclear

Las imágenes por emisión se basan en la detección de la radiación emitida por cada punto de un órgano después de administrar al paciente una sustancia que incluye un radioelemento, el cual tiene un periodo de vida muy corto y se degrada hasta convertirse en una sustancia inerte. Esta modalidad generalmente proporciona imágenes cuya intensidad cuantifica el funcionamiento del órgano, proporcionando información acerca de la capacidad de tal órgano de asimilar o transformar la sustancia que se ha inyectado. Cada radioelemento es específico para el órgano y la función que se desea estudiar (por ejemplo, el isótopo de yodo se utiliza para estudiar la glándula tiroides, mientras que el talio se usa para estudiar los tejidos cardiacos).

La adquisición se efectúa midiendo la distribución del radioelemento en el órgano. Generalmente se usan tres esquemas de adquisición: la ***gammagrafía*** en donde la emisión de radioelemento se detecta en un arreglo de sensores fijo produciendo una imagen 2D. Los otros dos esquemas son de tipo tomográficos y corresponden a la emisión de fotones gamma únicos (*Single Photon Emision Tomography:* ***SPECT***) e imágenes por emisión de positrones (***PET***: *Positron Emission Tomography*). En el primer caso (***SPECT***), la fuente de radiación es un emisor gamma. Los fotones gamma se detectan usando una cámara radioluminiscente rotativa. La distribución de fotones gamma emitidos según una dirección dada constituye una proyección. Repitiendo el proceso para múltiples direcciones es posible determinar la distribución del radioelemento en el interior del cuerpo al utilizar técnicas de reconstrucción tomográfica. En la imaginología por emisión de positrones (***PET***), el trazador emite positrones (rayos beta) los cuales no se pueden medir directamente, sin embargo, cuando un positrón choca con un electrón emite dos fotones gamma que se propagan en direcciones opuestas. La detección casi simultanea de los dos fotones a ambos lados del paciente, permite localizar espacialmente la ubicación del choque de las partículas. La principal limitación de esta modalidad consiste en requerir una infraestructura muy costosa que debe incluir un acelerador de partículas cargadas para generar sustancias radioactivas con un periodo de vida que está entre 5 minutos y media hora aproximadamente. Adicionalmente se requiere de un laboratorio químico que permita la síntesis de sustancias asimilables por el organismo las cuales son marcadas con el trazador radioactivo.

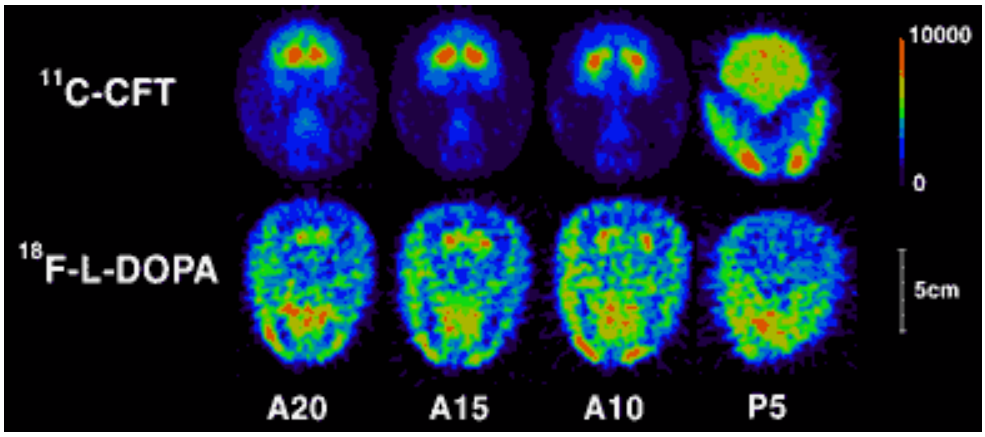


Figura 1.9: PET cerebral de un paciente con la enfermedad de Parkinson

1.2.4.5. La radiología

Los *rayos X* son una clase de radiación electromagnética similar a la luz para la cual la longitud de onda es más pequeña. Sus propiedades físicas más importantes son su capacidad para atravesar la materia, producir fosforescencia e imprimir películas con emulsiones fotográficas. Los rayos X pueden también producir cambios en los tejidos biológicos y son capaces de producir la ionización de los materiales gaseosos. Los rayos X son a menudo utilizados para producir imágenes médicas y para desarrollar efectos terapéuticos en algunos pacientes.

Las modalidades de imaginología médica basadas en la utilización de rayos X incluyen la *radiografía convencional*, la *mamografía*, la *video-angiografía o fluoroscopia* y la *tomografía*. El desarrollo de estas modalidades se apoya fundamentalmente en los avances logrados en la tecnología de procesamiento digital de imágenes, lo cual ha permitido mejorar su valor diagnóstico y terapéutico. La imaginología por rayos X es en la actualidad una de las modalidades más utilizadas en el dominio médico en general.

La ***radiología convencional*** permite obtener sobre una película radiográfica, la imagen de una parte del cuerpo humano expuesta a un haz de rayos X en donde la absorción que sufre el haz de rayos X depende del espesor de la sustancia atravesada y de los átomos que la constituyen. La visualización de las diferencias de densidad dentro de los tejidos o los órganos se logra al ubicar una película radiográfica detrás de la parte del cuerpo a estudiar, en relación con la fuente de rayos X. Se podría decir entonces, que la radiografía es la sombra producida por el cuerpo cuando se utiliza como fuente de iluminación un haz de rayos X. En la representación 2D de la imagen obtenida (imagen de proyección), la intensidad de las diferentes zonas de la imagen da una medida de la intensidad de los órganos o estructuras anatómicas iluminadas, mientras que la apariencia de sus sombras da una indicación de la forma de la estructura y de su posición espacial.

Si bien las características de las radiografías son dependientes de los aspectos técnicos relacionados con la potencia y el desplazamiento del sistema fuente detector, hay un conjunto común de atributos característicos que limitan la calidad de la imagen tales como el hecho de que las imágenes tengan poco contraste debido al bajo valor del coeficiente de atenuación de la mayor parte de tejidos del cuerpo y adicionalmente que puedan estar afectadas por diversas fuentes de ruido difíciles de controlar.

Mamografía

Particularizando para mamografías, una ***mamografía*** es una radiografía de baja radiación, que examina el tejido de la mama. Es decir, es una radiografía específica de la mama. Normalmente se realizan dos radiografías de cada mama, una de lado y otra desde arriba. La mama tiene que ser aplastada un poco entre dos placas para que la imagen sea clara y nítida; resulta algo incómodo para la paciente pero es cuestión de pocos segundos.

En algunos tumores, que no en todos, se pueden visualizar las lesiones antes de que proporcionen síntomas o puedan palparse. Así, se ven quistes y diminutos granitos de calcio (*microcalcificaciones*), que, aunque en la mayoría de las ocasiones tienen carácter benigno, pueden alertar precozmente del cáncer. Que, como es sabido, puede presentarse en cualquier tejido del organismo humano y sobreviene cuando las células se dividen y multiplican de forma anárquica, sin orden ni control, como si se hubieran escapado al control natural del organismo. Cuando las células se multiplican de este modo, se origina una masa se células que adquiere el temido nombre de tumor, pudiendo ser estos o bien benignos o bien malignos.

Estas pruebas pueden completarse con otras más específicas, para detectar la gravedad del cáncer, tales como una resonancia magnética, o en último caso con una biopsia, en la que se toma parte del tejido afectado para analizar en el laboratorio las características de las células cancerosas.

1.3. Cáncer de mama y diagnosis asistida

El cáncer de mama es el tumor maligno más frecuente entre las mujeres. Su incidencia en España varía entre 40 y 75 por 100.000 mujeres, según los datos obtenidos por los distintos registros españoles de cáncer de base poblacional. El cáncer de mama constituye la primera causa de muerte por cáncer en mujeres, con una tasa de mortalidad de 28´2 por 100.000, lo que representa el 18,4% del total de muertes por cáncer en mujeres [1]. En la actualidad, realmente se desconocen con exactitud los factores que determinan la aparición del cáncer de mama, además de la forma de prevenirlo, pero la detección precoz y su tratamiento posterior constituye la manera más efectiva de reducir el número de muertes. La detección precoz, cuando el tumor no está extendido ni evolucionado, hace que el porcentaje de curación se eleve casi al 90%.

1.3.1. Tipos de anormalidades en una mamografía

La mayoría de las anomalías que se presentan en la mama se pueden descubrir mediante un programa de detección precoz, y así poder tratarlas adecuadamente en una etapa previa al desarrollo de la anomalía. Se pueden clasificar en [2]: Opacidades infraclínicas[1] y Microcalcificaciones.

A. ***Opacidades infraclínicas***, es decir distorsiones de la estructura mamaria, y masas irregulares. Se clasifican en estrelladas y redondas.

A.1. Estrelladas

- Estrelladas con centro denso. Suponen el 20-30% de los cánceres infraclínicos.

[1] *Infraclínico*: No detectado tras examen clínico

- Estrelladas sin centro denso (distorsiones localizadas) representan el 10-20% de los cánceres infraclínicos. El 60% de estas lesiones sometidas a biopsia son malignas.

A.2. Redondeadas

- Representa el 10-20% de los cánceres infraclínicos, pero más del 50% de ellas, cuando se someten a biopsia resultan ser benignas

B. ***Microcalcificaciones.***

Las microcalcificaciones son minúsculos depósitos de calcio que aparecen como pequeños puntos blancos y brillantes en la mamografía (Figura 1.10).

Pueden presentarse aisladas o en grupos. Los cluster de microcalcificaciones se definen por contener dentro de un $1cm^2$ de la región de interés (ROI, *region of interest*) al menos 5 microcalcificaciones. Las microcalcificaciones agrupadas en mamografías son un signo temprano de una lesión maligna en casi la mitad de los cánceres de mama. Su tamaño varia desde los 0.1 mm hasta 5 mm de diámetro, por tanto un radiólogo deberá analizar exhaustivamente la mamografía con lupa para poder localizar las microcalcificaciones, por tanto constituyen uno de los problemas de diagnósticos más difíciles del tratamiento de cáncer de mama.

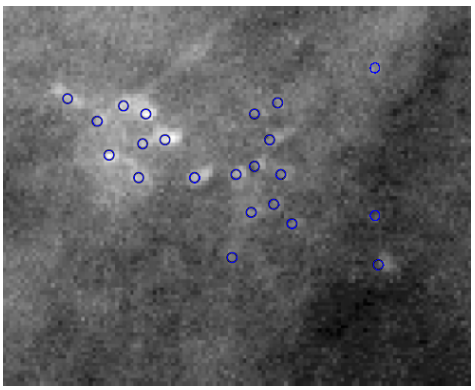


Figura 1.10: Sección de una mamografía con múltiples microcalcificaciones (marcadas con círculos)

La clasificación de estas microcalcificaciones divide estas lesiones en 5 tipos:

Tipo I: Microcalcificaciones anulares, redondeadas, de centro claro.(0% Malignidad).

Tipo II: Microcalcificaciones puntiformes regulares, redondeadas, (10% Malignidad).

Tipo III: Microcalcificaciones "en polvo", muy finas, sin poder precisar su forma ni su número (19% Malignidad).

Tipo IV: Microcalcificaciones puntiformes irregulares, son poliédricas, en grano de sal (29% Malignidad).

Tipo V: Microcalcificaciones vermiculares, son alargadas, como un árbol sin hojas (72% Malignidad).

Además de esta clasificación, se han considerado otros factores a la hora de valorar qué tipo de microcalcificaciones son más susceptibles de corresponder a carcinomas:

- Número de microcalcificaciones por cm^2: Más de 20 por cm^2 es más frecuente en lesiones malignas.
- Número total de microcalcificaciones: Más de 30 microcalcificaciones en total es sospechoso de malignidad.
- Irregularidad de la densidad y del tamaño: Más común en los tumores malignos.
- Distribución lineal ó en ramas: Típico de tumor maligno.

1.3.2. Programas de screening con mamografía

Para reducir de lesiones cancerigenas es necesario una gran participación de mujeres en los programas de *screening* mamográficos (programa de diagnosis para la detección precoz del cáncer de mama). La mamografía de detección se utiliza para buscar una enfermedad de la mama en mujeres asintomáticas, es decir, que no parecen tener ningún problema en las mamas. En ellos se separan las mamografías normales de las anormales, siendo estas ultimas estudiadas detenidamente hasta llegar a un diagnostico final, que determine en último lugar la necesidad de una biopsia para analizar la malignidad del tumor. Como estándar de mamografías se obtienen cuatro imágenes de rayos X, con dos vistas en cada mama (dos vistas cráneo-caudal (*CC*), y dos medio-lateral-oblicuo (MLO)). Todas ellas son empleadas para detectar anormalidades y valorar su malignidad o benignidad.

El diagnostico precoz tiene muchas ventajas, consiguiendo predecir el cáncer en muchos casos, pero en otros se puede llevar a realizar biopsias innecesarias [3]. Esto es debido a dos factores, el primero de ellos es el aumento de realización de mamografías en mujeres asintomáticas, y el segundo, la complejidad que presenta determinar la malignidad de las lesiones incluso para radiólogos expertos. Otro problema que se presenta es la baja participación de mujeres debido a la creencia de la baja fiabilidad del proceso y la incomodidad física, además de existir un alto coste para poder llevar a cabo un programa de screening. Las imágenes de una mamografía son difíciles de interpretar incluso para un radiólogo experto; una razón de ello, es que una mamografía contiene una estructura con una textura 3D que se ha proyectado sobre un plano en 2D, además de tener muy bajo contraste para mantener bajos los niveles de radiación a los que se exponen las pacientes. Por tanto la detección precoz del cáncer de mama depende de la capacidad que tenga el radiólogo de identificar lesiones sutiles y de pequeño tamaño, que sólo son visibles si la imagen es de buena calidad. La mayor parte de los errores que se producen en la detección y caracterización de estas lesiones son debidos al uso de una técnica radiológica inadecuada, a la alta densidad de la mama explorada o a un error u omisión en la lectura de la mamografía, viéndose esta última muy influenciada por las otras dos anteriores.

Los programas de screening con mamografía presentan típicamente una *sensibilidad*[2] , o también fracción de verdaderos positivos (*FVP*), entre el 30 y el 75 por ciento. La exactitud del diagnóstico puede aumentar si dos radiólogos revisan cada mamografía o un radiólogo que realice doble revisión [4]. Se ha demostrado que así se consigue disminuir la fracción de falsos negativos (canceres omitidos) y aumentar la FVP (canceres diagnosticados).

[2] Ver Apéndice A

1.3.3. Diagnosis Asistida por Ordenador

En los últimos 15 años se han venido desarrollando proyectos de diagnóstico asistido por ordenador (CAD, Computer Arded Diagnosis) en Radiología para la detección automática de lesiones y la caracterización de patrones normales y anormales con el objetivo de mejorar la precisión y consistencia diagnóstica de los radiólogos.

Una vez que la imagen radiológica está en formato digital, los ordenadores pueden procesar la imagen para mejorar la percepción del observador (visión humana) o realizar funciones de análisis de imagen (visión artificial) de forma diferente a los observadores humanos, complementándolos.

Los proyectos de diagnóstico asistido por ordenador que actualmente se están desarrollando, tienen los siguientes objetivos:

1. Mejorar el rendimiento del observador, conociendo previamente el tamaño o la localización hipotética del objeto. Se trata de métodos que dirigen la atención del radiólogo a regiones sospechosas, intentando resolver los problemas de omisión de hallazgos.
2. Aportar una segunda opinión sobre el grado de correlación de las características de una imagen con un diagnóstico determinado.
3. Realizar comparaciones detalladas entre varias imágenes con el objeto de aportar información en una misma exploración (por ejemplo, con y sin contraste) o en varias exploraciones diferidas de un mismo paciente.
4. Proporcionar sistemas de autoevaluación y reciclaje para radiólogos con diversos grados de experiencia y de aprendizaje para residentes en periodo de formación.

Las posibilidades del CAD son especialmente útiles cuando la precisión diagnóstica es inferior a lo deseable o cuando el diagnostico se basa en apreciaciones y mediciones subjetivas. El uso de CAD produce un aumento en la *sensibilidad* (fracción esperada de microcalcificaciones que son clasificados correctamente) para un nivel de *especificidad* (fracción esperada de casos normales que son clasificados correctamente). Uno de los objetivos finales del CAD es desarrollar métodos que proporcionen un segundo observador, objetivo e infatigable, que indique al radiólogo la zona que debe revisar, preservando para éste la decisión final, además de tener más tiempo disponible para examinar un mayor numero de casos sin un aumento de coste. Actualmente existen sistemas CAD comercializados para trabajar en mamografía de 'screening'.

1.3.4. Estudios y sistemas CAD propuestos por otros investigadores

Existe un gran interés entre la comunidad investigadora, universidades, hospitales y gobiernos en el estudio de los sistemas CAD aplicados a la detección de microcalcificaciones. Los sistemas CAD implementados han seguido distintos criterios y enfoques.

En trabajos como los de Strikland [5], Karssemeijer [6] se detecta las microcalcificaciones usando la transformada wavelet en combinación con un criterio donde se analiza la forma, el tamaño y el número de microcalcificaciones por unidad de área. Posteriormente [7], se han aplicado técnicas de ecualización para ajustar la distribución de los datos, lo cual es determinante para aumentar la FVP y minimizar la FFP (fracción de falsos positivos). Una gran cantidad de estudios realizan la detección

de anomalías combinando desde la detección de contornos hasta la extracción de características por técnicas multiresolución. Otros trabajos como los de Nishikawa (1992-1997) *et al.* [8,9,10] se centran en el tratamiento de la mamografía digitalizada, como el filtrado para eliminar el fondo y aumentando la relación señal a ruido que permita el realzado de la región de interés, para después identificar las posibles anomalías realzadas a través técnicas de umbralización local y global de los niveles de gris, mientras que la detección de microcalcificaciones se hace mediante la extracción de características como tamaño, distribución espacial de las microcalcificaciones, análisis de la textura. Se consigue detectar un 85% de verdaderos positivos.

Distintos investigadores han centrado sus trabajos en clasificadores basados en redes neuronales, algoritmos genéticos, funciones de base radial,…En los trabajos realizados por Verma (1998) [11] se propone un sistema basado en redes neuronales consiguiendo detectar un 87 % de verdaderos positivos. Kupinski (1997) *et al.* [12] propone como los algoritmos genéticos para extraer las mejores características de cada píxel sobre una ROI, para posteriormente clasificarlos como microcalcificaciones empleando una red neuronal basada en la función de error entropía cruzada. Los resultados se presentan a través de curvas ROC[3] y presenta un área bajo la curva A_z=0.96. Mientras Hojjatoleslami (1997) *et al.* [13] realiza un comparación de varios métodos de clasificación tras aplicar una etapa de preprocesado a las mamografías, compara un clasificador RBF con un perceptrón multicapa y un clasificador K-NN, proporcionando mejores resultados el clasificador RBF. Por otro lado, Patrocinio (2001) *et al.* [14] tras una etapa de segmentación aplicada a la mamografía y la extracción y selección de características dentro de una ROI (como irregularidad, perímetro, área,…), emplea un red neuronal entrenada con retropropagación, y finalmente consigue hasta un 92 % como FVP y un área bajo la curva ROC de A_z=0.96.

Otros trabajos se centran en las técnicas de multiresolución. Wang (1998) *et al.* [15] propone la detección de microcalcificaciones usando wavelets. A partir la mamografía digital, realiza un descomposición multiresolución basada en filtros de análisis wavelet, a continuación elimina las subbandas de baja frecuencia y por último reconstruye la mamografía que únicamente contiene componentes de alta frecuencia, incluyendo microcalcificaciones. Songyang (2000) *et al.* [16] combina las prestaciones entre el análisis de características de multiresolución de la imagen con el uso de la transformada wavelet y las características de la estructura de niveles de gris y las redes neuronales para la detección de microcalcificaciones, en concreto una red GRNN (General Regression Neural Network). El sistema propuesto por Vega (2004) [17] es explicado en el apartado 1.2 y está dentro de esta línea de trabajos.

Trabajos recientes de desarrollo de sistemas CAD para la detección de microcalcificaciones se han basado en el uso de SVM (Support Vector Machine). Bazzani (2000) *et al.* [18] combina dos métodos distintos, el primero para descubrir microcalcificaciones gruesas y el segundo de ellos es capaz de descubrir microcalcificaciones mas finas empleando técnicas de multiresolución mediante transformada wavelet. Por ultimo realiza la clasificación empleando un clasificador SVM. Se consigue una sensibilidad del 94’6 %. Mientras, El-Naqa (2002) *et al.* [19] propone un método basado en SVM para detectar microcalcificaciones y lo compara

[3] Receiver Operating Characteristics (Ver Apéndice A)

con otras técnicas empleadas por otros investigadores. Directamente usa ventanas de $M \times M$ píxeles centradas en la ROI como entradas del clasificador SVM, aunque antes suprime el fondo de la imagen filtrando a altas frecuencias, y la salida del clasificador es ± 1 (si es microcalcificación o es tejido sano) para cada píxel de la ventana de M^2 píxeles filtrada. Consigue una sensibilidad de hasta un 94%.

Antes de desarrollar el modelo propuesto por Vega[4], en el cual está basado este proyecto fin de carrera, cabe destacar que todos estos estudios y trabajos dependen mucho de la base de datos sobre la que se trabaje y de la resolución de las mamografías, que suele ir desde 8 bits por píxel hasta incluso 16 bits. Además para poder comparar los resultados se debe analizar tanto el valor que proporciona una mejor relación FVP y FFP como el área bajo la curva ROC.

1.4. Modelo VECO

En este sistema se pretende clasificar cada píxel entre dos clases: tejido sano (clase normal) y microcalcificación (clase anormal), a partir de dos conjuntos (uno para cada clase) de patrones multidimensionales extraídos de las mamografías digitalizadas.

El modelo VECO se compone de cinco etapas [17]:

- **Primera etapa:** Consiste en una segmentación de la imagen previamente diagnosticada con patología de microcalcificación (las regiones con lesiones son conocidas y servirán como ejemplos de entrenamiento para nuestro sistema de detección). En esta etapa, se analiza el contraste de los niveles gris de la imagen segmentada desde la mamografía a tamaño completo. En este proceso se hace un re-escalado de la ROI, para realizar un análisis a diferentes resoluciones de niveles de gris.
- **Segunda etapa:** Se usan técnicas de procesado de imagen para realzar las singularidades de la ROI a diferentes escalas o frecuencias, donde se obtiene una subimagen filtrada por cada banda de frecuencia mediante la transformada wavelet. En este proceso, se suprime el ruido y se refuerzan las singularidades frente al fondo de la ROI. Además, se realiza un análisis de la estructura de los niveles de gris, considerando el entorno de cada píxel (contexto) de la imagen para extraer las mejores características que definen a los píxeles que pertenecen a una microcalcificación y los que pertenecen al tejido sano o normal.
- **Tercera etapa:** En esta etapa, se construyen dos conjuntos de patrones o vectores de características por píxel con cada uno de los tipos de características extraídas: un conjunto de vectores no-contextuales y un conjunto de vectores contextuales. En este caso, cada subimagen obtenida en el procesado de la segunda etapa, contribuye con una característica por píxel al patrón. En este proceso, se agrupan y etiquetan (sobre dos posibles clases) los vectores no contextuales por medio de un método de clasificación no supervisado para segmentar el tejido sano y las posibles microcalcificaciones. La calidad de la segmentación de las ROIs, en cierta forma es supervisada en el proceso de clasificación, debido al conocimiento a priori sobre la patología de la misma.
- **Cuarta etapa:** Una vez obtenidas las etiquetas de los vectores no contextuales, estas se usarán sobre el conjunto de vectores contextuales para determinar las mejores características en el patrón. Se aplica una técnica de selección de características sobre el conjunto de características contextuales. Con esta técnica, se

[4] A partir de ahora al sistema CAD desarrollado por A. Vega Corona nos referiremos como **VECO**

realiza un análisis de correlación entre los conjuntos de características de los vectores contextuales para verificar su relevancia. En esta etapa, se reduce la dimensión de los vectores de características a una dimensión que minimiza el error de clasificación.

- **Quinta etapa:** En esta etapa, se entrena un clasificador neuronal, el cual se encarga de discriminar los patrones pertenecientes a los píxeles de las microcalcificaciones frente a los vectores de los píxeles pertenecientes al tejido sano sobre una región de interés. El clasificador, es entrenado con los conjuntos de entrenamiento de los patrones formados desde una mezcla de las características no contextuales y contextuales. La evaluación de las prestaciones del método propuesto se realiza sobre la curva ROC. Una vez que el clasificador es diseñado, se puede generar una imagen segmentada desde cada ROI presentada de forma arbitraria gracias a la capacidad de generalización de las redes neuronales.

Para realizar su investigación, Vega usó 50 casos de mamografías previamente diagnosticadas con patología de microcrocalcificaciones desde la base de datos DDSM. Cada región de microcalcificaciones ha sido marcada por un radiólogo experto. Además el escáner empleado en la digitalización (HOWTEK 960) presenta una resolución espacial a 43.5 μm a 12 bits por píxel. De esos 50 casos, trabaja sobre 120 regiones de interés de 256×256 píxeles, que es aproximadamente un área de 1 cm^2. Por tanto disponía de 65536 vectores de características por cada ROI analizada. En total 7864320 vectores de características que pueden tener microcalcificaciones o no.

1.4.1. Preprocesado de mamografías

Vega demuestra que los resultados obtenidos en un sistema de detección son muy sensibles a la digitalización de la mamografía. Esto se debe a que el tamaño de las microcalcificaciones en la imagen esta relacionado con la resolución espacial y los niveles de gris del proceso de digitalización de la mamografía realizado por un escáner. Las microcalcificaciones pueden ser tan pequeñas como 100 μm o incluso más pequeñas, por tanto este valor mínimo que debe presentar la resolución de imagen. En concreto, ha trabajado con imágenes que presentan una resolución de 43'5 μm.

Para evaluar el *efecto de los niveles de gris* realiza un escalado de estos desde una resolución alta a otra mas baja desplazando hacia la derecha los bits más significativos y rellenando con ceros por la izquierda.

Para evaluar el *efecto de la resolución* [17], realiza un escalado espacial promediando los valores de la ventana que cubre un píxel de la más alta resolución sobre una ventana de un píxel a baja resolución por medio de la siguiente expresión:

$$I(P_n) = \sum_i \left(A_{n_i} \times I\left(P_i\right) \right) \qquad (1.2)$$

donde $I\left(P_n\right)$ es el valor del nuevo píxel de gris, A_{n_i} es el peso que se le asigna cada píxel en función de su contribución en una ventana a la nueva resolución espacial y $I\left(P_i\right)$ es el nivel de intensidad de los píxeles que cubren la ventana.

Aunque se digitaliza la mamografía completa, solo se analiza la ROI donde tenemos un conocimiento a priori de la presencia de anomalías. El procesado de imagen usada para la segmentación y el realzado de las ROI se realiza sobre la función de

histograma. El histograma de los niveles de gris es una función de distribución que muestra el número de píxeles para cada nivel de gris. Es útil para conseguir información relevante de las microcalcificaciones pues estas se encuentran en los niveles más altos del histograma. Frecuentemente se encuentran entre los niveles de gris superiores al 75% de la escala de gris del histograma. También es empleado para eliminar el fondo de la mamografía, pues se encuentra en los niveles más bajos de los niveles de grises y esto se puede emplear para eliminar el fondo. Por tanto, interesa suprimir los niveles de gris de los píxeles que tienen un valor de amplitud pequeña y realzar aquellos con un valor de nivel de gris elevado, para ello se coloca un umbral para la discriminación [17]. Esto lo hace mediante la aplicación de una transformación no-lineal que nos permite controlar de manera eficiente la razón de contraste y el umbral de forma adaptativa.

La función sigue la expresión siguiente:

$$F(f) = \alpha\left[sigm\left(\kappa\left(f - \beta\right)\right) - sigm\left(-\kappa\left(f + \beta\right)\right)\right] \qquad (1.3)$$

donde $f = f(x, y)$ es el nivel de gris de un píxel de la imagen de entrada y $F(f)$ es la función de transformación no-lineal por puntos, donde α está definida como

$$\alpha = \frac{1}{sigm\left(sigm\left(\kappa\left(1-\beta\right)\right) - sigm\left(-\kappa\left(1+\beta\right)\right)\right)} \qquad 0<\beta<1 \qquad (1.4)$$

y $sigm(x)$ se define como

$$sigm(x) = \frac{1}{1+e^{-x}} \qquad (1.5)$$

donde β y κ son dos parámetros que representan el control del umbral y la proporción del contraste de la imagen respectivamente. Para el desarrollo de su trabajo, Vega empleo como valores óptimos β = 0.3 y κ=20.

1.4.2. Procesado de imagen

En esta etapa, realiza un análisis a través de la transformada wavelet en 2D, obteniéndose en concreto 4 imágenes subbanda, a partir de las cuales se obtendrán cuatro características distintas para cada píxel. Resumiendo, a una ROI de 256x256 píxeles le aplica la transformada wavelet, obteniéndose 4 imágenes del mismo tamaño a distintas frecuencias, teniendo el píxel i-ésimo 4 características procedentes del valor del píxel i-ésimo de las cuatro imágenes subbanda. Estas características se explicarán más adelante y se denotan por **características no contextuales.**

También, realiza un procesado a la mamografía digitalizada que pretende realzar las singularidades y suprimir el ruido dentro de la ROI, esto se hace durante la etapa de etiquetado, donde se determina la pertenencia de un píxel a una clase o a otra.

1.4.2.1. Análisis mediante transformada wavelet

VECO usa la transformada wavelet para la extracción de características debido a las ventajas que presenta frente a la transformada de Fourier, pues no es adecuado su uso en problemas donde existen señales que sólo toman valores en intervalos de corta duración, como es la transición de los niveles de gris de una microcalcificación a los del fondo o al tejido sano de la mama,.ya que las funciones base del análisis de Fourier no son cero sobre todo su dominio, se dice que *carecen de soporte compacto*. Para

solventarlo es necesario emplear funciones base de duración limitada, y por ello, además de las funciones wavelet, se puede emplear también STFT (Short Time Fourier Transform) que intenta resolver el problema descomponiendo las señales como suma de exponenciales de duración limitada por ventanas, pero presenta el inconveniente de que dicha anchura temporal es fija, y por tanto localiza por igual cualquier porción de la señal, ya sean transiciones rápidas o lentas. Así se llega a la transformada wavelet, que tiene ventanas de anchura variable, estrechándose temporalmente para evaluar altas frecuencias y ensanchándose para bajas frecuencias, por lo cual se puede decir que las wavelets se comportan como un sistema de enfoque "tipo zoom".

VECO emplea una detección de contornos a diferentes escalas o bandas de frecuencia (en concreto en 4 escalas distintas de resolución) para extraer las características que representan a las microcalcificaciones desde los máximos wavelet. Este modelo se basa en la detección de extremos locales de la transformada wavelet discreta [17]. Pero se presenta el problema de que esta transformación discreta no es invariante a la translación, es decir, cuando un patrón es trasladado por medio del escalado sus posiciones son modificadas de manera piramidal, lo que dificulta la construcción de nuestro algoritmo de extracción de características y la posterior clasificación de los patrones. Para evitar este fenómeno, Vega aplica un muestreo adaptativo de la transformada wavelet basado en su máximo local [17], el cual es invariante a la traslación, y en este caso el máximo indica la posición de las variaciones rápidas en la amplitud de los niveles de gris, por tanto este muestreo es equivalente a una detección de contornos multiescala. A partir de esta técnica de detección de máximos en una transformada wavelet, se obtienen las características wavelet o no contextuales. Para dicha extracción de características wavelet sobre los máximos locales se aplica un método que combina un filtrado tipo Gradiente-Laplaciano, con el filtrado de Gradiente analizamos los máximos y con el de Laplaciano los cruces por cero.

Para la implementación del análisis por sub-bandas de frecuencia, aplica una combinación de filtros pasa bajas H y pasa altas G, que son conocidos como filtros espejo en cuadratura. Para la reconstrucción o síntesis desde las imágenes sub-banda se aplica un proceso inverso, usando unos filtros distintos $\tilde{H}$ y $\tilde{G}$, así se obtiene una representación invariante a la translación. En el caso de imágenes (dos dimensiones) como es el caso bajo estudio, el procedimiento consiste en dos etapas, primero se filtra por renglones y se diezma por columnas y la segunda etapa consiste en filtrar por columnas y diezmar por reglones.

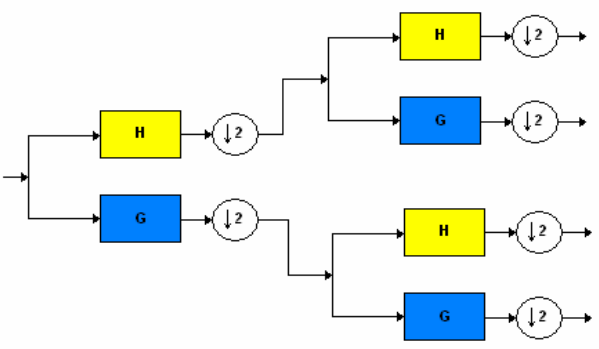


Figura 1.11: Esquema de filtros wavelet

1.4.2.2. Supresión de ruido y realce de singularidades

La *eliminación del ruido* es una tarea difícil, ya que debe existir un equilibrio entre la reducción de ruido y la preservación de las características de la imagen. Para ello, VECO usa un método de umbralización suave para eliminar los coeficientes de ruido a partir del umbral calculado. La combinación del método de supresión de ruido Gradiente-Laplaciano y el proceso de realce no lineal, presenta buenos resultados, sobre todo para las características que se encuentran ocultas entre el fondo de la imagen y que de forma visual, no se perciben o se confunden con tejidos o vasos, debido a un bajo contraste. Con este método, se ha logrado separar el fondo de las características de interés y por tanto, y es el que utiliza en el extractor de características.

La agregación de un píxel fantasma a la imagen bajo estudio, es una técnica que sirve de referencia para verificar la respuesta de los filtros a diferentes escalas y por tanto, a diferentes frecuencias. De forma visual podemos verificar que la singularidad no se pierde, aunque la ajustemos a un valor inferior al máximo nivel de gris de la ROI. Esta técnica resulta muy adecuada para aumentar el contraste entre el fondo y las singularidades.

1.4.3. Extracción de características

En esta parte se aborda la extracción de características wavelet y de los niveles de gris sobre una ROI de la mamografía, es decir, cada patrón o vector de características es construido a partir de un conjunto de variables extraídas de las características wavelet y de las características de la estructura de los niveles de gris de la imagen. Se corresponde un patrón para cada píxel de la ROI. A priori se conoce que en cada una de las ROIs usadas existen píxeles que pertenecen a microcalcificaciones y píxeles que pertenecen a tejido sano, por tanto, podrán pertenecer a una clase u a otra en función de las características de cada uno de estos patrones, quedando el conjunto inicial de patrones Q dividido en dos subconjuntos de patrones Q_o (clase normal) y Q_1 (clase anormal).

VECO emplea dos tipos de características:

- ***Características no contextuales***
- ***Características contextuales***.

Las características no contextuales son extraidas de la descomposición de la imagen original, a través de la transformada wavelet redundante en dos dimensiones, en 4 sub-imágenes correspondientes a cada nivel de descomposición wavelet, en concreto, se extrae el nivel de gris de cada píxel perteneciente a cada sub-imagen. Pero si solamente se emplean estas características para la clasificación se obtienen resultados pobres, lo que lleva a añadir otro tipo de características. El conjunto de los vectores de características no contextuales se denota como X_c.

Las características contextuales son aquellas que contienen información relevante sobre los niveles de gris del entorno de cada píxel analizado, es decir, la información extraída desde cada píxel y sus vecinos es usada para formar un nuevo vector de características que contiene información local sobre los niveles de gris del entorno. El conjunto de los vectores de características contextuales se denota como X_ω. Permiten hacer frente al problema de la detección de microcalcificaciones de una

manera mas realista, ya que para resolverlo no es suficiente considerara características obtenidas desde las singularidades, sino que es necesario agregar otras que tienen que ver con la estructura de los niveles de gris del entorno de los píxeles y así el clasificador aprenda y sea capaz de generalizar. En concreto, toma dos tipos características contextuales [17]: el *contraste local* y el *contraste local normalizado*. El contraste local se define como

$$C_l(x,y) = Sf(x,y) - M(x,y) \qquad (1.6)$$

donde $Sf(x,y)$ es el nivel de gris del píxel de la ROI en la posición (x,y) y $M(x,y)$ es el promedio de los niveles de gris de los píxeles sobre una región determinada centrada en la posición (x,y). Mientras el contraste normalizado se define como

$$C_n(x,y) = \frac{C_l(x,y)}{D(x,y)} \qquad (1.7)$$

donde $C_l(x,y)$ es el contraste local del píxel centrado en la posición (x,y) y $D(x,y)$ es la desviación estándar de los niveles de gris de los píxeles sobre una región determinada centrada en la posición (x,y).

El problema se encuentra en determinar la ventana o el tamaño de la región desde donde son extraídas las características contextuales. Vega lo resuelve en la etapa posterior de selección de mejores características, proponiendo diversos tamaños de ventana y escogiendo aquel que produce mejores resultados.

1.4.3.1. Etiquetado de patrones

Queda por determinar a que clase pertenece cada patrón, es decir, **el etiquetado de los vectores de características**. Como ya se ha comentado, de las ROIs sólo se conoce que son sensibles a tener microcalcificaciones, pero no se sabe que píxeles pertenecen a cada clase (normal o anormal). Por tanto, el sistema debe etiquetar como si en realidad fuese un *"verdadero radiologo"* el que examinase cada píxel.

Para ello, emplea las imágenes obtenidas **tras la supresión de ruido y el realce de singularidades** a partir de un determinado umbral, en definitiva, los píxeles que son sensibles a pertenecer a la clase anormal (los que se encuentran por encima del umbral estimado) son realzados, mientras que aquellos que son sensibles a pertenecer a la clase normal son atenuados.

Tras este proceso comentado ya en el apartado 1.4.2.2, usa un método de **agrupamiento auto-organizado** (no supervisado) para asignar las etiquetas a los vectores de características no contextuales **(**tras realizar la supresión de ruido y realce de singularidades**)**, en concreto se emplea el ***k*-medias**. Como es conocido que los píxeles pertenecientes a microcalcificaciones se encuentran en la parte alta del histograma de los niveles de gris, se agrupan los vectores en K clusters agregando aquellos que representan un nivel de gris elevado a la clase Q_1 (clase anormal), el resto son asignados a Q_o (clase normal). Como dispone de un elevado número de vectores, y teniendo en cuenta además que el numero de píxeles que pertenecen a microcalcificación es mucho menor que el total de los píxeles que se encuentran en la ROI, realiza al final del agrupamiento una partición aleatoria de los píxeles que pertenecen a Q_o, y consiguiendo así el conjunto definitivo de entrenamiento $\{X_\omega, Y_\omega\}$, donde Y_ω es el conjunto de etiquetas asignadas a cada vector de características del conjunto X_ω.

Para encontrar el **número óptimo de clusters**, Vega emplea un criterio de separabilidad basado en el análisis del discriminante de Fisher, que mide el nivel de separación entre cluster mediante las dos matrices de dispersión, matriz de dispersión *intraclase* y matriz de dispersión *entreclase*. La matriz intraclase representa la dispersión de las muestras del cluster alrededor de su centro. La matriz entreclase representa la dispersión entre las muestras alrededor de de la media de la mezcla de los clusters. Definiendo la métrica de separabilidad como $J_d = tr\left(S_w^{-1} \cdot S_b\right)$, así, J_d será grande cuando la dispersión entre-clases es grande o la dispersión intra-clases es pequeña. Por tanto, el subconjunto que genere una J_d más grande, es considerado como el mejor subconjunto para el proceso de agrupamiento, y se escogera aquel valor mínimo de K que maximice la separabilidad.

Como conclusión, cabe destacar que al realiza el etiquetado a diferentes resoluciones, este cambia para cada resolución, aumentando el número de vectores asignados a Q_1 conforme decrece la resolución, por consiguiente, es necesario definir los conjuntos de entrenamientos a distintas resoluciones $\{X_\omega, Y_\omega\}_{b=12,11,10,9,8}$.

En el caso del etiquetado de los vectores de características contextuales, se les asigna la misma etiqueta que la del vector no contextual, es decir, $\{X_c, Y_\omega\}$

1.4.4. Selección de características

En el proceso de generación de características aparece el problema de la dimensionalidad que aparece en la literatura acerca del reconocimiento de patrones [20,21], debido a que este conjunto de característica depende del tamaño de la ventana sobre la cual se analiza el entorno de cada píxel. Por tanto es necesario determinar el conjunto características que presentan mejores resultados para cada tipo de ventana empleada, es decir, **seleccionar las mejores características contextuales** para cada tamaño de ventana. En concreto dispone de una dimensión de 14 características y prueba 10 pares distintos de tamaño para las ventanas empleadas en el cálculo del contraste local y el contraste local normalizado [17].

Para escoger las mejores características que represente mejor la separabilidad entre las dos clases es necesario medir el grado de correlación que hay entre las características del conjunto de los vectores analizados. En concreto, VECO utiliza la combinación de un método de búsqueda de Selección Secuencial Directa (**SFS**, *Sequential Forward Selection*) con una red neuronal de regresión general (**GRNN**, *General Regresión Neural Network*) para estimar el error de predicción como criterio de selección. Para la medida del error, emplea la suma del error cuadrático medio.

El ***método de selección SFS*** es un procedimiento de búsqueda de abajo hacia arriba donde se adiciona una característica cada vez al conjunto de características actual. En cada iteración, la característica a ser incluida en el conjunto de características, es seleccionada del conjunto de posibles características, y así, el nuevo conjunto extendido de características debe producir un error mínimo de clasificación comparado con la adición de cualquier otra característica. El algoritmo debe detenerse en un punto donde añadiendo más características al conjunto de características actual, se incremente el error de clasificación.

La ***GRNN*** es un tipo de red neuronal con memoria, parecida en su funcionamiento a las redes neuronales de función de base radial y se basa en la estimación no paramétrica de la función de densidad de probabilidad usando funciones tipo núcleo (Kernel). Puede emplearse en cualquier problema de regresión y entrenarse con un número mínimo de muestras, pues en caso de tener un gran número de muestras los resultados no son satisfactorios, ya que la superficie de regresión es definida de manera instantánea en cualquier lugar (incluso con solo una muestra). Posee una estructura altamente paralela. La ventaja que presenta esta red es la velocidad de convergencia, pues necesita un solo paso para aprender y alcanzar las prestaciones óptimas en la clasificación. Como inconveniente destaca la cantidad de memoria y cálculos requeridos para entrenar el sistema al estimar un nueva salida, por lo que es adecuada para conjuntos de patrones no muy elevados.

Sean $\boldsymbol{x_c}$ e y, un vector de variables aleatorias y una variable aleatoria respectivamente y sea $f_{xy}(x_c, y)$, la función de probabilidad conjunta de $\boldsymbol{x_c}$ e y. El valor esperado de y dado $\boldsymbol{x_c}$ es

$$E\left[y \middle| x_c\right] = \frac{\int_{-\infty}^{\infty} y \cdot f_{xy}(x_c, y)\, dy}{\int_{-\infty}^{\infty} f_{xy}(x_c, y)\, dy} \tag{1.8}$$

En la práctica, la función de densidad de probabilidad es generalmente desconocida, por lo tanto, esta debe ser estimada desde los valores muestra $\boldsymbol{x_c^{(q)}}$ y su valor deseado $y^{(q)}$. La estimación, se realiza por medio de un estimador de probabilidad conjunta $f_{xy}(x_c, y)$, utilizando una ventana de Parzen, mediante la cual se obtiene una estimación de $\hat{y}$ dado el vector $\boldsymbol{x_c}$ como sigue:

$$\hat{y}(x_c) = \frac{\sum_{q=1}^{M} y^{(q)} \cdot \exp\left(-\frac{D_q^{\,2}}{2\sigma^2}\right)}{\sum_{q=1}^{M} \exp\left(-\frac{D_q^{\,2}}{2\sigma^2}\right)} \tag{1.9}$$

donde σ, es el ancho del núcleo del estimador, M es el número de muestras y $D_q^{\,2}$ es el cuadrado de la distancia entre el vector de entrada $\boldsymbol{x_c}$ y el vector muestra $\boldsymbol{x_c^{(q)}}$ y es definida como

$$D_q^{\,2} = \left(x_c - x_c^{(q)}\right)^T \left(x_c - x_c^{(q)}\right) \tag{1.10}$$

La estructura de la ***GRNN*** (Figura 1.12) que emplea, presenta cuatro capas; una *capa de entrada*, una *capa intermedia*, una *capa de suma* y una *capa de salida*. La función de la *capa de entrada* es simplemente dar paso al vector de variables de entrada $\boldsymbol{x_c}$ a todas las unidades de la capa intermedia. La *capa intermedia* contiene un número de nodos igual al de todas las muestras de entrenamiento $x_c^{(1)}, ..., x_c^{(q)}, ..., x_c^{(M)}$. Así que, cuando un patrón desconocido $\boldsymbol{x_c}$ es presentado a la entrada de la **GRNN**, se calcula el cuadrado de la distancia entre el patrón desconocido $\boldsymbol{x_c}$ y la q-ésima muestra de entrenamiento $\boldsymbol{x_c^{(q)}}$ y ésta es evaluada por la función núcleo siguiente

$$\exp\left(-\frac{D_q^{\,2}}{2\sigma^2}\right) \tag{1.11}$$

La *capa de suma* tiene dos nodos, el A y B. En el nodo A, se calcula el sumatorio de todos los núcleos, evaluados y ponderados con su valor deseado En el nodo B, se calcula solamente el sumatorio de todos los núcleos evaluados

$$\sum_{q=1}^{M}\exp\left(-\frac{D_q^{\,2}}{2\sigma^2}\right) \tag{1.12}$$

En el *nodo de salida*, se calcula el cociente entre los resultados de los nodos A y B para estimar el valor $\hat{y}$. La estimación de $\hat{y}$ se presenta como una razón de probabilidad, por tanto, en el sentido de Bayes, este clasificador cumple con las condiciones de un observador ideal [22].

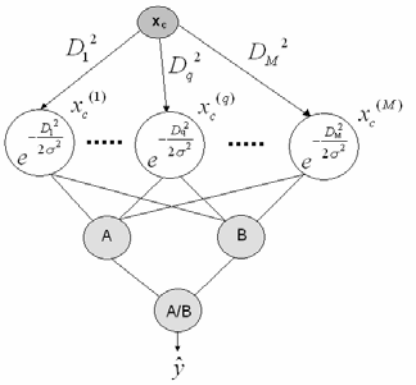


Figura 1.12: Red GRNN empleada en la selección de características

Para la selección de mejores características, Vega escoge una relación entre el número de vectores pertenecientes a la clase Q_1 de 1:2 con respecto al número de vectores de la clase Q_o. En concreto, toma dos conjuntos según lo anterior para cada resolución y evalúa que características son mejores para la etapa posterior de clasificación para cada resolución. Analizando los resultados que obtiene [17], se comprueba que a menor resolución mayor cantidad de características son necesarias para modelar el problema.

Para construir los vectores de entrenamiento se deben considerar las características no contextuales (provenientes del proceso de análisis-síntesis wavelet) y las característica contextuales (provenientes del procesado del contraste local y el contraste local normalizado). Luego, unas representan por un lado la información de los niveles de gris del a singularidades a diferentes escalas y por el otro, la información contextual de esos píxeles. Por tanto, el vector usado para la clasificación $\boldsymbol{x}^{(q)}$ proviene de la mezcla de los vectores no contextuales $\boldsymbol{x}_{\omega}^{(q)}$ y los vectores contextuales $\boldsymbol{x}_{c}^{(q)}$.

1.4.5. Clasificación

Tras haber realizado la extracción de características y seleccionado las mejores, queda el problema de la clasificación. El objetivo principal del clasificador es aproximarse al observador ideal, que es aquel que logra una eficiente discriminación

entre los casos positivos y negativos, lo cual se traduce en un aumento de la sensibilidad y de la especificidad en el proceso de clasificación.

El trabajo propuesto por Vega se centra en el refinamiento de la extracción de características de la imagen, en cambio no estudia y desarrolla en profundidad el proceso de clasificación, lo cual será el objeto de estudio de este proyecto fin de carrera como se comentará en el apartado 1.5.

El clasificador debe ser un clasificador de patrones binario, pues los patrones pueden pertenecer únicamente a dos clases distintas, microcalcificación o tejido sano. Para ello, Vega emplea dos clasificadores y los compara, en concreto, una red ***GRNN*** y una red neuronal progresiva (***FFNN***, *Feed Fordward Neural Network*).

La red ***GRNN*** empleada para la clasificación presenta diferencias con respecto a la red empleada para la extracción de mejores características. Esta red se compone ahora de 4 capas, y en lugar de usar el total de los patrones de entrenamiento para estimar las distancias de los núcleos al patrón de entrada, se estima la distancia usando solo los patrones prototipo de cada cluster, así ahorrando carga computacional. Además el ancho del núcleo del estimador se calcula convenientemente para cada nodo de la capa intermedia, con el objetivo de conseguir un mínimo error de clasificación. La estructura de la red **GRNN** es la siguiente:

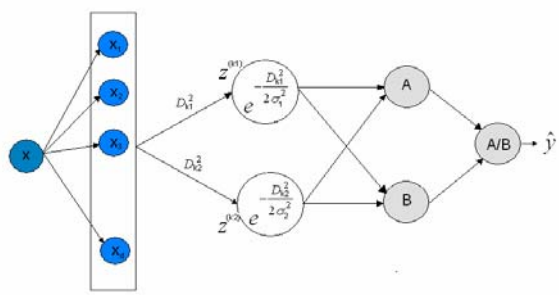


Figura 1.13: Red GRNN empleada en la etapa de clasificación

La red neuronal empleada es la clásica red progresiva ***FFNN*** con tres capas. La capa de entrada presenta 6 nodos, la intermedia 4 nodos y tiene un nodo en la capa de salida. Cada uno de los nodos se compone de funciones de activación de tipo sigmoidal.

En general la red **GRNN** presenta ligeramente mejores resultados que la red **FFNN** para todas las resoluciones de la imagen. Los mejores resultados que se consigue para cada una de las resoluciones (12 bits, 11 bits, 10 bits y 8 bits por píxel) son:

	12 bits	11 bits	10 bits	9 bits	8 bits
Área bajo la curva ROC	0.9568	0.9559	0.9522	0.9107	0.9137
Fracción de falsos positivos	0.2050	0.2050	0.2050	0.2050	0.2050
Fracción de verdaderos positivos	0.9404	0.9394	0.9354	0.8776	0.8825

Tabla 1.1: Resumen de resultados obtenidos por VECO

1.5. Objetivos del proyecto

Tras analizar el modelo VECO, se procede a centrar el objetivo principal del proyecto. El trabajo realizado en este proyecto fin de carrera pretende, a partir del método que se emplea en [17] para la extracción y generación de características para cada píxel de la ROI bajo estudio, realizar un estudio comparativo de distintos tipos de clasificadores, abarcando desde las clásicas redes neuronales progresivas hasta las maquinas de vectores de soporte aplicadas a clasificación, intentando mejorar los resultados obtenidos durante su investigación en términos de sensibilidad y especificidad. En concreto, pretende ampliar la etapa de clasificación del sistema VECO, analizando comparativamente los resultados obtenidos para distintos tipos de clasificadores sobre los conjuntos de patrones derivados de la etapa de preprocesado y extracción de características, teniendo como último objetivo conseguir mejores resultados.

Para lo cual se han desarrollado las siguientes arquitecturas neuronales:

- Perceptrón Multicapa (MLP, *MultiLayer Perceptron*)
- Redes de Funciones de Base Radial (RBF, *Radial Basis Functions*)
- Maquinas de vectores de soporte (SVM, *Support Vector Machines*)

Este proyecto se estructura básicamente en 5 capítulos, donde se describen principalmente los fundamentos teóricos de este trabajo y los resultados y conclusiones obtenidas:

En este primer capitulo se ha planteado e introducido el problema bajo estudio, analizando brevemente el cáncer de mama, además de explicar las ventajas que supone emplear un sistema CAD de detección de microcalcificaciones. También se han expuesto brevemente las distintas investigaciones y publicaciones similares realizadas, viendo la variedad de enfoques que se han presentado hasta la fecha. Por ultimo se ha explicado el trabajo en el cual está centrado este proyecto, el modelo VECO.

En el segundo capitulo se analiza el conjunto de patrones empleado en esta investigación, así como también se describe la base de mamografías DDSM, la mas importante base de datos de mamografía digitalizada, de la que proceden las imágenes de las que extraído los patrones a clasificar.

El tercer capitulo se centra en explicar los fundamentos teóricos de las arquitecturas neuronales que se han implementado en el presente proyecto, analizando uno a uno los distintos clasificadores.

En el cuarto capitulo se describen la estructura de cada una de las arquitecturas empleadas y las diferentes variantes que se han simulado. También se exponen los resultados obtenidos en términos de sensibilidad y especificidad a través del uso de las curvas ROC para los distintos clasificadores implementados durante el desarrollo de este proyecto.

Por último, un quinto capítulo donde se analizan los resultados finales obtenidos, dando las conclusiones y proponiendo las líneas futuras de este trabajo.

1.5.1. Modelo propuesto

A modo de conclusión en la Figura 1.14 se describe mediante un diagrama de bloques del sistema de detección diseñado:

Figura 1.14: Sistema de detección de microcalcificaciones diseñado

Como ya se ha comentado anteriormente, el preprocesado y la extracción de características se realiza de manera similar al modelo VECO, siendo el objetivo mejorar la etapa de clasificación, lo que llevará a una mejora de los resultados finales.

Capítulo 2

Bases de Datos y Conjunto de Patrones

2.1. Introducción

En este segundo capitulo se pretende describir las bases de datos de mamografías digitalizadas y su importante papel en la detección precoz del cáncer de mama. Como ya se ha explicado en el capítulo anterior, actualmente el método más efectivo para la detección el cáncer de mama en un estado precoz es el screening con mamografía. Los signos mamográficos que indican la presencia de la enfermedad, son muchas veces muy sutiles, esto hace que puedan pasar desapercibidos, por ello desde que se comenzó a realizar los screening de mama se consideró que sería interesante desarrollar programas de detección automática que sirvieran de ayuda al radiólogo en su trabajo. Por esta razón, durante el primer Congreso Internacional de Mamografía Digital que se celebró en 1993 en San José (California), se tomó la iniciativa de crear una base de datos de mamografía digital común, con anotaciones fidedignas, para facilitar la comparación de los diferentes métodos de procesado de imagen publicados por los investigadores.

Para concluir este capitulo, se describe el conjunto de patrones empleado en este proyecto, los cuales se han extraído a través de las técnicas propuestas en [17] de la base de datos de la Clínica Puerta de Hierro. Se analiza el formato, procedencia, tipo de características, etc.

2.2. Bases de datos sobre mamografía digitalizada

El objetivo principal de una base de datos sobre mamografía digitalizada es la creación de algoritmos de ayuda en la detección de lesiones en los programas de screening. Otro de los objetivos de las bases de datos es el desarrollo de algoritmos que sirvan de ayuda en el diagnóstico y el desarrollo de programas de aprendizaje o de entrenamiento asistido. Hasta el momento varias instituciones han creado bases de datos de este tipo y numerosas investigaciones las están utilizando en sus trabajos, entre ellos esta investigación.

A continuación se indican las bases de datos de mamografía digitalizada más importantes (los datos relativos a las diferentes bases de datos, se han obtenido a traves de Internet):

- Digital Database for Screening Mammography (DDSM). University of South Florida.
- The Mammographic Image Analysis Society (MIAS). Digital Mammography Database.
- Nijmegen Digital Mammogram Database (NDB). Ha sido incluida recientemente dentro de la base de datos DDSM.

- Lawrence Livermore National Laboratories/University of California at San Francisco (LLNL).
- Washington University Digital Mammography Database.(WDB)

2.2.1. Base de datos DDSM

La base de datos DDSM viene respaldada por el Programa de Investigación del Cáncer de Mama de la Armada de los Estados Unidos, y en este proyecto colaboran el Hospital General de Massachussets, la Universidad de Washington y la Universidad de Wake Forest entre otros.

Esta base de datos es la más completa, se actualiza con mucha frecuencia. Cuando se creo en septiembre de 1999 contaba con 9556 imágenes mamográficas, actualmente (septiembre de 2004) la constituyen 10480 imágenes, correspondientes a 2620 casos, estructurados a su vez en 43 volúmenes. El formato de las imágenes es LJPEG sin perdidas de compresión. Pretende reproducir una casuística completa de 'screening' de mama para aplicarla incluso a estudios de detección y aprendizaje. Todos los casos están clasificados según el sistema BI-RADS[5] (American College of Radiology -ACR- 1998) y se organizan en cuatro tipos de volúmenes para facilitar la clasificación:

- Normal (12 volúmenes, 695 casos)
- Cáncer (15 volúmenes, 914 casos)
- Benigna (14 volúmenes, 870 casos)
- Benigna sin rellamada (2 volúmenes, 141 casos)

Cada caso posee cuatro imágenes que pertenecen a dos proyecciones de cada mama (Medio-Lateral-Oblicuo y Cráneo-Caudal) e información adicional asociada al caso, siendo el numero de bits por píxel, dependiendo del escáner usado, entre 12 y 16. El contenido de cada caso está organizado en un directorio que incluye los diferentes tipos de archivo que se describen a continuación.

- **Archivo ".ics"**. Aporta información adjunta a cada caso:
  - Fecha del estudio
  - Edad de la paciente
  - Fecha de digitalización de las imágenes
  - Tipo de digitalizador utilizado
  - Lista de los archivos de imagen
  - Clasificación de la densidad de la mama (según ACR)
- **Archivo ".LJPEG".** Las imágenes han sido almacenadas en este formato usando LOSSLESS JPEG compression. Incluso con la compresión cada imagen es muy grande debido a que las mamografías han sido escaneadas con una resolución entre 42 y 100 micras
- **Archivos ".overlay".** Cada caso posee entre uno a cuatro archivos ".overlay". Los casos normales no tienen este tipo de fichero. Son ficheros que contienen la siguiente información correspondiente a una imagen:
  - El número de anormalidades presentes

[5] BI-RADS : Breast Imaging Reporting and Data System.
Ver Apéndice A

  - El tipo de lesión según el BI-RADS
  - La categoría de la lesión, de 1 a 5, según BI-RADS
  - El grado de sutileza. Valoración independiente del BI-RADS, que indica el grado de dificultad de detección de la lesión de 1(sutil) a 5 (obvio).
  - El resultado histopatológico
  - La descripción del contorno del hallazgo marcado como anormal. Cada contorno se especifica como una cadena código.
  - La descripción del contorno del hallazgo marcado como anormal. Cada contorno se especifica como una cadena código.

- **Archivos ".16_PGM".** Son cuatro imágenes concatenadas a baja resolución, almacenados en PGM (Portable Gray Map) a 16 bits. Permite una visión rápida de las imágenes del caso

Todas las imágenes han sido digitalizadas mediante escáner láser. El tamaño del pixel, así como los niveles de gris, dependen del escáner utilizado para digitalizar las imágenes:

- DBA: 42 µm, 16 bits
- HOWTEK: 43,5 µm, 12 bits
- LUMISYS: 50 µm, 12bits

El formato ".LJPEG" puede leerse mediante software de la USF, que funciona en entorno Unix. Para visualizarlos en entorno Windows hay que convertir el formato a otro más estándar como ".TIFF".

La base de datos puede adquirirse uno o varios volúmenes en cintas EXABYTE de 8mm o pueden descargarse vía ftp anónimo. El precio ronda los 20-30 $ por volumen. La forma de adquisición, así como una amplísima información sobre la base de datos puede encontrarse en

http://marathon.csee.usf.edu/Mammography/Database.html

Merece destacar otros aspectos interesantes de una página web muy elaborada, ya que contiene un motor de búsquedas según el tipo de caso y la posibilidad de visualizar cada uno de ellos en páginas tipo "thumbnails" con la información correspondiente.

En la Figura 2.1 y en las Tablas 2.1, 2.2 y 2.3 se muestra un ejemplo de la información que proporciona la base de datos DDSM.

Como puede apreciarse en la información proporcionada en la base de datos, los especialistas radiólogos han marcado la mamografía sobre una *región de interés* (**ROI**). La **ROI** es el área que analizamos en las imágenes usadas en este proyecto, puesto que contienen la información relevante sobre la diagnosis.

La base de datos, es uno de los elementos indispensables en el diseño de sistemas CAD para la detección de microcalcificaciones. En esta investigación se han elegido los volúmenes definidos como BCRP_CALC_0 y BCRP_CALC_1 de la base de datos DDSM, para casos benignos y malignos de microcalcificaciones.

Base de datos de Mamografía Digitalizada
DDSM

Volume: cancer-11 Case: A-1252-1

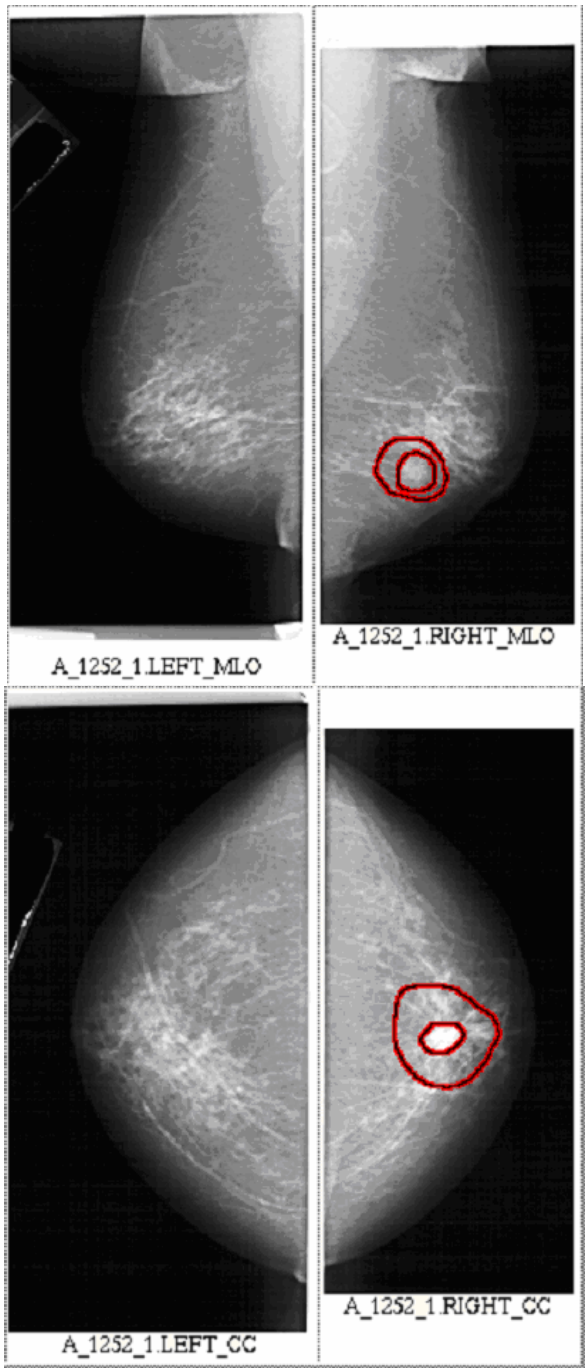


Figura 2.1: Vistas estándar utilizadas en la base de datos DDSM

```
ics_version 1.0
filename A-1252-1
DATE_OF_STUDY 16 10 1991
PATIENT_AGE 74
FILM
FILM_TYPE REGULAR
DENSITY 3
DATE_DIGITIZED 23 7 1998
DIGITIZER HOWTEK 43.5
SEQUENCE
LEFT_CC LINES 5956 PIXELS_PER_LINE 2776 BITS_PER_PIXEL 12 RESOLUTION 43.5 NON_OVERLAY
LEFT_MLO LINES 6871 PIXELS_PER_LINE 3106 BITS_PER_PIXEL 12 RESOLUTION 43.5 NON_OVERLAY
RIGHT_CC LINES 5251 PIXELS_PER_LINE 2326 BITS_PER_PIXEL 12 RESOLUTION 43.5 OVERLAY
RIGHT_MLO LINES 6226 PIXELS_PER_LINE 2701 BITS_PER_PIXEL 12 RESOLUTION 43.5 OVERLAY
```

Tabla 2.1: Información sobre la imagen y datos generales de la paciente

```
FILE: A_1252_1.RIGHT_MLO.OVERLAY

TOTAL_ABNORMALITIES 1
ABNORMALITY 1
LESION_TYPE CALCIFICATION TYPE PLEOMORPHIC DISTRIBUTION CLUSTERED
LESION_TYPE MASS SHAPE IRREGULAR MARGINS SPICULATED
ASSESSMENT 5
SUBTLETY 4
PATHOLOGY MALIGNANT
TOTAL_OUTLINES 2
BOUNDARY
        CORE
```

Tabla 2. 2: Información adicional sobre los *overlays* marcados por el Radiólogo sobre la imagen de la vista Medio Lateral Oblicua

```
FILE: A_1252_1.RIGHT_CC.OVERLAY

TOTAL_ABNORMALITIES 1
ABNORMALITY 1
LESION_TYPE CALCIFICATION TYPE PLEOMORPHIC DISTRIBUTION CLUSTERED
LESION_TYPE MASS SHAPE IRREGULAR MARGINS SPICULATED
ASSESSMENT 5
SUBTLETY 4
PATHOLOGY MALIGNANT
TOTAL_OUTLINES 2
BOUNDARY
        CORE
```

Tabla 2. 3: Información adicional sobre los *overlays* marcados por el Radiólogo sobre la imagen de la vista Cráneo Caudal

2.3. Conjunto de patrones

Como ya se ha comentado anteriormente, en este proyecto se emplean 118 ROIs extraídas de la base de datos DDSM, y como cada ROI tiene un tamaño de 256x256, se dispone un total de **7864320** patrones (casi 8 millones de patrones).

A este conjunto de patrones se le aplica la etapa de procesado y extracción de características explicado en el capítulo 1, obteniéndose un total de **15819** píxeles etiquetados como anormal o microcalcificación, siendo el resto etiquetados como normal o tejido sano. Cada patrón presenta un total de **6** características, **4** de ellas proceden del análisis wavelet y las **2** restantes son el contraste local y el contraste local normalizado, definidos en el capitulo 1, usando ventanas de 5x5 y 3x3 respectivamente, puesto que estos tamaños de ventanas son los óptimos para la clasificación de acuerdo al modelo VECO.

En definitiva, se dispone del conjunto total de patrones etiquetados

$$\{X,T\}=\{(X_w,X_c),T\} \tag{2.1}$$

donde X_w corresponde al conjunto de **características no contextuales** y X_c corresponde al conjunto de características contextuales e T representa el conjunto de etiquetas, siendo -1 para la clase anormal y +1 para la clase normal.

El conjunto total X se corresponde con una matriz bidimensional, donde las columnas están asociadas a los *N* patrones y las filas a las 6 características, por tanto tendrá la forma:

$$X=\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} & x_{1,6} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N,1} & x_{N,2} & x_{N,3} & x_{N,4} & x_{N,5} & x_{N,6} \end{bmatrix} \tag{2.2}$$

Mientras el conjunto total T se corresponde con una vector columna con las *N* etiquetas para cada patrón:

$$T=\begin{bmatrix} t_1 & t_2 & \dots & t_N \end{bmatrix} \tag{2.3}$$

De los dos conjuntos totales $\{X,T\}$ se debe extraer un conjunto de patrones etiquetado para el entrenamiento y otro conjunto de patrones etiquetado que sirva como conjunto de test. El primero de ellos, *conjunto de entrenamiento*, sirve para diseñar y entrenar las distintas arquitecturas neuronales, mientras que el segundo conjunto, *conjunto de test*, sirve para la obtención de resultados y evaluar la capacidad de generalización de la arquitectura.

Para la obtención del *conjunto de entrenamiento*, $\{X_{Tr},T_{Tr}\}$, se ha escogido **9000** patrones aleatoriamente del conjunto total $\{X,T\}$, de los cuales **3000** están etiquetados como anormales y los **6000** restantes como normales.

Para la obtención del *conjunto de test*, $\{X_{Ts},T_{Ts}\}$, se ha escogido los **15819** patrones etiquetados como anormales, y **31681** patrones etiquetados como normales escogidos aleatoriamente del conjunto total $\{X,T\}$, quedando un conjunto de test formado por **47500** patrones.

Capítulo 3
Análisis de arquitecturas neuronales

3.1. Introducción.

En este capitulo se aborda el estudio de las distintas arquitecturas neuronales empleadas en el desarrollo de este proyecto. Como ya se comento en el capítulo de introducción, los clasificadores que se van a implementar son:

- ***Perceptrón Multicapa*** (**MLP**, *MultiLayer Perceptron*)
- ***Redes de Funciones de Base Radial*** (***RBF***, *Radial Basis Functions*)
- ***Maquinas de vectores de soporte*** (***SVM***, *Support Vector Machines*)

Cada uno de ellos es estudiado en profundidad durante este capitulo, incidiendo profundamente en los aspectos teóricos claves en los que se basan los clasificadores diseñados. Y como en el problema bajo estudio se pretende distinguir entre dos clases, tejido sano o microcalcificación, se centrará la mayoría del capítulo en el caso particular del problema genérico de clasificación binaria.

3.2. Perceptrón Multicapa

3.2.1. Conceptos Básicos

Las Redes Neuronales Artificiales (RNA) son sistemas implementados de manera tal que funcionen como las neuronas biológicas de los seres vivos. El cerebro humano continuamente recibe señales de entrada de muchas fuentes y las procesa a manera de crear una apropiada respuesta de salida. Nuestros cerebros cuentan con millones de neuronas que se interconectan para elaborar “Redes Neuronales”. Las RNAs fueron originalmente una simulación abstracta de los sistemas nerviosos biológicos, formados por un conjunto de unidades llamadas “neuronas” o “nodos” conectadas unas con otras. Estas conexiones tienen una gran semejanza con las dendritas y los axones en los sistemas nerviosos biológicos.

3.2.1.1. Breve reseña histórica

La investigación en RNAs ha experimentado tres periodos de extensa actividad. El primer pico en los años 1940s fue debido al trabajo iniciado por McCulloch y Pitts [23]. McCulloch era un psiquiatra y neuroanatomista de formación; gastó unos 20 años pensando acerca de la representación de un evento en el sistema nervioso. Pitts era un prodigio matemático, quien se unió a McCulloch en 1942. En su artículo clásico [24], McCulloch y Pitts describen un cálculo lógico de las redes neuronales que reunían los estudios de neurofisiología y lógica matemática. Con un número suficiente de tales unidades simples y conexiones sinápticas ajustadas apropiadamente y operando

sincronizadamente, McCulloch y Pitts mostraron que una red neuronal constituida así, computaría en principio cualquier función computable. Este fue un resultado muy significativo y a partir de su descubrimiento se ha acordado en general el nacimiento de las disciplinas en redes neuronales e inteligencia artificial [25].

El segundo pico ocurrió en los 1960s con el Teorema de Convergencia del Perceptrón de Rosenblatt [26] y el trabajo de Minsky y Papert mostrando las limitaciones de un perceptrón simple [27]. Los resultados de Minsky y Papert desalentaron el entusiasmo de la mayoría de los investigadores.

El resultante estancamiento en la investigación en redes neuronales duró casi veinte años, aunque existió una actividad importante que emergió en los 1970s llamada *mapas autoorganizativos* usando aprendizaje competitivo. El trabajo de simulación por computador realizado por von der Malsburg (1973) fue tal vez el primero en demostrar la autoorganización [28]. En 1976 Willshaw y von der Malsburg publicaron el primer artículo sobre la formación de mapas autoorganizativos, motivados por mapas ordenados topológicamente en el cerebro [29].

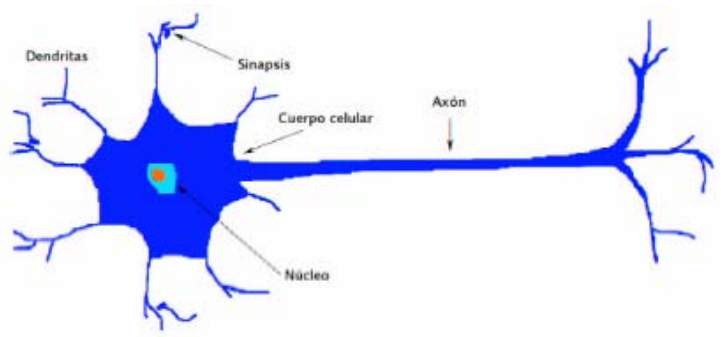
Desde comienzos de 1980 las redes neuronales artificiales han recibido un interés considerable. Los mayores desarrollos tras este resurgimiento incluyeron el enfoque de energía de Hopfield en 1982 [30] y el algoritmo de aprendizaje de retropropagación para perceptrones multicapa propuesto inicialmente por Werbos [31], reinventado varias veces, y después popularizado por Rumelhart y otros en 1986 [32].

Otro importante desarrollo en 1982 fue la publicación del artículo de Kohonen [33] sobre mapas autoorganizativos usando una estructura de malla uni o bidimensional, que fue diferente en algunos aspectos al trabajo previo de Willshaw y von der Malsburg. El modelo de Kohonen ha recibido mucha mayor atención en un contexto analítico y con respecto a aplicaciones.

En los 90s, Vapnik y sus colaboradores inventaron una clase computacionalmente poderosa de redes de aprendizaje supervisado, llamadas ***máquinas con vector de soporte*** (***support vector machine***) para la solución de problemas de reconocimiento de patrones, regresión y estimación de densidad. Una característica novedosa de éstas es la forma natural en que la dimensión de *Vapnik-Chervonenkis* (***VC***) es incorporada en su diseño. La dimensión ***VC*** provee una medida de la capacidad de una red neuronal de aprender de un conjunto de ejemplos [25].

3.2.1.2. La neurona biológica

Una neurona (o célula nerviosa) es una célula biológica especial que procesa información (Figura 3.1). Está compuesta de un cuerpo celular, o *soma*, y dos tipos de ramas que llegan al exterior: el *axón* y las *dendritas*. El cuerpo celular tiene un núcleo que contiene información acerca de rasgos hereditarios y un plasma que sostiene el equipo molecular para producir material necesario para la neurona. Una neurona recibe señales (impulsos) de otras neuronas a través de sus dendritas (receptores) y transmite señales generadas por su cuerpo celular a lo largo del axón (transmisor), el cual eventualmente se divide en ramales y subramales. En los extremos de estos ramales están las *sinapsis*. Una sinapsis es una estructura elemental y una unidad funcional entre dos neuronas (un ramal axón de una neurona y una dendrita de otra). Cuando el impulso alcanza el extremo de la sinapsis, ciertos químicos llamados neurotransmisores son



liberados. Los neurotransmisores se difunden a través del espacio sináptico, para aumentar o inhibir, dependiendo del tipo de la sinapsis, la tendencia de la neurona receptora a emitir impulsos eléctricos. La efectividad de la sinapsis puede ser ajustada por las señales que pasan a través de ella de tal manera que las sinapsis pueden *aprender* de las actividades en las que participan. Esta dependencia de la historia actúa como una memoria, la cual es posiblemente responsable de la memoria humana [23].

Figura 3.1: Diagrama de una neurona biológica

La corteza cerebral contiene cerca de 10^{11} neuronas, masivamente conectadas. Cada neurona está conectada con otras 10^3 a 10^4 neuronas. En total, el cerebro humano contiene aproximadamente 10^{14} a 10^{15} interconexiones. Las neuronas se comunican a través de un corto tren de pulsos, normalmente de milisegundos de duración, a una velocidad de conmutación mucho más lenta que en circuitos electrónicos. Las neuronas artificiales usadas para construir RNAs son mucho menos complejas que las encontradas en el cerebro.

3.2.1.3. La neurona artificial

La unidad básica de una RNA es la neurona. Aunque hay varios tipos de neuronas diferentes, la más común es la de tipo McCulloch-Pitts (1943). En la siguiente figura puede verse una representación de la misma

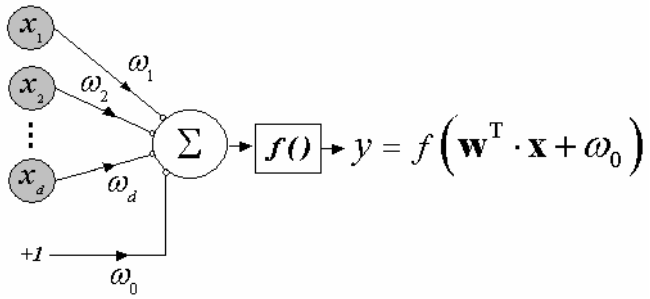


Figura 3. 2: Diagrama del perceptrón simple

Este modelo se conoce como ***perceptrón simple*** de McCulloch-Pitts, y es la base de la mayor parte de la arquitectura de las RNA. Una neurona artificial es un procesador elemental, en el sentido de que procesa un vector de entrada $\mathbf{x} = \{x_1, x_2, ..., x_d\}$ y produce una respuesta o salida única y. Los elementos clave de una neurona artificial los podemos ver en la figura anterior y son los siguientes:

- Las ***entradas*** **x** que reciben los datos de otras neuronas a las que esta conectada

- Los ***pesos sinápticos*** ω_i. En una neurona artificial a las entradas que vienen de otras neuronas se les asigna un peso. El parámetro ω_0 se suele conocer como *bias* o *sesgo*, y puede ser considerado como otro peso de una entrada extra fijada permanentemente a +1.

- Una regla de propagación. Con esas entradas y los pesos sinápticos, se realiza una suma ponderada.

- Una función de activación. El valor obtenido con la regla de propagación, se filtra a través de una función conocida como función de activación y es la que nos da la salida de la neurona. Según para lo que se desee entrenar, se suele escoger una función de activación u otra. En la Tabla 3.1 se muestran las funciones de activación mas usuales

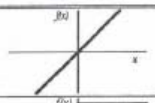
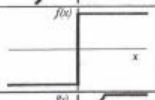
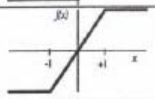
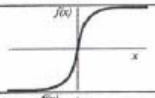
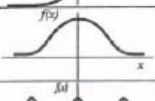
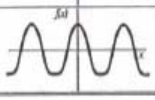
	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = sign(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Lineal a tramos	$y = \begin{cases} -1, & si\ x < -1 \\ x, & si\ +1 \le x \le -1 \\ +1, & si\ x > +1 \end{cases}$	$[-1, +1]$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = tgh(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0,+1]$	
Sinusoidal	$y = A\,\mathrm{sen}(\omega x + \varphi)$	$[-1,+1]$	

Tabla 3.1: Funciones de activación habituales

El perceptrón es capaz tan sólo de resolver funciones definidas por un *hiperplano* (objeto de dimensión N-1 contenida en un espacio de dimensión N) que corte un espacio de dimensión N. Un ejemplo de una función que no puede ser resuelta es el operador lógico XOR. Una explicación más sencilla de un hiperplano sería, hablando en un plano de dos dimensiones, una línea que separa a los elementos existentes en dos grupos. El perceptrón sólo puede resolver una función, si todos los posibles resultados del problema pueden separarse de ésta forma (en dos secciones) es decir, que no se combinen entre sí.

3.2.1.4. Arquitectura de las redes neuronales artificiales

Desde un punto de vista matemático, se puede ver una red neuronal como un grafo *dirigido y ponderado* donde cada uno de los nodos son neuronas artificiales y los arcos que unen los nodos son las conexiones sinápticas. Al ser *dirigido*, los arcos son unidireccionales, es decir la información se propaga en un único sentido, desde una neurona origen a neurona destino. Por otra parte es ponderado, lo que significa que las conexiones tienen asociado un número real, un *peso*, que indica la importancia de esa conexión con respecto al resto de las conexiones.

Lo usual es que las neuronas se agrupen en capas de manera que una RNA esta formada por varias capas de neuronas. Aunque todas las capas son conjuntos de neuronas, según la función que desempeñan, suelen recibir un nombre especifico. Las más comunes son las siguientes:

- *Capa de entrada*: Las neuronas de la capa de entrada, reciben los datos que se proporcionan a la RNA para que los procese.

- *Capas ocultas*: Estas capas introducen grados de libertad adicionales en la RNA. El número de ellas puede depender del tipo de red que estemos considerando. Este tipo de capas realiza gran parte del procesamiento.

- *Capa de salida*: Esta capa proporciona la respuesta de la red neuronal. Normalmente también realiza parte del procesamiento.

El número de capas en una red multicapa se define como el número de capas de pesos, en vez de capas de neuronas. Es decir, un MLP con una capa de entrada, dos capas ocultas y una capa de salida, se dice que es una red de 3 capas, pues presenta 3 capas de pesos.

Si la arquitectura de la red no presenta ciclos, es decir, no se puede trazar un camino de una neurona a sí misma, la red se llama unidireccional (*feedforward*). A partir de ahora siempre nos referiremos a redes neuronales unidireccionales (***FFNN***, *FeedForward Neural Network*).

3.2.2. El perceptrón multicapa

La forma más habitual de organizar las neuronas y capas de una RNA es la que se denomina ***perceptrón multicapa*** (***MLP***, *MultiLayer Perceptron*). El **MLP** (Rosenblatt, 1969) es el modelo más típico de las redes neuronales artificiales con aprendizaje supervisado. El entrenamiento de estas redes, se basa en la presentación sucesiva y de forma reiterada, de pares de vectores en las capas de entrada y salida (vectores entrada y salida deseada). La red crea un modelo a base de ajustar sus pesos en función de los vectores de entrenamiento, de forma que a medida que se pasan estos patrones, para cada vector de entrada la red producirá un valor de salida más similar al vector de salida esperado. Estas redes también se llaman de ***retropropagación*** (***backpropagation***), nombre que viene dado por el tipo de aprendizaje que utilizan.

En general, una red estándar de *L*-capas consiste de una capa de entrada, *L-1* capas ocultas, y una capa de salida de neuronas todas sucesivamente conectadas (completamente o parcialmente) con conexiones de alimentación hacia adelante, pero sin enlaces entre unidades de la misma capa o enlaces retroalimentados entre capas.

La Figura 3.3 muestra un diagrama con su topología típica. La información se propaga de capa en capa (de izquierda a derecha), por medio de las conexiones entre las neuronas de cada capa. En los perceptrones multicapa cada neurona emplea generalmente la función de activación de tipo sigmoide.

Un MLP con una capa oculta puede formar cualquier región convexa en este espacio. Las regiones convexas se forman mediante la combinación de las regiones formadas por cada neurona. Un MLP con dos capas ocultas puede generar regiones de decisión arbitrariamente complejas (Lippmann, 1987). El proceso de separación en

clases que se lleva a cabo consiste en la partición de la región deseada en pequeños hipercubos. Cada hipercubo requiere *2d* neuronas en la primera capa oculta (siendo *d* el número de entradas a la red), una por cada lado del hipercubo, y otra en la segunda capa oculta, que lleva a cabo la operación lógica AND de la salida de los nodos del nivel anterior. La salida de los nodos de este segundo nivel se activarán solo para las entradas de cada hipercubo. Los hipercubos se asignan a la región de decisión adecuada mediante la conexión de la salida de cada nodo del segundo nivel solo con la neurona de salida (tercera capa) correspondiente a la región de decisión en la que este comprendido el hipercubo llevándose a cabo una operación lógica OR en cada nodo de salida. Este procedimiento se puede generalizar de manera que la forma de las regiones convexas sea arbitraria, en lugar de hipercubos.

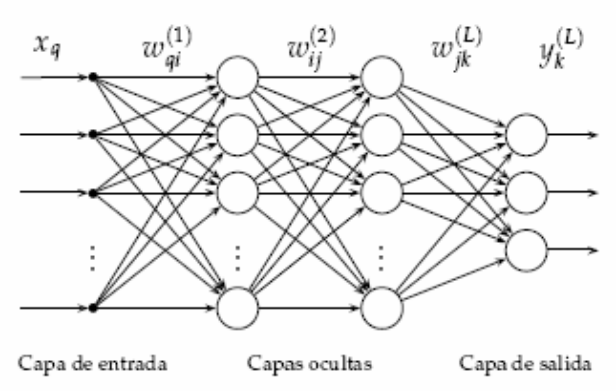


Figura 3.3: Topología genérica de un MLP

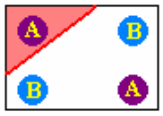
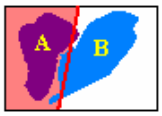
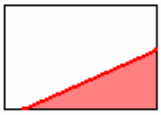
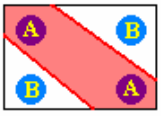
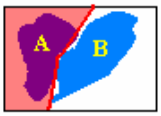
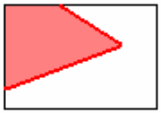
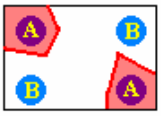
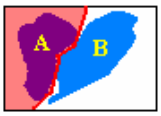
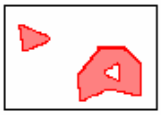
ARQUITECTURA	REGIÓN DE DECISIÓN	EJEMPLO 1: XOR	EJEMPLO 2: CLASIFICACIÓN	REGIONES MÁS GENERALES
Sin capa oculta	Hiperplano (2 regiones)			
Una capa oculta	Regiones polinómicas convexas			
Dos capas ocultas	Regiones arbitrarias			

Tabla 3.2: Relación de arquitecturas típicas del MLP y sus características

3.2.3. Método de entrenamiento : Algoritmo Backpropagation

Tras 20 años de parón en el desarrollo de RNA, surge el ***algoritmo de retropropagación*** de los errores (***BP***, ***BackPropagation algorithm***) que se ha convertido sin duda en el algoritmo más empleado a partir de que en 1986, Rumelhart y McClelland popularizaran este método. La idea básica, sencilla en su filosofía, requirió un desarrollo matemático fuerte para conseguir su convergencia. Se consideran los errores en las capas ocultas, que al propagarse hasta la capa de salida, producen los errores que realmente se miden. En definitiva, con este algoritmo se retropropagan los errores de la capa de salida

hacia las capas anteriores, y emplear estos errores retropropagados para ajustar los pesos de las entradas de la correspondiente capa.

El algoritmo **BP** para **MLPs** es una generalización del algoritmo **LMS**[6] (explicado posteriormente en el apartado 3.3.5.1), pues ambos algoritmos realizan la actualización de los pesos en base al error cuadrático medio. Este algoritmo trabajo de modo supervisado, por tanto es necesario tener un conjunto de *N* patrones de entrenamiento junto con sus *N* salidas deseadas o *targets*. En la Figura 3.4 se describe la red multicapa sobre la que se analizará el algoritmo de retropropagación. Las entradas (procedente del patrón *n*-ésimo, con $n = 1 \ldots N$) de la neurona *j*-ésima de la primera capa oculta se denotan como $y_i(n)$, siendo $\omega_{ij}(n)$ los pesos asociados a la conexión de cada uno de los pesos a la neurona *j*-ésima. La combinación lineal de las entradas en la neurona *j*-ésima ponderadas con los respectivos pesos se denota $\upsilon_j(n)$ y $\varphi(\cdot)$ es la función de activación empleada, generalmente la función sigmoide.

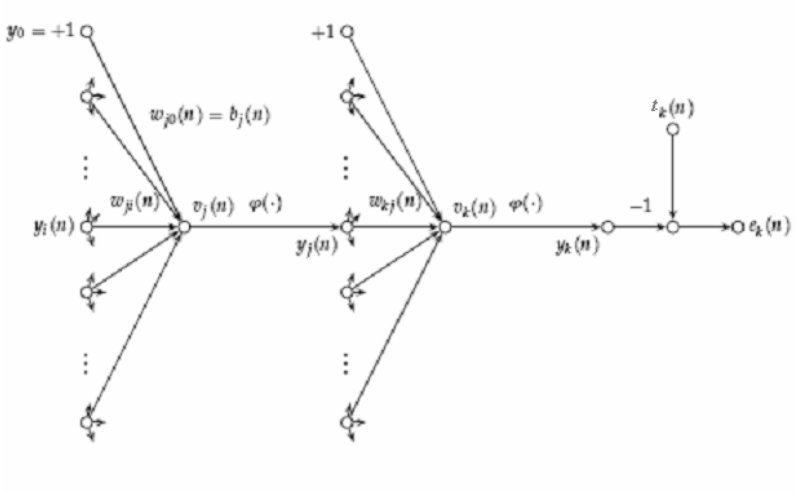


Figura 3.4: Diagrama de conexiones y pesos de un MLP con dos capas ocultas

Se define la señal de error en la salida de la neurona *k* (perteneciente a la capa de salida) en la iteración *n* (esto es, la presentación del *n*-ésimo patrón de entrenamiento) por:

$$e_k(n) = t_k(n) - y_k(n) \tag{3.1}$$

El valor instantáneo del error para la neurona *k* es $\frac{1}{2}e_k^2(n)$. Así mismo, el valor instantáneo del error total *E(n)* se obtiene sumando todos los errores correspondientes a cada una de las neuronas de la capa de salida, esto es

$$E(n) = \frac{1}{2}\sum_k e_k^2(n) \tag{3.2}$$

donde *k* va desde uno hasta el número de neuronas de la capa de salida de la red. El error cuadrático promedio, teniendo en cuenta los *N* patrones ($n = 1 \ldots N$), es obtenida sumando todos los *E(n)* sobre todos los *n* y normalizando con respecto al tamaño del conjunto, de forma que

$$E_{prom} = \frac{1}{N}\sum_n E(n) \tag{3.3}$$

[6] *LMS*, Least Mean Squares

Tanto $E(n)$ como E_{prom} están en función de los parámetros libres (pesos sinápticos y niveles de sesgo) de la red. Para un conjunto dado de entrenamiento, E_{prom} representa la función de coste como medida de rendimiento del aprendizaje. El objetivo del proceso de aprendizaje es ajustar los parámetros libres de la red para minimizar E_{prom}. Para esto, se considera un método simple de entrenamiento en el que los pesos se actualizan presentando uno a uno los patrones hasta que se termine una *época*, esto es, hasta que se presente completamente el conjunto de entrenamiento. Los ajustes a los pesos se hacen de acuerdo a los errores respectivos calculados para cada patrón presentado a la red.

Considérese una neurona *k* que está recibiendo un conjunto de señales producidas por una capa de neuronas a su izquierda (ver Figura 3.4). El algoritmo de retropropagación aplica la corrección $\Delta\omega_{kj}(n)$ al peso $\omega_{kj}(n)$, que es proporcional a la derivada parcial $\partial E(n)/\partial \omega_{kj}(n)$. Esta derivada parcial determina la dirección de búsqueda en el espacio de los pesos para el peso $\omega_{kj}(n)$.

La corrección $\Delta\omega_{kj}(n)$ aplicada a $\omega_{kj}(n)$ está definida por la regla delta:

$$\Delta\omega_{kj}(n) = -\eta \frac{\partial E(n)}{\partial \omega_{kj}(n)} \tag{3.4}$$

donde η es la tasa de aprendizaje del algoritmo. El signo menos describe el descenso del gradiente en el espacio de pesos (es decir, busca una dirección que el cambio de peso reduzca el valor de $E(n)$). Finalmente se tiene que

$$\Delta\omega_{kj}(n) = \eta \cdot \delta_k(n) \cdot y_j(n) \tag{3.5}$$

donde el gradiente local $\delta_k(n)$ está definido por

$$\delta_k(n) = e_k(n)\varphi_k'(\upsilon_k(n)) \tag{3.6}$$

donde la prima en $\varphi_k'(\upsilon_k(n))$ denota diferenciación.

De acuerdo con las dos anteriores ecuaciones existe un factor involucrado que es la señal de error $e_k(n)$. En este contexto, se pueden identificar dos casos distintos, dependiendo donde este localizada la neurona. En el caso descrito, la neurona *k* pertenece a la capa de salida, por lo tanto se puede determinar el gradiente local de una manera simple, ya que se tiene una respuesta deseada $d_k(n)$ y $e_k(n)$, que se puede calcular fácilmente.

El segundo caso se presenta cuando una neurona *j* está localizada en una capa oculta de la red, cuando no se ha especificado una respuesta deseada para esa neurona. De acuerdo a esto, la señal de error para una neurona oculta se determina recursivamente en términos de las señales de error de todas las neuronas con las cuales esa neurona oculta está directamente conectada. Considérese la situación descrita en la Figura 3.5. Se redefine el gradiente local $\delta_j(n)$ para la neurona oculta como

$$\delta_j(n) = -\frac{\partial E(n)}{\partial y_j(n)}\varphi_j'(\upsilon_j(n)) \tag{3.7}$$

Finalmente se obtiene la formula de retropropagación para el gradiente local $\delta_j(n)$ así:

$$\delta_j(n) = \varphi_j'(\upsilon_j(n))\sum_k \delta_k(n)\omega_{kj}(n) \tag{3.8}$$

De esta forma la corrección aplicada $\Delta\omega_{ji}(n)$ a l peso que conecta la neurona *i* con la neurona *j* está definido por la regla delta:

$$\Delta\omega_{ji}(n) = -\eta \cdot \delta_j(n) \cdot y_i(n) \qquad (3.9)$$

En la aplicación del algoritmo de retropropagación se distinguen dos pasos distintos. El primer paso es referido como el paso hacia adelante, y el segundo como paso hacia atrás. En el *paso hacia adelante* los pesos se mantienen inalterables y las salidas de los nodos de la red se calculan neurona por neurona. Esta fase de cómputo comienza en la primera capa oculta presentándole el vector de entrada, calculando las respectivas salidas y pasando esta información hacia las demás capas ocultas (si existen); la fase termina cuando se calcula la señal de error (comparar la salida de la neurona con su respuesta deseada) para cada neurona en la capa de salida.

El *paso hacia atrás*, por otra parte, comienza en la capa de salida pasando las señales de error hacia la izquierda a través de la red, capa por capa, y recursivamente calculando el δ (gradiente local) para cada neurona. Este proceso recursivo permite que los pesos sinápticos de la red sufran cambios de acuerdo con la regla delta (ecuación 3.9). Para una neurona en la capa de salida su gradiente local se calcula con la ecuación 3.6. Por consiguiente, esta se usa para calcular los cambios en los pesos que alimentan la capa de salida. Dados estos δs se usa la ecuación 3.8 para calcular los gradientes de todas las neuronas de la penúltima capa y así aplicar las correcciones a los respectivos pesos. Este cómputo recursivo se realiza capa por capa hasta ajustar los pesos de toda la red. La figura siguiente muestra la representación como grafo de la ecuación 3.8, asumiendo que la capa de salida consta de m_L neuronas.

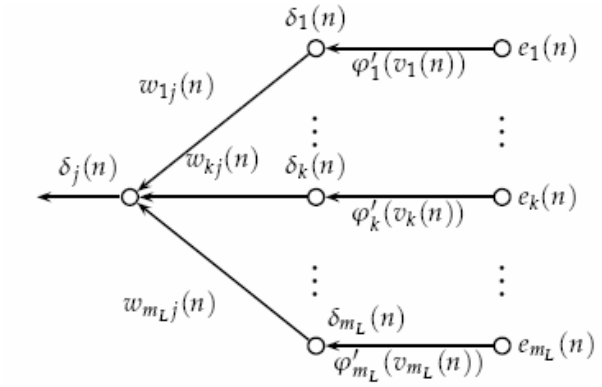


Figura 3.5: Representación de la propagación hacia atrás de los errores en una neurona

3.2.3.1. Resumen del algoritmo de retropropagación

En el modo de entrenamiento secuencial o estocástico [25], la actualización de pesos se efectúa después de la presentación de cada patrón de entrenamiento. Para ser específicos, considérese una época consistente en *N* patrones de entrenamiento organizados en el orden $\{(\mathbf{x}_1,\mathbf{t}_1),(\mathbf{x}_2,\mathbf{t}_2),...,(\mathbf{x}_N,\mathbf{t}_N)\}$ (pares entrada-respuesta deseada). El primer patrón se presenta a la red, y se realiza la secuencia de operaciones hacia adelante y hacia atrás, descrita anteriormente, y como resultado se tiene un cierto ajuste de pesos. Luego se presenta el segundo patrón de igual forma y así sucesivamente hasta el último patrón, momento en el cual se concluye una época.

Para este modo de operación secuencial, el algoritmo realiza ciclos a través de la muestra de entrenamiento $(\mathbf{x}_n,\mathbf{t}_n)_{n=1}^{N}$ de la siguiente manera [25]:

1. *Inicialización*. Obtener los pesos sinápticos de una distribución de probabilidad uniforme, cuya varianza y desviación estándar de los campos locales inducidos de las neuronas se encuentren en la transición entre las partes lineales y saturadas de la función de activación sigmoide.
2. *Presentaciones de los ejemplos de entrenamiento*. Presentar una época de ejemplos de entrenamiento a la red. Realizar los pasos hacia adelante y hacia atrás, descritos en los puntos 3 y 4, respectivamente.
3. *Propagación hacia adelante*. Para un ejemplo de entrenamiento $(\mathbf{x}_n, \mathbf{d}_n)$, calcular las sumas ponderadas. La suma ponderada $\upsilon_j(n)$ para la neurona *j* en la capa *l* es:

$$\upsilon_j^{(l)}(n) = \sum_{i=0}^{m} \omega_{ji}^{(l)}(n) \cdot y_i^{(l-1)}(n) \qquad (3.10)$$

donde $y_i^{(l-1)}(n)$ es la salida de neurona *i* en la capa anterior *l-1* en la iteración *n*, $\omega_{ji}^{(l)}(n)$ es el peso de la neurona *j* en la capa *l* que es alimentada por la neurona *i* en la capa *l-1*, y *m* es la cantidad de neuronas de la capa anterior. En este caso se tienen en cuenta las entradas para los sesgos. Asumiendo el uso de una función sigmoide, la salida de la neurona *j* en la capa *l* es

$$y_j^{(l)}(n) = \varphi_j\left(\upsilon_j^{(l)}(n)\right) \qquad (3.11)$$

Si la neurona *j* está en la capa de entrada (esto es, *l = 0*), entonces

$$y_j^{(0)}(n) = x_j^n \qquad (3.12)$$

donde x_j^n es el j-ésimo elemento del vector de entrada $\mathbf{x}_n$. Si la neurona *j* está en la capa de salida (esto es, *l = L*, donde *L* es el índice de la última capa de la red), entonces el error se calcula así:

$$e_j(n) = t_j(n) - y_j^{(L)}(n) \qquad (3.13)$$

donde $d_j(n)$ es el *j*-ésimo elemento del vector de respuestas deseadas $\mathbf{d}_n$

4. *Propagación hacia atrás.* Calcular los gradientes locales de la red, definidos por

$$\delta_j^{(l)}(n) = \begin{cases} \text{Para la neurona } j \text{ en la capa de salida } L: \\ e_j^{(L)}(n) \cdot \varphi_j'\left(\upsilon_j^{(L)}(n)\right) \\ \text{Para la neurona } j \text{ en la capa oculta } l: \\ \varphi_j'\left(\upsilon_j^{(l)}(n)\right) \sum_k \delta_k^{(l+1)}(n) \cdot \omega_{kj}^{(l+1)}(n) \end{cases} \qquad (3.14)$$

El ajuste en los pesos de la red de la capa *l* se realiza de acuerdo a la regla delta generalizada

$$\omega_{ji}^{(l)}(n+1) = \omega_{ji}^{(l)}(n) - \eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n) \qquad (3.15)$$

donde η es la tasa de aprendizaje. Escoger un valor adecuado influye en la convergencia del algoritmo, por lo que en general se toman valores de pequeños, pero esto significa que tendrá que hacer un gran número de iteraciones. Si la tasa de de aprendizaje es grande, el resultado son cambios

drásticos en los pesos, avanzando muy rápidamente por la superficie de error, con el riesgo de estar oscilando alrededor del mínimo. Este efecto se puede contrarrestar con tasa adaptativa o con la adición del término de momento (ver apartado 3.2.2.2).

5. *Iteración.* Iterar los cómputos hacia adelante y hacia atrás de los puntos 3 y 4 presentando nuevas épocas de ejemplos de entrenamiento a la red hasta que el criterio de terminación se cumpla. El orden de presentación de los ejemplos de entrenamiento debería ser aleatorio de una época a otra. Los parámetros de momento y de tasa de aprendizaje son habitualmente ajustados (usualmente decrementados) cuando el número de iteraciones se incrementa.

3.2.4. Problemas y variantes del algoritmo de retropropagación

Como puede verse en la literatura relativa a RNAs, se ha escrito mucho acerca de los problemas y excelencias de los perceptrones multicapa, así como sobre sus aplicaciones. Surgen muchas variantes para aplicaciones concretas, o para resolver algunos de los problemas. A continuación se incluye un resumen de los problemas más importantes de los MLPs y variantes del algoritmo BP.

3.2.4.1. Problemas de los MLPs y algoritmo BP

- *Numero de capas y de neuronas*. No hay procedimientos claros para decidir el mejor número de capas y/o neuronas en el MLP. Hay publicaciones sobre cotas mínimas y máximas, generalmente para problemas de clasificación, en función del número de patrones de entrenamiento, pero en la práctica no se utilizan. Generalmente se determina de forma intuitiva y/o experimental, pero sin duda la experiencia del investigador sirve de gran ayuda.

- *Conjunto de entrenamiento*. Se puede decir casi lo mismo que en el apartado anterior. Lo que está claro, es que si se pretende un buen aprendizaje del dominio del problema, o una buena generalización de los patrones, estos deben cubrir todo el espectro del dominio. El número de patrones a emplear dependen del problema y la experiencia del investigador ayuda. Se suele determinar casi siempre (a veces solo se dispone de patrones concretos) de forma experimental.

- *Preparación de los datos*. En general, los datos no se usan directamente en la forma en que se obtienen o los proporciona el problema. Casi siempre se adaptan los datos reales al modelo usado. Por ejemplo, las salidas hay que normalizarlas en [0,1] si se usa activación sigmoide, porque sino, el error nunca podrá reducirse de forma satisfactoria. También es habitual normalizar las entradas, aún cuando no sea necesario. Obviamente habrá un post-procesamiento de resultados, para hacerlos válidos en la realidad del problema.

- *Velocidad de convergencia*. El algoritmo de retropropagación es lento y se han propuesto muchas modificaciones y variantes para mejorar la velocidad del entrenamiento. Algunas se reflejan en el apartado de variantes, y otras como la adición de un término de momento ya se han expuesto anteriormente.

- *Mínimos locales*. Algo importante que se debe tener en cuenta es la posibilidad de existencia de mínimos locales en la superficie de error (ver Figura 3.6).El algoritmo de retropropagación con descenso de gradiente no garantiza que se alcance un mínimo global, una vez que se alcance un mínimo, el algoritmo se detiene, quedando a veces "atrapado" en un mínimo local, del que no puede salir, y por tanto el aprendizaje no se hace bien. Una forma de obviar esto es realizar el aprendizaje varias veces, partiendo de pesos diferentes cada vez, y seleccionar la ejecución que mejor resuelve el problema. Algunas de las variantes buscan usar otros procesos de optimización no lineal más seguros que el de gradiente descendente.

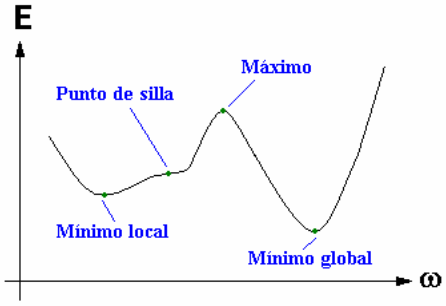


Figura 3.6: Función de error E en función de los pesos ω

- *Sobreentrenamiento*. Es el problema de que se aprende muy bien los ejemplos de entrenamiento, pero cuando se usa el MLP entrenado, con muestras que no ha aprendido, no es capaz de dar buenas respuestas. Se dice entonces que la red no generaliza bien. Se puede evitar esto seleccionando muy bien el conjunto de entrenamiento y parando el entrenamiento cuando el error es suficientemente pequeño pero no excesivamente pequeño.

- *Saturación*. Se puede producir cuando las salidas esperadas en cada neurona de salida son 0 o 1. Al pretender acercar las salidas de la red a estos valores, nos ponemos en las zonas donde la función de activación tiene tangente de pendiente casi cero y entonces no se produce apenas modificación de los pesos y por tanto del error. La forma de resolverlo es cambiar los valores de salidas esperadas a 0.1 y 0.9. Esto puede requerir adaptación de los datos reales. También se produce si los pesos se hacen muy grandes, por que los resultados de la red ya no varían.

- *Estabilidad y robustez*. Un tema interesante, sobre todo para las implementaciones en hardware de la RNA entrenada, es la estabilidad del aprendizaje sobre variaciones en los pesos (y/o las entradas). Hay medidas de estabilidad estadística, que se pueden usar para seleccionar el MLP entrenada más estable de entre varias ejecuciones. También hay métodos que tienen en cuenta estas medidas en el entrenamiento de la red, aunque son más lentos.

3.2.4.2. Variantes del algoritmo de retropropagación

Han surgido muchas variantes del algoritmo de retropropagación, en general intentando disminuir el tiempo de computación, o para resolver alguno de los problemas del algoritmo. La cantidad de estas variantes hace casi imposible reflejarlas todas, por lo que indicaremos algunas de las más destacadas.

- *Tasa de aprendizaje adaptativa.* Con el algoritmo de retropropagación basado en descenso de gradiente, la tasa de aprendizaje η se mantiene constante durante el entrenamiento. Como ya se ha comentado más arriba, el algoritmo es muy sensible al valor de η. Para evitarlo, se emplea η variable durante el proceso de entrenamiento. Pretende mantener un paso de aprendizaje elevado pero garantizando un aprendizaje estable, sin oscilaciones. Primero, se calcula la salida inicial de la red y el valor del error. En cada época nuevos pesos son calculados usando la nueva tasa, y se calcula de nuevo la salida de la red y el nuevo error. Si el nuevo error excede el de la anterior época en un valor mayor que una relación predefinida (p.e. 1.04), entonces los pesos de la época actual son descartados y la tasa de aprendizaje se reduce multiplicando por un valor constante (p.e. 0.7). En el otro caso, si el nuevo error es menor que el anterior, el valor de la tasa de aprendizaje es incrementada. (p.e. 1.05).

- *Retropropagación con término de momento.* Para evitar oscilaciones en torno a un mínimo y para conseguir una mayor velocidad de convergencia, se suele añadir un termino de *momento* a la formula de actualización de los pesos por descenso de gradiente. De tal manera que la ecuación de actualización de pesos queda:

$$\omega_{ji}^{(l)}(n+1) = \omega_{ji}^{(l)}(n) - \eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n) + \mu \cdot \Delta\omega_{ji}^{(l)}(n-1) \qquad (3.16)$$

  Siendo

$$\Delta\omega_{ji}^{(l)}(n-1) = \omega_{ji}^{(l)}(n) - \omega_{ji}^{(l)}(n-1) \qquad (3.17)$$

  donde μ es el parámetro de momento y está comprendido entre 0 y 1. Presenta el inconveniente de que se introduce un segundo parámetro μ cuyo valor debe ser escogido.

- *Método del gradiente conjugado*. Este algoritmo pertenece a los métodos de búsqueda de línea, y pretende minimizar el número de iteraciones necesarias para alcanzar el mínimo de la superficie de error buscando la dirección y magnitud óptimas de actualización de los pesos, de manera que, para superficies cuadráticas se demuestra que se alcanza la solución en *d* pasos, siendo *d* la dimensión del espacio de los pesos.

  El algoritmo básico **BP** ajusta los pesos en la dirección negativa a la del gradiente, pues está es la dirección en la cual decrece la función de error. Sin embargo, el uso del vector gradiente en cada iteración como dirección de búsqueda del mínimo global no es la mejor opción. La solución está en escoger sucesivas direcciones de búsqueda tal que, en cada paso del algoritmo, el gradiente no cambie en esa dirección y sea ortogonal al anterior, de tal manera que se dice que las direcciones de actualización sucesivas son conjugadas. El método del gradiente conjugado, sin aplicarlo al algoritmo de retropropagación, se rige por las siguientes ecuaciones:

$$\boldsymbol{\omega}(n+1) = \boldsymbol{\omega}(n) + \alpha(n) \cdot \mathbf{d}(n) \qquad (3.18)$$

$$\mathbf{d}(k) = -\mathbf{g}(n+1) + \beta(n) \cdot \mathbf{d}(n) \qquad (3.19)$$

$$\beta(n)=\begin{cases}\dfrac{\mathbf{g}(n+1)^T\cdot\left[\mathbf{g}(n+1)-\mathbf{g}(n)\right]}{\mathbf{g}(n)^T.\mathbf{g}(n)} & \text{(Polak-Ribiere)}\\ \dfrac{\mathbf{g}(n+1)^T\cdot\mathbf{g}(n+1)}{\mathbf{g}(n)^T.\mathbf{g}(n)} & \text{(Fletcher-Reeves)}\end{cases} \qquad (3.20)$$

donde $\alpha(n)$ se obtiene por un procedimiento de búsqueda en linea y $\mathbf{g}(n)$ es el gradiente de la función de error en la iteración n.

- *Métodos de Quasi-Newton*. La idea del algoritmo de Newton es evitar el cálculo directo de la matriz Hessiana (pues implica un alto coste computacional) en la obtención de la dirección de Newton: $\mathbf{d}=-\mathbf{H}^{-1}\cdot\nabla\mathbf{E}$. Para ello calcula **H** en cada paso del algoritmo tomando únicamente información de primer orden, es decir, empleando tan sólo el valor de los pesos y del gradiente. Además, dicho algoritmo garantiza en cada paso que el Hessiano sea definido positivo (evitando que converja a un máximo de la función de error). A continuación se explica el proceso que dicho algoritmo realiza:

  1. En primer lugar se parte de una aproximación de $\boldsymbol{H}^{-1}$: $\boldsymbol{G}=\boldsymbol{I}$.

  2. Se actualizan los pesos según la siguiente expresión:
  $$\boldsymbol{\omega}(n)=\boldsymbol{\omega}(n)-\lambda(n)\cdot\mathbf{G}(n)\cdot\nabla\mathbf{E}(n) \qquad (3.21)$$
  En dicha actualización se emplea una búsqueda en línea para la determinación del parámetro λ en cada paso. Esto es lo que garantiza que no se produzca la inestabilidad del algoritmo.

  3. A continuación se actualiza la matriz **G** en función de la información de primer orden (simplificando de esta manera considerablemente el cálculo de $\boldsymbol{H}^{-1}$). G es actualizada de la forma
  $$\mathbf{G}(n+1)=\mathbf{G}(n)+F\left[\mathbf{G}(n),\boldsymbol{\omega}(n+1),\boldsymbol{\omega}(n),\nabla E(n+1),\nabla E(n)\right] \qquad (3.22)$$

  4. $\mathbf{G}(n)$ debe satisfacer la *"Condición Quasi-Newton"* o *"Condición Secante"*. Dicha condición es la que garantiza precisamente que la matriz Hessiana quede definida positiva en cada paso del algoritmo. Esta condición es la siguiente
  $$\boldsymbol{\omega}(n+1)-\boldsymbol{\omega}(n)=-\mathbf{H}^{-1}\cdot\left(\nabla E(\boldsymbol{\omega}(n+1))-\nabla E(\boldsymbol{\omega}(n))\right) \qquad (3.23)$$
  Esta condición se satisface para dos puntos cercanos al mínimo.

  Una fórmula de la estima **G** propuesta que satisface la condición de Quasi-Newton es la propuesta por *Broyden-Fletcher-Goldfard-Shanno* (*BFGS*):
  $$\mathbf{G}(n+1)=\mathbf{G}(n)-\frac{\mathbf{p}\cdot\mathbf{p}^T}{\mathbf{p}^T\cdot\mathbf{v}}-\frac{\left(\mathbf{G}(n)\cdot\mathbf{v}\right)\cdot\mathbf{v}^T\cdot\mathbf{G}(n)}{\mathbf{v}^T\cdot\mathbf{G}(n)\cdot\mathbf{v}}+\left(\mathbf{v}^T\cdot\mathbf{G}(n)\cdot\mathbf{v}\right)\cdot\mathbf{u}\cdot\mathbf{u}^T \qquad (3.24)$$
  Donde
  $$\mathbf{p}=\boldsymbol{\omega}(n+1)-\boldsymbol{\omega}(n) \qquad (3.25)$$
  $$\mathbf{v}=\nabla E(n+1)-\nabla E(n) \qquad (3.26)$$

$$\mathbf{u} = \frac{\mathbf{p}}{\mathbf{p}^T \cdot \mathbf{v}} - \frac{\mathbf{G}(n) \cdot \mathbf{v}}{\mathbf{v}^T \cdot \mathbf{G}(n) \cdot \mathbf{v}} \qquad (3.27)$$

Si ponemos $\mathbf{G}$ en la condición de Quasi-Newton anterior veríamos que ésta queda satisfecha. Vemos además que $\mathbf{G}(n+1)$ es función de los pesos y del gradiente, esto es, de la información de primer orden, así como de $\mathbf{G}$ en la etapa *k*-ésima. Se evita de esta manera calcular la matriz ***H*** de manera directa.

Por último, si la superficie de error es cuadrática y aplicamos el algoritmo Quasi-Newton, se demuestra que dicho algoritmo converge al mínimo en *d* pasos (con *d* número de dimensiones del espacio de pesos) y $\mathbf{G}$ converge a ***H***, siempre y cuando λ sea calculada mediante una búsqueda en línea exacta.

3.3. Redes de Funciones de Base Radial

3.3.1. Introducción

Las ***redes de funciones de base radial*** o **RBF** (***Radial Basis Functions***) constituyen un modelo de red de neuronas que cuentan con un gran ámbito de aplicaciones prácticas. Este tipo de redes presentan algunas características que lo diferencian, como es natural, de otros modelos clásicos como las redes neuronales.

3.3.1.1. Arquitectura de las redes RBF

La arquitectura genérica de una red **RBF** consta de tres capas de neuronas con conexiones de tipo *feedforward* como puede verse en la Figura 3.7.

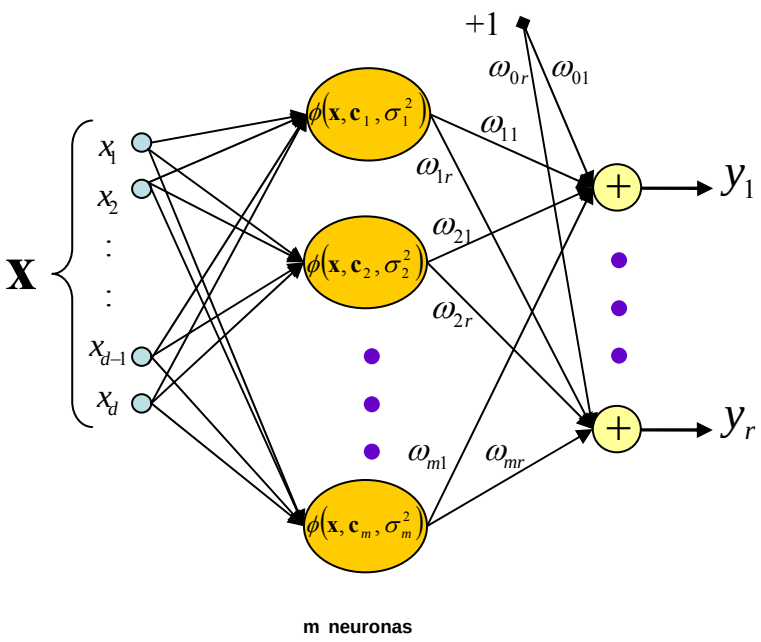


Figura 3.7: Arquitectura típica de una red RBF

Capa Entrada. La primera capa está formada por las neuronas de entrada, representando cada una de estas neuronas cada una de las características que componen

el vector o patrón de entrada $\mathbf{x} = \{x_1, x_2, ..., x_d\}$, es decir, la neurona i-ésima de la capa de entrada está asociada a la característica i-ésima x_i del patrón de entrada $\mathbf{x}$. Esta primera capa no realiza ningún procesado de los datos de entrada, pasándolos tal cual a la siguiente capa; es por esta razón por la que algunos autores consideran a estas redes como de dos capas, sólo considerando aquellas capas que realizan algún tipo de procesado de los datos. La capa de entrada simplemente realiza una función de simple transmisión de información.

Capa Oculta. La siguiente es la capa oculta, que al contrario que en otros tipos de redes, es única. Cada una de las neuronas de la capa oculta está asociada a un vector sináptico llamado comúnmente *centroide*. Al contrario de lo que suele ser habitual en otras redes, en vez de calcular la suma ponderada de las entradas (producto escalar del vector de entrada y de pesos sinápticos) y aplicarla una cierta función de activación, calcula la distancia euclídea entre el vector de las entradas $\mathbf{x} = \{x_1, x_2, ..., x_d\}$ y el centroide de cada una de las neuronas ocultas, y a esa distancia se le aplica una *función de base radial* que realiza una transformación no lineal, y cuya salida pasara a la siguiente capa. La función más utilizada es la gaussiana, aunque existen otras funciones que se pueden utilizar. La respuesta de las neuronas de la capa oculta es *localizada*, pues en contra de lo que ocurre en otras redes, las neuronas de la capa oculta sólo responden con una intensidad apreciable en el caso de que el vector de entrada presentado y el centroide pertenezcan a una zona próxima en el espacio de entrada.

Capa Salida. Finalmente, las neuronas de la capa de salida, realizan una combinación lineal de las funciones de base radial, es decir, las salidas proporcionadas por las neuronas de la capa oculta son multiplicadas por el peso de su conexión y sumadas todas ellas, para a continuación el resultado aplicarlo a una función de activación como por ejemplo la lineal (generalmente) o la sigmoide.

3.3.2. Modelo genérico de una red RBF

Para una ***red neuronal de base radial*** con d neuronas en la capa de entrada, m neuronas en la capa oculta, r neuronas en la capa de salida y salida lineal, la formula que describe la salida de la neurona *k*-ésima de la capa oculta, y_k, viene dada por la siguiente expresión:

$$y_k(\mathbf{x}) = \sum_{j=1}^{m} \omega_{kj} \cdot \phi_j(\mathbf{x}) + \omega_{k0} \qquad k = 1...r \tag{3.28}$$

donde ω_{kj} representa el **peso** asociado a la conexión que une la neurona *j*-ésima de la capa oculta con la neurona *k*-ésima de la capa de salida, ω_{k0} es el **umbral** asociado a la neurona de salida *k*-ésima, $\mathbf{x}$ es la entrada de la red y $(\phi_j)_{j=1...m}$ son las **funciones de base radial o de activación** de cada uno de las neuronas de la capa oculta.

Las **funciones de base radial** se expresan normalmente como traslaciones de una función prototipo de la siguiente forma:

$$\phi_j(\mathbf{x}) = \phi\left(\frac{\|\mathbf{x} - \mathbf{c}_j\|}{d_j}\right) \tag{3.29}$$

donde $\mathbf{c}_j$ es el centro asociado a la neurona *j*-ésima de la capa oculta, d_j es la desviación, anchura o factor de escala para la distancia euclídea entre el vector de entrada $\mathbf{x}$ y el centroide, $\|\mathbf{x}-\mathbf{c}_j\|$.

Como función prototipo se suele emplear una gaussiana de la forma:

$$\phi(r)=e^{-\frac{r^2}{2\sigma^2}} \tag{3.30}$$

donde r es la distancia entre el centroide y el vector de entrada (ver Figura 3.8).

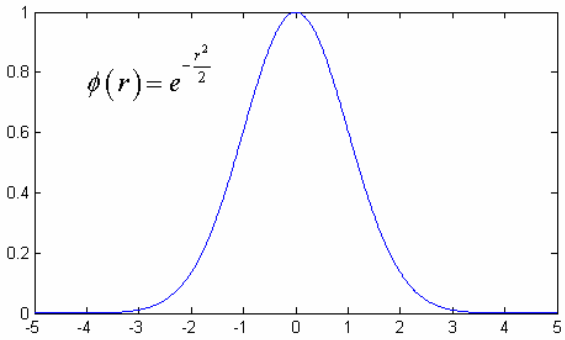


Figura 3.8: Función de base radial gaussiana

Aunque también se pueden emplear otro tipo de funciones, como se presenta en la siguiente tabla:

Nombre	Expresión	Parámetros
Gaussiana	$\phi(r)=e^{-\frac{r^2}{2\sigma^2}}$	Con parámetro de normalización $\sigma>0$
Cauchy	$\phi(r)=\frac{1}{1+\frac{r^2}{\sigma^2}}$	Con parámetro de normalización $\sigma>0$
Multi-Cuadraticas	$\phi(r)=\left(r^2+\sigma^2\right)^{\frac{1}{2}}$	Con parámetro de normalización $\sigma>0$
Multi-Cuadráticas Generalizadas	$\phi(r)=\left(r^2+\sigma^2\right)^{\beta}$	Con parámetro de normalización $\sigma>0 \quad 0<\beta<1$
Cúbica	$\phi(r)=r^3$	Con parámetro de normalización $\sigma>0$

Nota: En caso de tener exponente negativo en las funciones multi-cuadráticas, se obtienen las llamadas funciones multi-cuadráticas inversas y multi-cuadráticas generalizadas inversas.

Tabla 3. 3: Relación de funciones de base radial

La mayoría de estas funciones tienen similar dependencia radial, caracterizándose porque alcanzan un máximo en torno al origen ($r=0$) y decrecen hacia cero $\phi(r)\to 0$ cuando $r\to\infty$. Por tanto, estas funciones se caracterizan porque la respuesta es *localizada*. Se dice que además son *aproximadores universales* ya que pueden aproximar arbitrariamente bien cualquier función continua y acotada (Poggio & Girosi, 1990)

También se han usado funciones de base radial con una dependencia distinta, como por ejemplo la multicuadrática, cuya expresión se encuentra en la tabla anterior, pero en el que la raíz está en el numerador. En este caso la respuesta no es localizada.

El ***parámetro de normalización*** σ es un factor de escala que nos da la anchura de la función. Viene a ser como el radio de influencia de la neurona en el espacio de las entradas. En algunos casos se pueden escoger diferentes valores de σ para cada función de cada neurona, con lo que estas funciones $\phi_j(r)$ dejan de tener simetría radial y adquieren formas elipsoidales.

Como ya se ha comentado, la salida de la red es generalmente lineal aunque también existen otras posibilidades como la función de activación de forma sigmoidea (Figura 3.9)

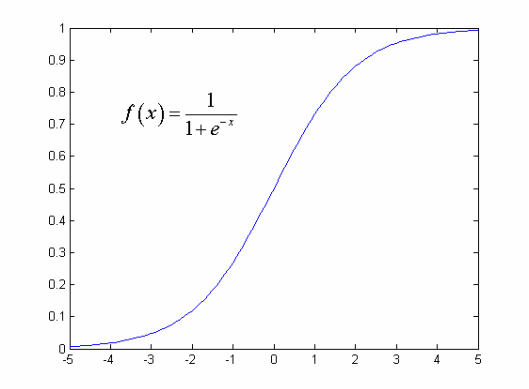


Figura 3.9: Función sigmoide

El principal problema de este tipo de redes se produce con la alta dimensionalidad del espacio de entrada, es decir, cuando el número de neuronas en la capa de entrada es muy alto, se produce el problema llamado ***curse of dimensionality***. Este problema provoca que el número de neuronas en la capa oculta, necesarias para obtener una buena generalización, crezca de forma exponencial.

Obviamente, y debido en gran parte a este problema, dependiendo del número de neuronas en la capa oculta se conseguirán resultados muy diferentes. La determinación de las neuronas necesarias para resolver un problema suele ser un factor crítico, aunque no el único, a la hora de conseguir una buena generalización de la red.

Normalmente la determinación de la topología de la red (número de neuronas de la capa oculta), se hace de forma artesanal, es decir, el investigador utilizando su experiencia, determina el número de neuronas a utilizar, si bien el método de prueba y error es uno de los más utilizados cuando no se dispone de esta experiencia.

3.3.3. Aprendizaje en redes de funciones de base radial

El aprendizaje de las **redes RBF** se caracteriza porque en general, y al contrario que otros tipos de redes es un aprendizaje por etapas, es decir, que se realiza por separado para cada una de las dos capas de procesamiento (oculta y de salida).

Los métodos de aprendizaje para las redes **RBF** deben de dar respuesta a varios puntos que determinan las características de dicho aprendizaje. Esencialmente los algoritmos deben determinar lo siguiente:

- ***Número de neuronas de la capa oculta***
- ***Selección de centroides en la capa oculta***
- ***Ajuste de anchuras en la capa oculta***
- ***Entrenamiento de la capa de salida***

En concreto, se han empleado en este trabajo la familia de métodos que intentan resolver el problema usando una visión de agrupamiento (o ***clustering***). Para el entrenar los pesos de la capa de salida se ha usado el algoritmo ***LMS*** (ver apartado 3.3.5.1)

3.3.4. Aprendizaje basado en agrupamiento

A este tipo de aprendizaje basado en métodos de agrupamiento se le suele denominar como *Autoorganización de centros*. Muy resumidamente se puede decir que su funcionamiento consiste en localizar los *m* centroides en los puntos más representativos de la señal de entrada.

3.3.4.1. Selección del número de neuronas en la capa oculta

Una cuestión importante a la hora de trabajar con redes **RBF** es la elección del *número de neuronas ocultas*. Cada neurona, cubre una parte del espacio de entrada, de modo que la elección correcta del número de neuronas se basa en que cubran suficientemente (para una aplicación dada) dicho espacio.

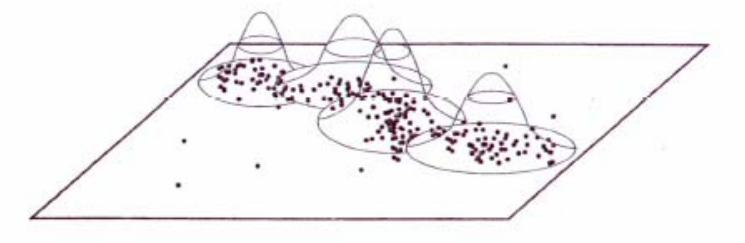


Figura 3.10: Distribución de neuronas en el espacio de entrada

Sin embargo, es usual trabajar con espacios de entrada de muchas variables, lo que provoca que el número de nodos necesario crezca exponencialmente al hacerlo la dimensión.

Al final, habrá que llegar a un compromiso entre el número de neuronas seleccionado, el error que se alcanza en el ajuste de los patrones de aprendizaje y la capacidad de generalización, ya que puede aparecer sobreajuste al tomar demasiadas neuronas ocultas.

Como se ha comentado con anterioridad, es necesario fijar previamente el número de neuronas presentes en la capa oculta, de modo que el entrenamiento se realiza con ese número de neuronas. En este proyecto, se analizan los resultados obtenidos con distinto número de neuronas en la capa oculta.

3.3.4.2. Selección de centroides en la capa oculta

En el entrenamiento de las neuronas debe en primer lugar determinarse el valor de los centroides. Existen varios algoritmos para realizar esto que se basan en el concepto de ***agrupamiento*** o ***clustering***, que consiste en que se busca que los centroides estén asociados a zonas en el espacio de entrada donde hay una mayor densidad de patrones de aprendizaje. Entre dichos algoritmos destacan el algoritmo clásico *k*-medias, y los algoritmos de aprendizaje competitivo propuestos por Kohonen.

3.3.4.2.a. Algoritmo FSCL

En concreto, en este trabajo se ha empleado el método ***FSCL***, *Frequecy Sensitive Competitive Learning*, para el cálculo de los centroides. Es un algoritmo de aprendizaje competitivo no supervisado. En un algoritmo competitivo, solo se actualiza una centroide en cada momento, compitiendo los centroides entre si para resultar "vencedor" ante un patrón de entrada determinado, por lo que es necesario determinar el criterio de selección del centroide ganador, que normalmente es la distancia del centroide a la entrada, así el centroide más próximo al patrón de entrada es el vencedor, actualizando su valor convenientemente. Además por ser no supervisado, solo se dispone de los patrones de entrada y no de las salidas deseadas.

El algoritmo **FSCL** consigue equipar la capacidad de victoria entre todos los centroides, evitando así el fenómeno de los "centroides muertos", que son aquellos que no se actualizan por encontrarse muy alejados de los datos y no resultan vencedores. La manera de conseguirlo es teniendo en cuenta durante la selección del centroide vencedor el número de veces que cada uno de los centroides resulta "vencedor", es decir, cada vez que un centroide resulta vencedor, se le penaliza multiplicando la distancia del centroide al patrón de entrada por el número de veces que dicho centroide resulta vencedor. Con ello se consigue que la probabilidad de victoria de cada uno de los centroides sea aproximadamente idéntica, e igual a $1/m$ siendo m el numero de centroides. A continuación se describe el algoritmo que se ha implementado:

1. Se escoge el número de centroides m y el número de iteraciones, y se inicializan. Una manera de inicializarlos es situar los centroides iniciales en torno a la media de la distribución de los datos de entrada.
2. Inicializar el valor del parámetro de actualización μ, y el número de veces que resulta vencedor cada centroide $\mathbf{v} = \{v_1, v_2, ..., v_m\}$ a la unidad.
3. Escoger un patrón de entrada $\mathbf{x}$ aleatoriamente del conjunto total de N patrones $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, y calcular las distancias d_j de dicho patrón a los distintos centroides $\mathbf{c}_j$
4. Aplicar el criterio de penalización, es decir, multiplicar el numero de victorias v_j de cada centroide por la distancia obtenida d_j, y tomar k según
$$k = \arg\min_j \left\{ v_j \cdot \left\| \mathbf{x} - \mathbf{c}_j \right\|^2 \right\} \qquad (3.31)$$
5. Actualizar el centroide k-ésimo según la ecuación siguiente:
$$\mathbf{c}_k = \mathbf{c}_k + \mu \cdot (\mathbf{x} - \mathbf{c}_k) \qquad (3.32)$$
6. Volver al paso 3 hasta que el numero de actualizaciones sea igual a N
7. Actualizar el parámetro de actualización según:
$$\mu = \alpha \cdot \left(e^{-i \cdot \beta} - e^{-N \cdot \beta} \right) \quad donde$$
$$\alpha = \frac{\eta}{\left(e^{-\beta} - e^{-N \cdot \beta} \right)} \quad con \quad \eta = 0.3 \quad \beta = 0.0005 \qquad (3.33)$$
8. Volver al paso 2 hasta acabar el numero de iteraciones, entonces el proceso finaliza

3.3.4.2.b. Aprendizaje por cuantificación vectorial. LVQ

Para conseguir mejor distribución de los centroides en el espacio de entrada, en este proyecto además de aplicar **FSCL**, se aplica un ***aprendizaje por cuantificación vectorial***, ***LVQ***, *Learning Vector Quantization*. En concreto, se emplea el ***LVQ3***. Con dicho aprendizaje se pretende distribuir mejor los centroides obtenidos mediante **FSCL** e incluso consiguiendo eliminar algunos de ellos. Por tanto, **LVQ3** proporciona una manera automática de obtener el número de centroides y su situación. A continuación se explica el funcionamiento de los **LVQ**.

La ***cuantificación vectorial*** es un método clásico que produce una aproximación de una **fdp**[7] $p(\mathbf{x})$ del vector de entrada **x** usando un número finito de vectores $\boldsymbol{m_i}$, que los llamaremos a partir de ahora centros, con i =1,2,…,k. Una vez escogido los centros, la aproximación de **x** implica encontrar el centro $\boldsymbol{m_c}$ más cercano a **x**. Se pretende que con la distribución óptima de los centros se minimice el error E dado por

$$E = \int \left\| \mathbf{x} - \mathbf{m}_\mathbf{c} \right\| \cdot p(\mathbf{x})\, d\mathbf{x} \qquad (3.34)$$

donde $d\mathbf{x}$ es el volumen diferencial en el espacio de **x** y el índice $\boldsymbol{c}$ es el índice de aquel centro más cercano a **x**. Se dice que el centro $\boldsymbol{m_c}$ es el "vencedor". No existe una solución cerrada para la óptima situación de los centros.

Kohonen propuso una serie de procedimientos iterativos, conocidos como ***LVQs***, que resolvían el problema. Cada uno de ellos mueve los centros intentando conseguir la mejor clasificación de un conjunto de entrenamiento dado usando la regla 1-nn[8]. Los patrones son analizados uno a uno, es decir, hay tantas iteraciones como número de patrones del conjunto de entrenamiento, y los centros son actualizados después de que cada iteración.

El método original ***LVQ1*** emplea la siguiente regla de actualización. El centro mas cercano a al patrón de entrada **x**, $\boldsymbol{m_c}$, es actualizado por

$$\begin{aligned} &\mathbf{m}_\mathbf{c} \leftarrow \mathbf{m}_\mathbf{c} + \alpha(t)\cdot\left[\mathbf{x} - \mathbf{m}_\mathbf{c}\right] \text{ } \textit{si } \mathbf{x} \textit{ es clasificado correctamente por } \boldsymbol{m_c} \\ &\mathbf{m}_\mathbf{c} \leftarrow \mathbf{m}_\mathbf{c} - \alpha(t)\cdot\left[\mathbf{x} - \mathbf{m}_\mathbf{c}\right] \text{ } \textit{si } \mathbf{x} \textit{ es clasificado incorrectamente} \end{aligned} \qquad (3.35)$$

mientras que el resto de centros no son actualizados. Inicialmente para $\alpha(t)$ se escoge un valor pequeño y es reducido linealmente hacia cero conforme aumenta el número de iteraciones. El efecto de esta regla de adaptación es mover el centro hacia las muestras más cercanas de su misma clase, y alejarlo de las otras clases.

La anterior regla de adaptación fue modificada con ***LVQ2***. Asumiendo que dos centros $\boldsymbol{m_i}$ y $\boldsymbol{m_j}$, que pertenecen a clases diferentes, son los mas cercanos a un determinado patrón de entrada **x**, se define una *ventana* simétrica, de anchura mayor que cero, entre ambos centros, de tal manera que $\boldsymbol{m_i}$ y $\boldsymbol{m_j}$ solo serán actualizados si **x** cae dentro de dicha ventana (ver Figura 3.11). **LVQ2** se basa en la idea de aproximar las fronteras de decisión hacia la frontera ideal de Bayes.

[7] fdp: función de distribución de probabilidad

[8] Regla 1-nn: Un patrón de entrada **x** de clase C_i es clasificado correctamente si el centro más cercano tiene su misma clase C_i

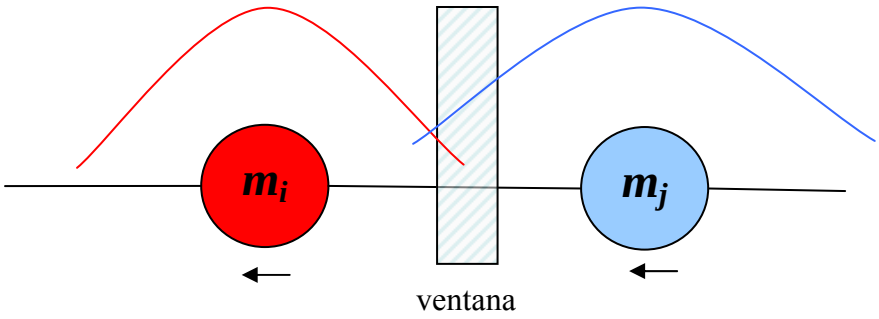


Figura 3.11: Representación de la ventana empleada en LVQ2

Las ecuaciones de actualización son las siguientes:

Si C_i es la clase del centro más cercano, pero **x** *pertenece a $C_j \neq C_i$, donde C_j es la clase del segundo del centro más cercano y además* **x** *tiene que "caer" dentro de la "ventana", entonces:*

$$\begin{aligned} \mathbf{m}_i &\leftarrow \mathbf{m}_i - \alpha(t)\cdot[\mathbf{x}-\mathbf{m}_i] \\ \mathbf{m}_j &\leftarrow \mathbf{m}_j + \alpha(t)\cdot[\mathbf{x}-\mathbf{m}_j] \end{aligned} \qquad (3.36)$$

en todos los demás casos, no hay actualización:

$$\mathbf{m}_k \leftarrow \mathbf{m}_k$$

El tamaño óptimo de la ventana debe ser determinado experimentalmente y depende del número de muestras. Resulta adecuado anchuras de ventana entre 10% y 20% de la diferencia entre $\boldsymbol{m_i}$ y $\boldsymbol{m_j}$. También resulta apropiado dar valores de α entre 0.02 y 0.05, pero decreciente con el número de iteraciones.

LVQ2 presenta una serie de inconvenientes. El primero de ellos, es debido a que las correcciones son proporcionales a la distancia entre x al centro $\boldsymbol{m_i}$ o $\boldsymbol{m_j}$, pero la corrección debida a $\boldsymbol{m_i}$ (clase correcta) siempre será mayor que la corrección debida a $\boldsymbol{m_j}$ (clase incorrecta). Una manera de compensar este efecto es actualizar para todos los patrones que caigan dentro de la ventana, con la única condición de que $\boldsymbol{m_i}$ y $\boldsymbol{m_j}$ deben pertenecer uno a la clase correcta y el otro a la incorrecta. El segundo problema surge del hecho de que en ese caso, los centros no se alejarán, y no se ajustarán a las distribuciones de los patrones. Combinando estas ideas, surge **LVQ3**, cuyas ecuaciones de actualización son las siguientes:

Si $\boldsymbol{m_i}$ y $\boldsymbol{m_j}$ son los centros mas cercanos a la muestra **x**, *y* **x** *y $\boldsymbol{m_j}$ pertenecen a la misma clase C_j, mientras que $\boldsymbol{m_i}$ pertenece a $C_i \neq C_j$ y además* **x** *tiene que "caer" dentro de la "ventana", entonces:*

$$\begin{aligned} \mathbf{m}_i &\leftarrow \mathbf{m}_i - \alpha(t)\cdot[\mathbf{x}-\mathbf{m}_i] \\ \mathbf{m}_j &\leftarrow \mathbf{m}_j + \alpha(t)\cdot[\mathbf{x}-\mathbf{m}_j] \end{aligned} \qquad (3.37)$$

y su **x**, $\boldsymbol{m_i}$ *y* $\boldsymbol{m_j}$ *pertenecen a la misma clase:*

$$\mathbf{m}_k \leftarrow \mathbf{m}_k + \varepsilon\cdot\alpha(t)\cdot[\mathbf{x}-\mathbf{m}_k]$$

en todos los demás casos, no hay actualización:

$$\mathbf{m}_l \leftarrow \mathbf{m}_l$$

El valor óptimo de ε depende del tamaño de la ventana, siendo menor conforme decrece el tamaño de ventana.

3.3.4.3. Ajuste de parámetro de normalización σ

Hasta ahora sólo se han ajustado las posiciones de los centroides en el espacio de los patrones de entrada. Sin embargo, se hace necesario además ajustar los valores de los ***parámetros de normalización*** o desviaciones σ_j de cada neurona. Estos valores son importantes porque determinan el área o zona de influencia de cada neurona de la capa oculta. Cuanto mayor sea la desviación asociada a una neurona, mayor será el campo de acción de esa neurona, de tal forma que una neurona da mayores valores de salida para un patrón que está cerca de esa neurona (centroide), y tanto más cuanto más grande es la desviación. Lo que ocurre es que por ejemplo, una neurona puede dar valores próximos a cero para un determinado patrón (de ahí el carácter local de las neuronas), pero si se aumenta la anchura o desviación asociada a la neurona, entonces la salida para ese patrón puede variar enormemente.

Los valores de estos parámetros deberán ser tales que se cubra bien el espacio de entrada. Una posible forma de conseguir esto será haciendo que el radio σ de una neurona fuese igual a la máxima distancia del conjunto de los patrones asociados a él con respecto al centroide. Esta forma, presenta el inconveniente de polarizar los pesos en torno a los patrones de entrenamiento (y por tanto se puede obtener una mala generalización) y además en el caso de que el conjunto de patrones asociados a una neurona estuviese formado por un único patrón, este parámetro sería nulo y solo daría salida significativa en el caso de que un patrón coincidiese con el centroide.

En la práctica se recurre a fijar ese valor de una forma heurística, utilizándose diversas técnicas de las que destacamos:

- Tomar como valor de σ_j, un valor fijo para todos los casos e igual

$$\sigma^2 = \frac{d^2}{2m} \qquad (3.38)$$

  siendo *d* la máxima distancia entre centroides. Este método es simulado en este proyecto.

- Tomar como valor de σ un promedio de las distancias del centroide con respecto al resto de los centroides. Existen varias posibilidades:

  · Media geométrica de la distancia euclídea del centroide j-ésimo a los dos centroides más cercanos

$$\sigma_j = \sqrt{d_{j,1} \cdot d_{j,2}} \qquad (3.39)$$

  donde $d_{j,1}$ y $d_{j,2}$ son las distancias a los dos centros más cercanos.

  · Estimación a partir de las patrones por cada clase

$$\sigma_j^2 = \frac{1}{\#C_j} \sum_{\mathbf{x} \in C_j} \left\| \mathbf{x} - \mathbf{c}_j \right\|^2 \qquad (3.40)$$

  donde C_j es la clase a la que pertenece cada centroide $\mathbf{c}_j$. Pero antes es necesario establecer la pertenencia de cada centroide a una determinada clase, para ello se asigna a cada centroide los patrones que estén a la menor distancia, y la clase C_j es aquella que sea mayoritaria dentro del grupo de patrones asignados a cada centroide $\mathbf{c}_j$. Este método también se ha empleado en el proyecto.

Las desviaciones pueden influir mucho en la *capacidad de generalización* de una red neuronal u otra. Para ver esto con mayor claridad, piénsese primero en obtener unos errores de aprendizaje lo más bajos posibles para lo cual lo más sencillo es tener tantas neuronas como patrones de entrada; si los valores de las anchuras son altos, las neuronas tendrán poco carácter local con lo que una neurona dará valores altos para patrones que estén alejados de ella, interfiriendo en la salida que sobre ese patrón de una neurona que esté más cerca de ese patrón, y aumentando el error de aprendizaje. Si las anchuras son pequeñas, posiblemente las salidas que las neuronas den para un patrón alejado serán próximas a 0, dando valores altos sólo para los patrones que estén más cerca de esa neurona y el error de aprendizaje baje.

El problema es que cuando se entrena la red hay que atender también a los *errores de validación o test*, y este error es el que determina la capacidad de generalización de la red, que no es más que la reacción de la red ante nuevos datos no aprendidos, es decir que para valores de entrada no aprendidos, la salida de la red ha de ser lo más próxima a la salida real. Si las desviaciones no son lo suficientemente grandes como para atender a esos nuevos patrones, las neuronas devolverán valores muy pequeños, y la salida de la red no se va a adecuar a los valores reales. En definitiva, entender que la desviación asignada a una neurona es algo muy importante, y que con desviaciones pequeñas, el error de entrenamiento puede ser pequeño pero el de test puede ser muy grande, con lo que hay que buscar que ambos errores sean parecidos y lo más bajos posibles.

3.3.5. Entrenamiento de la capa de salida

Tras acabar el entrenamiento de los nodos ocultos, se procede a entrenar las neuronas que conforman la capa de salida. Para ello se usaran las distintas salidas $\phi_1, \phi_2, ..., \phi_m$ que produce la capa de nodos ocultos para los patrones de entrada $\mathbf{x}$. Estas salidas son en general función de los valores de entrada, los centroides obtenidos en el entrenamiento de la capa oculta y los radios σ. Se pueden emplear distintos métodos de optimización como algoritmos de descenso de gradiente, de gradiente conjugado, de Newton, de Quasi-Newton…

En este proyecto, se ha utilizado para el entrenamiento de los pesos de la capa de la salida ω_{kj} el algoritmo LMS en su versión bloque, el cual se explica brevemente a continuación para nuestro caso particular donde solo hay una salida y.

3.3.5.1. Algoritmo LMS

El ***algoritmo LMS***, *Least Mean Squares*, también conocido como algoritmo de *Widrow-Hoff* o *regla delta* (ya que trata de minimizar una delta o diferencia entre el valor observado y el deseado en la salida de la red). Pertenece a la familia de los algoritmos de gradiente estocástico. Con el término "estocástico" se pretende distinguir este algoritmo del *Steepest Descent*, que utiliza un gradiente determinista para el cálculo de los pesos. Una característica importante del LMS es su simplicidad.

Dado el conjunto $\mathbf{O} = \{\mathbf{o}_1, ..., \mathbf{o}_N\}$,siendo en nuestro caso, $(\mathbf{o}_n)_{n=1...N}$ la salida de cada neurona de la capa oculta debido a cada patrón de entrada $(\mathbf{x}_n)_{n=1...N}$, y sus N salidas deseadas $\mathbf{T} = \{t_1, ..., t_N\}$, este algoritmo pretende encontrar los pesos $\mathbf{w}$ que minimizan la función de error cuadrático medio dado por:

$$E = \frac{1}{2}\sum_{n=1}^{N}\varepsilon_n^2 \quad \textit{siendo } \varepsilon_n = y\left(\mathbf{o}^n;\mathbf{w}\right) - t^n \tag{3.41}$$

con y la salida lineal de la red,

$$y = \mathbf{w}^T \cdot \mathbf{o} = \sum_i w_i \cdot o_i \tag{3.42}$$

La función de error es una función matemática definida en el espacio de pesos multidimensional para un conjunto de patrones dados. Es una superficie que tendrá muchos mínimos (global y locales), y el algoritmo va a buscar el punto en el espacio de pesos donde se encuentra el mínimo global de esta superficie. Aunque la superficie de error es desconocida, el método LMS consigue obtener información local de dicha superficie a través del gradiente. Con esta información se decide qué dirección tomar para llegar hasta el mínimo global de dicha superficie.

Basándose en el método del gradiente decreciente, se obtiene una regla para modificar los pesos de tal manera que hallamos un nuevo punto en el espacio de pesos más próximo al punto mínimo. Es decir, las modificaciones en los pesos son proporcionales al gradiente decreciente de la función error:

$$\Delta w_j = -\alpha \frac{\partial E}{\partial w_j} \tag{3.43}$$

Por tanto, si se deriva la función error con respecto a los pesos y si se aplica la regla de la cadena para el cálculo de dicha derivada, se obtiene la siguiente ecuación de actualización de los pesos (versión bloque):

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \Delta\mathbf{w}(k) = \mathbf{w}(k) + \frac{\alpha}{2}\cdot\sum_{n=1}^{N}\left(t^n - y\left(\mathbf{o}^n;\mathbf{w}(k)\right)\right)\cdot\mathbf{o}^n \tag{3.44}$$

El valor del parámetro α tiene una gran influencia sobre el entrenamiento. Si α es demasiado grande, la convergencia es posible que no se produzca, debido a que se *darán saltos* en torno al mínimo sin alcanzarlo. Si α es demasiado pequeño, alcanzaremos la convergencia, pero la etapa de aprendizaje será más larga.

En cuanto al momento en el que se debe detener el entrenamiento, depende de los requerimientos del sistema. El entrenamiento se detiene cuando el error observado es menor que el admisible. Se suele tomar el error cuadrático medio como la magnitud que determina el instante en el que el algoritmo ha convergido.

3.4. Máquinas de vectores soporte

3.4.1. Introducción

Muchas redes neuronales se diseñan para encontrar un hiperplano que separe las muestras de las distintas clases. Normalmente se comienza con un hiperplano aleatorio que se va desplazando hasta que todos los datos del entrenamiento quedan en la parte correcta. Esto deja, inevitablemente, algunos puntos del entrenamiento muy cerca del hiperplano lo cual no es un resultado óptimo a la hora de generalizar.

Al igual que los perceptrones multicapa y que las redes de funciones de base radial, las ***máquinas de vectores soporte*** (*Support Vector Machine*, *SVM*) pueden usarse tanto para clasificación de patrones como para regresión no lineal. Aquí nos centraremos

en el uso de las máquinas de vectores soporte para el reconocimiento de patrones. Las máquinas de vectores soporte presentan un buen rendimiento al generalizar en problemas de clasificación, pese a no incorporar conocimiento específico sobre el dominio.

La formulación de las máquinas de vectores soporte se basa en el principio de minimización estructural del riesgo que ha demostrado ser superior al principio de minimización del riesgo empírico, que es el empleado por muchas de las redes neuronales convencionales. Esto es analizado en el apartado 3.5.2 del presente trabajo.

Para una mayor análisis sobre las **SVM** se recomienda leer [34] y [35], de los cuales se han obtenido los conceptos aquí presentados.

3.4.2. Minimización del error

Para una determinada tarea de aprendizaje, con una cantidad finita de datos de entrenamiento, el mejor rendimiento al generalizar se conseguirá si se equilibra la balanza entre la *exactitud* obtenida sobre ese conjunto de entrenamiento y la *capacidad de generalización* de la maquina, esto es, la habilidad de la máquina de aprender cualquier conjunto de entrenamiento sin error.

3.4.2.1. Riesgo esperado y riesgo empírico

Sea un conjunto de N patrones junto con sus salidas deseadas $\{(\mathbf{x}_1,t_1),(\mathbf{x}_2,t_2),...,(\mathbf{x}_N,t_N)\}$. Para simplificar las fórmulas siguientes, la salida esperada t_n será +1 o -1.

Ahora supongamos que existe una distribución de probabilidad desconocida $P(\mathbf{x},t)$ de la que se han tomado esas muestras. Esto es más general que asignar un t fijo a cada $\mathbf{x}$. Con esta distribución, las etiquetas t_n se asignan según una distribución de probabilidad condicional en $\mathbf{x}_i$. Más adelante, sin embargo, se asumirá un t fijo para un $\mathbf{x}$ dado.

Supongamos que tenemos una máquina para aprender la transformación $\mathbf{x}_i \mapsto t_i$. La máquina está definida por $\mathbf{x} \mapsto f(\mathbf{x},\zeta)$ donde ζ es un conjunto de parámetros ajustables (por ejemplo, en una red neuronal los pesos y los umbrales). Una elección particular de ζ genera lo que llamamos una *máquina entrenada*.

La esperanza del error de evaluación para una máquina entrenada, llamada también *riesgo real*, *riesgo esperado* o simplemente *riesgo*, es

$$R(\zeta) = \int \frac{1}{2}\left|t - f(\mathbf{x},\zeta)\right| dP(\mathbf{x},t) \qquad (3.45)$$

Sin embargo, $P(\mathbf{x},t)$ es desconocida normalmente.

El *riesgo empírico* es el error medio medido sobre el conjunto de entrenamiento:

$$R_{emp}(\zeta) = \frac{1}{2}\sum_{i=1}^{N}\left|t_i - f(\mathbf{x}_i,\zeta)\right| \qquad (3.46)$$

Notese que aquí no aparece ninguna distribución de probabilidad. A la cantidad

$$\frac{1}{2}\left|t_i - f(\mathbf{x}_i,\zeta)\right| \qquad (3.47)$$

se le denomina *perdida*. Ahora si se escoge un ρ tal que $0 \le \rho \le 1$. Entonces, la siguiente cota se cumple con probabilidad $1-\rho$:

$$R(\zeta) \le R_{emp}(\zeta) + \sqrt{\frac{h\left(\log\left(2N/h\right)+1\right) - \log\left(\rho/4\right)}{N}} \qquad (3.48)$$

donde *h* es un número entero no negativo llamado dimensión de ***Vapnik-Chervonenkis (VC)*** y es una medida de la idea de capacidad de generalización mencionada anteriormente. A la parte derecha de la desigualdad se le llama ***cota del riesgo*** y al segundo término de la cota del riesgo se le llama ***confianza de VC***.

Algunas notas sobre esta cota:

- es independiente de $P(\mathbf{x},t)$,
- normalmente no es posible calcular la parte izquierda de la desigualdad,
- si se conociera *h*, se podría calcular la parte derecha. De esta forma, dadas diferentes máquinas de aprendizaje (familias de funciones $f(\mathbf{x},\zeta)$) y eligiendo un valor de ρ lo suficientemente pequeño, si seleccionamos la máquina que minimice la parte derecha de la desigualdad, estaremos eligiendo la máquina que da la cota superior más pequeña del riesgo real.

El último punto da la idea básica para la minimización del riesgo estructural discutida más adelante.

3.4.3. La dimensión de Vapnik-Chervonenkis

La dimensión **VC** es una propiedad de un conjunto de funciones $\{f(\zeta)\}$ y puede definirse para distintas clases de funciones f. Aquí nos centraremos en funciones de la forma $f(\mathbf{x},\zeta) \in \{+1,-1\} \quad \forall \mathbf{x},\zeta$.

Si un conjunto de *N* patrones puede etiquetarse de 2^N formas distintas, y para cada etiquetado puede encontrarse un elemento del conjunto $\{f(\zeta)\}$ que asigne correctamente estas etiquetas, diremos que el conjunto de patrones es *roto* por ese conjunto de funciones. La dimensión **VC** del conjunto de funciones $\{f(\zeta)\}$ se define como el máximo número de patrones de entrenamiento que pueden ser rotos por $\{f(\zeta)\}$. Nótese que si la dimensión **VC** es *h*, entonces existe como mínimo un conjunto de patrones que puede romperse, pero en general no todos los conjuntos de *h* patrones podrán romperse.

3.4.3.1. Dimensión VC del conjunto de hiperplanos orientados

Supongamos que el espacio en que viven los datos es $\mathbb{R}^2$ y que $\{f(\zeta)\}$ está formado por líneas rectas orientadas. Aunque es posible encontrar tres puntos que pueden romperse con este conjunto de funciones (Figura 3.12), es imposible encontrar cuatro. Por lo tanto la dimensión **VC** del conjunto de líneas rectas orientadas en $\mathbb{R}^2$ es siempre 3. Nótese de paso que, aunque los puntos de la Figura 3.12 pueden romperse, no

todos los conjuntos de tres puntos $\mathbb{R}^2$ en pueden romperse. Por ejemplo, si los tres puntos son colineales, es imposible romperlos con líneas orientadas.

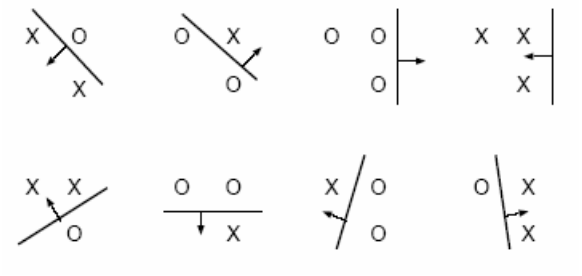


Figura 3.12: Tres puntos en $\mathbb{R}^2$ separados por líneas rectas

El resultado anterior se generaliza en el siguiente teorema y corolario:

- *Teorema*. Consideremos un conjunto de patrones de $\mathbb{R}^n$ y tomemos cualquiera de ellos como origen. Los *N* patrones pueden romperse con un hiperplano orientado si y solo si los vectores de posición de los restantes patrones son linealmente independientes.
- *Corolario*. La dimensión **VC** del conjunto de hiperplanos orientados en $\mathbb{R}^n$ es *n+1*, ya que siempre pueden encontrarse *n* patrones linealmente independientes (dado un origen cualquiera), pero nunca *n+1*.

La dimensión **VC** formaliza la idea la capacidad de un conjunto de funciones. Intuitivamente cabe esperar que máquinas con muchos parámetros tengan una dimensión **VC** alta y que máquinas con pocos parámetros su dimensión sea baja. Aunque, esto suele ser así, hay un contraejemplo: una maquina con un único parámetro y con dimensión **VC** infinita (puede romper cualquier conjunto de *N* patrones, independientemente de *N*):

$$f(x,\zeta)=\theta\left(sen(\zeta x)\right),\quad x,\zeta\in\mathbb{R} \tag{3.49}$$

donde θ es la función escalón definida como

$$\theta(x)=\begin{cases}+1 & x>0\\ -1 & x\le 0\end{cases} \tag{3.50}$$

Como curiosidad, aunque con esta función se puede romper un número arbitrario de puntos, es posible encontrar cuatro puntos que no pueden ser rotos (Figura 3.13).

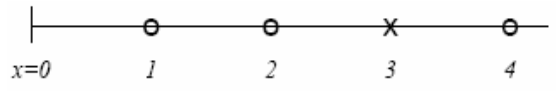


Figura 3.13 : Cuatro puntos que no pueden ser separados a pesar de tener dimensión VC infinita

3.4.3.2. Acotación del riesgo

La Figura 3.14 muestra la evolución con un nivel de confianza del 95% $(\rho=0.05)$ de la confianza **VC** al aumentar la dimensión **VC**. Como puede verse, la confianza es monótonamente creciente con *h*. Aunque la gráfica está generada para un valor de *N* = 10000, lo anterior es cierto para cualquier valor de *N*.

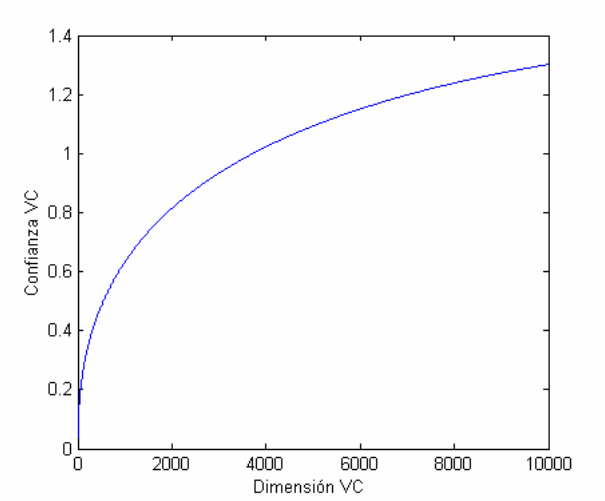


Figura 3.14: Confianza VC con respecto a dimensión VC

Por lo tanto, si tenemos un conjunto de maquinas de aprendizaje cuyo riesgo empírico es cero, nos quedaremos con aquella de menor dimensión **VC**, es decir, aquella que minimiza la cota del riesgo.

De cualquier forma, es importante tener en cuenta que la ecuación 3.48 da (con una determinada probabilidad) una cota superior del riesgo real, y que esto no impide que una determinada máquina con el mismo R_{emp} y cuyo conjunto de funciones asociado presente una dimensión VC mayor tenga mejor rendimiento. Un ejemplo de sistema de este tipo que da un buen rendimiento pese a tener dimensión **VC** infinita es el de un clasificador basado en los k vecinos más cercanos con $k = 1$. Este conjunto de funciones tiene dimensión **VC** infinita y riesgo empírico cero, ya que cualquier conjunto de patrones etiquetados arbitrariamente será aprendido correctamente por el algoritmo. En este caso la cota no da ninguna información. Con todo, los clasificadores basados en el vecino más cercano funcionan bien. Así, una capacidad infinita no implica un rendimiento pobre.

3.4.4. Minimización estructural del riesgo

La confianza **VC** de la ecuación 3.44 depende de la clase de funciones escogidas, mientras que el *riesgo empírico* y *el riesgo ideal* dependen de la función particular elegida por el algoritmo de entrenamiento.

El objetivo es encontrar un subconjunto del conjunto de funciones que minimice la *cota de riesgo*. Para ello se divide la clase completa de funciones en subconjuntos anidados. Para cada conjunto se debería poder calcular h o, al menos, establecer una cota de su valor. La ***minimización estructural del riesgo*** consiste en encontrar el subconjunto de funciones que minimiza la cota de error actual. Entonces se toma aquella máquina entrenada de la serie con menor valor para la suma del riesgo empírico y la confianza VC.

3.4.5. Máquinas de vectores soporte lineales

Comenzaremos con **máquinas lineales entrenadas con datos linealmente separables** (el caso general, máquinas no lineales entrenadas con datos no separables, se resuelve con un problema de programación cuadrática muy similar).

Consideremos el conjunto de entrenamiento

$$\{(\mathbf{x}_1,t_1),...,(\mathbf{x}_i,t_i),...,(\mathbf{x}_N,t_N)\} \quad t_i \in \{+1,-1\} \quad \mathbf{x}_i \in \mathbb{R}^d \qquad (3.51)$$

y supongamos que existe un hiperplano que separa los puntos de ambas clases. Se verifica que los puntos $\mathbf{x}$ sobre el hiperplano satisfacen la ecuación $\boldsymbol{\omega}^T\mathbf{x}+b=0$ donde el vector $\boldsymbol{\omega}$ es normal al hiperplano, $|b|/\|\boldsymbol{\omega}\|$ es la distancia perpendicular del hiperplano al origen y $\|\boldsymbol{\omega}\|$ es la norma euclídea de $\boldsymbol{\omega}$.

Sean d_+ (d_-) la distancia más corta al hiperplano de separación al patrón perteneciente a la clase +1 (perteneciente a la clase -1, respectivamente) más cercano. Definamos el *margen* del hiperplano como la suma $d_+ + d_-$. En el caso linealmente separable, el algoritmo buscara simplemente el hiperplano con mayor margen. A continuación se formula este concepto.

Supongamos que todos los datos de entrenamiento satisfacen

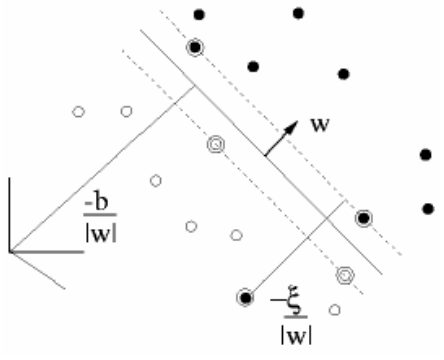
$$\boldsymbol{\omega}^T\mathbf{x}_i + b \geq +1 \quad para\ t_i = +1 \qquad (3.52a)$$

$$\boldsymbol{\omega}^T\mathbf{x}_i + b \leq -1 \quad para\ t_i = -1 \qquad (3.52b)$$

esto es

$$t_i\left(\boldsymbol{\omega}^T\mathbf{x}_i + b\right) - 1 \geq 0 \quad \forall i \qquad (3.53)$$

Ahora consideremos los patrones para los que se cumple la igualdad en la ecuación 3.49a. Estos patrones están sobre el hiperplano H_1: $\boldsymbol{\omega}^T\mathbf{x}_i + b = +1$ con normal $\boldsymbol{\omega}$ y distancia al origen $|1-b|/\|\boldsymbol{\omega}\|$. De forma similar, para el hiperplano H_2: $\boldsymbol{\omega}^T\mathbf{x}_i + b = -1$, la distancia al origen es $|-1-b|/\|\boldsymbol{\omega}\|$. Por tanto, $d_+ = d_- = 1/\|\boldsymbol{\omega}\|$ y el margen es simplemente $2/\|\boldsymbol{\omega}\|$. Nótese que H_1 y H_2 son paralelos (tienen la misma normal) y que no hay patrones de entrenamiento entre ellos. Podemos en definitiva, encontrar el par de hiperplanos que dan el máximo margen minimizando la función de coste $1/2\|\boldsymbol{\omega}\|^2$ con las restricciones de la ecuación 3.53.



Nota: Los vectores soporte están rodeados por un circulo

Figura 3.15: Separación mediante hiperplanos lineales para el caso separable.

Ahora pasaremos a una formulación lagrangiana del problema. Hay dos razones importantes para hacer esto:

- la primera es que las restricciones de la ecuación 3.53 se sustituirán por restricciones sobre los multiplicadores de Lagrange, que serán más fáciles de manejar,
- la segunda es que con esta reformulación del problema, los patrones de entrenamiento solo aparecen en forma de productos escalares entre vectores. Esta propiedad es crucial para poder generalizar el procedimiento al caso no lineal.

Por lo tanto, introduzcamos *N* multiplicadores de Lagrange que denotaremos por $\alpha_1, \alpha_2, ..., \alpha_N$, uno para cada una de las restricciones de la ecuación 3.53. La regla es que para restricciones de la forma $c_i \geq 0$ las ecuaciones que las definen se multiplican por multiplicadores de Lagrange positivos y se restan de la función objetivo para formar el lagrangiano. En el caso de restricciones de la forma $c_i = 0$, los multiplicadores de Lagrange no tienen restricciones. Lo anterior da el lagrangiano

$$L_P = \frac{1}{2}\|\boldsymbol{\omega}\|^2 - \sum_{i=1}^{N} \alpha_i t_i \left(\boldsymbol{\omega}^T \mathbf{x}_i + b\right) + \sum_{i=1}^{N} \alpha_i \qquad (3.54)$$

A continuación se debe minimizar L_P con respecto a $\boldsymbol{\omega}, b$ y a la vez exigir que las derivadas de L_P con respecto a todos los α_i se anulen, todo sujeto a las restricciones $\alpha_i \geq 0$ (restricciones C_1). Puede demostrarse que se trata de un problema de *programación cuadrática convexo*. Esto quiere decir que podemos resolver de forma equivalente el siguiente problema *dual*: maximizar L_P sujeto a la restricción de que el gradiente de L_P con respecto a $\boldsymbol{\omega}$ y *b* se anule y sujeto también a la restricción de que $\alpha_i \geq 0$ (restricciones C_2). Esta formulación particular del problema se conoce como *dual de Wolfe* y presenta la propiedad de que el máximo de L_P con las restricciones C_2 ocurre en los mismos valores de $\boldsymbol{\omega}$, *b* y α que el mínimo de L_P con las restricciones C_1.

Al requerir que se anule el gradiente de L_P con respecto a $\boldsymbol{\omega}$ y *b*, obtenemos las condiciones:

$$\boldsymbol{\omega} = \sum_i \alpha_i t_i \mathbf{x}_i \qquad (3.55)$$

$$\sum_i \alpha_i t_i = 0 \qquad (3.56)$$

Ya que estas restricciones aparecen como igualdades, podemos sustituirlas en la ecuación 3.54 para obtener

$$L_D = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j t_i t_j \mathbf{x}_i \mathbf{x}_j \qquad (3.57)$$

La solución se obtiene minimizando L_P o maximizando L_D y viene dado por la ecuación 3.54.

Nótese que hay un multiplicador de Lagrange α_i para cada patrón de entrenamiento. Tras obtener una solución, aquellos puntos para los que $\alpha_i \geq 0$ se denominan ***vectores soporte*** y yacen sobre los hiperplanos H_1 y H_2. El resto de patrones tienen $\alpha_i = 0$ y satisfacen la ecuación 3.53, quizá con la igualdad. Con estas máquinas, los vectores soporte son los elementos críticos del conjunto entrenamiento: son los más cercanos a la frontera de decisión y si el resto de patrones no se consideraran en un nuevo entrenamiento, el algoritmo encontraría el mismo hiperplano de separación.

Nótese de paso como $\boldsymbol{\omega}$ está determinado explícitamente por el algoritmo de entrenamiento, pero no es este el caso del umbral *b*, aunque su obtención es inmediata: basta tomar en la ecuación 3.53 cualquier *i* para el que $\alpha_i \neq 0$ y despejar *b*; por ejemplo, si $t_i = 1$ (vector positivo):

$$b = 1 - \boldsymbol{\omega}^T \mathbf{x}_i \qquad (3.58)$$

En cualquier caso, siempre es más seguro tomar el valor medio de *b* que resulte de todas las ecuaciones.

3.4.5.1. Fase de evaluación

Una vez que hayamos entrenado una **SVM**, para clasificar un patrón de entrenamiento $\mathbf{x}$ basta determinar en qué parte de la frontera de decisión se encuentra y asignarle la etiqueta de la clase correspondiente, es decir, asignamos a $\mathbf{x}$ la clase $\text{sgn}\left(\boldsymbol{\omega}^T \mathbf{x} + b\right)$, donde *sgn* es la función signo.

3.4.6. El caso no separable linealmente

En el caso no linealmente separable nos interesa poder relajar las restricciones de las ecuaciones 3.55, pero únicamente cuando sea necesario, es decir, nos interesa añadir un nuevo coste a la función de objetivo. Una posible forma de hacerlo es introduciendo variables débiles ξ_i, $i = 1,...,N$, en las restricciones para convertirlas en:

$$\boldsymbol{\omega}^T \mathbf{x}_i + b \geq +1 - \xi_i \quad para\ t_i = +1 \qquad (3.59a)$$

$$\boldsymbol{\omega}^T \mathbf{x}_i + b \leq -1 + \xi_i \quad para\ t_i = -1 \qquad (3.59b)$$

$$\xi_i \geq 0 \quad \forall i \qquad (3.59c)$$

Con esto, para que una muestra quede mal clasificada, el correspondiente ξ_i debe ser mayor que la unidad. La suma $\sum_i \xi_i$ es, por tanto, una cota superior en el número de errores sobre el entrenamiento. Una forma de añadir el coste de los errores a la función objetivo es intentar minimizar $1/2\|\boldsymbol{\omega}\|^2 + \gamma\left(\sum_i \xi_i\right)$ donde γ es un parámetro ajustable, un valor grande de γ equivale a una mayor penalización de los errores.

El problema anterior es de nuevo un problema de programación cuadrática con la propiedad de que ni los ξ_i ni sus multiplicadores de Lagrange asociados, aparecen en el problema *dual de Wolfe*, que es ahora

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j t_i t_j \mathbf{x}_i \mathbf{x}_j \qquad (3.60)$$

sujeto a las restricciones

$$0 \le \alpha_i \le \gamma \tag{3.61}$$

$$\sum_i \alpha_i t_i = 0 \tag{3.62}$$

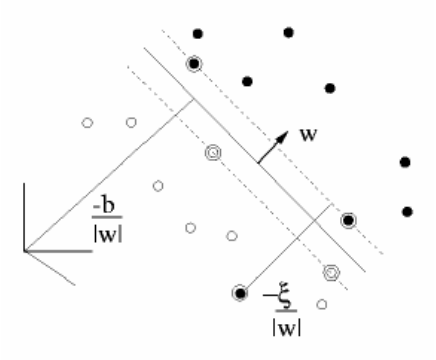
La solución viene dada nuevamente por:

$$\boldsymbol{\omega} = \sum_{i=1}^{N^S} \alpha_i t_i \mathbf{s}_i \tag{3.63}$$

donde ahora N^S es el número de vectores soporte y $\mathbf{s}_i$ es el i-ésimo vector soporte. Por lo tanto, la única diferencia con el caso del hiperplano óptimo es que los valores de α_i están ahora acotados superiormente por γ. La Figura 3.16 ilustra gráficamente algunas de las ideas anteriores.

Aunque no se demostrará, el calculo del umbral *b* sigue de nuevo la expresión $b = 1 - \boldsymbol{\omega}^T \mathbf{x}_i$ con $0 < \alpha_i < \gamma$. Al igual que en el caso linealmente separable, es más adecuado tomar la media sobre todos los vectores soporte.

En cuanto al parámetro γ, conforme $\gamma \to 0$, la solución converge a la obtenida con el hiperplano óptimo (sobre el problema no separable). Por otro lado conforme $\gamma \to \infty$, la solución converge hacia una en la que domina el término de maximización del margen y se hace énfasis en minimizar el error de clasificación.



Nota: Los vectores soporte están rodeados por un circulo

Figura 3.16: Separación mediante hiperplanos lineales para el caso no separable.

3.4.7. Máquinas no lineales de vectores soporte

Para extender el concepto a clasificadores no lineales, se realiza una transformación del espacio de entrada a otro espacio de alta dimensionalidad en el que los datos son separables linealmente

En primer lugar, observemos que la única forma en que aparecen los datos en las ecuaciones 3.60 – 3.63 es como productos escalares $\mathbf{x}_i \cdot \mathbf{x}_j$. Supongamos entonces que llevamos los datos a un nuevo espacio euclídeo H (posiblemente de *dimensión infinita*) mediante una transformación $\aleph$ (ver la Figura 3.17)

$$\aleph : \mathbb{R}^m \mapsto \mathrm{H} \tag{3.64}$$

Con esto, el algoritmo de entrenamiento solo dependerá de los datos a través de productos escalares en H, es decir, de funciones de la forma $\aleph(\mathbf{x}_i)\cdot\aleph(\mathbf{x}_j)$. Si existiera una función núcleo (***kernel***) K tal que $K(\mathbf{x}_i,\mathbf{x}_j)=\aleph(\mathbf{x}_i)\cdot\aleph(\mathbf{x}_j)$, podríamos usar únicamente en el algoritmo de entrenamiento sin tener que conocer explícitamente $\aleph$.

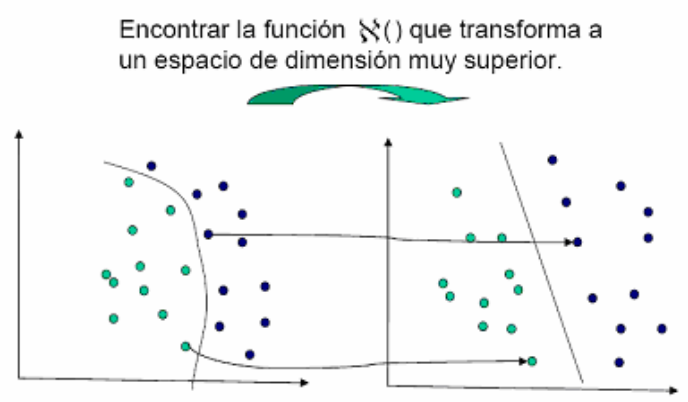


Figura 3.17: Transformación del espacio de entrada a un espacio de dimensión muy superior

Un ejemplo de función núcleo es

$$K(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{3.65}$$

en este ejemplo H es de dimensión infinita y no sería sencillo trabajar explícitamente con $\aleph$. Sin embargo, si sustituimos $\mathbf{x}_i\cdot\mathbf{x}_j$ por $K(\mathbf{x}_i,\mathbf{x}_j)$ a lo largo de todo el algoritmo de entrenamiento, obtendremos una **SVM** que vive en un espacio de dimensión infinita y que opera en prácticamente la misma cantidad de tiempo. La separación sigue siendo lineal, pero en un espacio diferente.

En la fase de evaluación, la **SVM** se usa calculando el producto escalar de un patrón de entrenamiento dado $\mathbf{x}$ con $\boldsymbol{\omega}$ y calculando el signo de

$$f(\mathbf{x})=\sum_{i=1}^{N^S}\alpha_i t_i\aleph(\mathbf{s}_i)\cdot\aleph(\mathbf{x})+b=\sum_{i=1}^{N^S}\alpha_i t_i K(\mathbf{s}_i,\mathbf{x})+b \tag{3.66}$$

donde $\mathbf{s}_i$ son los vectores soporte. De nuevo, por tanto, podemos evitar el calculo directo de $\aleph(\mathbf{x})$.

En la bibliografía sobre máquinas de vectores soporte se denomina normalmente a H como ***espacio de Hilbert***.

3.4.7.1. Condición de Mercer

La ***condición de Mercer*** permite determinar para qué núcleos existe un con las propiedades descritas anteriormente y para cuáles no:

- ***Teorema de Mercer***. Existe una transformación $\aleph$ y una expansión en series:

$$K(\mathbf{x},\mathbf{y})=\sum_i\aleph(\mathbf{x})_i\aleph(\mathbf{y})_i \tag{3.67}$$

si y solo si para cualquier $g(\mathbf{x})$ para la que la integral

$$\int g(\mathbf{x})^2 d\mathbf{x} \qquad (3.68)$$

sea finita, se tiene

$$\int K(\mathbf{x},\mathbf{y})\, g(\mathbf{x})\, g(\mathbf{y})\, d\mathbf{x}\, d\mathbf{y} \geq 0 \qquad (3.69)$$

Al usar un núcleo que no cumpla la *condición de Mercer*, no estamos asegurando que el problema de programación cuadrática que tenemos que resolver tenga solución. Aún así, puede darse de trabajar correctamente con un conjunto de datos para los cuales la hessiana no este indefinida, y que por tanto, la máquina funcione bien. Aunque la solución sea satisfactoria, en este caso en particular la interpretación geométrica que estamos dando al núcleo K no tiene sentido.

3.4.7.2. Funciones kernel habituales

Los **kernels** o núcleos más utilizados para el reconocimiento de patrones son los siguientes:

- ***Polinomial***
- ***Función de base radial gaussiana***

Para poder comprender mejor el funcionamiento de cada uno de estos núcleos, se va a representar los resultados que se consiguen al emplearlos para la clasificación de los datos del problema **IRIS**. Este conjunto de datos contiene cuatro atributos de distintos lirios, pero nos hemos restringido a las dos características que contienen la mayoría de información acerca de la clase, que son el largo del pétalo y la anchura del pétalo.

3.4.7.2.a. Polinomial

El uso de un núcleo polinomial para resolver distintos problemas de clasificación es muy popular. La ecuación 3.70 proporciona un clasificador que es un polinomio de grado p sobre los datos.

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y}+1)^p \qquad (3.70)$$

En la Figura 3.18 se representan las fronteras de decisión que se han obtenido para la resolución del problema **IRIS** empleando núcleos polinomiales de grado 2 y asumiendo un valor de $\gamma = \infty$, siendo el número de vectores soporte 10 (8.3% de los 120 datos del problema).

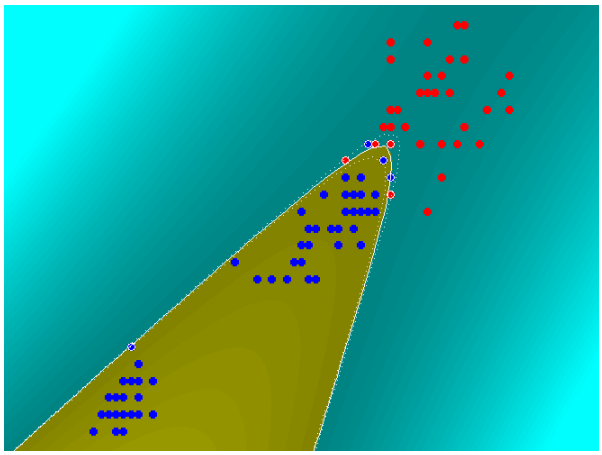


Figura 3.18: Fronteras de decisión obtenidas mediante núcleos polinomiales de grado 2 $(\gamma = \infty)$

Para ver el efecto del valor de γ, en la Figura 3.19 se vuelve a representar los resultados para el caso de $\gamma = 10$, con lo que se esta aumentando la probabilidad de que se cometa error, pero se comprueba que se consigue una mejor generalización que en el caso anterior, donde se producía un sobreajuste.

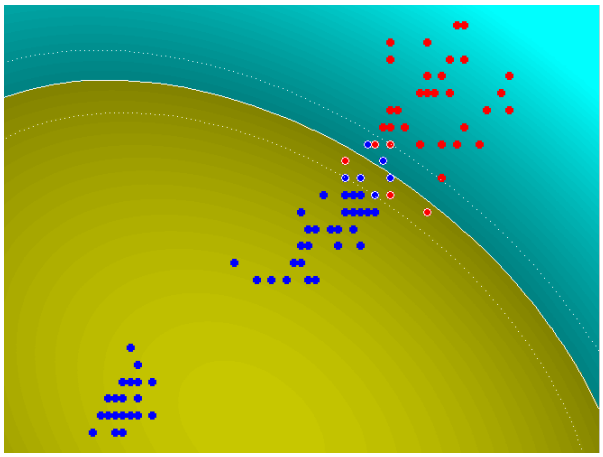


Figura 3.19: Fronteras de decisión obtenidas mediante núcleos polinomiales de grado 2 $(\gamma = 10)$

3.4.7.2.b. Funciones de base radial gaussiana

Las funciones de base radial han sido estudiadas en profundidad, más concretamente las de tipo gaussiano,

$$K(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right) \qquad (3.71)$$

Las técnicas clásicas que emplean funciones de base radial utilizan métodos para determinar los centroides. Una atractiva característica de las **SVM** es que esta selección de centros está implícita en la propia máquina, donde cada uno de los vectores soportes esta asociado a cada una de las funciones núcleo. En resumen, el número de centros (N^S), los centros en si (los $\mathbf{s}_i$), los pesos (α_i) y el umbral (b) son determinados automáticamente en el entrenamiento de la **SVM** y dan resultados excelentes en comparación con las RBF gaussianas clásicas.

En este caso, los resultados para el problema **IRIS** se representan en la Figura 3.20 empleando $\sigma = 10 \quad \gamma \to \infty$. El número de vectores soporte es de 12 (10%).

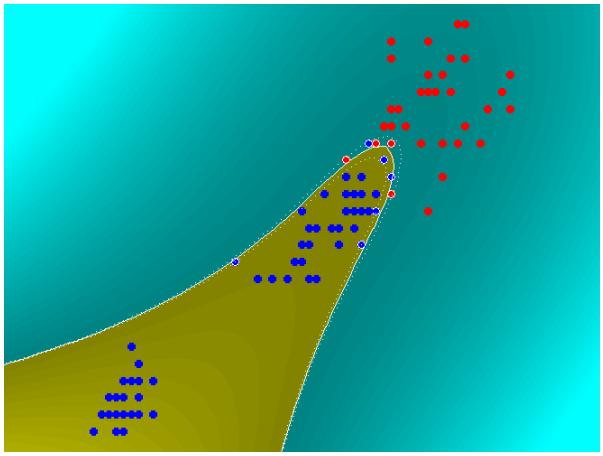


Figura 3.20: Fronteras de decisión obtenidas mediante núcleos RBF gaussianos $(\sigma = 10; \gamma \to \infty)$

Capítulo 4
Resultados en la detección

4.1. Introducción

En este capitulo, se describen las distintas topologías empleadas en la etapa de clasificación. Dentro de cada tipo de clasificador (**MLP**, **RBF** o **SVM**) se han implementado distintas arquitecturas y entrenamientos, con el objetivo de encontrar el mejor clasificador, siendo este aquel que presente mejores resultados en términos de la *fracción de verdaderos positivos* y la *fracción de falsos positivos*, es decir, mejores resultados de acuerdo al análisis mediante curvas **ROC**.

Mientras, en el apartado 4.3 se exponen los resultados obtenidos para cada clasificador.

4.2. Topologías de las distintos clasificadores

4.2.1. Clasificador mediante MLP

Para implementar los clasificadores basados en **MLP**, siempre se ha elegido la arquitectura con dos capas ocultas, pero se ha variado el número de neuronas en dichas capas y el algoritmo de entrenamiento de los pesos.

El diseño, entrenamiento y validación de los **MLP** se ha realizado con el *ToolBox NeuralNetwork* de **MATLAB**.

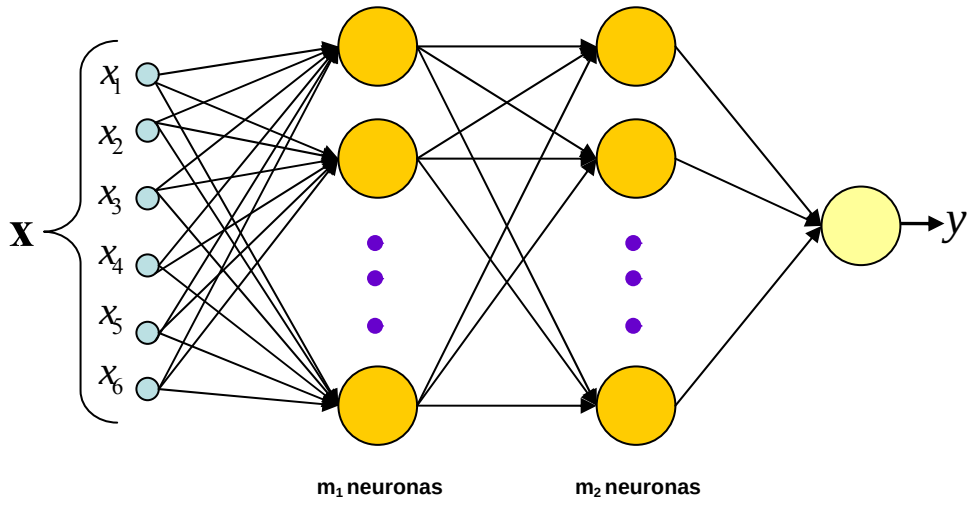


Figura 4.1: Topología genérica del clasificador basado en MLP

En la Figura 4.1 se representa la topología genérica de clasificador basado en **MLP** con m_1 neuronas en la primera capa oculta y m_2 neuronas en la segunda capa oculta, teniendo a la entrada patrones con 6 dimensiones y una única salida, que puede ser +1 (normal o tejido sano) o -1 (anormal o microcalcificación). Las funciones de activación de las neuronas de las capas ocultas siguen la expresión:

$$f(x) = \tanh(x) \qquad (4.1)$$

mientras que la función de activación de la neurona de la capa de salida es la función identidad:

$$f(x) = x \qquad (4.2)$$

Como ultimo comentario, en todas las arquitecturas implementadas, la función de error empleada es la suma de errores cuadráticos (*MSE*, *mean squared error*), y empleándose un número de épocas durante el entrenamiento igual a 2000.

En la Tabla 4.1 se describe las distintas configuraciones que se han implementado:

Nombre del clasificador	m_1	m_2
MLP-1	6	4
MLP-2	6	3
MLP-3	6	2
MLP-4	5	4
MLP-5	5	3
MLP-6	5	2
MLP-7	4	4
MLP-8	4	3
MLP-9	4	2

Tabla 4.1: Relación de clasificadores MLP implementados

Mientras en la Tabla 4.2, se muestra los distintos algoritmos de entrenamiento empleados en cada uno de los **MLPs** anteriores:

Nombre del entrenamiento	Entrenamiento de pesos
traingda	BP con tasa de aprendizaje variable
traingdm	BP con termino de momento
traingdx	BP con tasa de aprendizaje variable y termino de momento
traincgf	BP con gradiente conjugado (Fleetcher-Reeves)
traincgp	BP con gradiente conjugado (Polar-Ribiere)
trainbfg	BP con método Quasi Newton (BFGS)

Tabla 4.2: Relación de tipos de entrenamiento

4.2.2. Clasificador mediante red RBF

En este proyecto se ha empleado una red **RBF** que presenta (ver Figura 4.2):

- *Capa de entrada*: Un determinado patrón de entrada **x** presenta las 6 características.
- *Capa oculta*: Formada por *m* neuronas. Se ha evaluado las prestaciones de la red para distinto número de neuronas en la capa oculta, presentando de manera comparativa los resultados obtenidos.
- *Capa de salida*: 1 salida lineal, *y*, que puede ser +1 (normal o tejido sano) o -1 (anormal o microcalcificación).

Las funciones de cada una de las neuronas de la capa oculta son gaussianas centradas en sus respectivos centroides $\mathbf{c}_j$ y varianza σ_j^2:

$$\phi_j(\mathbf{x}) = e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|}{2\sigma_j}} \qquad (4.3)$$

Además para el calculo de los centroides se ha usado el algoritmo **FSCL** (se han usado 200 iteraciones) mientras que para el calculo de la varianza se ha empleado dos métodos, el primero, donde tomamos varianza fija para todos los centros, y el segundo, donde se estima la varianza para cada centroide teniendo en cuenta los patrones y su clase, ambos explicados ambos en el apartado 3.3.4.3. Finalmente, el entrenamiento de los pesos de la capa de salida se hace mediante **LMS.**

En la tabla siguiente se describe las distintas configuraciones de red **RBF** que se han empleado, indicando el número de neuronas de la capa oculta para cada una:

Nombre del clasificador	m
RBF-1	4
RBF-2	8
RBF-3	16
RBF-4	32
RBF-5	64

Tabla 4. 3: Relación de clasificadores RBF implementados

Mientras en la Tabla 4.4, se muestra las dos posibles opciones para la estimación de la varianza. Las dos opciones se han implementado con todas las redes **RBF** anteriormente citadas.

Nombre de estimación de σ_j^2	Estimación de σ_j^2
varf	Varianza fija para todos los centroides
vard	Varianza distinta para cada centroide

Tabla 4.4: Relación de métodos empleados para la estimación de la varianza σ_j^2

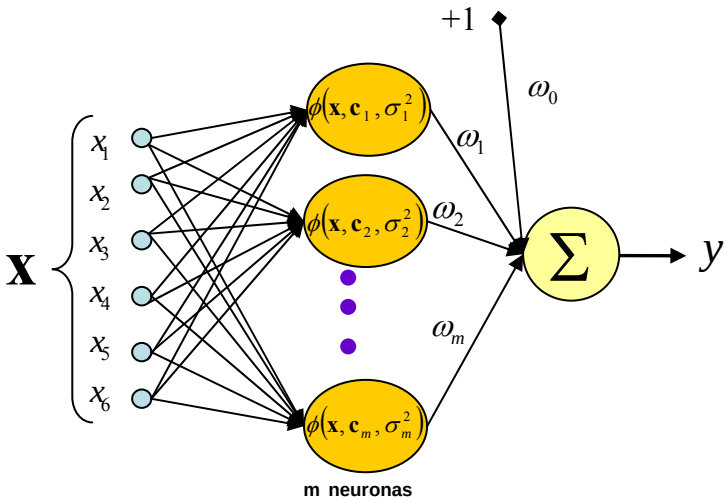


Figura 4.2: Topología genérica del clasificador basado en RBF

4.2.3. Clasificador mediante SVM

Las arquitecturas implementadas basadas en **SVM** presentan una capa oculta de *m* neuronas que implementan un *kernel* determinado. En función de dicho *kernel* tendremos un clasificador u otro. Otro parámetro a tener en cuenta es el parámetro de generalización γ, este determina si asumimos o no el caso separable, $\gamma \to \infty$, y en caso de estar en el problema no separable indica la capacidad de generalización, como ya se analizó en el capitulo tres. Cabe destacar que el número de neuronas *m* depende del numero de vectores soporte, y este a su vez depende del tipo de red empleada.

En la Figura 4.3 se representa la topología genérica de una red basada en **SVM**.

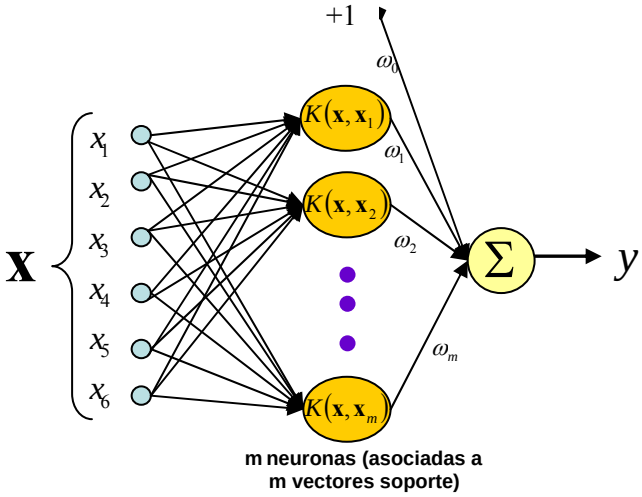


Figura 4.3 : Topología genérica del clasificador basado en SVM

En la tabla siguiente se describe las distintas configuraciones del clasificador basado en **SVM** que se han empleado, indicando el tipo de núcleo o *kernel* utilizado, así como también el valor del parámetro de generalización γ.

Nombre del clasificador	Tipo de núcleo	γ
SVM-1	Polinómico (grado 2)	∞
SVM-2	Polinómico (grado 2)	20
SVM-3	Polinómico (grado 3)	∞
SVM-4	Polinómico (grado 3)	20
SVM-5	RBF (varianza 0.25)	∞
SVM-6	RBF (varianza 0.25)	20
SVM-7	RBF (varianza 0.5)	∞
SVM-8	RBF (varianza 0.5)	20

Tabla 4.5: Relación de clasificadores MLP implementados

La implementación de los clasificadores basados en **SVM** se ha realizado mediante el ToolBox **SVM** de la Universidad de Southampton, disponible en:

http://www.isis.ecs.soton.ac.uk/ resources/svminfo/

Por ultimo, destacar que los resultados obtenidos en el caso SVM se han tenido que obtener de un conjunto de entrenamiento y test de tamaño inferior, pues requería para poder resolverlo (en el caso de conjuntos mayores) casi 3 GB de memoria RAM y durante la elaboración de esta investigación no se disponía de tal cantidad de memoria, pues este proyecto esta realizado sobre un Pentium IV a 2.6 GHz con 1024 MB de RAM. Luego, los resultados que se obtienen en este caso no pueden ser comparados cuantitativamente con los clasificadores RBF y MLP, pues estos dos son diseñados con conjuntos de entrenamiento y test distintos, pero si se puede realizar una comparativa cualitativa.

4.3. Resultados obtenidos en la detección de microcalcificaciones

En este apartado se representan los resultados obtenidos en términos de curva ROC para los distintos clasificadores implementados, junto con las áreas bajo dichas curvas y los valores de (*FFP, FVP*) para un valor fijo de *especificidad*. Dado que cada arquitectura proporciona curvas ROC muy distintas, se deben tomar valores de especificidad distintos, y así poder comparar cuantitativamente cada clasificador dentro de un mismo tipo de arquitectura neuronal.

4.3.1. Resultados del clasificador MLP

A continuación se mostrarán las curvas ROC obtenidas para el **MLP-1**,…, **MLP-9** con los distintos algoritmos de entrenamiento. En este caso, para poder comparar cuantitativamente las prestaciones de cada clasificador, para un valor fijo de *FFP*, en términos de la *FVP*, se ha tomado un valor de **especificidad** igual a **95.0 %.**

En la Figura 4.4 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-1,** para todos los entrenamientos implementados. Cabe notar, que las curvas resultantes para *traincgp* y *trainbfg* están muy juntas, y por ello no se distinguen una de otra.

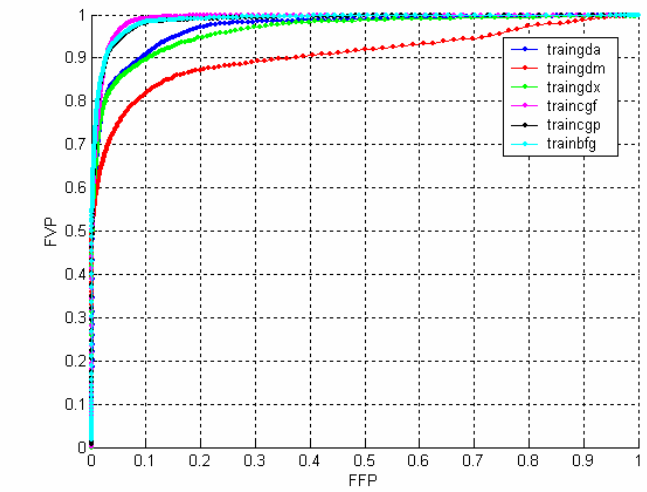


Figura 4.4: Curvas ROC generadas por el clasificador MLP-1 para los distintos entrenamientos

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos en el caso de **MLP-1** se presentan en la Tabla 4.6.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.855490233	0.9685
traingdm	0.05	0.748779449	0.9065
traingdx	0.05	0.849700668	0.9624
traincgf	0.05	0.959877979	0.9896
traincgp	0.05	0.940533467	0.9864
trainbfg	0.05	0.945134047	0.9877

Tabla 4.6: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-1

En la Figura 4.5 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-2,** para todos los entrenamientos implementados. Cabe notar, que las curvas resultantes para *traingda*, *traincgf* y *trainbfg* están muy juntas, y por ello no se distinguen una de otra.

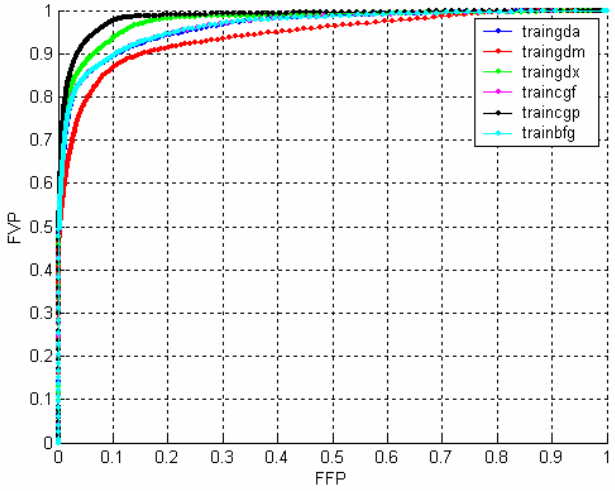


Figura 4.5: Curvas ROC generadas por el clasificador MLP-2 para los distintos entrenamientos

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos en el caso de **MLP-2** se presentan en la Tabla 4.7.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.845588355	0.9619
traingdm	0.05	0.786886082	0.9407
traingdx	0.05	0.879382745	0.9744
traincgf	0.05	0.847441368	0.9619
traincgp	0.05	0.928243726	0.9836
trainbfg	0.05	0.847410803	0.9620

Tabla 4.7: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-2

En la Figura 4. 6 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-3,** para todos los entrenamientos implementados. Como se puede observar, las curvas resultantes para los algoritmos *trainbfg, traingda y traincgf* están solapadas.

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos, en el caso de **MLP-3**, se presentan en la.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.845282317	0.9620
traingdm	0.05	0.800059857	0.9474
traingdx	0.05	0.868542963	0.9713
traincgf	0.05	0.847019335	0.9620
traincgp	0.05	0.910293142	0.9819
trainbfg	0.05	0.847410803	0.9620

Tabla 4.8: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-3

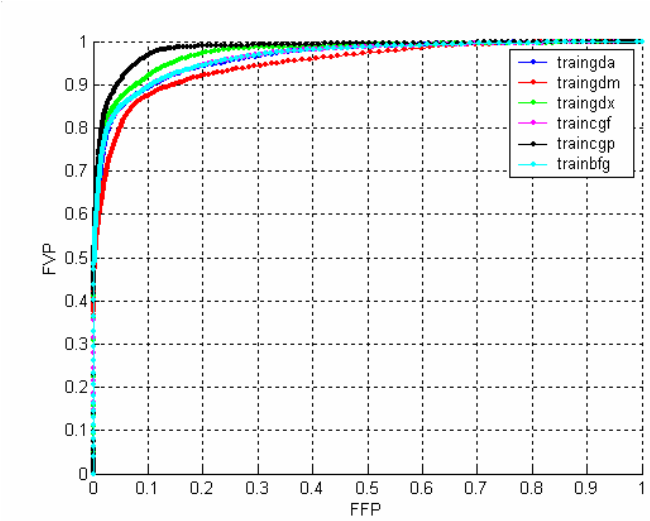


Figura 4. 6: Curvas ROC generadas por el clasificador MLP-3 para los distintos entrenamientos

En la Figura 4.7 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-4,** para todos los entrenamientos implementados. Como se puede observar, las curvas resultantes para la mayoría de los algoritmos están muy solapadas, excepto para los casos *traincgf* y *traingdm*.

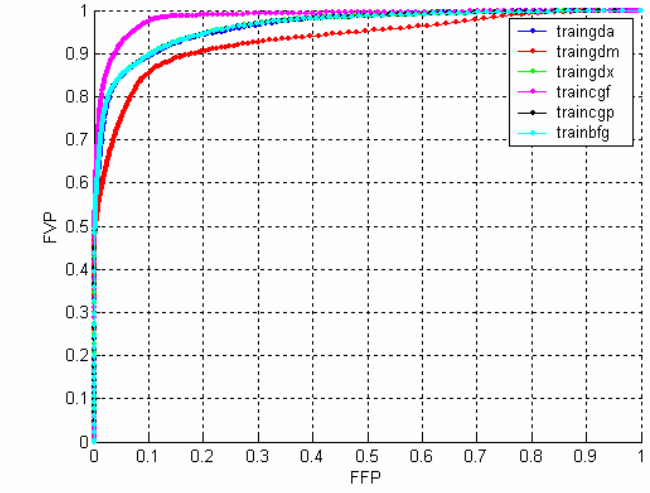


Figura 4.7: Curvas ROC generadas por el clasificador MLP-4 para los distintos entrenamientos

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos, en el caso de **MLP-4**, se presentan en la Tabla 4.9.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.845282317	0.9620
traingdm	0.05	0.751741370	0.9315
traingdx	0.05	0.848231198	0.9624
traincgf	0.05	0.922671191	0.9832
traincgp	0.05	0.847730218	0.9620
trainbfg	0.05	0.847727614	0.9620

Tabla 4.9: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-4

En la Figura 4.8 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-5,** para todos los entrenamientos implementados. Como se puede observar, las curvas resultantes para la mayoría de los algoritmos están muy solapadas, excepto para los casos *traingda* y *traingdm*.

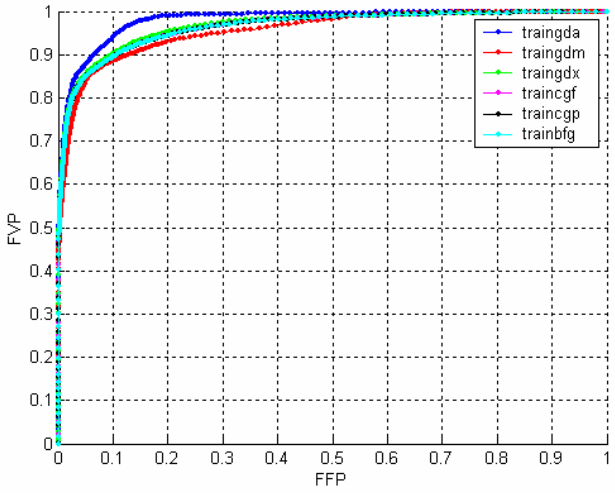


Figura 4.8: Curvas ROC generadas por el clasificador MLP-5 para los distintos entrenamientos

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos, en el caso de **MLP-5**, se presentan en la Tabla 4.10.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.876789552	0.9783
traingdm	0.05	0.836157795	0.9548
traingdx	0.05	0.853060727	0.9649
traincgf	0.05	0.847361164	0.9620
traincgp	0.05	0.847454338	0.9620
trainbfg	0.05	0.846916858	0.9620

Tabla 4.10: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-5

En la Figura 4.9: se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-6,** para todos los entrenamientos implementados. Como se puede observar, las curvas resultantes para los casos *traincgf, traincgp* y *trainbfg* son prácticamente identicas.

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos, en el caso de **MLP-6**, se presentan en la Tabla 4.11.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.874446712	0.9772
traingdm	0.05	0.763846640	0.9318
traingdx	0.05	0.864787269	0.9710
traincgf	0.05	0.847102249	0.9621
traincgp	0.05	0.847091534	0.9620
trainbfg	0.05	0.846803027	0.9620

Tabla 4.11: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-6

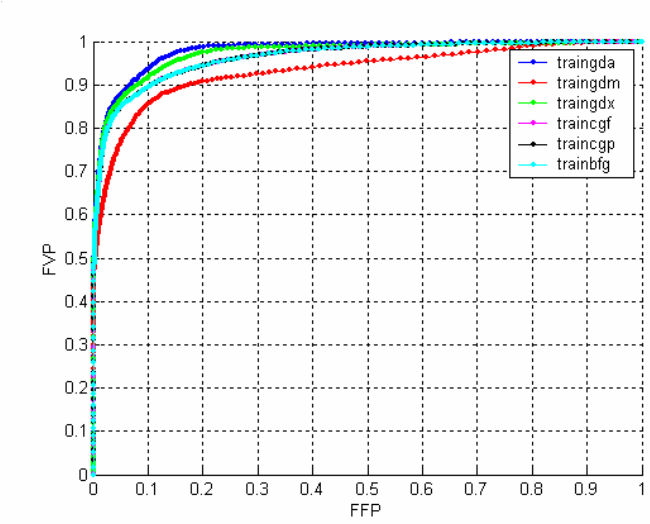


Figura 4.9: Curvas ROC generadas por el clasificador MLP-6 para los distintos entrenamientos

En la Figura 4.10 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-7,** para todos los entrenamientos implementados. Como se puede observar, las curvas resultantes para la mayoría de los algoritmos están muy solapadas, excepto para el caso de *traingdm* y *trainbfg*.

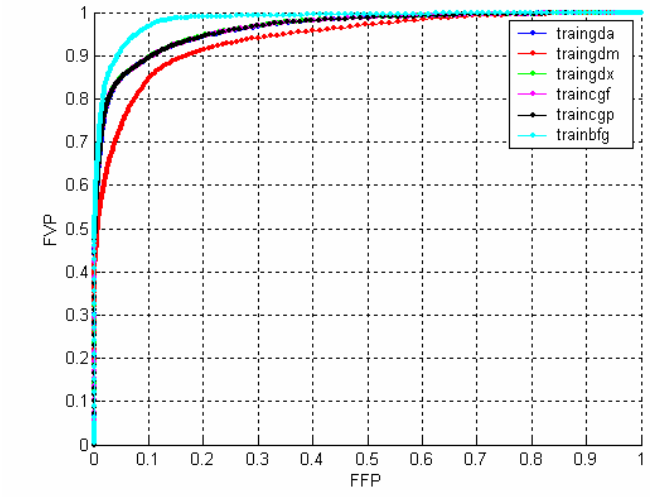


Figura 4.10: Curvas ROC generadas por el clasificador MLP-7 para los distintos entrenamientos

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos, en el caso de **MLP-7**, se presentan en la Tabla 4.12.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.846130331	0.9620
traingdm	0.05	0.735346033	0.9389
traingdx	0.05	0.848240249	0.9627
traincgf	0.05	0.846863740	0.9620
traincgp	0.05	0.846939986	0.9620
trainbfg	0.05	0.910303547	0.9822

Tabla 4.12: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-7

En la Figura 4.11 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-8,** para todos los entrenamientos implementados. Como se puede observar, las curvas resultantes para la mayoría de los algoritmos son prácticamente iguales, excepto para el caso de *traingdm* que presenta peores resultados.

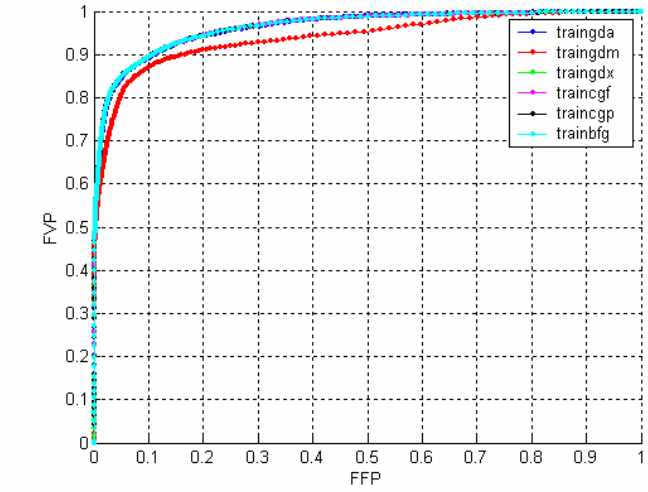


Figura 4.11: Curvas ROC generadas por el clasificador MLP-8 para los distintos entrenamientos

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos, en el caso de **MLP-8**, se presentan en la Tabla 4.13

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.842823944	0.9610
traingdm	0.05	0.804155491	0.9381
traingdx	0.05	0.847054919	0.9617
traincgf	0.05	0.847235177	0.9620
traincgp	0.05	0.847003152	0.9620
trainbfg	0.05	0.847120887	0.9620

Tabla 4.13: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-8

En la Figura 4. 12 se ha representado los resultados obtenidos, cuando se emplea como clasificador **MLP-9,** para todos los entrenamientos implementados. Como se puede observar, las curvas resultantes se pueden considerar casi idénticas en todos los casos

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los entrenamientos, en el caso de **MLP-9**, se presentan en la Tabla 4.14.

Entrenamiento	FFP	FVP	A_z
traingda	0.05	0.844556501	0.9620
traingdm	0.05	0.807735976	0.9560
traingdx	0.05	0.846068294	0.9619
traincgf	0.05	0.846358273	0.9619
traincgp	0.05	0.846022188	0.9619
trainbfg	0.05	0.846430798	0.9619

Tabla 4.14: Valores de FFP vs. FVP y áreas bajo la curva ROC para el clasificador MLP-9

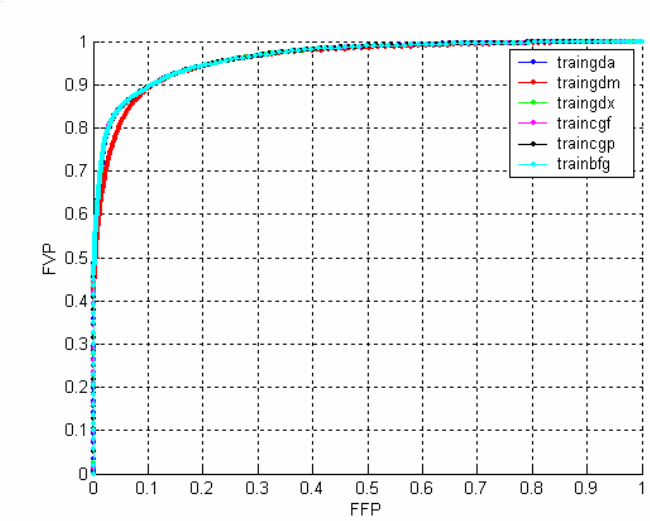


Figura 4. 12: Curvas ROC generadas por el clasificador MLP-9 para los distintos entrenamientos

4.3.2. Resultados del clasificador RBF

En este caso, para poder comparar cuantitativamente las prestaciones de cada clasificador, para un valor fijo de *FFP*, en términos de la *FVP*, se ha tomado un valor de **especificidad** igual a **79.50 %,** que como se puede apreciar, es sensiblemente inferior al tomado en el apartado anterior, esto es debido a que en general las curvas ROC obtenidas mediante MLP son mejores que las que se obtienen en este apartado

En la Figura 4.13 se ha representado los resultados obtenidos cuando se emplean como clasificador la red **RBF** en el caso de usar una varianza fija y constante para todos los centros. Como era previsible, los valores de varianza resultaron menores conforme aumenta el número de centroides.

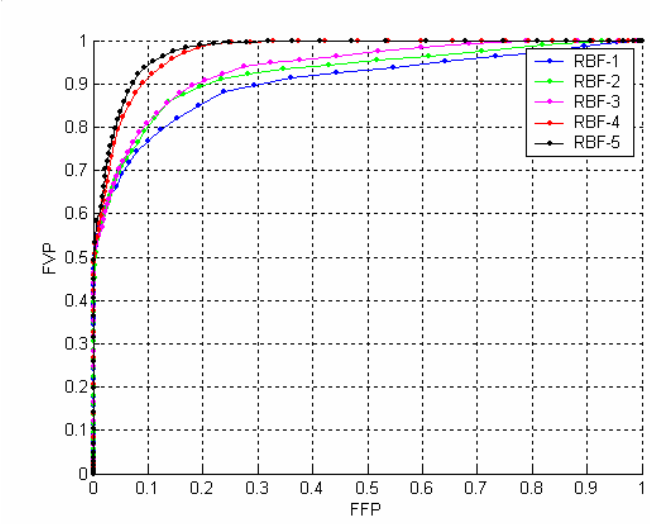


Figura 4.13: Curvas ROC generadas por las redes RBF con varianza fija

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los clasificadores **RBF,** en el caso de varianza fija para todos los centroides, se presentan en la Tabla 4.15.

Clasificador	FFP	FVP	A_z
RBF-1	0.205	0.858500566	0.9030
RBF-2	0.205	0.897319332	0.9230
RBF-3	0.205	0.908246944	0.9363
RBF-4	0.205	0.986602749	0.9712
RBF-5	0.205	0.990306388	0.9762

Tabla 4.15: Valores de FFP vs. FVP y áreas bajo la curva ROC para clasificadores RBF con varianza fija

En la Figura 4.14 se muestran las curvas ROC obtenidas para el clasificador **RBF** en caso de emplear una varianza variable y distinta para cada centro.

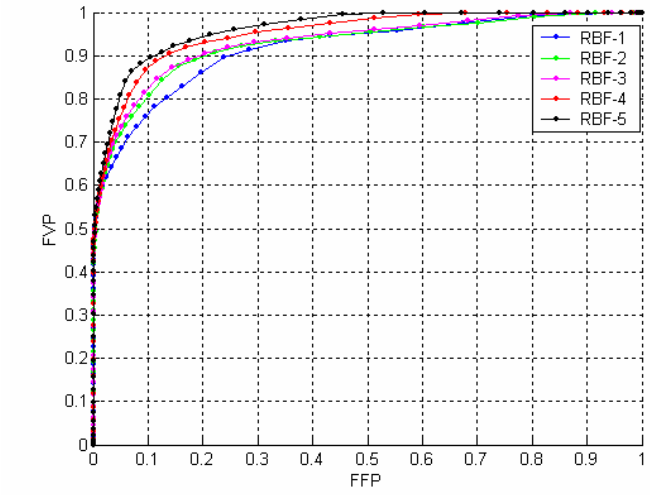


Figura 4.14: Curvas ROC generadas por las redes RBF con varianza distinta

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los clasificadores **RBF,** en el caso de varianza distinta para cada uno de los centroides, se presentan en la Tabla 4.16.

Clasificador	FFP	FVP	A_z
RBF-1	0.205	0.867788690	0.9167
RBF-2	0.205	0.900073765	0.9262
RBF-3	0.205	0.905335572	0.9322
RBF-4	0.205	0.931555054	0.9514
RBF-5	0.205	0.944772541	0.9616

Tabla 4.16: Valores de FFP vs. FVP y áreas bajo la curva ROC para los clasificadores RBF con varianza distinta

4.3.3. Resultados del clasificador SVM

A continuación se mostrarán las curvas ROC obtenidas para el **SVM-1**,…, **SVM-8**. En este caso, para poder comparar cuantitativamente las prestaciones de cada clasificador, para un valor fijo de *FFP*, en términos de la *FVP*, se ha tomado un valor de **especificidad** igual a **95.0 %.** Como ya se ha comentado anteriormente, los resultados obtenidos mediante los clasificadores SVM proceden de emplear **conjuntos de entrenamiento y test de tamaño inferior** al resto de redes, en concreto, el tamaño del **conjunto de entrenamiento** para los clasificadores SVM es de *1000 patrones*, mientras que el **conjunto de test** tiene un tamaño de *5000 patrones*.

En la Figura 4.15 se muestran las curva ROC obtenidas en el caso de emplear núcleos polinómicos de grado 2, para el caso de usar $\gamma = \infty$ (SVM-1)o $\gamma = 20$ (SVM-2).

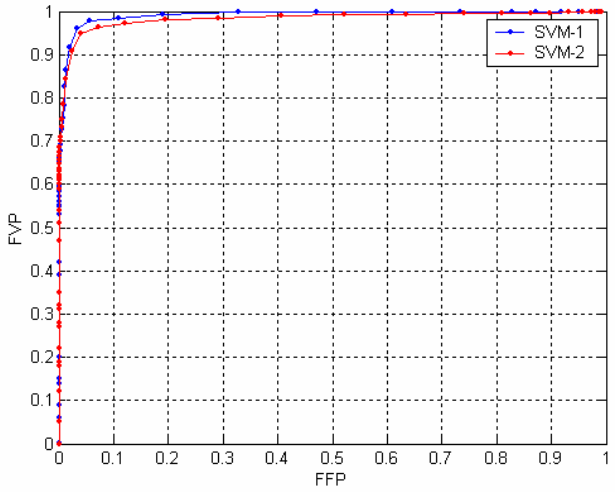


Figura 4.15: Curvas ROC generadas por las redes SVM con núcleo polinomial de grado 2

En la Figura 4.16 se muestran las curva ROC obtenidas en el caso de emplear núcleos polinómicos de grado 3, para el caso de usar $\gamma = \infty$ (SVM-3) o $\gamma = 20$ (SVM-4).

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los clasificadores **SVM,** en el caso de usar núcleos de tipo polinómico, se presentan en la Tabla 4.17.

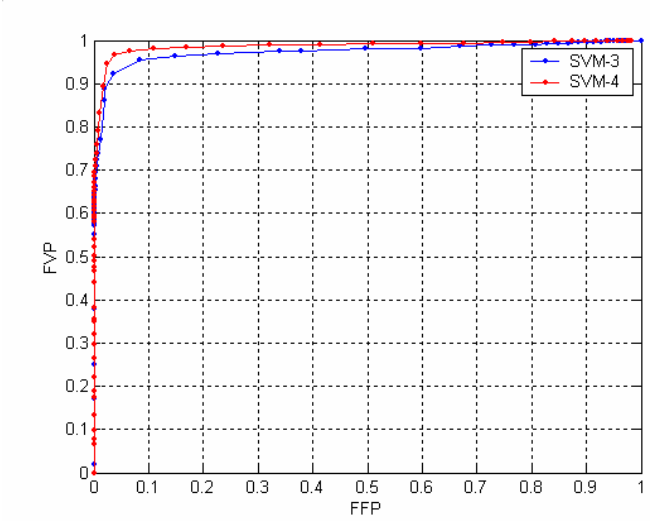


Figura 4.16: Curvas ROC generadas por las redes SVM con núcleo polinomial de grado 3

Clasificador	FFP	FVP	A_z
SVM-1	0.05	0.941197802	0.9806
SVM-2	0.05	0.930073431	0.9740
SVM-3	0.05	0.931323078	0.9728
SVM-4	0.05	0.968910518	0.9696

Tabla 4.17: Valores de FFP vs. FVP y áreas bajo la curva ROC para los clasificadores SVM con núcleo polinomial

En la Figura 4.17 se muestran las curva ROC obtenidas en el caso de emplear núcleos RBF de varianza 0.25, para el caso de usar $\gamma = \infty$ (SVM-5)o $\gamma = 20$ (SVM-6).

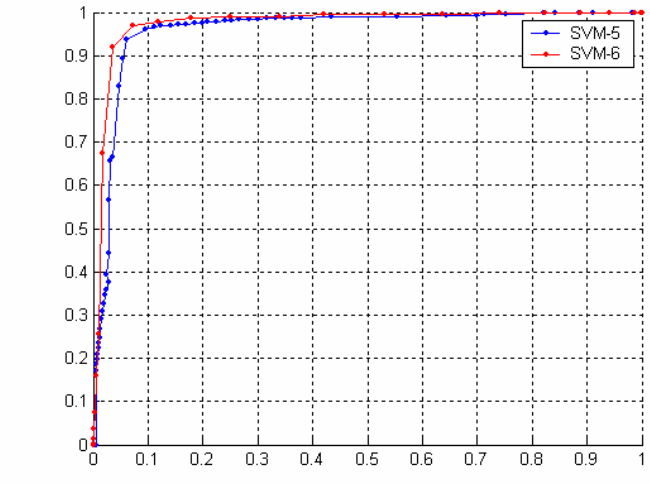


Figura 4.17: Curvas ROC generadas por las redes SVM con núcleo RBF de varianza 0.25

En la Figura 4.18 se muestran las curva ROC obtenidas en el caso de emplear núcleos RBF de varianza 0.50, para el caso de usar $\gamma = \infty$ (SVM-7)o $\gamma = 20$ (SVM-8).

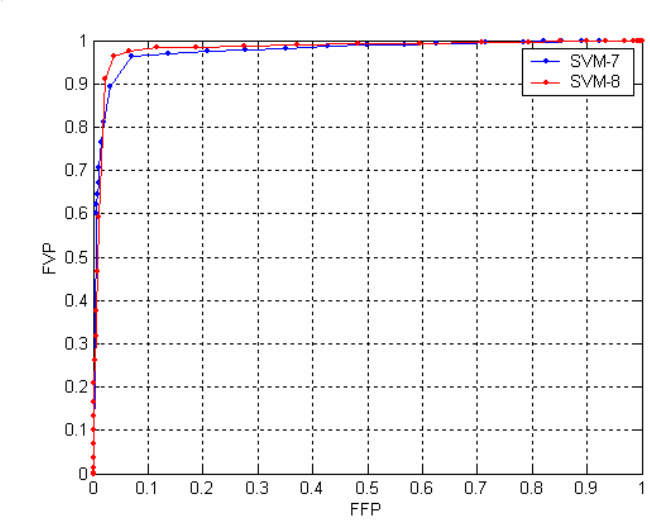


Figura 4.18: Curvas ROC generadas por las redes SVM con núcleo RBF de varianza 0.50

Las áreas bajo la curva ($\mathbf{A_z}$) y los valores de *FFP* y *FVP* para cada uno de los clasificadores **SVM,** en el caso de usar núcleos de tipo RBF, se presentan en la

Clasificador	FFP	FVP	A_z
SVM-5	0.05	0.909619048	0.9652
SVM-6	0.05	0.938228571	0.9750
SVM-7	0.05	0.926644713	0.9756
SVM-8	0.05	0.969698413	0.9809

Tabla 4.18: Valores de FFP vs. FVP y áreas bajo la curva ROC para los clasificadores SVM con núcleo RBF

Por ultimo, en la siguiente tabla se muestran el número de vectores soporte que presenta cada clasificador SVM.

Clasificador	Número de vectores soporte	%
SVM-1	152	15.2
SVM-2	145	14.5
SVM-3	62	6.2
SVM-4	124	12.4
SVM-5	127	12.7
SVM-6	224	22.4
SVM-7	78	7.8
SVM-8	152	15.2

Tabla 4.19: Relación de vectores soporte obtenidos para cada clasificador SVM

4.4. Análisis de prestaciones de los clasificadores

En este apartado se pretende analizar los resultados obtenidos en este proyecto para cada tipo de clasificador implementado, para ello se analiza inicialmente cada arquitectura por separado, se propone el *"mejor"* clasificador en cada arquitectura y por ultimo se realiza una comparativa global de todos las arquitecturas y los resultados obtenidos.

4.4.1. Análisis de resultados para los clasificadores MLPs

Primero se van a analizar globalmente los resultados para los algoritmos de entrenamientos implementados. A priori, no se puede indicar que algoritmo dará los mejores resultados, pero tras estudiar los resultados, se llegan a las siguientes conclusiones. El algoritmo por descenso de gradiente con término de momento, ***traingdm***, es el que *peores resultados* presenta, tanto en términos de área A_z, como en términos de ***FVP*** y ***FFP***, para todos los casos. Mientras en los demás entrenamientos, los resultados dependen de la topología de la red implementada, pero se puede concluir que el algoritmo de gradiente conjugado, en cualquiera de sus dos versiones ***traincgf*** y ***traincgp***, es el que *mejores resultados* proporciona. Los resultados de los restantes entrenamientos dependen del número de neuronas en cada capa.

Una vez que se ha escogido el tipo de entrenamiento que presenta mejores resultados globalmente, queda determinar la influencia del número de neuronas en cada capa oculta, m_1 y m_2. Para ello se han analizado distintas topologías, variando el número de neuronas m_1 y m_2.

En las Figuras 4.19, 4.20 y 4.21 se ha representado para el entrenamiento *traincgf*, el área A_z para un número fijo de neuronas en la primera capa, seis, siete y ocho respectivamente, variando el número de neuronas en la segunda capa. Se ha aumentado el numero de neuronas en la primera capa con respecto al máximo usado hasta este momento, ya que el valor mayor de m_1 que se ha utilizado en el apartado 4.3.1 era m_1**=6**, para ver como varían el valor del área A_z. Además analizando las curvas ROC, los resultados van empeorando considerablemente conforme se empieza a reducir m_1, por tanto, a partir de este momento se fijarán valores mayores a 4 para el número de neuronas en la primera capa

Observando dichas gráficas, podemos considerar como valor óptimo para el numero de neuronas de la segunda capa, m_2**=4**. Pues los resultados no mejoran prácticamente en caso de tomar un valor mayor, incluso empeoran, además hay que tener en cuenta que aumentar el número de neuronas implica un aumento del tiempo computacional que no supone en nuestro caso un incremento del área A_z, luego podemos concluir que con 4 neuronas en la segunda capa se obtienen los mejores resultados empleando el entrenamiento *traincgf*.

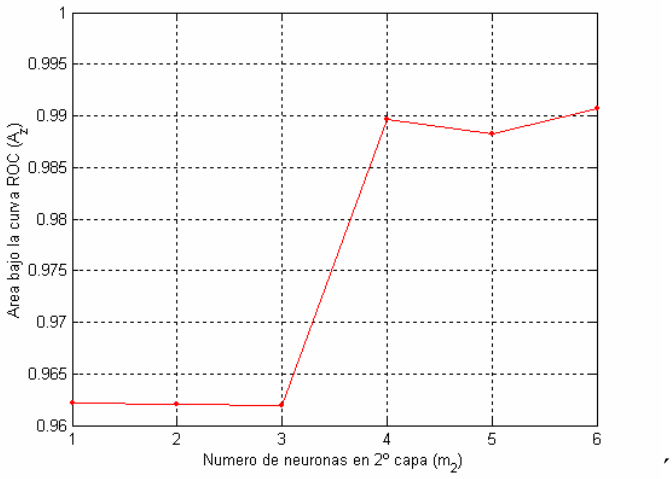


Figura 4.19: Área A_z vs. Número de neuronas de la segunda capa m_2, con m_1=6 y usando *traincgf*

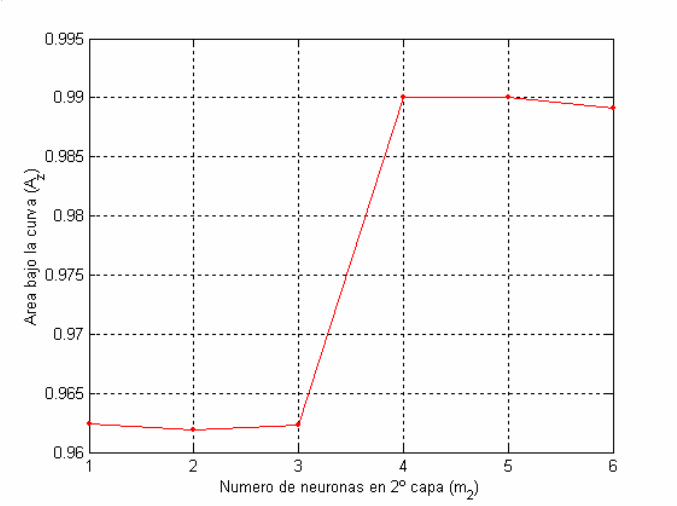


Figura 4.20: Área A_z vs. Número de neuronas de la segunda capa m_2, con m_1=7 y usando *traincgf*

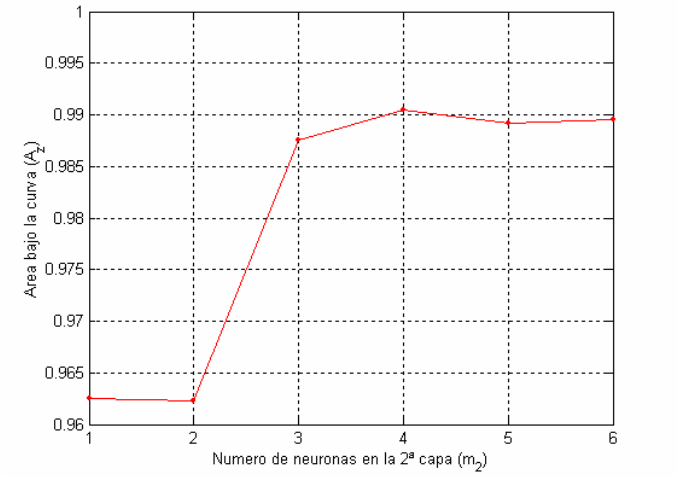


Figura 4.21: Área A_z vs. Número de neuronas de la segunda capa m_2, con m_1=8 y usando *traincgf*

Queda por determinar el número de neuronas en la primera capa, $\boldsymbol{m_1}$, para ello se realizara un procedimiento similar al anterior. En la Figura 4.22 se ha representado el valor de $\boldsymbol{A_z}$ con respecto a $\boldsymbol{m_1}$, con $\boldsymbol{m_2}$**=4** y ***traincgf***.

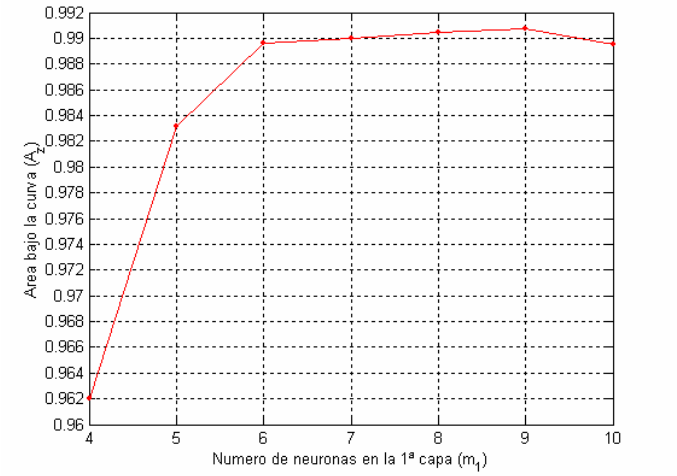
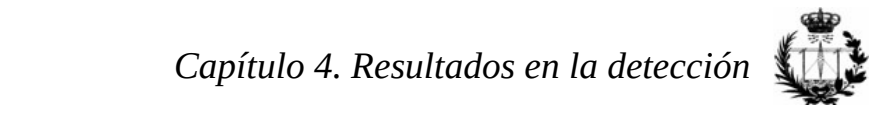


Figura 4.22: Área A_z vs. Número de neuronas de la primera capa m_1, con m_2=4 y usando *traincgf*



A la vista de los resultados, teniendo en cuenta que en el caso de escoger m_1=**9** se consigue un mayor valor del área bajo la curva luego, se puede afirmar que la arquitectura **MLP** que presenta *mejores resultados* es la red con **9** neuronas en la primera capa oculta, **4** neuronas en la segunda capa y 1 neurona en la capa salida, y que además es entrenada mediante el algoritmo de **gradiente conjugado**. En la Figura 4.23 se ha representado su curva ROC.

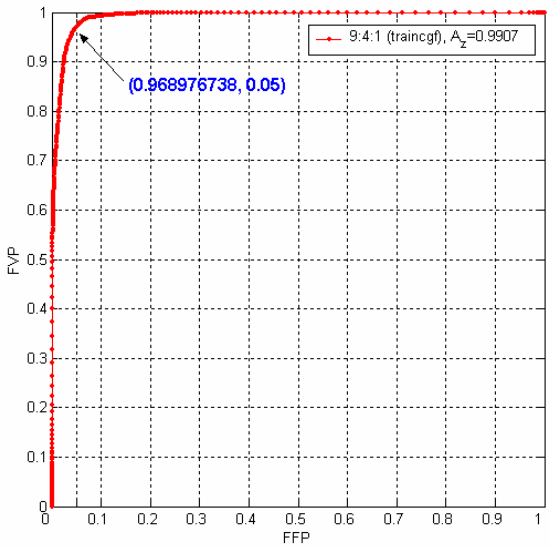


Figura 4.23: Curvas ROC generadas por el clasificador MLP 8:4:1 entrenando mediante gradiente conjugado

4.4.2. Análisis de resultados para los clasificadores RBFs

Como se puede comprobar examinando las curvas obtenidas en el apartado 4.3.2, se verifica que el uso de **varianza constante** para todas las neuronas de la red es apropiado en aquellos casos donde se tiene un gran número de centroides; mientras que el uso de **varianza adaptativa** para cada neurona conlleva mejores resultados en aquellas topologías donde se emplean muy pocos centroides. En definitiva, es conveniente usar varianza adaptativa en topologías con un muy bajo número de neuronas, pero conforme el número de neuronas aumenta, se consiguen mejores resultados empleando varianza fija. En la Figura 4.24, se muestra la evolución del área bajo la curva ROC conforme aumenta el número de centros, tanto para varianza constante como para varianza adaptativa para cada neurona. En ella se puede apreciar como se obtiene menor área en el caso de varianza adaptativa que en el caso de varianza conforme aumenta el número de centros, y viceversa.

Otra cuestión importante que queda es la ***selección automática del número de centroides***, ya que con **FSCL** tenemos que imponer un número inicial de centros, y mejorar la distribución de éstos sobre el espacio de entrada. Para conseguirlo se ha propuesto aplicar **LVQ3** a los centroides obtenidos mediante **FSCL**, ya que **LVQ3** es un *método supervisado*, donde se conocen las clases de los centroides, con el que podemos distribuir de una manera óptima los centroides en el espacio de entrada.

Primeramente se emplea **FSCL** para un número fijo de centros. A continuación hay que determinar la clase de los centroides, esto se hace asignándole a cada uno aquella clase que sea mayoritaria entre los patrones más cercanos a dicho centroide.

Tras esto se emplea **LVQ3** sobre el conjunto de centroides etiquetado procedente de **FSCL**. Puede ocurrir que algún de los nuevos centros no tenga asignado[9] ningún patrón, pues al usar **LVQ3** se han desplazado los centroides de manera tal que se pueda conseguir una mejor clasificación. Si esto ocurre, el centroide es eliminado, logrando así una selección automática del número de centros. Queda establecer las varianzas de los nuevos centroides. Se probaran nuevamente los dos casos, varianza constante e igual para todos los centroides y varianza distinta para cada centroide. Mediante este proceso se han conseguido mejores resultados en términos de curvas ROC. Presenta el inconveniente de tener que fijar el valor de tres parámetros que se usan en el **LVQ3**, estos son:

- α : Primer "learning rate" del LVQ3
- ε : Segundo "learning rate" del LVQ3
- *steps* : Numero de muestras que se pasan en el proceso LVQ3

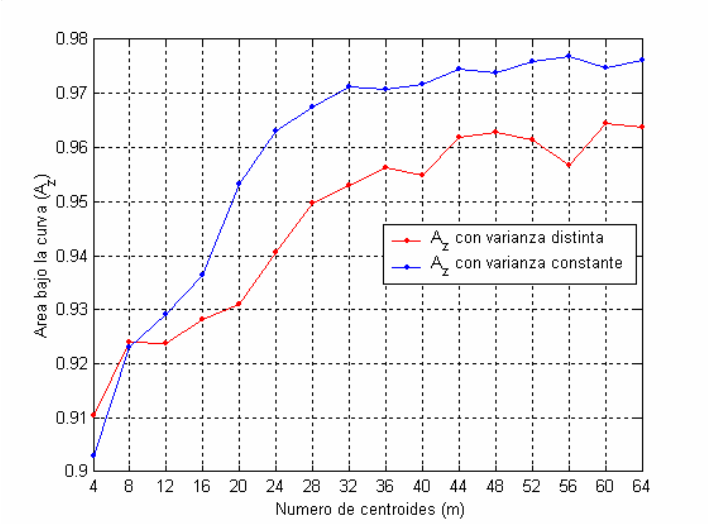


Figura 4.24: Evolución del área bajo la curva A_z vs. Numero de centroides (m)

Para conocer el valor adecuado de dichos valores han sido necesario varias simulaciones, y con ellas, se ha decidido tomar $\alpha = 0.05$ y *steps* = 50. Queda por determinar el valor del parámetro ε, el cual está relacionado con el tamaño de ventana que se emplea. Dicho valor influye sensiblemente en los resultados, por lo cual en la figura se muestra su influencia. En la Figura 4.25se representa el valor del área de la curva ROC para distintos valores de ε. Los resultados proceden de emplear **FSCL** para *48* centroides, y a continuación utilizar **LVQ3**, hay que tener en cuenta que con **LVQ3** se reduce el número de centros en cada caso, de tal manera, que para *vard* se ha reducido hasta 41 centros, mientras que para *varf* se reduce hasta 37 centros.

Como se puede observar en la Figura 4.25 para cada método de estimación de la varianza se obtiene un valor óptimo de ε diferente. En concreto, en el caso de usar **varianza fija** para todas las neuronas, el mejor resultado en términos de $\boldsymbol{A_z}$, $\boldsymbol{A_z}$ = **0.9785**, se consigue con $\boldsymbol{\varepsilon}$ = **0.005**, mientras que en el caso de emplear **varianza distinta** para cada neurona, el mejor resultado términos de $\boldsymbol{A_z}$, $\boldsymbol{A_z}$ = **0.9745**, se consigue con $\boldsymbol{\varepsilon}$ = **0.0325**. Si se comparan estos valores con los mejores valores de $\boldsymbol{A_z}$ de las tablas

[9] La asignación se realiza teniendo en cuenta la distancia euclídea, asignando a cada centroide aquellos patrones que tiene más cercanos

4.15 y 4.16, se comprueba que aplican **LVQ3** se han conseguido aumentar el valor de A_z en ambos casos.

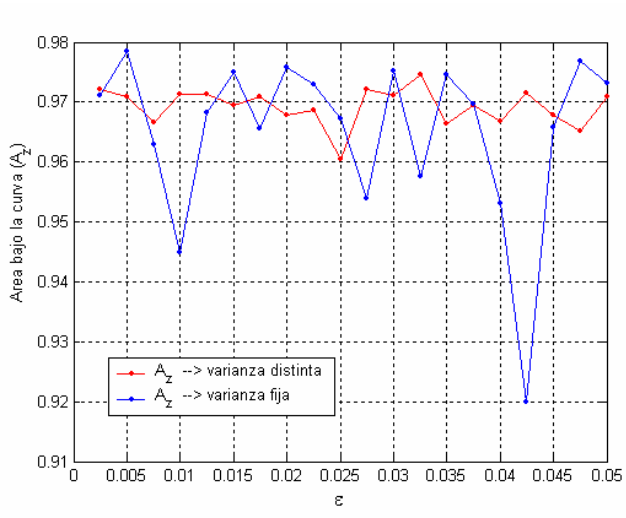


Figura 4.25 : Área bajo la curva A_z vs. el parámetro ε

Con lo cual, las redes **RBF** que presentan mejores resultados, para cada método de estimación de la varianza, han resultado ser las siguientes:

- ***RBF-varf* : m=37, α = 0.05, ε = 0.0325, *steps* = 50**
- ***RBF-vard* : m=41, α = 0.05, ε = 0.0050, *steps* = 50**

Para finalizar, en la Figura 4.26 se ha representado las curvas ROC de los dos mejores clasificadores **RBF**:

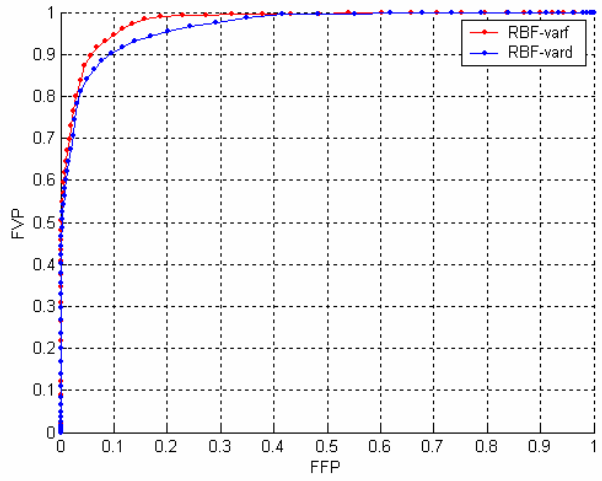


Figura 4.26 : Curvas ROC generadas por los clasificadores RBF-*varf* y RBF-*varv*

4.4.3. Análisis de resultados para los clasificadores SVMs

Los resultados obtenidos en el apartado 4.3.3 no se pueden comparar directamente con los de los apartados 4.3.1 y 4.3.2 ya que proceden de conjuntos de entrenamiento y test distintos, por lo cual nos limitaremos a analizar los resultados solo cualitativamente, pues no se pretende en este proyecto hacer un estudio completo de los

clasificadores SVM debido a la gran cantidad de carga computacional que supone, si no que únicamente se pretende comprobar su funcionamiento en un problema de detección de microcalcificaciones.

En el caso de los núcleos polinomiales se observa que el efecto de aumentar el grado del polinomio, es decir, ajustar aun más las fronteras de decisión al conjunto de patrones de entrenamiento, lo que se denomina como sobreajuste, no conlleva una mejora de resultados, ya que en general se está perdiendo generalización. El efecto de la generalización se puede apreciar también al reducir a 20 el parámetro γ, ya que en la mayoría de los casos, excepto para polinomios de grado 2, al reducir γ aumenta el área bajo la curva, esto es debido, a que con ello se permite cometer un mayor número de errores y así se evita el sobreajuste.

En el caso de los núcleos RBF, encontramos el clasificador SVM que presenta mejores resultados, siendo este el clasificador **SVM-8**, que presenta núcleos RBF con varianza 0.5 y γ=20.

4.4.4. Análisis comparativo de curvas ROC generadas por los mejores clasificadores

En este apartado se quiere concluir el estudio de las curvas ROC generadas los distintos clasificadores representando en un misma gráfica las curvas de los mejores clasificadores en cada tipo de arquitectura neuronal implementada. Como se puede observar en la figura siguiente, el ***MLP 9:4:1*** es el que presenta mejores resultados, tras el, el ***SVM-8*** y por ultimo la ***RBF-varf***. Cabe comentar que esta comparativa es únicamente cualitativa y no cuantitativa, pues los clasificadores **SVM** han sido entrenados con conjuntos de patrones distintos al resto de arquitecturas.

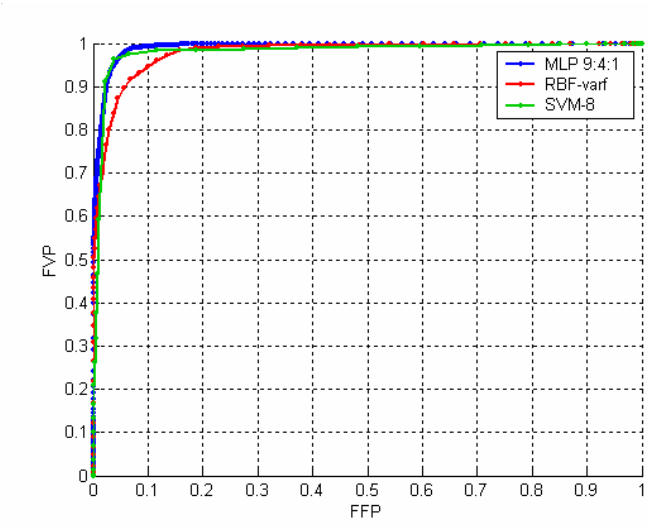


Figura 4.27: Curvas ROC generadas por los mejores clasificadores

4.5. Resultados visuales en la detección de microcalcificaciones

A continuación, para varias ROIs sensibles a contener microcalcificaciones se presentan los resultados visuales que se obtienen tras la etapa de clasificación. El objetivo de este ejemplo es mostrar las diferencias de los resultados que se perciben visualmente entre unos clasificadores u otros.

En la Figura 4.28 se muestra la vista cráneo-caudal de una mamografía digitalizada con un diagnostico de cáncer de mama. Se ha seleccionado una ROI de 128 x 128 marcada por un radiólogo experto. A simple vista se observan multitud de microcalcificaciones en dicha ROI.

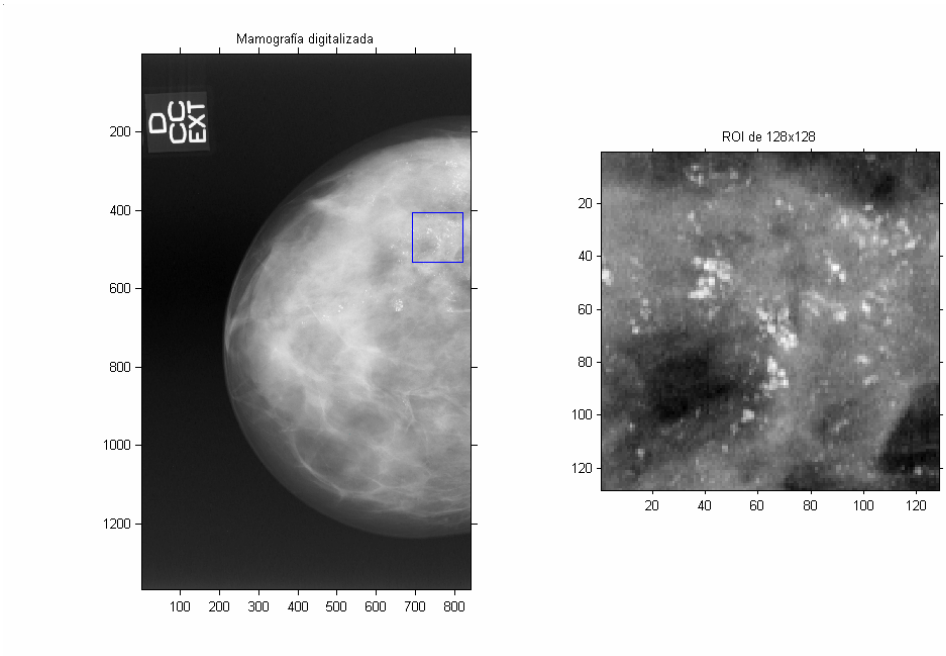


Figura 4.28: Mamografía digitalizada y ROI bajo estudio

Los resultados obtenidos se muestran en la Figura 4.29, **Figura 4.30** y Figura 4.31 cuando se aplican las mejores arquitecturas en la etapa de clasificación. Analizando los resultados obtenidos, se observa que se detectan todas las microcalcificaciones que son visibles y otras que no son visibles claramente. Además se puede ver como el clasificador **MLP 9:4:1** proporciona una mejor detección que el clasificador **RBF-*varf***, pues como se demostró en los apartados anteriores, las prestaciones del **MLP** son superiores a las de la red **RBF**, y por tanto proporcionará una detección más perfeccionada, es decir, segmenta las microcalcificaciones del resto de la imagen de una forma más refinada. En el caso de **SVM-8** funciona de una manera similar a la **RBF**, pero a la vez proporciona una detección más refinada que esta.

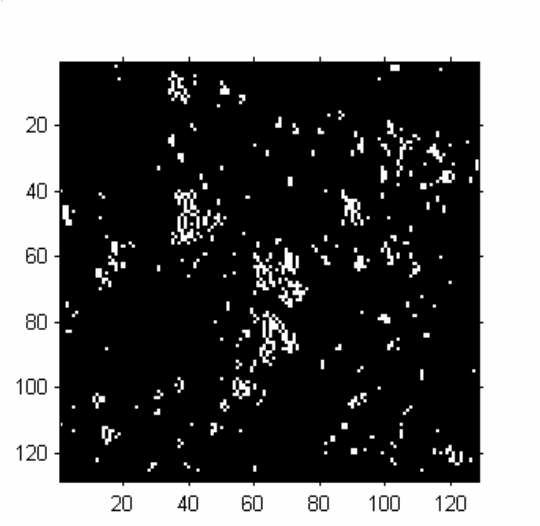


Figura 4.29: Microcalcificaciones detectadas por el clasificador MLP 9:4:1

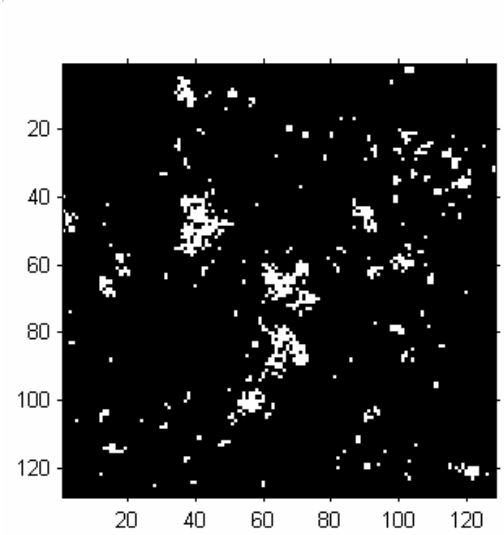


Figura 4.30: Microcalcificaciones detectadas por el clasificador RBF-varf

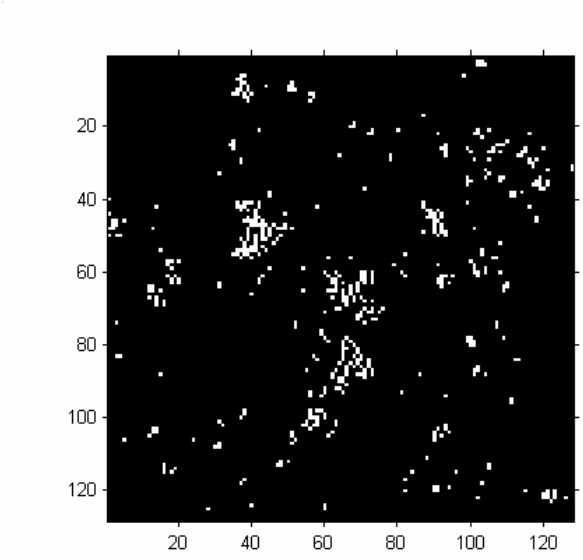


Figura 4.31: Microcalcificaciones detectadas por el clasificador SVM-8

En la Figura 4.32 se muestra la vista cráneo-caudal de una mamografía digitalizada con un diagnostico de caso normal. Se ha seleccionado una ROI de 128 x 128 que no es sospechosa, y que en apariencia corresponde a tejido sano. En este caso no existen microcalcificaciones en la ROI. Los resultados obtenidos cuando se aplican las mejores arquitecturas en la etapa de clasificación se muestran en la Figura 4.33. Analizando los resultados, se observa como en ningún caso se ha detectado la presencia de microcalcificaciones en la ROI bajo estudio.

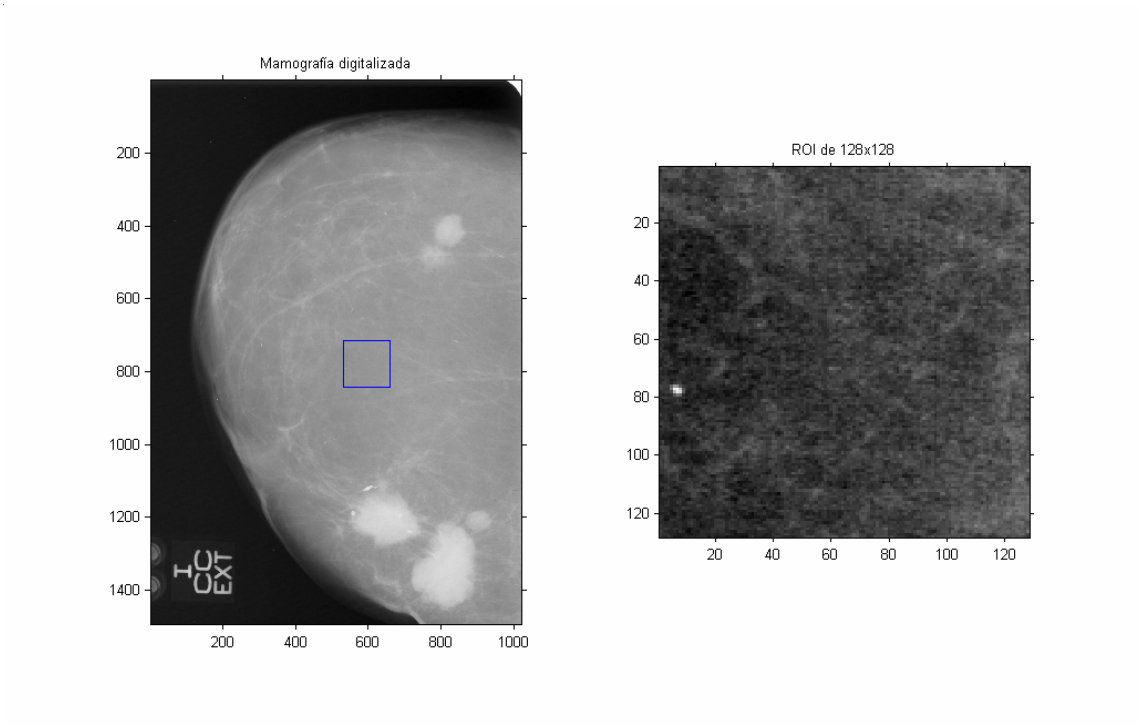


Figura 4.32: Mamografía digitalizada y ROI bajo estudio

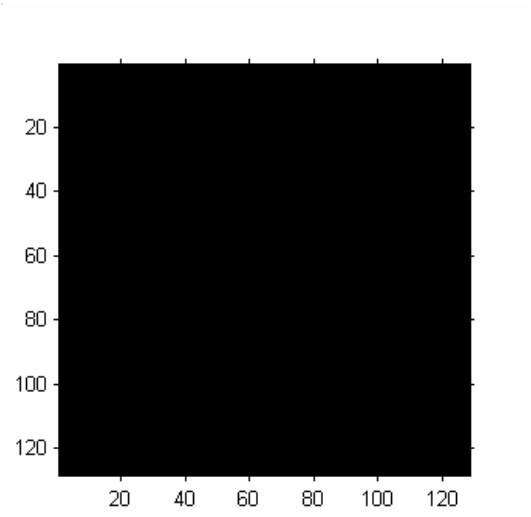


Figura 4.33: Microcalcificaciones detectadas por MLP 9:4:1, RBF-*varf* y SVM-8

Capítulo 5
Conclusiones y líneas futuras

5.1. Introducción

En este proyecto, se ha propuesto una mejora en la etapa de clasificación de un sistema CAD para la detección de microcalcificaciones sobre mamografía digitalizada. Se han refinado los resultados considerablemente, superando sensiblemente los mejores resultados que se obtuvieron en el modelo VECO. Esto es debido al perfeccionamiento de la etapa de clasificación, al emplear arquitecturas neuronales optimas para la detección, lo que ha supuesto un avance en la detección de microcalcificaciones.

5.2. Conclusiones finales

Para realizar este proyecto, se ha usado la metodología propuesta durante el desarrollo del modelo VECO para la detección, junto con una base de datos de imágenes con diagnostico previo, es decir, de estas se tiene un conocimiento a priori de la presencia o no de microcalcificaciones. Tras aplicar el procesado y extracción de características a las imágenes bajo estudio, se obtuvieron los conjuntos de entrenamiento y test que han servido para el desarrollo de los distintos modelos neuronales con los que se ha trabajado. A continuación, nos centramos en la etapa clasificación, objetivo principal de este proyecto. Para ello, se emplearon perceptrones multicapa, redes de funciones de base radial y, para concluir, maquinas de vectores de soporte. Se ha estudiado cada arquitectura de manera individual, refinando los algoritmos de entrenamiento, e intentando conseguir el mejor clasificador en cada caso, para luego poder comparar globalmente los resultados que se generan para cada modelo. La consecuencia directa del perfeccionamiento de la etapa de clasificación ha llevado a mejorar las prestaciones del sistema de detección, llegando a unos valores muy altos de sensibilidad y especificidad, así como también para el área bajo las curvas ROC, consiguiendo valores muy próximos a 1, que es el valor que se obtiene para el caso ideal, donde se predicen todos los casos que son cáncer y todos los casos que son tejido sano.

A continuación se describen brevemente las conclusiones finales a las que se ha llegado en este proyecto:

1º. Se han propuesto distintas alternativas para la etapa de la clasificación de patrones en un problema de detección de microcalcificaciones, consiguiendo optimizar dicha etapa.

2º. Se ha encontrado la red neuronal que da mejores resultados dentro de cada arquitectura

3º. Para cada arquitectura se han implementado distintos algoritmos que conllevan una mejora sustancial en los resultados finales

4º. Se han conseguido mejores resultados tanto en terminas de sensibilidad y especificidad, como en términos del área bajo la curva ROC que en el modelo VECO (ver Tabla 1.1)

5º. Se ha propuesto el uso de los SVM dentro de la etapa de clasificación, y se han analizado las prestaciones de esta novedosa arquitectura

5.3. Líneas futuras

Con la experiencia acumulada durante el desarrollo del proyecto, se pueden plantear las siguientes líneas futuras:

1º. Completar la etapa de clasificación con la aplicación de los SVM sobre los conjuntos totales de entrenamiento y test, para lo cual es necesario gran cantidad de memoria RAM, como ya se ha comentado.

2º. Intentar mejorar las redes RBF con algoritmos más complejos para seleccionar los centros y sus respectivas varianzas.

3º. Realizar un estudio comparativo del efecto de la resolución sobre la etapa de clasificación, pues en este trabajo siempre se han trabajado sobre imágenes de 12 bits

4º. Incorporar en la etapa final del proceso de diagnosis, la clasificación de imágenes entre malignas y benignas teniendo en cuenta las microcalcificaciones detectadas por el sistema, su distribución, número,…, incluyendo además de las características radiológicas, las características clínicas, tales como la edad de las pacientes y sus antecedentes hereditarios.

Apéndice A

Terminología clínica y curvas ROC

A.1. Introducción

La toma de decisiones clínicas es un proceso extremadamente complejo en el que deberá finalmente ser valorada la utilidad para el manejo del paciente de cualquier prueba diagnóstica. En este contexto, es imprescindible conocer detalladamente la *exactitud* de las distintas pruebas diagnósticas, es decir, su capacidad para clasificar correctamente a los pacientes en categorías o estados en relación con la enfermedad (típicamente hay *dos estados*: estar o no estar enfermo, respuesta positiva o negativa a la terapia, benignidad o malignidad, sano o anormal,…). En este proyecto se usan algunos términos que son ampliamente conocidos en el argot médico en el proceso de toma de decisiones y se usan para medir los resultados al realizar la clasificación entre casos normales o anormales (tejido sano o microcalcificación). Por ello, en este apéndice se da un resumen de las definiciones más importantes.

A.2. Terminología de Screening

La mayoría de las mediciones que se realizan en los programas de screening envuelven términos clínicos que no son muy conocidos en el lenguaje que se maneja en ingeniería, por tanto a continuación se dan algunas definiciones que ayudaran a entender estos términos:

- *Valor de predicción positiva* (*VPP*): Es una medida del poder discriminante de los métodos de diagnostico utilizados. Se define como:

$$\frac{\text{Número de cánceres verdaderos}}{\text{Número de pruebas llamadas positivas}}$$

  Esta definición se usa para medir el poder discriminante de las lesiones mamográficas y sirve como medida de la agresividad de la intervención si se recomienda una biopsia, por tanto, se tiene que

$$VPP_{\substack{\text{para recomendación}\\ \text{de biopsia}}} = \frac{\text{Número de cánceres diagnosticados}}{\text{Número de lesiones recomendadas para biopsia}}$$

- Los resultados *verdaderos positivos*, son lesiones que son llamadas cáncer y se prueba que es cáncer.
- Los resultados *falsos positivos*, son lesiones que son llamadas cáncer y se clasifican como benignas.
- Los resultados *falsos negativos*, son lesiones que son llamadas benignas (negativas) y se prueba que son cáncer.

- Los resultados *verdaderos negativos*, son lesiones que son llamadas benignas y se prueba que son benignas.

Antes de explicar los conceptos de sensibilidad y especificidad, se procede a describir en la siguiente sección el sistema BI-RADS.

A.2.1. Sistema BI-RADS

El sistema BI-RADS (Breast Imaging Reporting and Data System), creado por el Colegio Americano de Radiología (ACR) en 1993 (con ediciones posteriores en 1995 y 1998), es un sistema estandarizado de descripción de las lesiones mamográficas y de elaboración del informe radiológico.

La clasificación de BIRADS nace en 1993 como una necesidad de estandarizar los informes, establecer un grado de sospecha de malignidad y permitir la toma de decisiones para así maximizar la detección del cáncer en estado precoz y reducir los falsos positivos en las biopsias. Esta clasificación otorga un valor pronóstico establecido por estudios que correlacionaron radiología y biopsias, determinando el porcentaje de malignidad para cada categoría BI-RADS.

En la Tabla A.1 se describe el sistema BI-RADS, que como ya se ha mencionado sirve para clasificar los hallazgos mamográficos y orientar la actitud del doctor:

Categoría BI-RADS	Recomendación
BI-RADS 1 - Negativa (ningún hallazgo)	Revisión Rutinaria
BI-RADS 2 - Apariencia Benigna	Revisión Rutinaria
BI-RADS 3 - Apariencia probablemente benigna	Seguimiento a los 6 meses y durante 2 años *
BI-RADS 4 - Hallazgos sospechosos de cáncer	Considerar Biopsia
BI-RADS 5 - Hallazgos muy sospechosos de cáncer	Misma pauta que en grupo 4
BI-RADS 0 - Estudio incompleto	Pruebas adicionales, comparar previas
* Puede realizarse biopsia si la paciente lo desea ó está preocupada o si la lesión durante el control no permanece estable	

Tabla A.1: Clasificación BI-RADS de la American College of Radiology (ACR)

A.2.2. La sensibilidad y la especificidad

Generalmente, la *exactitud diagnóstica* se expresa como ***sensibilidad*** y ***especificidad*** diagnósticas. Cuando se utiliza una prueba dicotómica (una cuyos resultados se puedan interpretar directamente como *positivos* o *negativos*),

- La ***sensibilidad*** es la probabilidad de clasificar correctamente a un individuo cuyo estado real sea el definido como positivo respecto a la condición que estudia la prueba, razón por la que también es denominada fracción de verdaderos positivos (*FVP*).
- La ***especificidad*** es la probabilidad de clasificar correctamente a un individuo cuyo estado real sea el definido como negativo. Es igual al resultado de restar a uno la fracción de falsos positivos (*FFP*).

Cuando los datos de una muestra de pacientes se clasifican en una tabla de contingencia (o matriz de confusión, ver Tabla A.2) por el resultado de la prueba

(predicción) y su estado real respecto a la enfermedad, es fácil *estimar* a partir de ella la *sensibilidad* y la *especificidad* de la prueba.

		Verdadero Diagnóstico	
		Enfermo	**Sano**
Resultado de la prueba	**Prueba Positiva**	Verdadero Positivo (VP)	Falso Positivo (FP)
	Prueba Negativa	Falso Negativo (FN)	Verdadero Negativo (VN)
	Total	**VP+FN**	**FP+VN**

Tabla A.2: Matriz de Confusión: Resultado de una prueba y su estado respecto a la enfermedad

Conviene insistir en que lo que realmente obtenemos son *estimaciones* de los verdaderos valores de sensibilidad y especificidad para una población teórica de la que suponemos que nuestro grupo de pacientes constituye una muestra aleatoria. Por tanto, un tratamiento estadístico correcto de cantidades como las calculadas por el método descrito por la Tabla A.3 necesita incluir medidas de su precisión como estimadores, y, mejor aún, utilizarlas para construir intervalos de confianza para los verdaderos valores de sensibilidad y especificidad

Concepto	Definición
Sensibilidad (FVP)	$\frac{VP}{VP+FN}$
Especificidad (FVN)	$\frac{VN}{VN+FP} = 1 - FFP$

Tabla A.3: Sensibilidad y Especificidad

A.3. Las curvas ROC

La limitación principal del enfoque hasta ahora expuesto es la exigencia de que la respuesta proporcionada por la prueba diagnóstica sea de tipo *dicotómico*, por lo que en principio quedaría excluida una gran cantidad de pruebas diagnósticas cuyos resultados se miden en una escala nominalmente continua. Un ejemplo de ello es una prueba realizada sobre las mamografías, donde los resultados se expresen empleando las categorías "seguramente normal", "probablemente normal", "dudoso", "probablemente anormal" y "seguramente anormal".

La generalización a estas situaciones se consigue mediante la elección de distintos *niveles de decisión* o *valores de corte o umbrales* que permitan una clasificación dicotómica de los valores de la prueba según sean superiores o inferiores al valor elegido. La diferencia esencial con el caso más simple es que ahora contaremos no con un único par de valores de sensibilidad y especificidad que definan la exactitud de la prueba, sino más bien con un conjunto de pares correspondientes cada uno a un distinto nivel de decisión.

Este procedimiento constituye la esencia del análisis de curvas ROC (*Receiver Operating Characteristics*), una metodología desarrollada en el seno de la Teoría de la Decisión en los años 50 y cuya primera aplicación fue para analizar y reducir el efecto del ruido sobre señales radar (aunque el detalle pueda parecer anecdótico, son situaciones equivalentes el operador que interpreta los picos en la pantalla del radar para

decidir sobre la presencia de un misil y el médico que emplea el resultado de una prueba diagnóstica para decidir sobre la condición clínica del paciente).

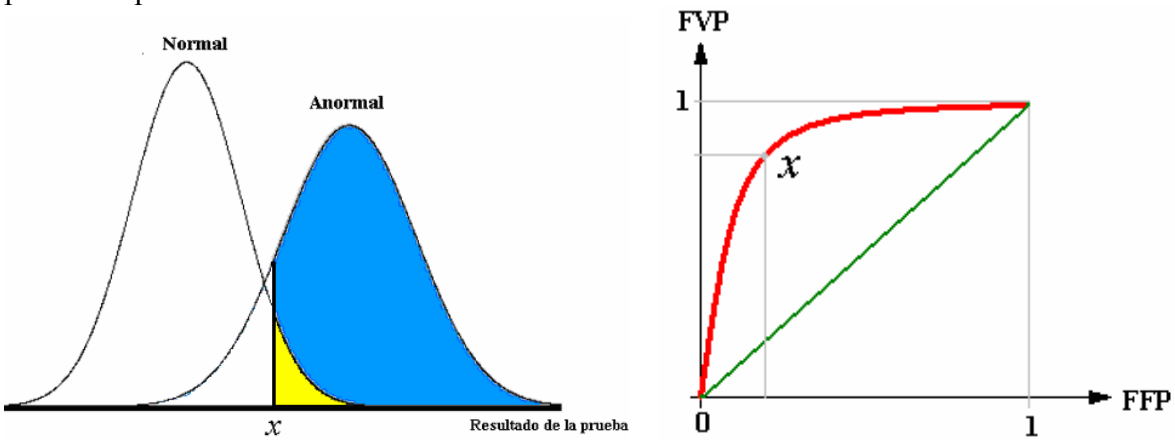
La investigación de Pickett y Swets [36] marcó el comienzo de la difusión del uso de las curvas ROC en el área de la **Biomedicina**, inicialmente en Radiología, donde la interpretación subjetiva de los resultados se recoge en una escala de clasificación.

A.3.1. Pruebas dicotómicas

Antes de describir como se generan las curvas ROC, se supone la existencia de dos poblaciones distintas, una normal (sana) y otra anormal (enferma), y para ambos casos, la variable de decisión que representa el resultado de la prueba diagnóstica sigue una distribución normal $\sim N\left(\mu,\sigma^2\right)$, con media μ y desviación típica σ conocidas.

En la Figura A.1(a) se muestran las funciones de densidad de probabilidad (fdp) para ambas variables, que presentarán un determinado nivel de solapamiento. Si consideramos un valor arbitrario del resultado de la prueba, x,(al que, en adelante, aludiremos como valor de corte), la FVP (Sensibilidad) y la FFP (1-Especificidad) se corresponderán respectivamente con el área a la derecha de ese punto bajo la función de densidad de probabilidad de la población enferma (áreas azul y amarilla) y de la población sana (área amarilla).

La curva ROC se obtiene representando, Figura A.1(b), para cada posible elección de valor de corte x, la FVP en ordenadas y la FFP en abscisas. Mediante esta representación de los pares (FFP, FVP) obtenidos al considerar todos los posibles valores de corte de la prueba, la curva ROC (en rojo) proporciona una representación global de la exactitud diagnóstica. La curva ROC es necesariamente creciente, propiedad que refleja el compromiso existente entre sensibilidad y especificidad: si se modifica el valor de corte x para obtener mayor sensibilidad, sólo puede hacerse a expensas de disminuir al mismo tiempo la especificidad. Si la prueba no permitiera discriminar entre grupos, la curva ROC sería la diagonal (en verde) que une los vértices inferior izquierdo y superior derecho. La exactitud de la prueba aumenta a medida que la curva se desplaza desde la diagonal hacia el vértice superior izquierdo. Si la discriminación fuera perfecta (100% de sensibilidad y 100% de especificidad) pasaría por dicho punto.



(a) **(b)**

Figura A.1. (a) Distribución de los resultados de una prueba en dos poblaciones: normal y anormal. (b) Curva ROC correspondiente a la distribución teórica de los resultados de la prueba. Se muestra el punto correspondiente al valor de corte x

Este modelo es aplicable en principio a datos continuos, pudiendo generalizarse al caso en que los datos se obtienen por algún sistema de clasificación en una escala discreta. Para ello basta suponer la existencia de unas variables latentes con distribución normal y de unos límites fijos que marcan los extremos de cada categoría. La Figura A.2 muestra esquemáticamente este modelo para un ejemplo con cinco categorías.

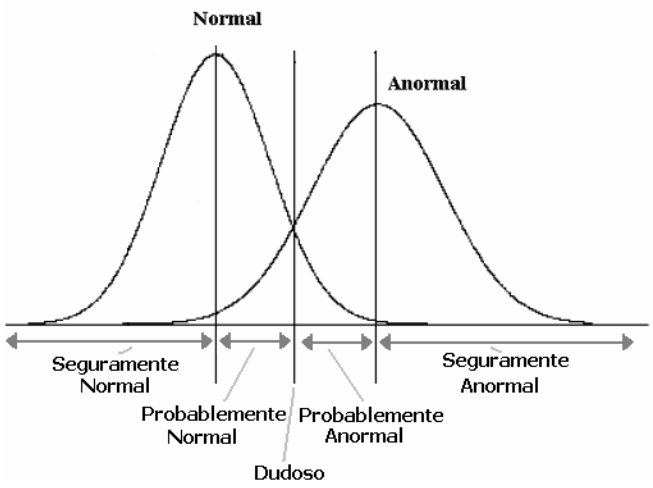


Figura A.2. Representación esquemática de un modelo para clasificación en cinco categorías

Obviamente, el escenario en que hemos presentado la curva ROC es completamente teórico, por dos razones relacionadas entre sí:

- en la práctica no disponemos de las poblaciones (abstractas) de enfermos y sanos, sino simplemente de una muestra de ellas,
- en general, no conocemos las distribuciones de los valores de la prueba diagnóstica en dichas poblaciones.

Estas limitaciones nos obligan a considerar el problema práctico de la construcción de curvas ROC, que a continuación tratamos, desde un punto de vista típicamente estadístico.

A.3.2. Construcción de la curva ROC

Un primer grupo de métodos para construir la curva ROC lo constituyen los llamados ***métodos no paramétricos***. Se caracterizan por no hacer ninguna suposición sobre la distribución de los resultados de la prueba diagnóstica. El más simple de estos métodos es el que suele conocerse como *empírico*, que consiste simplemente en representar todos los pares (FFP, FVP), es decir, todos los pares para todos los posibles valores de corte que se puedan considerar con la muestra particular de que dispongamos. Desde un punto de vista técnico, este método sustituye las funciones de distribución teóricas por una estimación no paramétrica de ellas, a saber, la función de distribución empírica construida a partir de los datos. Informalmente, es como si en la Figura A.2(a) sustituyéramos las funciones de densidad por histogramas obtenidos a partir de la muestra de pacientes sanos y enfermos y construyéramos la curva ROC a partir de ellos.

En la Figura A.3 se representa la curva ROC obtenida por el método empírico y semiparamétrico para un conjunto de datos obtenidos en un grupo de mamografías investigadas con el fin de establecer un diagnóstico de cáncer de mama mediante la determinación del número de agrupaciones por unidad de área por imagen (ROI).

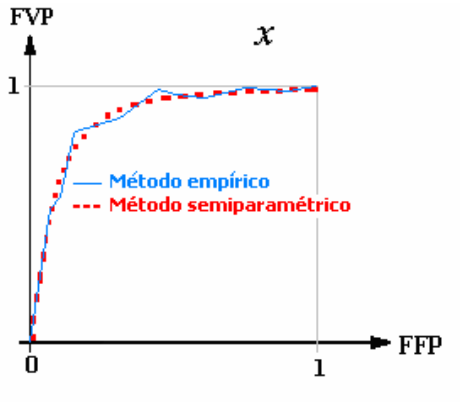


Figura A.3: Curvas ROC calculadas por los métodos empírico y semiparamétrico

La representación obtenida por el método *empírico* tiene forma aproximadamente en **escalera,** mientras que en el caso de emplear el método *semiparamétrico* tiene una forma **suavizada**. Para cada variación mínima del valor umbral que produzca cambios en sensibilidad o especificidad, al menos un caso pasa a ser considerado bien como verdadero positivo, lo que se corresponde con un trazo vertical, bien como falso positivo, lo que da lugar a un trazo horizontal. Existe aún otra posibilidad, derivada de la posibilidad de que se produzcan empates, es decir, dos o más casos con el mismo valor de la prueba: si el empate ocurre entre un caso del grupo enfermo y otro del grupo sano aparecerá un trazo diagonal en la representación. En este caso la apariencia dentada de la curva es menos acusada conforme crece el tamaño de la muestra, y por tanto, idealmente, en el límite se tendría una curva suave como la curva ROC teórica de la Figura A.2(b).

Existen otros métodos no paramétricos [37] aplicables a datos continuos que permiten obtener curvas ROC suavizadas, en contraposición con la forma dentada de la curva obtenida por el método empírico. La idea es básicamente obtener estimaciones no paramétricas suavizadas de las funciones de densidad de las dos distribuciones de resultados de la prueba empleando generalmente estimadores de tipo núcleo. A partir de dichas densidades (en lugar de a partir de los histogramas, como en el método anterior) se obtiene directamente la curva ROC que, como se dijo anteriormente, será suave.

Los ***métodos paramétricos*** se basan en asignar un determinado tipo de distribución para la variable de decisión en las dos poblaciones que se trata de distinguir [36]. El modelo más frecuentemente utilizado es el *binormal*, que supone la normalidad de las variables tanto en la población normal como la anormal, pero existen muchos otros modelos posibles que surgen al considerar distintas distribuciones, similares a la normal como la logística (modelo *bilogístico*) o no, como la exponencial negativa. El problema ahora se reduce a estimar los parámetros de cada distribución por un método estadísticamente adecuado, en general el método de máxima verosimilitud. Se obtiene así una curva ROC suave, pero puede ocurrir una falta de ajuste si las supuestas distribuciones resultan ser erróneos [38,39].

Varios investigadores han examinado el modelo binormal para clasificación de datos, sin encontrar situaciones en las que el modelo fallara seriamente [40,41]. De hecho, esta última observación constituye la base para un método aplicable tanto a datos continuos como de clasificación, debido a Metz et al. [42][10]. Según este método,

[10] Para mayor referencia, http://xray.bsd.uchicago.edu/krl/KRL_ROC/

primero se agrupan los datos en categorías ordenadas y después se aplica un algoritmo paramétrico para crear una curva ROC suave. Del método se dice que es *semiparamétrico* [37,43], porque aunque supone la existencia de una transformación que haga que las dos distribuciones sean aproximadamente normales, ésta se deja sin especificar.

A.3.3. Análisis de resultados usando las curvas ROC

Como se ha explicado en el apartado A.3.2., la mayor exactitud diagnóstica de una prueba se traduce en un desplazamiento ***"hacia arriba y a la izquierda"*** de la curva ROC. Esto sugiere que el área bajo la curva ROC (A_z) se puede emplear como un índice conveniente de la exactitud global de la prueba: la exactitud máxima correspondería a un valor de A_z de 1 y la mínima a uno de 0.5 (si fuera menor de 0.5 debería invertirse el criterio de positividad de la prueba).

En términos probabilísticos, si X_A y X_N son las dos variables aleatorias que representan los valores de la prueba en las poblaciones anormal y normal, respectivamente, puede probarse que el A_z de la *"verdadera"* curva ROC (intuitivamente, aquella que obtendríamos si el tamaño de la muestra fuera infinito y la escala de medida continua) es precisamente, la probabilidad de que, si se eligen al azar un caso anormal y otro normal, sea mayor el valor de la prueba en aquél que en éste [44], es decir $\theta = \Pr\{X_A > X_N\}$

Cuando la curva ROC se genera por el método empírico, independientemente de que haya empates o no, el área puede calcularse mediante la *regla trapezoidal*, es decir, como la suma de las áreas de todos los rectángulos y trapecios (correspondientes a los empates) que se pueden formar bajo la curva. Estadísticamente, la observación es importante, puesto que permite hacer contrastes de significación y dar intervalos de confianza para la verdadera área bajo la curva, así que el área calculada por el método geométrico anterior coincide con el valor del estadístico de suma de rangos de Wilcoxon ($\mathcal{W}$) [45].

Bibliografía

[0] R.A.Robb. "Biomedical imagining, visualization and análisis". Wiley-Liss, 2000

[1] "Factores pronósticos del cáncer de mama." http://www.opolanco.es/Apat/Boletin13/pronosti.htm . Abril 2000

[2] A.J. Alberro. "Manejo Clínico de las lesiones no palpables: diagnostico y tratamiento". http://www.cirugest.com/revisiones/. Febrero 2002

[3] J.C.Miller. "Screening Mammography. Who needs it ? What are the benefits?" Radiology Rounds, Agosto 2003.

[4] D.B. Kopans. "Breast Imaging". Philadlephia, 1998

[5] R.N. Strickland. "Wavelet transforms for detecting microcalcifications in mamography". Image Processing. Proceedings. ICIP94. IEEE International Conference,1:402-406.1994

[6] G. te Brake and N. Karssemeijer. "Detection criteria for evaluation of computer aided diagnosis systems". Engineering in Medicien and Biology Society, 1996. Bridging Disciplines for Biomedicina, 18th Annual Internacional Conference of the IEEE, 3:1157-1159,1997.

[7] W.J.H. Veldkamp and N. Karssemeijer. "Normalization of local contrast in mammograms". IEEE Transactions on Medical Imaging, 19(7): 731-738, 2000.

[8] R.M. Nishikawa, Y. Jiang, M.L. Giger, K. Doí, C.J. Vyborny, and R.A. Schmidt. "Computer aided detection of clustered mícrocalcifications". IEEE International Conference on Man and Cybernetics Systems.1992, 2:1375-1378, 1992.

[9] K. Doi, M. Giger, H. MacMahon, R.Nishikawa, R. Schmidt, K. Hoffmann, S. Katsuragawa, S. Sanada, F. Behien, and D. Sinis. "Development of digital processing techniques for computer aided diagnosis in radiographic images." IEEE Proceedings. The Third International Conference on Image Management and Communication in Patient Care.IMAC93, pages 110-115, 1993.

[10] Y. Jiang, R.M. Nishikawa, D.E. Wolverton, C.E. Metz, R.A. Schmidt, and K.Doi. "Computerized classification of malignant and benign clustered microcalcifications in mammograms". Engíneerirg in Medicine and Biology Society,1997. Proceedings of the l9th Annual International Conference of thee IEEE, 2:521-523, 1997.

[11] B. Verma. "A neural network bse technique to locate and classify microcalcifications in digital mammograms" 1790-1793. IEEE 1998

[12] M.A. Kupinski and M.L. Giger. "Feature selection and classifiers for the computerized detection of mass lessions in digital mammography". 2460-2463. IEEE 1997.

[13] A. Hojjatoleslami, L. Sardo and J. Kitler "An RBF based classifier for detection of microcalcifications in mammograms with outlier rejection capability" 1379-1384. IEEE 1997.

[14] A.C. Patrocinio and H. Schiabel. "Classifing clusters of microcalcifications in digitized mammograms by artificial neural network" 266-272. IEEE 2001.

[15] T.C. Wang and N.B. Karayiannis. "Detection of microcalcifications in digital mammograms using wavelets". IEEE 1998.

[16] Y.Songyang and G.Ling "A cad system for automatic detection of clustered microcalcifications in digitized mammograms films". IEEE Transactions on Medical Imaging, 2(2):115-126. 2000.

[17] A. Vega Corona. "Contribución a la diagnosis asistida sobre mamografía digitalizada mediante la aplicación sinérgica de la transformada wavelet y clasificadores neuronales". U. Politécnica de Madrid, Tesís Doctoral. 2004.

[18] A. Bazzaní, A. Bevilacqua, D. Bollini, R. Brancaccio, R. Campanini, N. Lanconcili, A. Riccardi, D. Romani and G.Zamboni. "Automatic detection of clustered microcalcifications in digital mammograms using an SVM classifier" ESANN2000 proceedings - European Symposium on Artificial Neural Networks. Bruges (Belgium), 26-28 Abril 2000, 195-200

[19] I. El-Naqa, Y. Yang, M.N. Wernick, N.P. Galatsanos and R.M. Nishikawa "A support vector machine approach for detection of microcalcifications". IEEE 2002. 1552-1563.

[20] C.M. Bishop. "Neural Network for Pattern Recognition" Oxford University Press.1995

[21] R.O. Duda, P.E. Hart, D.G. Stork. " Pattern classification". John Wiley & Sons. 2001

[22] M. Kupinski, D. Edwards, M. Giger, and C. Metz. "Ideal observer approximation using bayesian classification neural networks". IEEE Transaction on Medical Imaging, 20(9):886-889, September 2001.

[23] A.K. Jain, J. Mao, and K.M. Mohiuddin. "Artificial Neural Networks: A Tutorial". IEEE Computer Magazine, 29(3):31.44, 1996.

[24] W.S. McCulloch and W. Pitts. "A logical calculus of the ideas immanent innervous activity". Bulletin of Mathematical Biophysics, 5:115.133, 1943.

[25] Simon Haykin. "Neural Networks: A Comprehensive Foundation". Prentice Hall, Upper Saddle River, N.J., 1999.

[26] F. Rosenblatt. "Principles of Neurodynamics". Spartan Books, Washington, D.C., 1962.

[27] M.L. Minsky and S.A. Papert. "Perceptrons: An Introduction to Computational Geometry". MIT Press, Cambridge, Mass., 1969.

[28] C. von der Malsburg. "Self-organization of orientation sensitive cells in the striate cortex". Kybernetik, 14:85.100, 1973.

[29] D.J. Willshaw and C. von der Malsburg. "How patterned neural connections can be set up by self-organization". In Proceedings of the Royal Society of London, volume 194 of Series B: Biological Sciences, pages 431.445, 1976.

[30] J.J. Hopfield. "Neural networks as physical systems with emergent collective computational abilities". In Proceedings of the National Academy of Sciences, USA, volume 79, pages 2554.2558, 1982.

[31] P.J. Werbos. "Beyond Regression: New tools for prediction and analysis in the behavioral sciences". PhD thesis, Dept. of Applied Mathematics, Harvard University, Cambridge, Mass., 1974.

[32] David E. Rumelhart and James L. McClelland, editors. "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", volume 1. MIT Press, Cambridge, Mass., 1986.

[33] T. Kohonen. "Self-organized formation of topologically correct feature maps" Biological Cybernetics, 43:59.69, 1982.

[34] Burges, Cristopher J. C., "A tutorial on support vector machines for pattern recognition",http://svm.research.bell-labs.com/papers/tutorial_web_page.ps.gz.

[35] Gunn, Steve, "Support vector machines for classification and regression ", inf. téc., Image Speech & Intelligent Systems Group, University of Southampton, mayo 1998, http://www.isis.ecs.soton.ac.uk/ resources/svminfo/svm.ps.gz.

[36] Swets JA, Pickett RM. "Evaluation of diagnostic systems: methods from signal detection theory". Nueva York: Academic Press; 1982.

[37] Zou KH, Hall WJ, Shapiro DE. "Smooth non-parametric receiver operating characteristic (ROC) curves for continuous diagnostic tests". Statist Med 1997; 16: 2143-2156.

[38]. Zweig MH, Campbell G. "Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine". Clin Chem 1993; 39: 561-577.

[39] Burgueño MJ, García-Bastos JL, González-Buitrago JM. "Las curvas ROC en la evaluación de las pruebas diagnósticas." Med Clin (Barc).1995; 104: 661-670.

[40] Hanley JA. "The robustness of the binormal model used to fit ROC curves". Med Decision Making.1988; 8: 197-203.

[41]. Swets JA. "Form of empirical ROCs in discrimination and diagnostic tasks: implications for theoryand measurement of performance". Psych Bull 1986; 99: 181-198.

[42] Metz CE, Herman BA, Shen, J. Maximum likelihood estimation of receiver operating characteristic (ROC) curves from continuously distributed data. Statist Med 1998; 17: 1033-1053.

[43] Hsieh F, Turnbull BW. Nonparametric and semiparametric estimation of the receiver operating characteristic curve. Ann Statist 1996; 24: 25-40.

[44] Hanley JA, McNeil BJ. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology 1982; 143: 29-36.

[45] Bamber D. The area above the ordinal dominance graph and the area below the receiver operating graph. J Math Psych 1975; 12: 387-415.