




MÁSTER EN INGENIERÍA

DE SISTEMAS Y CONTROL

CURSO 2015-2016

# ESTUDIO APLICADO DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO PARA EL PROCESAMIENTO DE IMÁGENES DE CANCER DE MAMA

TRABAJO DE FIN DE MASTER

Alumno: Alfonso Areitio Pachón

Director: Sebastián Dormido Canto

# Agradecimientos

A mí querida Laura por su apoyo sin condiciones.

A mi amigo Jose por sus inestimables consejos e inyecciones de ánimo, en definitiva por su demostrada amistad

A mi director de proyecto, Sebastian Dormido, por darme la oportunidad de realizar este trabajo, por su inestimable ayuda y comprensión.

# Resumen

El cáncer de mama entre las mujeres es el tumor maligno más frecuente después de los canceres de piel no melanomas a nivel mundial, tanto en países desarrollados como los que están en vías de desarrollo.

En el año 2008 se diagnosticaron 1.380.000 nuevos casos en el mundo y en España 26.000 casos anuales, es decir el 30 % de los tumores del sexo femenino en nuestro país [1].

El número de casos y las tasas de incidencias aumentan en el mundo debido muy probablemente a un mayor envejecimiento de la población y a un diagnostico que cada vez se produce con mayor precocidad. Una de cada 8 mujeres padecerá cáncer de mama a lo largo de la vida.

El cáncer de mama es una de las pocas enfermedades cancerígenas que se pueden diagnosticar antes de que aparezcan los síntomas. Un diagnóstico temprano y acertado sobre este tipo de cáncer es de máxima importancia para proporcionar al paciente un tratamiento adecuado. Gracias a la investigación, ocho de cada 10 mujeres superan con éxito el cáncer de mama [14].

Este trabajo trata de revisar las técnicas más evolucionadas en redes de aprendizaje automático y sobre todo de las redes neuronales que existen en la actualidad, particularizando sobre el tratamiento de imágenes y su aplicación al campo del cáncer de mama basándose en el análisis de las mamografías realizadas para la prevención.

Para ello y para comprender el problema, primero haremos una introducción y descripción del cáncer, su clasificación y diagnóstico. Haremos un repaso de las técnicas de inteligencia computacional y su aplicación en el campo de la detección del cáncer de mama, y como resultado, un breve estado del arte de las técnicas que se están aplicando actualmente, trataremos de sacar conclusiones comparando las técnicas más utilizadas para este fin en este

momento, si bien es cierto que el estudio es limitado debido al reducido número de datos disponibles.

El objetivo final es apoyar la investigación en este campo y aportar un nuevo ensayo que pueda arrojar luz y dar nueva utilidad sobre el análisis y toma de decisiones en el diagnóstico del cáncer de mama en particular, de otras enfermedades y en múltiples campos que aún están por explorar.

# Índice General

*Agradecimientos* ..... *3*

*Resumen* ..... *4*

*Índice General* ..... *6*

*Índice de Figuras* ..... *9*

*Capítulo 1 Introducción y objetivos* ..... *18*

**1.1 Introducción** ..... **19**

1.1.1 ¿Qué es el cáncer? ..... 20

1.1.2 Anatomía de la mama ..... 22

1.1.3 Cáncer de mama. ..... 27

1.1.3.1 Incidencia del cáncer de mama ..... 28

1.1.3.2 Síntomas ..... 33

1.1.3.3 Clasificación ..... 33

1.1.3.4 Interés en los Tipos de micro calcificaciones ..... 34

1.1.3.5 Diagnostico ..... 43

1.1.3.6 Avances ..... 46

**1.2 Objetivos** ..... **47**

**1.3 Organización de la memoria.** ..... **51**

*Capítulo 2 Estado del arte* ..... *52*

**2.1 Historia y antecedentes** ..... **52**

2.1.1 Cronología de la mamografía ..... 52

2.1.2 Cronología de los Sistemas CAD ..... 55

2.1.3 Minería de Datos ..... 57

**2.2 Sistemas CAD aplicados a la mamografía** ..... **59**

2.2.1 Digitalización ..... 62

2.2.2 Preprocesamiento. ..... 65

2.2.3 Segmentación. ..... 72

2.2.4 Extracción de características y selección ..... 74

2.2.5 Clasificación ..... 81

2.2.6 Técnicas CAD aplicadas al problema de las microcalcificaciones ..... 81

**2.3 Técnicas utilizadas en inteligencia computacional ..... 89**
2.3.1 Cronología e Historia ..... 96
2.3.2 Clasificación: ..... 99
2.3.2.1 Tipos de aprendizaje ..... 99
2.3.2.2 Tipos de modelos ..... 100
2.3.2.3 Algoritmos y técnicas. ..... 102
2.3.2.3.1 Modelos tradicionales ..... 123
2.3.2.3.2 Método de ensamble: modelos promediados, modelos aditivos, modelos jerárquicos ..... 126
2.3.2.3.3 Métodos de kernel ..... 131
2.3.2.3.4 Redes Neuronales (RN). ..... 154
2.3.2.3.4.1 Historia y conceptos básicos ..... 154
2.3.2.3.4.2 Redes neuronales artificiales (RNA) ..... 165
2.3.2.3.4.2.1 Primera ola en redes neuronales artificiales. Perceptron y Adaline. ..... 177
2.3.2.3.4.2.2 Segunda ola: Redes de dos capas, back-propagation (BPN). ..... 188
2.3.2.3.4.2.3 Tercera Ola: Redes profundas, redes convolucionales. Pre-entrenamiento no supervisado. Dropout. Aplicaciones. ..... 201

**2.4 Bases de datos. ..... 219**
2.4.1 Magic-5 ..... 221
2.4.2 Mini-MIAS (Mini Mammographic Image Analysis Society) ..... 221
2.4.3 DDSM (Digital Database For Screening Mammography) ..... 223
2.4.4 NDB (Nijmegen Digital Mammogram Database) ..... 229
2.4.5 LLNL (Lawrence Livermore National Laboratories Database) ..... 229
2.4.6 IRMA (Image Retrieval in Medical Applications) ..... 231
2.4.7 AMDI Indexed Atlas of Digital Mammograms ..... 232
2.4.8 B-SCREEN (Bayesian Decision Support in Medical Screening) ..... 233
2.4.9 MammoGrid ..... 233
2.4.10 Comparativa y Elección ..... 235

*Capitulo 3 Modelo Práctico. ..... 237*

**3.1 Dificultad del Modelo ..... 237**

**3.2 Descripción del Modelo ..... 240**

**3.3 Casos practicos MatConvNet ..... 266**

*Capítulo 4 Conclusión modelo convolucional. ..... 288*

*Capítulo 5 Nuevas líneas de investigación y aplicaciones practicas. ..... 290*

*Referencias ..... 294*

# Índice de Figuras

**Figura 1** Aumento de la esperanza de vida entre 1900 y 2012..... 19

**Figura 2** Esperanza de vida en los 10 paises con mayores cifras ..... 19

**Figura 3** Número estimado de casos por muerte de cáncer..... 22

**Figura 4** Anatomía de la glándula mamaria..... 23

**Figura 5** Ligamentos de Cooper..... 23

**Figura 6** Anatomía pectoral, tejido adiposo y glandular, prolongación axilar ..... 24

**Figura 7** Puntos: 1. Caja torácica 2. Músculos pectorales 3. Lóbulos 4. Superficie del pezón 5. Aureola 6.Conducto de leche 7. Tejido adiposo 8. Piel..... 25

**Figura 8** Anatomía de la glándula mamaría..... 26

**Figura 9** Mortalidad de las mujeres en el año 2012 por tipo de Tumor..... 29

**Figura 10** Mortalidad en España por tipo de tumor en el año 2012..... 30

**Figura 11** Incidencia, mortalidad y prevalencia a 5 años estimadas para el año 2012 en España ambos sexos..... 31

**Figura 12** Cáncer mas frecuente en España en 2012..... 32

**Figura 13** Prevalencia a 5 años del cáncer de mama en España..... 32

**Figura 14** Proceso de un diagnostico precoz [127] ..... 32

**Figura 15** Tipo de micrcalcificaciones, morfología y malignidad ..... 36

**Figura 16** Descriptores Bi-Rads de distribución de las microcalcificaciones ..... 37

**Figura 17** Calcificaciones benignas.Tipos..... 37

**Figura 18** Calcificaciones con sospecha media: amorfas, toscas heterogéneas..... 38

**Figura 19** Calcificaciones con sospecha alta: Pleomorficas, lineales..... 38

**Figura 20** Descriptores BI-RADS de morfología..... 39

**Figura 21** Descriptores BI-RADS en Nódulos..... 39

**Figura 22** Atributos de las microcalcificaciones malignas..... 40

**Figura 23** mamografías con varias formas agrupadas, de distintos tamaños y con densidad variable..... 42

**Figura 24** Mamografo..... 44

**Figura 25** Comparación de los resultados obtenidos con mamografía analógica y digital [102] 45

**Figura 26** Una visión general del método aplicado a Imágenes de mamografía:..... 58

**Figura 27** Fases del procesamiento digital de imágenes..... 61

**Figura 28** Etapas en el diseño de un sistema de ayuda al diagnóstico de cáncer por ordenador a partir de imágenes médicas según Chen..... 61

**Figura 29** Imagen de la curva característica de una película Maidment et al. Sem in Roent 2003 [81] [82]..... 63

**Figura 30** comparativa entre mamografía convencional y digital..... 65

**Figura 31** Preprocesamiento digital de imágenes. Fuente: Álvarez et al., 2006..... 66

**Figura 32** Proceso de reducción del área de trabajo. Imagen tomada de Álvarez et al., 2006. 66

**Figura 33** a) Sección extraida de una imagen original de una mamografía con tejido graso. b) Histograma correspondiente a la sección "a)". c) Ecualización aplicada a la sección “a)” d) Histograma de la sección " c)”..... 69

**Figura 34** a,d). Sección extraida de la imagen original de dos mamografías con tejido graso "a)" y denso "d)". b,e). Realce del contraste por Bsplines en eje X. c,f). Realce del contraste por Bsplines en eje Y ..... 70

**Figura 35** Idea intuitiva de envolvente convexa de 17 puntos en el plano. ..... 76

**Figura 36** Un algoritmo para crear la envolvente convexa es por ejemplo Quick Hull [93]: ..... 77

**Figura 37** Resultado de la ejecución de algoritmo Quick Hull [94]: ..... 77

**Figura 38** Ilustración del proceso para detectar microcalcificaciones, (a) muestra la imagen después del paso I, (b) se aplica el umbral a la imagen previa y (c) se muestra un acercamiento a la región segmentada. ..... 78

**Figura 39** En estas imágenes se ilustran como se delimita un agrupamiento de microcalcificaciones, en (a) determinación de la región de interés y en (b) la región queda encerrada aplicando el método Quick Hull ..... 78

**Figura 40** En a) se muestra sobre una imagen de dimensiones 10x10 pixeles, una ventana móvil de 5x5 en dos posiciones y el pixel central que recibe el resultado. En b) los pixeles que reciben el resultado y las 2 filas y las 2 columnas del borde de la imagen que reciben el valor de textura calculados en los pixel cercanos. ..... 80

**Figura 41** Ejemplos de asimetría que se puede apreciar en la proyección MLO izquierda ..... 82

**Figura 42** Falso negativo, mamografías. a) Calcificaciones centrales previas a punción en mama izquierda. b) Control nueve meses después del procedimiento, se agregan calcificaciones y densidad del área biopsiada. ..... 85

**Figura 43** Estructuras de Wiston ..... 98

**Figura 44** Aprendizaje inductivo: métodos ..... 99

**Figura 45** Técnicas de aprendizaje supervisado ..... 105

**Figura 46** Aprendizaje inductivo no supervisado ..... 106

**Figura 47** Error de clasificación en espacio de instancias X .....108

**Figura 48** Sobre Ajuste.....109

**Figura 49** Sub Ajuste.....110

**Figura 50** Metodo K-fold CV. Una medida razonable es usar K=10 repitiendo el proceso .....110

**Figura 51** Resultados de los clasificadores para el conjunto de 23 atributos .....112

**Figura 52** Resultados de métricas de complejidad presentadas por el estudio realizado de la Salle.....113

**Figura 53** Diagrama con información general de las funcionalidades de Estudio de aprendizaje automático[106] .....114

**Figura 54** Clasificación hoja de referencia rápida de algoritmos de aprendizaje automático de Estudio de aprendizaje automático de Microsoft Azure. .....115

**Figura 55** comparativa algoritmos de aprendizaje automático .....119

**Figura 56** Clasificación de Aprendizaje automático con MATLAB .....119

**Figura 57** Entropía, efectividad y ganancia de información de un atributo.....124

**Figura 58** Algoritmo K-Vecinos.....125

**Figura 59** Método de enamble .....126

**Figura 60** Método Bagging regresión .....128

**Figura 61** Algoritmo AdaBoost .....130

**Figura 62** Figura de Cascada de clasificadores [137].....131

**Figura 63** Acontecimientos que han marcado la historia de las SVM.....136

**Figura 64** análisis de imágenes digitales que aplican la técnica SVM para su tratamiento .....137

**Figura 65** Vectores de soporte, determinación del hiperplano de margen máximo .....139

**Figura 66** Clasificación Vectores Soporte. La imagen de la izda muestra un caso separable linealmente . A la derecha se muestra otro caso no separable. .....141

**Figura 67** Hiperplano de Margen máximo .....141

**Figura 68** Restriccion de gradiente .....142

**Figura 69** Encontrar la función que transforma a un espacio de dimensión muy superior. .....147

**Figura 70** SVR con margen blando se muestra la relación entre las variables de holgura,.....153

**Figura 71** Esquema básico de una neurona artificial .....155

**Figura 72** Cerebro humano Vs computador [164] .....156

**Figura 73** Morfología de una neurona [172] .....159

**Figura 74** tasa de disparo e impulso eléctrico en la neurona .....159

**Figura 75** Analogía entre neurona biológica y artificial [164] El número, la longitud cuyo rango oscila desde 5 hasta 135 micrómetros, la función, la polaridad o el medidor químico brindan un método morfológico para la clasificación de las neuronas. .....160

**Figura 76** características y clasificación de las neuronas .....161

**Figura 77** Imagen de Red Hopfield de cuatro nodos[166] .....164

**Figura 78** Neurona artificial con la descripción de la función de activación. .....167

**Figura 79** Función Umbral [175].....167

**Figura 80** Función Sigmoidea .....168

**Figura 81** Tangente Hiperbólica .....169

**Figura 82** Capas entrada, oculta y salida .....170

**Figura 83** Estructura jerárquica basado en una RNA [164] ..... 171

**Figura 84** Comparativa redes neuronales biológicas y artificiales. ..... 171

**Figura 85** Tipos de redes heterasociativas ..... 173

**Figura 86** Tipos de redes autoasociativas ..... 174

**Figura 87** Llamada top-down ..... 175

**Figura 88** Representaciones del Perceptron de McCulloch-Pitts ..... 178

**Figura 89** Perceptrón simple ..... 179

**Figura 90** a) Patrones separables linealmente b) patrones no separables linealmente ..... 181

**Figura 91** Algoritmo del perceptrón simple ..... 182

**Figura 92** Capacidades del Perceptrón multicapa con dos y tres capas y con una única neurona en la capa de salida [185] ..... 185

**Figura 93** Representaciones del Perceptrón multicapa [187] ..... 186

**Figura 94** Imagen de una red neuronal artificial (RNA) perceptrón multicapa (MLP) ..... 186

**Figura 95** ¿cómo elegir el número de capas? ..... 189

**Figura 96** Diagrama de la Red back-propagation ..... 190

**Figura 97** Dibujo creativo de red neuronal ..... 202

**Figura 98** Arquitectura de la red de neuronas que detecta elementos de una imagen ..... 205

**Figura 99** Arquitectura de una red convolucional [225] ..... 208

**Figura 100** Arquitectura de una red convolucional [226] ..... 209

**Figura 101** Autoencoder ..... 211

**Figura 102** Unidades ocultas han aprendido a detectar bordes en diferentes posiciones .....212

**Figura 103** Gráfico de patrones de las microcalcificaciones realizado por neuronas de la capa oculta.....213

**Figura 104** Auto codificadores apilados.....213

**Figura 105** Convolución de una ventana de 2x2.....215

**Figura 106** Pooling de una ventana de 2x2 con cálculo del promedio .....216

**Figura 107** Ejemplo del caso 1252 de la BBDD DDSM. Imagenes.....228

**Figura 108** Ejemplo del caso 1252 de la BBDD DDSM. Información contextual.....229

**Figura 109** Caracteristicas mostradas por Matlab de la GPU.....238

**Figura 110** caracteristicas de la GPU NVIDIA GeForce GTX 860M.....239

**Figura 111** Aplicación de la herramienta de detección PFCM .....241

**Figura 112** Etapa de preprocesado .....242

**Figura 113** Mejora de contraste por acción de la transformada de Tophat [221] .....243

**Figura 114** Efecto de tophat.....244

**Figura 115** Sintaxis comando para CLAHE, adapthisteq.....246

**Figura 116** Imagen ecualizada y normalizada.....246

**Figura 117** Softmax Transfer Function .....247

**Figura 118** Corte manual de una imagen .....252

**Figura 119** Extracto de TophatprocesadoCNNnatconvnet390 y vista del espacio de trabajo del script .....252

**Figura 120** diferentes configuraciones de preporcesado utilizadas en el módelo .....253

**Figura 121** vista parcial de los directorios que contienen las imágenes preprocesadas con cáncer y benignas .....254

**Figura 122** Imágenes preprocesadas aplicando Tophat y CLAHE.....255

**Figura 123** Imdb390pre de tipo struct.....255

**Figura 124** Varias configuraciones probadas en distintas redes de nuestro modelo.....256

**Figura 125** *Caso* 1 30 iteraciones con ReLU en todas las capas y con Dropout y con 32 filtros de partida. .....267

**Figura 126** *Caso* 2 : 30 iteraciones con ReLU en todas las capas y con Dropout con 64 filtros .....269

**Figura 127** *Caso* 3: con las mimas condiciones probamos con 25 iteraciones con ReLU en todas las capas y con Dropout .....269

**Figura 128** *Caso* 4 con ReLU en todas las capas y con Dropout disminuyendo las ventanas de convolución.....270

**Figura 129** *Caso* 5 con los mismos datos que *caso* 4 con ReLU en todas las capas y con Dropout con 25 iteraciones.....271

**Figura 130** *Caso* 6 variando a 35 iteraciones con ReLU en todas las capas y con Dropout ...271

**Figura 131** *Caso* 7 40 iteraciones con ReLU en todas las capas y con Dropout y 16 filtros de partida.....273

**Figura 132** *Caso* 8 con ReLU en todas las capas y con Dropout alterando tamaño de las ventanas convolucionales y filtros de salida. .....275

**Figura 133** *Caso* 9 con ReLU en todas las capas , con Dropout y 32 filtros salida en primera capa.....277

**Figura 134** *Caso* 10 con ReLU en todas las capas , sin Dropout y 32 filtros salida en primera capa.....278

**Figura 135** *Caso* 11 sin Relu en capas intermedias, sin Dropout y 32 filtros salida en primera capa.....279

**Figura 136** *Caso* 12 sin Relu en capas intermedias , con Dropout y 32 filtros salida en primera capa.....280

**Figura 137** *Caso* 13 con 80 iteraciones y tophat 7:.....282

**Figura 138** *Caso* 14 con 27 iteraciones y tophat 7.....283

**Figura 139** *Caso* 15 con 27 iteraciones y tophat radio 8 en vez de 7.....284

**Figura 140** *Caso* 16 con 27 iteraciones y tophat radio 6.....285

**Figura 141** Imágenes tratadas con preprocesado Tophat de 3.....286

**Figura 142** Imágenes tratadas con preprocesado Tophat de 5.....287

**Figura 143** Resultado de red Matconvnet.....288

**Figura 144** Caracteristicas GPU TESLA.....291

# Capítulo 1 Introducción y objetivos

El principal objetivo de este trabajo fin de master es analizar las diversas técnicas de redes neuronales aplicadas a la clasificación de pacientes con cáncer de mama. Para ello es necesario conocer el problema y su dimensión haciéndose necesaria una introducción en la materia para conocer en que consiste el cáncer y más concretamente el cáncer de mama, conocer porque es tan importante avanzar en el estudio del diagnóstico temprano, describiremos en este capítulo su clasificación, los tipos de diagnóstico y finalmente los avances médicos y técnicos.

En nuestro trabajo nos basaremos en el análisis CAD de las imágenes de las mamografías realizadas para el diagnóstico de esta patología.

Nos apoyaremos en otros estudios realizados anteriormente para conocer que avances se han producido hasta la fecha, haremos una breve descripción de las bases de datos más importantes a nivel mundial que han sido creadas al efecto del estudio de esta patología y finalmente modelaremos y aplicaremos una las técnicas de redes neuronales actuales más importantes en este campo.

## 1.1 Introducción

La **esperanza de vida** en los últimos decenios ha crecido mucho en el mundo. Las personas nacidas en el año 2012 tendrán por término medio 72,7 si fuera niña y 68,1 en caso de ser niño es decir 6 años más que los nacidos en 1900 (OMS 2014) [2].

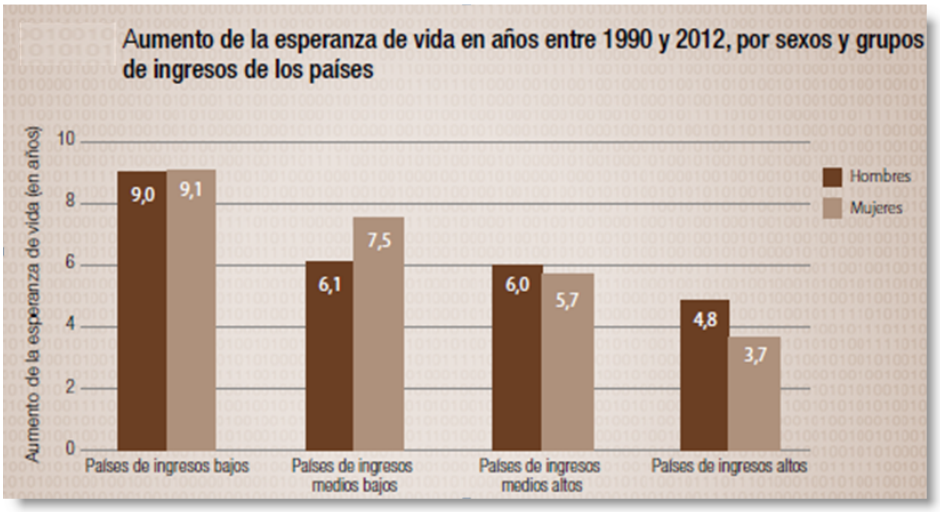



Figura 1 Aumento de la esperanza de vida entre 1900 y 2012

Esperanza de vida al nacer en hombres y mujeres en 2012 en los 10 países con mayores cifras

| Hombres | | | Mujeres | | |
|---|---|---|---|---|---|
| Orden | País | Esperanza de vida | Orden | País | Esperanza de vida |
| 1 | Islandia | 81,2 | 1 | Japón | 87,0 |
| 2 | Suiza | 80,7 | 2 | España | 85,1 |
| 3 | Australia | 80,5 | 3 | Suiza | 85,1 |
| 4 | Israel | 80,2 | 4 | Singapur | 85,1 |
| 5 | Singapur | 80,2 | 5 | Italia | 85,0 |
| 6 | Nueva Zelandia | 80,2 | 6 | Francia | 84,9 |
| 7 | Italia | 80,2 | 7 | Australia | 84,6 |
| 8 | Japón | 80,0 | 8 | República de Corea | 84,6 |
| 9 | Suecia | 80,0 | 9 | Luxemburgo | 84,1 |
| 10 | Luxemburgo | 79,7 | 10 | Portugal | 84,0 |

Los países con menos de 250 000 habitantes se han omitido debido a la incertidumbre de las estimaciones de la esperanza de vida.

Figura 2 Esperanza de vida en los 10 paises con mayores cifras

A pesar de esta buena noticia y de acuerdo con la **Sociedad Americana Contra el Cáncer**, actualmente mueren trece de cada cien personas por causa de enfermedades de tipo cancerígeno y esta puede afectar a personas de cualquier edad, fetos incluidos. El cáncer de mama es el que se presenta con mayor frecuencia en las mujeres y un **diagnostico precoz** aumentara la probabilidad de supervivencia siendo la piedra angular de la lucha contra este cáncer.

### 1.1.1 ¿Qué es el cáncer?

**El cáncer** es el nombre común que recibe un conjunto de enfermedades relacionadas en las que se observa un proceso descontrolado en la división de las células del cuerpo.[3]

**Etimológicamente**, el termino cáncer viene derivado del Latín y al igual que la palabra griega Karkinoma el significado es cangrejo. Se cree que la primera persona en utilizar el término carcinos fue el medico **Hipócrates** ya que relacionó como crecían las células con la forma que tiene el cangrejo.[1]

Esta enfermedad puede iniciarse de forma focalizada y extenderse a tejidos circundantes conduciendo a la muerte si el paciente no es tratado adecuadamente. Para entenderlo , se produce una aumento de la cantidad células que denominaremos tumor o neoplasia y una capacidad para colonizar y proliferar otros tejidos u órganos, proceso conocido como **metástasis**, enfermedad diseminada, enfermedad a distancia o enfermedad secundaria[4].

Al generarse un tumor en cualquier parte del organismo se vasculariza para alimentarse, crecer y subsistir (proceso de angiogénesis). Mediante este proceso, las células que son cancerígenas pasan al torrente sanguíneo viajando a través de la sangre y vasos linfáticos, expandiéndose hacia el resto del cuerpo asentándose en otros órganos si las células cancerígenas son capaces de traspasar los capilares. Si evolucionan adaptándose a este nuevo medio, se reproducen y entonces vuelven a producirse nuevos tumores (tumores secundarios).

El 90% de los tumores se generan por células epiteliales y los denominamos **carcinomas**. Los que se generan por células del Tejido conectivo o muscular son **sarcomas**. Si su procedencia son células sanguíneas entonces hablamos de **leucemia, mielomas y linfomas** y por último si es derivado del sistema nervioso son **neuroblastomas y gliomas**.

Dependiendo de si el cáncer está o no diseminado hay más o menos probabilidades de curación y tiene importantes consecuencias. Los eruditos en la materia estiman que aproximadamente el 90% de los fallecimientos se producen a causa de la enfermedad diseminada.

Si no hay metástasis entonces existe la posibilidad de aplicar tratamiento quirúrgico extirpando el tumor y en este caso se ofrece mejor pronóstico en la curación. Además se puede complementar con tratamiento de quimioterapia y radioterapia.

Si hay metástasis, no existe la opción de la cirugía y la terapia se basará en el uso de quimioterapia y radioterapia. Este tratamiento tendrá como objetivo aumentar la calidad de vida del paciente y su supervivencia, es de carácter paliativo Los tipos más habituales de **cáncer son el de piel, mama, pulmón y colorectal**. [5]

Como dato relevante, **en España** con una población de 46.771.000 el número personas afectadas en 2012 por cáncer fue de 215.534 [15]

La mortalidad debida al cáncer, se traduce en que en el año 2012 fallecieron 102.762 personas diferenciadas entre 63.579 casos en Hombres y 39.183 en las mujeres. En el 2020 según la previsión de naciones unidas aumentaran los casos en ambos sexos en un total de 15.097 hasta alcanzar la cifra de 117.859(73.424 hombres y 44.435 mujeres).

| Año | Número estimado de casos de muerte por cáncer | Hombre | Mujer | Ambos sexos |
|---|---|---|---|---|
| 2012 | | 63579 | 39183 | 102762 |
| | < 65 años | 16679 | 9995 | 26674 |
| | > = 65 años | 46900 | 29188 | 76088 |
| 2015 | | 67129 | 41261 | 108390 |
| | < 65 años | 17611 | 10399 | 28010 |
| | > = 65 años | 49518 | 30862 | 80380 |
| | Cambio demográfico | 3550 | 2078 | 5628 |
| | < 65 años | 932 | 404 | 1336 |
| | > = 65 años | 2618 | 1674 | 4292 |

ESPAÑA. Todos los cánceres excluyendo cáncer de piel no melanoma
Fuente: Ferlay J, Soerjomataram I, Ervik M, Dikshit R, Eser S, Mathers C, Rebelo M, Parkin DM, Forman D, Bray, F. GLOBOCAN 2012 v1.0, Cancer Incidence and Mortality Worldwide: IARC.

Figura 3 Número estimado de casos por muerte de cáncer

### 1.1.2 Anatomía de la mama

Los mamíferos alimentan a sus crías secretando leche de la glándula mamaria. Esta glándula es de origen ectodérmico.

Tres **tipos de tejidos** forman la glándula mamaria, glandular de tipo túbulo-alveolar, conjuntivo que conecta los lóbulos y adiposo que ocupa los espacios interlobulares.

La glándula mamaria femenina, anatómicamente se halla en la estructura llamada mama. Todas las especies comparten prácticamente la **histología de la glándula mamaria** y se podría decir que es prácticamente la misma:

Un **parénquima glandular** y un **estroma de soporte**. El parénquima se compone de alveolos y ductos que son conjuntos de estructuras en forma de tubo y huecas que ramifican en forma de árbol, cuyas luces confluyen progresivamente en canalículos cada vez más gruesos hasta terminar en uno de los doce a dieciocho vértices llamados **galactóforos**[12]; Cada unidad de secreción (célula alveolar) produce leche completa que está formada por las proteínas, grasas, hidratos de carbono, sales minerales, anticuerpos y el agua. Durante el embarazo y sobre todo en la lactancia, se produce el mayor desarrollo de la glándula mamaria. En el sexo masculino, también está presente la glándula mamaria pero es rudimentaria.

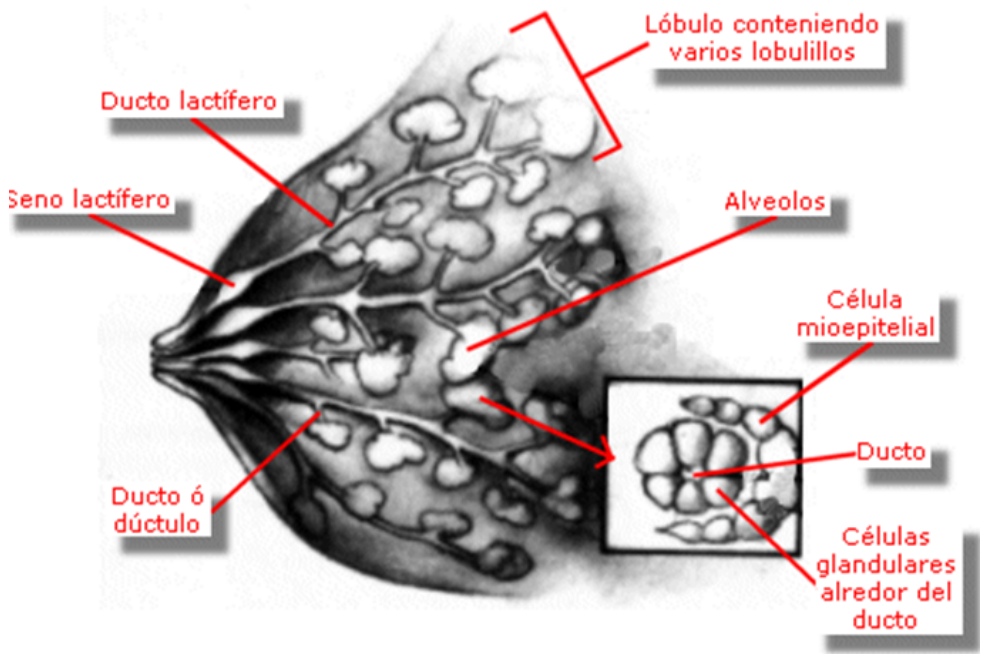



Figura 4 Anatomía de la glándula mamaria

Rodeando la glándula se encuentra el tejido celular subcutáneo, desde éste se dirigen hacia el interior buena cantidad de tabiques que constituyen los ligamentos suspensorios de la mama o **ligamentos de Cooper**.

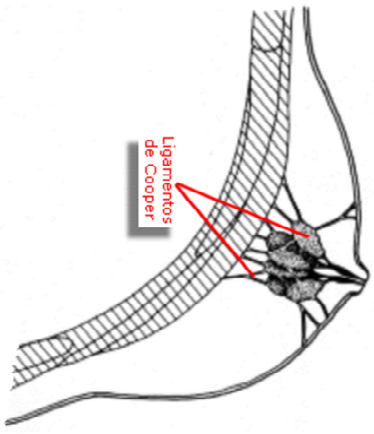



Figura 5 Ligamentos de Cooper

El tejido adiposo y glandular conforma la **masa de la mama**. Su extensión comprende desde el borde externo del esternón hasta la línea axilar media. El área súper externa de cada glándula se prolonga hacia la axila ("prolongación axilar").

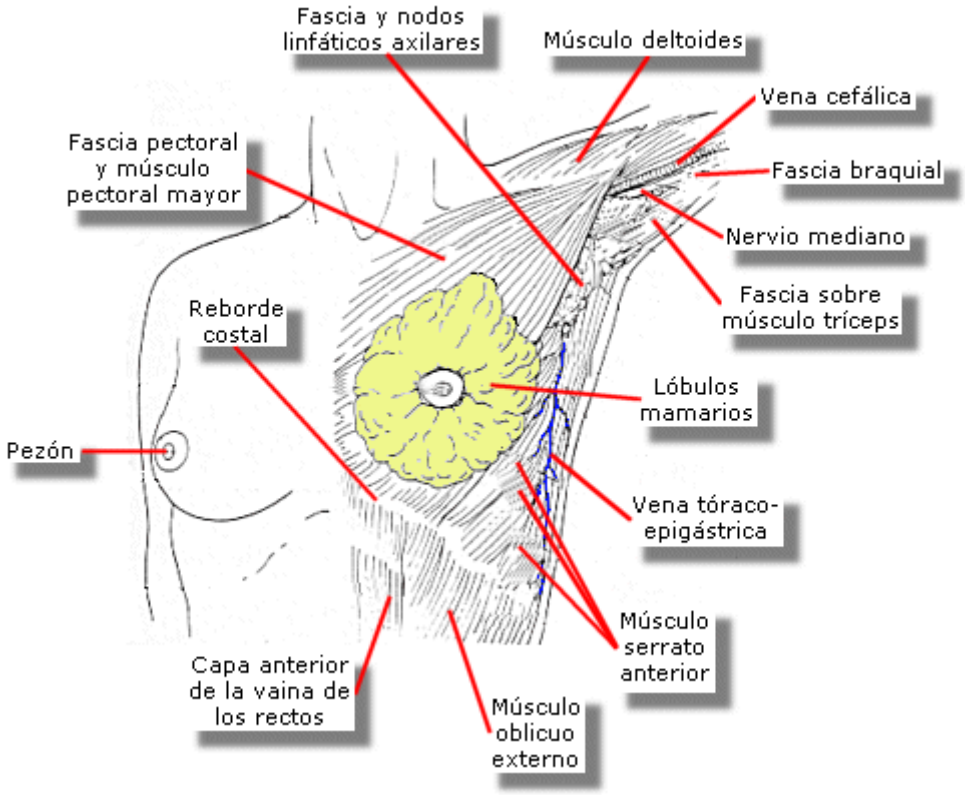



Figura 6 Anatomía pectoral, tejido adiposo y glandular, prolongación axilar

La superficie de la mama está recubierta por la piel. En el centro aproximadamente se halla el **pezón** cuya base está rodeada por una zona de piel híper pigmentada de dos centímetros y medio de tamaño que se denomina **areola**. El pezón contiene fibras musculares que facilitan la erección del mismo

La areola que es la piel coloreada que rodea el pezón que es más áspera que el resto. El tamaño medio ronda entre los 15-40 milímetros de diámetro y contiene glándulas sebáceas, entre ellas encontramos las denominadas **glándulas de Montgomery** que producen secreción grasa que lubrica la areola y el pezón.

A continuación se muestra la sección representativa normal de la anatomía del pecho con las flechas numeradas para una leyenda. [11]

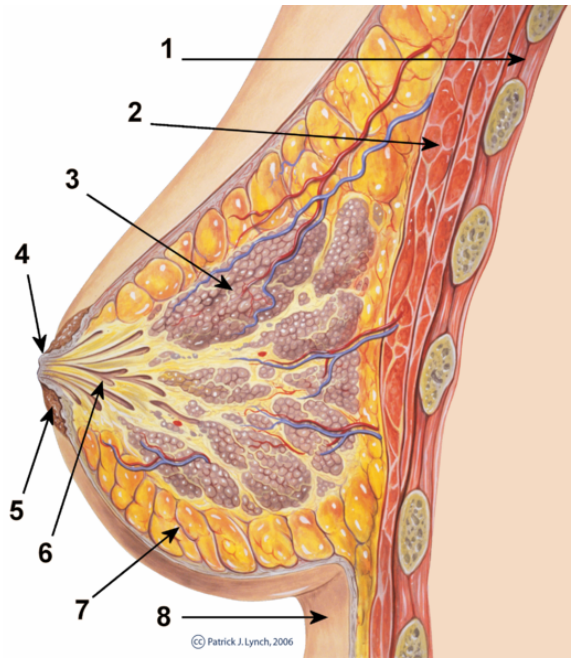



Figura 7 Puntos: 1. Caja torácica 2. Músculos pectorales 3. Lóbulos 4. Superficie del pezón 5. Aureola 6.Conducto de leche 7. Tejido adiposo 8. Piel

Observando un pezón adulto se pueden localizar entre 15 y 20 pequeñas aberturas (**conductos lactíferos**) dispuestas alrededor suyo que suministran leche mientras dura la lactancia.

La glándula mamaria es influenciada por la edad; por ejemplo tiene diferente estructura en pubertad y menopausia; también por su función de lactancia o estado de gestación de la mujer.

Salvo excepciones, que no dejan de ser patológicas, en el hombre se atrofia completamente.

En general, la glándula mamaria contiene **lóbulos mamarios** y estos a su vez contienen un máximo de 100 **acinos**. Los lóbulos mamarios excretan abriendo en el pezón a través del conducto lactífero. A su vez los lóbulos mamarios rodeados de células mioepiteliales y capilares sanguíneos están constituidos por **lobulillos** con conexión entre ellos por vasos

sanguíneos, tejido conectivo y conductos lactíferos. Los acinos están formados por células cuya misión es producir leche.

El **proceso de eyección** del lácteo es el siguiente:

El acino vacía el producto lácteo en un conducto terminal que funciona como los afluentes de los ríos. Varios de ellos confluyen en el conducto biliar y a su vez una agrupación de estos forman uno que denominamos **interlobulillar**. Agrupando de nuevo los interlobulillares, se conforma el conducto lobular que previo a la llegada a la areola se ensancha para finalmente acabar desembocando en el pezón. [13]

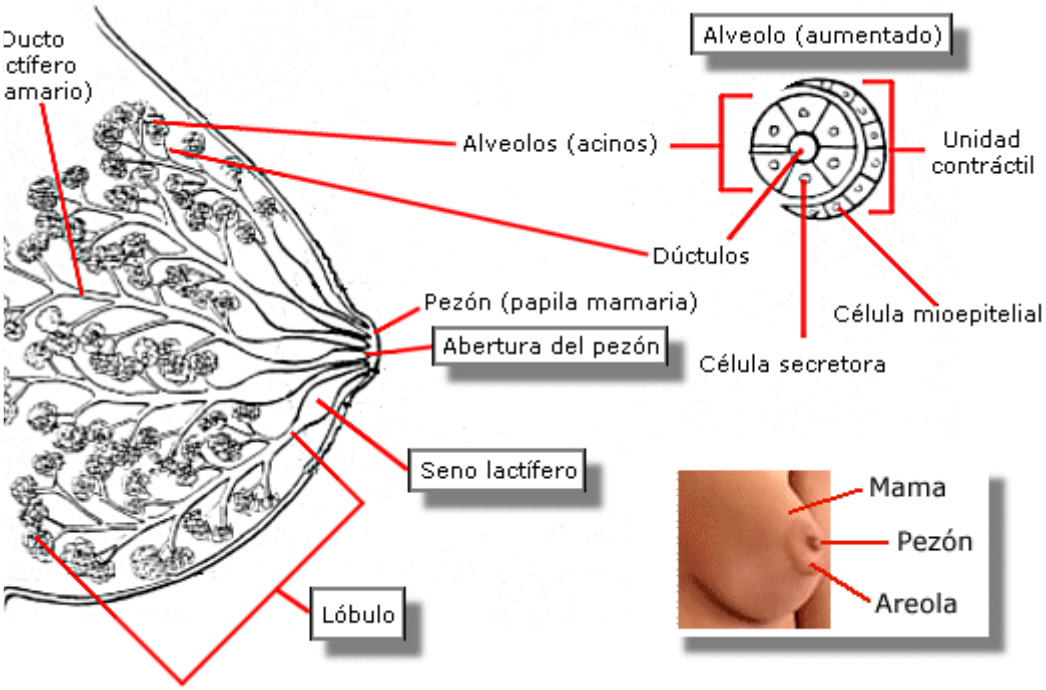



Figura 8 Anatomía de la glándula mamaría

### 1.1.3 Cáncer de mama.

## Introducción.

Se conoce descripción desde el año 1600 A.C aproximadamente. Es la neoplasia maligna que se presenta con mayor frecuencia en las mujeres independientemente del desarrollo del país en el que vivan. Del total de estos carcinomas, menos del 1 % ocurren en varones.

## Descripción

Es La enfermedad producida por la aparición de un tumor de carácter maligno en el tejido mamario.

El cáncer de mama se debe a una proliferación de las células epiteliales que revisten los conductos o lobulillos mamarios.

Las células, individualmente se clonan motivadas por una serie de mutaciones somáticas o de línea germinal y se descontrolan formándose un tumor cuyas células se han dividido de manera descontrolada y sin ningún tipo de orden. Si no se le detiene este comienza a invadir otros tejidos (**infiltración**) y a propagarse a otras zonas del cuerpo (**metástasis**).

El tumor que se ha generado, puede crecer y reproducirse de tres formas diferentes:

- **Crecimiento local**: crece por invasión directa, invadiendo otras estructuras vecinas y la piel.
- **Diseminación linfática**: a través de los vasos linfáticos de la mama a varios grupos ganglionares mediante drenaje de la linfa. Los ganglios con frecuencia más afectados son en la axila, la arteria mamaria interna y los ganglios supraclaviculares.
- **Diseminación hematógena**: En esta ocasión, se produce a través de los vasos sanguíneos y afecta piel, pulmón, huesos e hígado.

### 1.1.3.1 Incidencia del cáncer de mama

Definimos la incidencia como el número de casos nuevos que se presentan en un periodo de tiempo y de población.

En el cáncer de mama podemos definir como **factores de riesgo** importantes a tener en cuenta:

- El **riesgo** aumenta con la edad, también si el parto se produce a una edad avanzada, no haber alumbrado, una menstruación prematura, antecedentes en la familia (entre el cinco y el diez por ciento es causado por mutaciones genéticas que son heredadas), la raza (mayor riesgo en la blanca), consumir hormonas, consumir alcohol.
- Por tipo de cáncer de mama el **carcinoma ductal** infiltrante se da en el 80% de los casos y el **carcinoma lobulillar** infiltrante aparece con una frecuencia del 10 al 12 %.

## Incidencia Mundial

Se dan **1.7 millones de nuevos casos** de cáncer de mama al año en el mundo según estadísticas ofrecidas por World Cáncer Research Fund International en su sitio web, fuente del año 2012. Como cada año aumenta la esperanza de vida, la urbe y además se adoptan modos de vida occidentales, la incidencia de esta enfermedad se hace presente en mayor proporción. Lo que si podemos afirmar porque es un dato claro es que el éxito del tratamiento depende del nivel económico del país y del de su población ya que el desarrollo incide directamente como es lógico en el diagnostico precoz y su tratamiento por lo que naturalmente, estas variables afectan directamente a tasa de mortalidad de las ciudadanas de estos países que sufren este tipo de cáncer ya que el diagnostico se produce en fases más avanzadas. Observamos cifras inferiores al 40% de supervivencia en países de ingresos bajos [8] y alrededor del 80% o más en América del Norte, Suecia y Japón. Los países de ingresos medios rondan el 60%.

Las estrategias para prevenir y evitar el desarrollo de la enfermedad a fases más avanzadas pasa por el conocimiento de los primeros signos y síntomas, la exploración clínica, cribado mediante mamografía (cribado costoso) [9].

La incidencia de esta enfermedad, no es uniforme y varía mucho en el mundo con tasas del 99,4 por 100.000 en América del Norte. América del Sur, Europa oriental, África austral y Asia occidental. Siendo la más baja en países africanos.

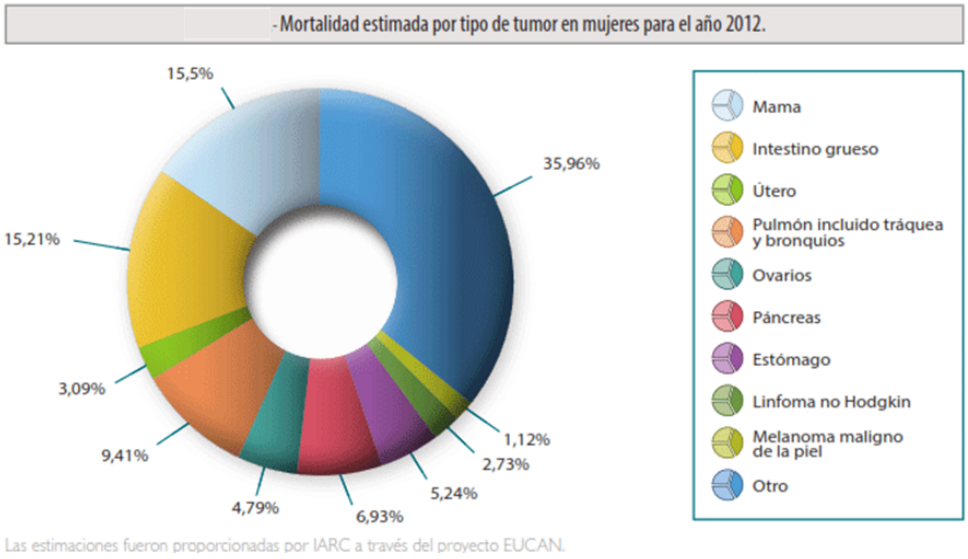



Figura 9 Mortalidad de las mujeres en el año 2012 por tipo de Tumor

El cáncer de mama es el cáncer de mayor prevalencia a 5 años a nivel mundial con un 19,2

Cuando el cáncer está localizado en la mama, la tasa de supervivencia es muy elevada; disminuyendo considerablemente si se ha diseminado a los ganglios linfáticos regionales y sobre todo si hay metástasis en otras partes del cuerpo.

Por ejemplo en estados unidos el porcentaje es del 99% en el primer caso, el 85 % en el segundo y de tan solo el 26% en el tercero en 5 años. [17].

## Incidencia en España

Respecto a la mortalidad de la población en España debido al cáncer, en la estimación de la mortalidad por tipo de tumor fue en primer lugar el cáncer de pulmón (21.118 muertes),

ocupando la segunda posición el cáncer de colon (14.700 muertes), seguidos a una gran distancia del cáncer de mama (6.075 muertes), cáncer de páncreas (5.720 muertes) y de próstata (5.481 muertes).

En 2012, 3 de cada 10 muertes en varones y 2 de cada 10 en mujeres en España fueron a causa del cáncer.

El cáncer es la segunda causa de muerte en mujeres y primera en hombres en nuestro país Y como veremos más adelante, el cáncer de mama es el tumor que más muertes produce en las mujeres Españolas

Sin embargo, de acuerdo con los datos del INE, la principal causa de muertes en los hospitales españoles fueron los tumores (24,2%) [6][7].

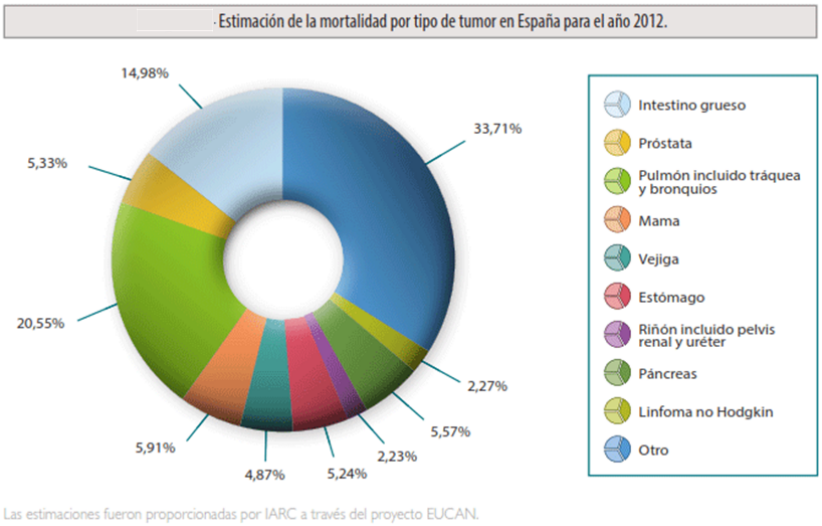



Figura 10 Mortalidad en España por tipo de tumor en el año 2012

En cualquier caso se demuestra que son cifras suficientemente preocupantes para que cualquier aportación que ayude a minorar las muertes por cáncer sea motivo de bienvenida.

| Cancer | Incidencia | | | Mortalidad | | | 5-year prevalencia | | |
|---|---|---|---|---|---|---|---|---|---|
| | Total | (%) | ASR | Total | (%) | ASR | Total | (%) | Prop. |
| Labios, cavidad oral | 4.098 | 1,9 | 4,7 | 1.117 | 1,1 | 1,2 | 11.811 | 2,0 | 29,8 |
| Nasofaringe | 350 | 0,2 | 0,5 | 188 | 0,2 | 0,2 | 909 | 0,2 | 2,3 |
| Faringe, otras localizaciones | 1.530 | 0,7 | 2,1 | 765 | 0,7 | 1,0 | 3.792 | 0,7 | 9,6 |
| Esófago | 2.090 | 1,0 | 2,5 | 1.728 | 1,7 | 1,9 | 2.238 | 0,4 | 5,7 |
| Estómago | 7.810 | 3,6 | 7,8 | 5.389 | 5,2 | 4,9 | 12.611 | 2,2 | 31,8 |
| Colorrecto | 32.240 | 15,0 | 33,1 | 14.700 | 14,3 | 12,3 | 89.705 | 15,4 | 226,3 |
| Hígado | 5.522 | 2,6 | 6,0 | 4.536 | 4,4 | 4,3 | 5.506 | 0,9 | 13,9 |
| Vesícula biliar | 2.002 | 0,9 | 1,7 | 1.174 | 1,1 | 0,9 | 1.905 | 0,3 | 4,8 |
| Páncreas | 6.367 | 3,0 | 6,3 | 5.720 | 5,6 | 5,5 | 3.437 | 0,6 | 8,7 |
| Laringe | 3.182 | 1,5 | 4,1 | 1.321 | 1,3 | 1,5 | 11.200 | 1,9 | 28,3 |
| Pulmón | 26.715 | 12,4 | 30,3 | 21.118 | 20,6 | 22,8 | 28.148 | 4,8 | 71,0 |
| Melanoma | 5.004 | 2,3 | 6,9 | 967 | 0,9 | 1,0 | 19.792 | 3,4 | 49,9 |
| Kaposi sarcoma | 316 | 0,1 | 0,5 | 24 | 0,0 | 0,0 | 880 | 0,2 | 2,2 |
| Mama | 25.215 | 11,7 | 67,3 | 6.075 | 5,9 | 11,9 | 104.210 | 17,9 | 516,2 |
| Cérvix | 2.511 | 1,2 | 7,8 | 848 | 0,8 | 2,1 | 8.867 | 1,5 | 43,9 |
| Cuerpo del Útero | 5.121 | 2,4 | 11,6 | 1.211 | 1,2 | 1,9 | 19.272 | 3,3 | 95,5 |
| Ovario | 3.236 | 1,5 | 7,7 | 1.878 | 1,8 | 3,7 | 7.925 | 1,4 | 39,3 |
| Próstata | 27.853 | 12,9 | 65,2 | 5.481 | 5,3 | 8,6 | 102.559 | 17,6 | 527,3 |
| Testículo | 823 | 0,4 | 3,5 | 42 | 0,0 | 0,1 | 3.242 | 0,6 | 16,7 |
| Riñón | 6.474 | 3,0 | 7,8 | 2.295 | 2,2 | 2,2 | 18.693 | 3,2 | 47,2 |
| Vejiga | 13.789 | 6,4 | 13,9 | 5.007 | 4,9 | 4,0 | 47.225 | 8,1 | 119,2 |
| Sistema nervioso | 3.717 | 1,7 | 5,1 | 2.668 | 2,6 | 3,3 | 3.187 | 0,5 | 8,0 |
| Tiroides | 2.059 | 1,0 | 3,4 | 286 | 0,3 | 0,3 | 8.097 | 1,4 | 20,4 |
| Linfoma de Hodgkin | 1.150 | 0,5 | 2,3 | 212 | 0,2 | 0,3 | 3.644 | 0,6 | 9,2 |
| Linfoma no Hodgkin | 6.130 | 2,8 | 7,5 | 2.337 | 2,3 | 2,2 | 16.342 | 2,8 | 41,2 |
| Mieloma múltiple | 2.420 | 1,1 | 2,3 | 1.675 | 1,6 | 1,4 | 5.730 | 1,0 | 14,5 |
| Leucemia | 5.190 | 2,4 | 6,5 | 3.212 | 3,1 | 3,0 | 11.040 | 1,9 | 27,9 |
| Todos los tumores (excl, tumores cutáneos no melanoma) | 215.534 | 100,0 | 249,1 | 102.762 | 100,0 | 98,1 | 581.688 | 100,0 | 1.467,6 |

Figura 11 Incidencia, mortalidad y prevalencia a 5 años estimadas para el año 2012 en España ambos sexos

En España son diagnosticados aproximadamente 25.000 nuevos cánceres de mama al año [10] A lo largo de su vida, de cada 8 mujeres una tendrá cáncer de mama.

Según se aprecia en la tabla, el de mama encabeza el ranking de los cinco tipos de canceres más frecuentes en España en las mujeres.

Cinco localizaciones de cáncer más frecuentes en España en 2012 (según total de nº de casos, siguiendo orden decreciente).

| | Hombre | Mujer | Ambos Sexos |
|---|---|---|---|
| 1º | Próstata | Mama | Colorrectal |
| 2º | Pulmón | Colorrectal | Próstata |
| 3º | Colorrectal | Cuerpo de Útero | Pulmón |
| 4º | Vejiga | Pulmón | Mama |
| 5º | Estómago | Ovario | Vejiga |

Fuente: Ferlay J, Soerjomataram I, Ervik M, Dikshit R, Eser S, Mathers C, Rebelo M, Parkin DM, Forman D, Bray, F. GLOBOCAN 2012 v1.0, Cancer Incidence and Mortality Worldwide: IARC.

Figura 12 Cáncer mas frecuente en España en 2012

También es el tipo de cáncer que más prevalencia a 5 años tiene por tipo y sexo con un 17,9% en España

| | Incidencia[1] | | | Mortalidad[1] | | | Prevalencia a 5 años[2] | | |
|---|---|---|---|---|---|---|---|---|---|
| Cáncer | Número | (%) | ASR (W) | Número | (%) | ASR (W) | Número | (%) | ASR (W) |
| Mama | 25215 | 11.7 | 67.3 | 6075 | 5.9 | 11.9 | 104210 | 17.9 | 516.2 |

Figura 13 Prevalencia a 5 años del cáncer de mama en España

Resumiendo podemos afirmar que es de vital importancia realizar un diagnóstico precoz para atacar la enfermedad en su estadio inicial.

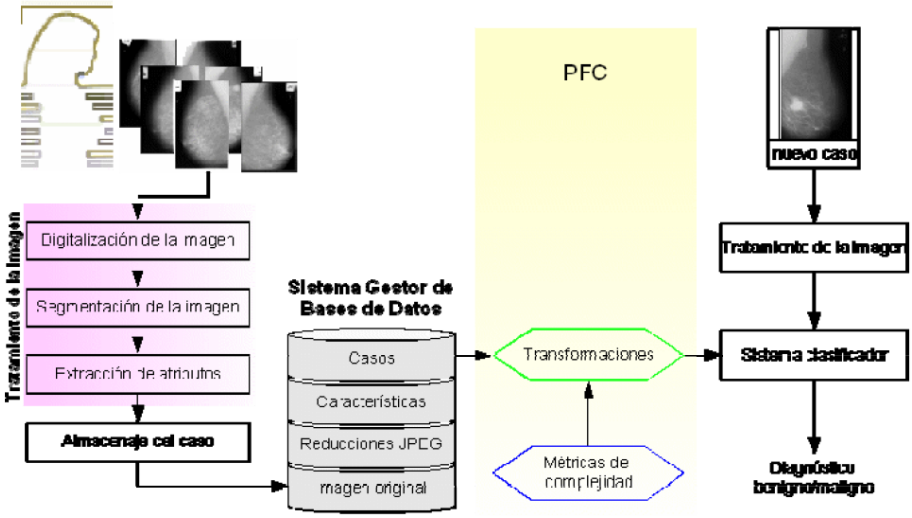



Figura 14 Proceso de un diagnostico precoz [127]

### 1.1.3.2 Síntomas

En las fases iniciales, la sintomatología no es apreciable y es difícil su detección. Si ocurre así es porque se realiza algún programa de cribado o mamografía en el cual participa la persona diagnosticada.

Cuando son apreciables, los más frecuentes en la mama son : Dolor a la palpación, nódulos, secreción repentina del pezón, hundimiento del pezón, llagas y otro cambios, alteración de la piel, ulceraciones, cambio de tamaño y de movilidad, nudo firme debajo del brazo.

### 1.1.3.3 Clasificación

La clasificación en estadio se va a basar principalmente en cuatro factores a tener en cuenta: Cual es el tamaño del tumor, si el cáncer que tenemos que clasificar es invasivo o no lo es, si ya ha afectado a los ganglios linfáticos, si hay metástasis en otras partes del cuerpo.

Vamos a describir los diferentes estadios por los que puede pasar y desarrollarse el cáncer de mama. Tiene 5 posibles etapas que vamos a numerar del 0 al IV. El 0 se corresponde con un cáncer que no ha invadido y el IV al cáncer que ha desembocado en metástasis y se ha propagado hacia otras partes del cuerpo.

- **Estadio 0**: son lesiones en las que las células del tumor se localizan únicamente en los lobulillos o en los conductos galactóforos. Se le denomina carcinoma in situ.
- **Estadio I:** a su vez consta de dos subestadíos:

  IA El tumor mide menos de dos centímetros y no ha afectado a los ganglios

  IB es invasivo pero con o sin tumor en la mama hay células cancerígenas inferiores a 2mm en los ganglios linfáticos o existe tumor.
- **Estadio II**: el tumor mide entre dos y cinco centímetros y puede afectar o no a los ganglios axilares.

- **Estadio III**: ha invadido el tumor los ganglios axilares, la piel o ambas y también la pared torácica.
- **Estadio IV**: afecta a otros órganos.

Dependiendo del estadio en el que se encuentre el cáncer, **la supervivencia** será mayor o menor en los siguientes 5 años. Prácticamente el 100% en el estadio I y el 20% en el IV.

Esta información se incluye en el informe patológico y previamente ha sido determinada durante la cirugía en la extirpación del tumor, el control de los ganglios linfáticos sobre todo en axila, análisis de sangre y estudios por diagnóstico de imagen como puede ser la mamografía, la ecografía y la resonancia magnética (RM).

También se utiliza el **sistema TGM** para clasificar el cáncer. T es la inicial de tamaño de tumor, G es, el grado de afectación de los ganglios linfáticos y M se refiere a si existe metástasis o propagación a otras partes corporales [18].

Las células malignas permiten una clasificación en **grados Histológicos**. Esta clasificación está basada en el grado de similitud entre las células cancerígenas y las sanas en la mama. Dependiendo del parecido se desarrollará con mayor o menor velocidad.

- **Grado 1** o bien diferenciadas: gran parecido y crecimiento lento.
- **Grado 2** o moderadamente diferenciadas: diferencias pequeñas y crecimiento más rápido que grado 1.
- **Grado 3** o indiferenciado: no se parecen y crecen rápidamente [19]

### 1.1.3.4 Interés en los Tipos de micro calcificaciones

La definición de calcificaciones en la mama se produce en mayor medida cuando las mujeres envejecen y se producen habitualmente por golpes o inflamaciones. Tienen un tamaño que no permiten detectarlas en la palpación. Las calcificaciones son en general benignas pero en

ocasiones indican cáncer. Podemos clasificarlas por su tamaño en macrocalcificaciones y microcalcificaciones.

Las que más nos interesan por su elevado porcentaje de cáncer cuando se muestran (un 25% aproximadamente) son las microcalcificaciones. Además por el estudio de la microcalcificaciones se detectan del 85 al 95% del **carcinoma ductal** in situ que a su vez se da con una frecuencia del 25-30% de los cánceres de mama. Estas se presentan en una cantidad mayor que las macrocalcificaciones y el radiólogo habrá de observar la distribución y por supuesto las formas para determinar su carácter. Se distinguen entre ellas por la **densidad, por su localización, agrupación, por su morfología, por su tamaño** menor de 1 mm.

La **composición química** es oxalato de calcio que se suele dar en las calcificaciones benignas o fosfato de calcio en las malignas.

Las malignas suelen estar agrupadas (más de 20 por cm2) y en formas lineales o en rama. La cantidad es importante ya que más de 30 en total es síntoma de malignidad.

Los tipos de calcificaciones para estudio de imagen son: **Microcalcificaciones, y Nódulos – masas**.

**Microcalcificaciones:**

En la mamografía aparecen como puntos diminutos y brillantes y cuando miden menos de 0,1mm.

Son muy difíciles de detectar. Con tamaño pequeño, el contraste es muy bajo. Según el sistema BI-RADS tiene 5 clasificaciones:

| Tipo | I | II | III | IV | V |
|---|---|---|---|---|---|
| Morfología | Microcalcificaciones anulares, redondeadas, de centro claro. | Microcalcificaciones puntiformes regulares, redondeadas. | Microcalcificaciones en polvo, muy finas, sin poder precisar su forma ni su número. | Microcalcificaciones puntiformes irregulares, son poliédricas, como un grano de sal | Microcalcificaciones vermiculares, son alargadas, como un árbol sin hojas |
| Malignidad % | 0 | 10 | 19 | 29 | 72 |

Figura 15 Tipo de micrcalcificaciones, morfología y malignidad

Por su **distribución:**

- **Agrupada**: mayor de cinco calcificaciones por cc en las 2 proyecciones. Se consideran benignas o malignas según la forma del grupo
- **Regional:** mayor a 2 CC. o presentes en más de un cuadrante. Suelen ser benignas
- **Difusa**: Las microcalcificaciones se encuentran dispersas a través de la mama. Suelen ser benignas
- **Segmentaria**: Calcio depositado en ductos en forma triangular. Apunta cáncer
- **Lineal**: el calcio se reparte linealmente aunque puede ramificarse. Sugiere cáncer

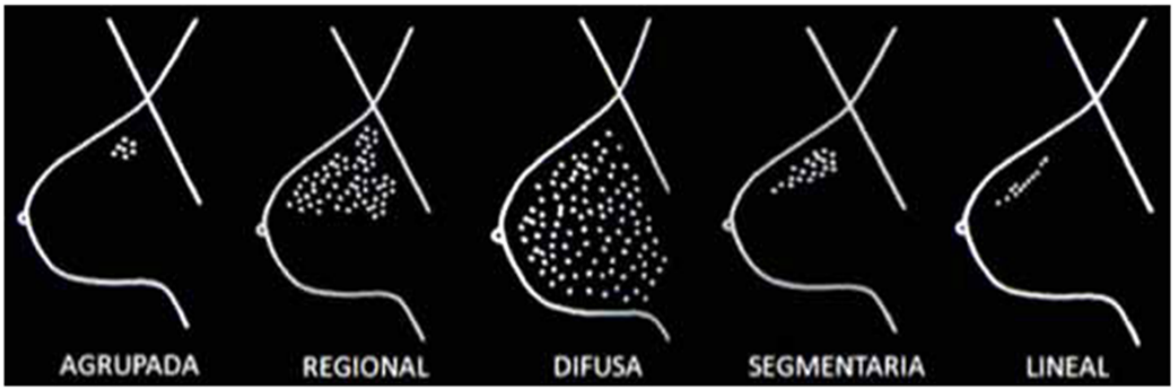



Figura 16 Descriptores Bi-Rads de distribución de las microcalcificaciones

Atendiendo a la **morfología:**

**Calcificaciones benignas**: vasculares, cutáneas, lechada de cal, secretoras, pop corn, distroficas, quistes oleosos, de sutura.

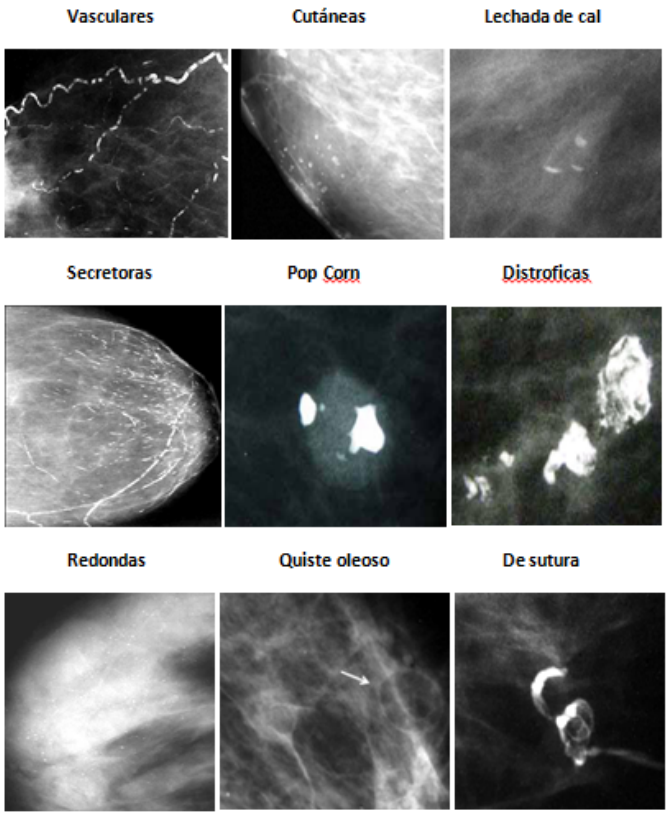



Figura 17 Calcificaciones benignas.Tipos

Amorfas Toscas heterogéneas

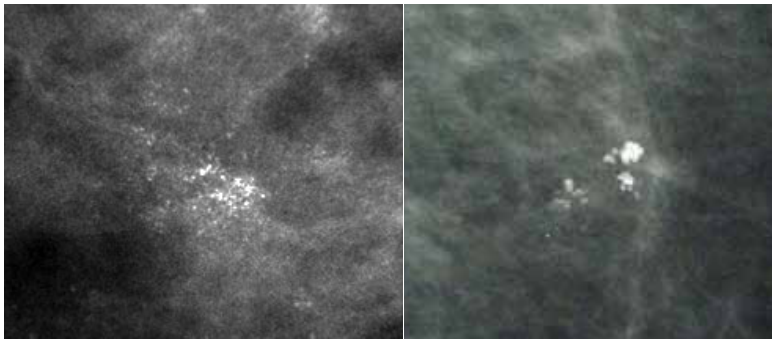


Figura 18 Calcificaciones con sospecha media: amorfas, toscas heterogéneas

**Calcificaciones con sospecha alta**: Pleomorficas, lineales

Pleomorficas Lineales.

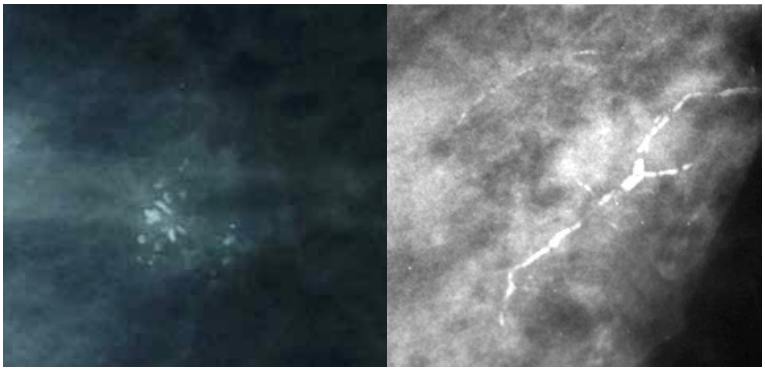


Figura 19 Calcificaciones con sospecha alta: Pleomorficas, lineales

Con las características antes mencionadas se realiza una clasificación en función del grado de **sospecha de malignidad**: se clasifican en BI-RADS por tipo de microcalcificaciones. Cuando hay mezcla de varias categorías se toma la de la mayor. La tabla de categorías es la siguiente:

**DESCRIPTOR BI-RADS**

| Categoría | Descriptor BI-RADS de morfología |
|---|---|
| BI-RADS 2 | Calcificaciones vasculares<br>Calcificaciones cutáneas<br>Calcificaciones en "lechada de cal"<br>Calcificaciones secretoras<br>Calcificaciones en "pop corn"<br>Calcificaciones redondas<br>Quistes oleosos<br>Calcificaciones de suturas |
| BI-RADS 3 | Calcificaciones redondas agrupadas |
| BI-RADS 4 | Calcificaciones amorfas<br>Calcificaciones toscas heterogéneas<br>Calcificaciones finas pleomórficas |
| BI-RADS 5 | Calcificaciones lineales / lineales ramificadas |

Figura 20 Descriptores BI-RADS de morfología

**Nódulos-Masas:** se pueden observar en dos proyecciones las masas son de tamaño superior a 2 cm y los nódulos son pequeños y se pueden aislar de aquello que le rodea. Tres características:

- **Forma**: margen indefinido o lobulado, maligno redondeado u oval , benigno
- **Contorno**: lisos y redondeado u ovoide son quistes o fibroadenomas
- **densidad** respecto al parénquima :Alta densidad , maligno y baja benigno

*DESCRIPTORES BI-RADS™ EN NÓDULOS*

| • *Forma* | • *Contorno* | • *Densidad respecto al parénquima* |
|---|---|---|
| – Redondeada **(R)** | – Bien definido **(D)** | – Superior **(+)** |
| – Oval **(O)** | – Oscurecido **(U)** | – Similar **(=)** |
| – Lobulada **(L)** | – Mal definido **(I)** | – Inferior **(–)** |
| – Irregular **(X)** | – Microlobulado **(M)** | – Con contenido graso **(O)** |
| | – Espiculado **(S)** | |

Figura 21 Descriptores BI-RADS en Nódulos

**Regiones de interés**: son regiones que se establecen sobre la mamografía de forma cuadrada. Se identifica con las coordenadas de su pixel central y para diferenciar un tumor del tejido sano la ventana debe ser ligeramente más pequeña que el tumor.

**Detección de lesiones**: Es primordial encontrar en un estadio o fase temprana, lesiones que permitan asegurar una estrategia que sea eficaz y así disminuir el riesgo de mortalidad asociada a la enfermedad del cáncer de mama.

El uruguayo **Raúl Leborgne** fue pionero en vincular cáncer de mama con microcalcificaciones. Según estadísticas cuyos porcentajes muestran una gran disparidad de cifras que oscilan entre el 20 y el 60%, las microcalcificaciones cuando se dan con una serie de características específicas, son el signo más evidente de cáncer.

Atendiendo a su malignidad:

| ATRIBUTO | Malignidad | |
|---|---|---|
| **Área** | MÍNIMO | Las microcalcificaciones pequeñas son más informativas en la diagnosis médica y acostumbran a ser más peligrosas. |
| **Aspereza** | MÁXIMO | Cuanto más rugosa y menos esférica, más sospechosa es. |
| **Compacidad** | MÁXIMO | Cuanto mayor es el valor, más indicios de malignidad. Una microcalcificación redonda tiene valor 1 y el índice se incrementa cuanto más alargada es. |
| **Elongación del feret** | MÁXIMO | Cuanto más alargada es la microcalcificación, más sospechosa desde un punto de vista médico. |
| **Número de agujeros** | MÁXIMO | Cuantos más agujeros presenta una microcalcificación, más sospechosa. |

Figura 22 Atributos de las microcalcificaciones malignas

Las microcalcificaciones se producen cuando se altera metabólicamente el parénquima.

**Stegner** [231] es de la opinión que las concreciones cálcicas se producen por degeneración de las células que se acumulan en su interior hasta que produce su estallido con consecuencias al diseminar en canalículos y conductos.

Entre las técnicas que se utilizan actualmente, la mamografía es una herramienta que juega un papel fundamental permitiendo la detección de carcinomas in situ que como vimos anteriormente están clasificadas en el estadio 0 y que con los tratamientos adecuados permiten una curación cercana al 100% evitando de esta manera que la enfermedad acabe desembocando en procesos más complejos, metástasis.

Mediante la mamografía podemos observar buenos indicadores como son las microcalcificaciones o masas tumorales. Estos nos indican que la enfermedad se hace presente pero su interpretación no deja de ser de una dificultad muy elevada en algunos casos, incluso para los expertos radiólogos. Esta se debe a la posible confusión entre tejidos además del diminuto tamaño de las anomalías que se presentan. Hay estudios que entre el 10% y el 25% de los tumores no son detectados por los radiólogos, es decir el porcentaje de aciertos no supera el 75% en la inspección visual.

Esta dificultad en el diagnostico supone en muchos casos la realización de pruebas suplementarias y biopsias. Para disminuir costes se hace necesario que el diagnostico sea preciso y en este caso las innovaciones en investigación y en tecnología juegan un papel fundamental.

Para mejorar el porcentaje en descubrimientos de tumores no detectados en una primera visualización realizada por un radiólogo en lectura simple, se utiliza la técnica recomendada de **doble lectura** que consiste en visualizar las mamografías por dos radiólogos.

**En España**, en el programa de cribado dependiendo de la comunidad que lo desarrolla es simple o doble.

**Lectura simple** en regiones como Aragón, Cantabria, Ceuta, Navarra, País Vasco y Murcia.

**Lectura doble** con o sin consenso:

- **Con consenso** se da cuando no hay acuerdo entre las interpretaciones de ambos radiólogos pero se reúnen y deciden el resultado entre ambos. Se aplica en Baleares, Canarias, Comunidad Valenciana, Extremadura, La Rioja, Madrid y Cataluña.
- **Sin consenso**, no se interpreta de manera acordada sino que la decisión pasa por asignar a la paciente el diagnostico peor, el que sería más perjudicial para la paciente. Se aplica en Andalucía y Galicia.

Hay otras comunidades como Asturias, Castilla-La Mancha y Castilla y León que utilizan un **sistema mixto** de tal forma que solo se realiza doble lectura en ciertos casos que son así son requeridos.[67]

Lógicamente la doble lectura tiene como inconveniente ser más costoso y generar una mayor carga de trabajo. Para evitar esta doble lectura entran en juego los sistemas CAD. Con ellos no es necesario que sea supervisado por el segundo radiólogo ya que esta función la realiza un sistema inteligente de proceso y clasificación de imágenes.

A continuación observamos imágenes de mamografías con varias formas agrupadas, de distintos tamaños y con densidad variable, Fueron observadas en mujeres que no presentaban sintomatología [20]

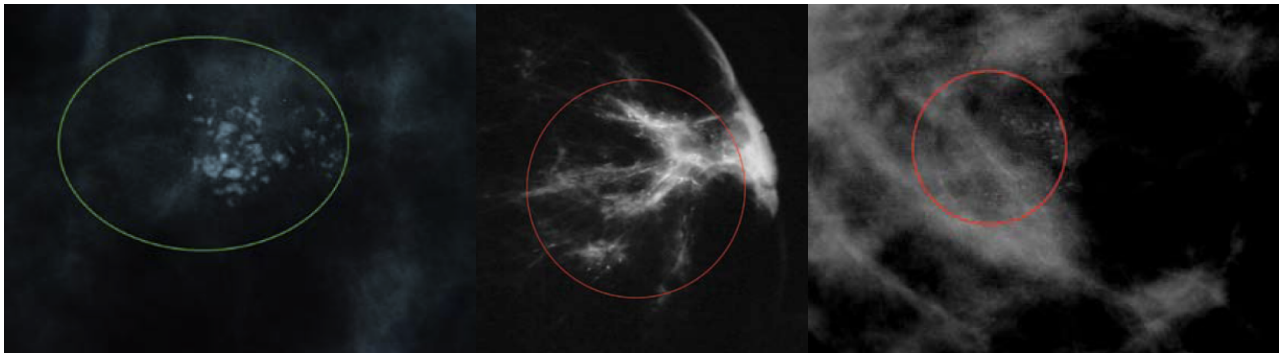


Figura 23 mamografías con varias formas agrupadas, de distintos tamaños y con densidad variable

Los investigadores centran su atención en mejorar los algoritmos destinados a detectar las lesiones patológicas, es decir, las calcificaciones, las masas, distorsiones de la arquitectura y asimetrías bilaterales.

### 1.1.3.5 Diagnostico

Las **técnicas para diagnosticar el cáncer** de mama son:

- **Exploración de la mama** que puede ser realizada por el facultativo o por la propia mujer.
- **Reconocimiento médico** periódico en función de la edad.
- **Técnicas de apoyo** cuando aún no es palpable el tumor con técnicas de imagen como resonancia magnética, la mamografía , mamografía con contraste , la ultrasonografía, Termografía, , tomosintesis para tratamiento de imágenes 3D, telemamografía, tomografía axial computerizada (TAC), escintigrafía[87], Tomografía por emisión de positrones (PET) y otros métodos intervencionistas (la Punción Aspiración con Aguja Fina (PAAF), la Biopsia con Aguja Gruesa (BAG) en sus diversas modalidades y la colocación de guía para biopsia quirúrgica) etc.

A pesar de todas las técnicas anteriores la detección del cáncer tiene dificultades en su estado de gestación o estadio inicial.

Como ya indicamos anteriormente, hay un porcentaje de tumores **en torno al 25% que no se detectan** por motivos como cansancio del especialista por largas jornadas de trabajo, falta de objetividad en el diagnóstico, mamas implantadas o con alta densidad, falta de procedimiento en la búsqueda o porque las lesiones se confunden con el tejido sano de la mama.

El **apoyo** del especialista **en sistemas CAD** aumenta la probabilidad de detención temprana en un 10%. [27].

La mamografía es la herramienta que presenta **mayor efectividad** en la detección antes de que se haga presente al tacto; [28] es una radiografía de la mama de baja intensidad.




Figura 24 Mamografo

Tiene además la **ventaja** de que ofrece dosis bajas de radiación y las imágenes son de una elevada calidad, la dosis actual es de **0,4 miliSierverts** que es equivalente a la radiación natural recibida por una persona en 7 semanas, es decir de riesgo bajo. Es más efectivo en mujeres mayores de 50 años ya que la densidad del tejido de una mujer más joven es más denso.

Hay **dos tipos de mamografías**, la convencional y de digital de campo completo. La digital permite un tratamiento a base de algoritmos para realizar análisis por computador. Se aprobó por la FDA (Food and Drug Administration) para su uso en ensayos clínicos en 2000 y la presencia en programas de cribado no ha sido todo lo rápido que se esperaba en un principio debido sobre todo a su elevado coste, y su menor resolución espacial. [97][98][99].

A día de hoy, de forma creciente se está reemplazando la analógica por la digital tanto en el uso clínico como en los programas de cribado [100]. Existen estudios que afirman que la mamografía digital de campo completo puede confundir y aumentar los falsos positivos debido a la interpretación de pseudo-lesiones y también mayores hallazgos de microcalcificaciones que son benignas. Además ofrece mayor contraste y resolución que si se comparan con pruebas anteriores analógicas puede hacer pensar de manera equivocada que han aparecido nuevas lesiones y, nuevos artefactos sometiendo a las pacientes a nuevas pruebas de descarte con lo que supone el elevado coste de estas y los efectos secundarios a los que por estas causa están sometidas y máxime teniendo en cuenta que los programas de cribado son masivos es un dato a tener presente. [101] [102]

| | MX analógica | | MX digital | Prueba $\chi^2$ |
|---|---|---|---|---|
| | N | Tasa (‰) (IC del 95%) | N | Tasa (‰) (IC del 95%) | p |
| *Mujeres cribadas* | 41.874 | | 38.264 | | |
| *Recitación* | 2.130 | 50,87 (48,80-53,01) | 2.449 | 64,00 (61,59-66,50) | <0,001 |
| *Control intermedio por microcalcificaciones* | 394 | 9,41 (8,53-10,38) | 715 | 18,69 (17,38-20,09) | <0,001 |
| *Biopsias por microcalcificaciones* | 72 | 1,88 (1,49-2,37) | 115 | 3,01 (2,50-3,61) | 0,002 |
| Negativas (FP) | 35 | 0,84 (0,60-1,17) | 57 | 1,49 (1,15-1,93) | 0,007 |
| Positivas (VP) | 36 | 0,86 (0,62-1,19) | 52 | 1,36 (1,03-1,79) | 0,033 |
| VPP | | 50,7% (39,3-62,0%) | | 47,7% (38,6-57,0%) | 0,698 |
| *CDIS* | 17 | 0,41 (0,25-0,66) | 34 | 0,89 (0,63-1,25) | 0,007 |

CDIS: carcinoma ductal *in situ*; FP: falsos positivos; IC: intervalo de confianza; Mx: mamografía; n: número de casos; VP: verdaderos positivos; VPP: valor predictivo positivo.

**Figura 25 Comparación de los resultados obtenidos con mamografía analógica y digital** [102]

La imagen por **resonancia magnética** tienen la ventaja de que no usan rayos x. Utiliza ondas de radio, imanes y un elemento de computación. Está indicado para mujeres con antecedentes familiares de cáncer de mama, que se hayan producido alteraciones en los genes BRCA1 o BRCA2 y síndromes genéticos. Esta técnica tiene una mayor detección que las mamografías pero es fácil que se detecten lesiones que no son tumores en realidad. [29].

La **termografía** se basa en el hecho de que los tumores cambian la temperatura de la piel. Mediante una cámara al efecto, se registra la temperatura en un ordenador que crea un mapa del pecho.

**Muestreo del tejido**: Se toman muestras celulares para análisis y observación al microscopio. Técnicas de muestreo son  Aspiración por aguja fina, Aspiración del pezón, Lavado ductal.

#### 1.1.3.6 Avances

Citaremos **avances** como:

- Detección de células tumorales circulantes (CTC)
- Gammagrafía mamaria.
- Cirugía oncoplástica
- Medicamentos de terapia dirigida
- Tratamientos dirigidos a los huesos.

Son **técnicas novedosas** que persiguen diferentes estrategias para abordar el problema del cáncer. Algunas detectan células separadas del tumor en la sangre, otras se basan en inyección de trazador radiactivo, hay técnicas combinadas de cirugía plástica y cirugía del cáncer, existen ya medicamentos que actúan en las células que sufren cambios genéticos, medicamentos inhibidores, medicamentos que evitan la propagación etc.

## 1.2 Objetivos

Después de exponer esta introducción es sencillo entender que **la vida de muchas personas** va a depender en el presente y el futuro de que se realice un buen diagnóstico precoz, preciso y veraz que contribuya a tratar eficazmente la enfermedad de cáncer de mama así como mejorar la calidad de vida de los pacientes.

Recordemos que el cáncer de mama es el cáncer más frecuente y el que **mayor mortalidad** produce entre las mujeres del mundo siendo un problema realmente preocupante de salud pública.

Los investigadores se esfuerzan en avanzar en este sentido con **líneas de investigación diversas**:

- Tratamiento de Imagen por computador.
- Métodos y cálculos estadísticos para predecir la evolución en base a la probabilidad de que la enfermedad evolucione hacia metástasis.
- Métodos basados en genética.
- Métodos de ayuda en la toma de decisiones del paciente.
- Estudios para encontrar el origen del cáncer como el que se está realizando en la actualidad por el National Institute of Environmental Health Sciences (NIEHS); El estudio se conoce como "El Estudio de Hermanas" (Sister Study), en el que participan 50,000 mujeres cuyas hermanas padecen cáncer de mama.
- Avances en los tratamientos con nuevos medicamentos que contienen bifosfonatos y estatinas. Medicamentos de acción dirigida.
- Estudios sobre suplementos dietéticos, extracto de semilla de uva, folato, acidos grasos omega_3, vitaminas B6 y B12.
- Nuevas técnicas de cirugía.

El **objetivo de este proyecto** es avanzar en la investigación para el diagnóstico precoz del cáncer de mama mediante la aplicación de técnicas de redes neuronales artificiales para el procesamiento de imágenes reales de mamografías (también llamada mastografía) realizada a mujeres y guardada en BBDD para su estudio e investigación.

Entre todos los procedimientos de detección, la mamografía nos ofrece una posibilidad de detección suficientemente oportuna.

**La célula cancerosa** se duplica con una frecuencia que oscila entre 100 y 300 días. Cuando el tumor alcanza el tamaño de 1 cm lleva en su haber 30 duplicaciones, es decir lleva evolucionando por un periodo de 7 años. Es deseable que las alteraciones se detecten cuando aún están en estado incipiente, es decir valorando si existen microcalcificaciones y si estas indican que hay cáncer. Esta **detección es sumamente compleja** incluso para los especialistas más entrenados ya que la constitución de la mama a menudo dificulta la apreciación porque los tejidos y lesiones son muy homogéneos y similares entre si y porque las lesiones buscadas por el radiólogo como indicadores de la posible presencia de un tumor, son realmente diminutas y en muchos casos inapreciables y se confunden con el tejido sano. La mamografía permitirá al especialista valorar la lesión para solicitar pruebas complementarias que determinen su malignidad.

Los errores en el diagnostico cuando son **falsos positivos** tienen consecuencias que se traducen en trastornos psicológicos, físicos y de índole económico y persisten a lo largo del tiempo (hasta tres años después de conocer que no es cáncer) [22] Estudio de 2013 de Annals of Family Medicine (Anales de medicina familiar). Lee "Long-Term Psychosocial Consequences of False-Positive Screening Mammography" (Consecuencias psicosociales a largo plazo del resultado falso positivo de una mamografía de detección).

Cuando son **falsos negativos** los efectos son aún más graves ya que retrasará el tratamiento con consecuencias que pueden ser fatales.

Con lo expuesto queda suficientemente justificado cualquier avance y contribución por pequeño que sea en la investigación del cáncer de mama en cualquiera de sus disciplinas.

Para este proyecto vamos a analizar y utilizar los últimos avances en **técnicas Deep Learnig** como aprendizaje automático; Geo Rey Hinton, fue uno de los pioneros en experimentar con éxito esta técnica y desde entonces, está evolucionando continuamente. Actualmente, un niño de 4 años tiene capacidades de reconocimiento de Imágenes que difícilmente podemos implementar en un ordenador actual pero esto va a cambiar en el futuro. Solo estamos en el comienzo y en este sentido veremos que es necesaria mucha capacidad de procesamiento y que queda mucho camino por recorrer.

Mediante el **aprendizaje profundo** se da respuesta a problemas que hasta ahora solo eran capaces de resolver las personas por ser conceptos muy complejos y que ahora comienzan a hacerlo también las maquinas, con programas que imitan redes neuronales, emulando el aprendizaje del cerebro de manera que las maquinas van mejorando su aprendizaje con la experiencia. Esta técnica está provocando una auténtica revolución. Es especialmente eficaz en sistemas no lineales para clasificar imágenes por lo que será objeto de estudio en nuestro proyecto.

Nos pasearemos por supuesto por el mundo de las **redes neuronales** Veremos que dentro de la división de las redes neuronales, las hay de **varios tipos** como son las recurrentes que no están estructuradas en capas y a las que se les puede añadir términos como memoria y temporalidad o las redes convolucionales que sí se basan en capas pero con la peculiaridad de que la neurona de una capa determinada no tiene porqué recibir conexiones de todas las de la capa anterior, favoreciendo así la especialización neuronal y ahondaremos en estas últimas que son una de las maneras más eficientes de implementar Deep Learning.

Afortunadamente para los investigadores, cada vez existen más **librerías y herramientas gratuitas** y en código abierto que implementan redes neuronales como por ejemplo libfann[25] que es una librería de "C"; Toolbox como Matconvnet: CNNs creada por

científicos de visión por ordenador de Oxford y buena candidata para implementar aplicaciones de análisis de imágenes con aprendizaje profundo. Esta toolbox está pensada para clasificar y codificar las imágenes, hacer el realce de imágenes y filtrado y está diseñada y soportada para **Matlab** que es un software reconocido para matemáticas ,ciencia e ingeniería o Tensorflow que recientemente ha abierto Google a terceros con el objetivo de crear una comunidad gratuita de orientación académica dedicada a la investigación y que según asegura Google, este modo de aprendizaje permite aprender 5 veces más rápido que los métodos más primitivos.[24].

Para entender la importancia de este mundo y el futuro de este tipo de redes, no hay más que observar que el gigante Google utiliza estas **técnicas de Inteligencia Artificial** para mejorar su capacidad de razonamiento en producto y en sus aplicaciones. Un ejemplo de esta aplicación es Street View reconociendo al 96% la búsqueda de los números del callejero que aparece en las fotografías que se realizan desde los vehículos en la toma de datos de sus calles [23]. Google quiere aplicar la disciplina de machine Learning y darle presencia a sus aplicaciones web y de correo mediante reconocimiento de patrones.

Analizaremos cómo funcionan estas redes, pondremos en práctica alguna de ellas y valoraremos los resultados aplicados a nuestro trabajo.

Para ponerlo en práctica, **analizaremos las más importantes BBDD de mamografías** públicas o privadas: Digital Database for Screening Mammography, IRMA, AMDI o B-SCREEN, (8), Nijmegen Digital Mammogram Database, Mammographic Image Analysis Society, Magic-5 MiniMIAS, Lawrence Livermore National Laboratories Database, construidas y puestas a disposición de los investigadores para su estudio en la detección de las microcalcificaciones y por supuesto analizaremos resultados empíricos con técnica de Deep learning sobre muestras de una de ellas elegida.

Repasaremos otras técnicas como las SVM´s aplicadas al propósito de este trabajo y compararemos resultados con los obtenidos en estudio de aprendizaje profundo. Existen

también librerías en Matlab como la desarrollada en la Universidad de Southampton MATLAB Support Vector Machine Toolbox disponible en la página Web

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

Finalmente propondremos trabajos futuros.

## 1.3 Organización de la memoria.

En el capítulo 1 vamos a plantear el problema, "el qué" y para ello es necesario que de manera sólida se expongan conocimientos médicos sobre el cáncer en general y del cáncer de mama en particular como base sólida para entender la magnitud a la que se enfrenta la ciencia y de esta manera exponer de manera clara "por qué" de este trabajo.

En el capítulo 2 hablaremos de la historia del arte , haremos un poco de historia , Cuales son y de que tratan las redes que vamos a utilizar para el estudio y las principales diferencias que existen entre ellas , las bases de datos existentes de mamografías , sus características , ventajas y desventajas , elección para este trabajo y motivación.

En el capítulo 3 describiremos "el cómo" de nuestro modelo, cual es la dificultad al aplicarlo a la mamografía, haremos una descripción detallada del mismo y mostraremos las pruebas realizadas en los ensayos para finalmente plasmar el resultado.

En el capítulo 4 concluiremos exponiendo las conclusiones del modelo práctico de red neuronal convolucional implementada.

En el capítulo 5 propondremos nuevas líneas de investigación.

Finalmente el último apartado lo dedicaremos a los apéndices y referencias.

# Capítulo 2 Estado del arte

## 2.1 Historia y antecedentes

Actualmente hay grandes organizaciones que se dedican a la investigación del cáncer de mama, que intentan resolver mezclando distintas disciplinas como BBDD, estadística, aprendizaje automático y tratamiento de imágenes [233]; como ejemplo podemos citar la NASA según Simoff S. [232] que además se apoyan en herramientas de empresas como Microsoft u Oracle que lideran la tecnología de BBDD. La mayoría de estas están aún en fase experimental.

En este punto vamos a hacer una exposición sobre literatura y trabajos de investigación que existen sobre técnicas de aprendizaje automático aplicado a la clasificación de mamografías. Comenzamos con una cronología sobre la aplicación de la mamografía, de los sistemas CAD, introduciremos y veremos la relación con la minería de datos. Dentro de este campo nos vamos a referir a los árboles de decisión, las máquinas de soporte vectorial (SVM) y las redes neuronales artificiales (ANN) y dentro de estas haremos especial hincapié en técnicas Deep Learnig. Previamente haremos un repaso histórico hasta llegar a los comprender los avances que disponemos actualmente.

### 2.1.1 Cronología de la mamografía

Año 1913, el médico cirujano **Albert Salomón** de nacionalidad alemana, con su publicación señala la importancia de hacer radiografías para comprobar si el tumor se ha extendido a los ganglios y describe variaciones radiográficas debidas a los cambios que se producen por la presencia de las microcalcificaciones cuando hay tumores de carcinoma infiltrante o circunscrito.

Año 1927 el cirujano **Otto Kleinschmidt** por vez primera indica en un libro de texto la mamografía.

Año 1930 el radiólogo **Stafford L. Warren** pública la técnica estereoscópica para las mamografías en vivo.

Año 1931 **Walter Vogel y Paul Seabold**, publican individualmente las investigaciones que realizan sobre mamografías diferenciando entre diagnóstico de enfermedad benigna y cáncer.

Año 1937 **Hicken** publica sus trabajos sobre uso del contraste con resultados asombrosos que permitían distinguir ductos lactoforos en sus diferentes estados anómalos y normales, [29]

Año 1956 **R. Egan** durante tres años estudia en 1000 pacientes mediante radiografías la mama con un alto porcentaje de aciertos en la diagnosis 238 casos detectados de 240 casos malignos. También advirtió que los falsos positivos eran el principal obstáculo [35]. A partir de este estudio la mamografía se aplica como técnica de diagnóstico.

Década 1970 se realizan los **primeros programas de cribado masivo** en personas sanas siendo pioneros en su aplicación los países nórdicos.

Estudios posteriores como **A. J. Evans** en su estudio de 208 casos en rango de edades de los 40 a los 48 arroja y pone en cuestión la eficacia de la técnica de cribado sobre todo en detección de microcalcificaciones con un 20, 16 resultados de falsos positivos [234].

**Gilbert** apunta que el cribado mamográfico es el programa que suscita mayor controversia a pesar de los profundos estudios realizados desde hace ya más de 50 años.

Esto se puede comprobar en **distintas ponencias y trabajos** realizados al respecto y un buen ejemplo de ello es la realizada en 2010 por Andreu Segura especialista en medicina preventiva, Director del área de salud Pública del Instituto de estudios de la Salud de la Generalidad de Cataluña, y Marina Álvarez Benito radióloga experta en mama, Jefe del servicio de radiodiagnóstico del HU Reina Sofía de Córdoba y presidenta de la Sociedad Española de Diagnóstico por Imagen de la Mama.

Marina Álvarez Benito se muestra a favor por motivos como su utilidad para disminuir la mortalidad, el diagnostico de otros tumores con ganglios negativos evitando terapias más agresivas,

En contra Andreu Segura argumenta las distintas investigaciones que se han postulado en el sentido contrario como son : Muir Gray[235] responsable de los cribados en el Servicio de Salud Nacional de Gran Bretaña y que afirma que todos los procesos de cribado hace daño en mayor o menor medida [36], Ned Calonge [236] responsable del grupo de trabajo de los servicios clínicos preventivos de EEUU argumentando que en una prueba de cribado pueden suceder varias cuestiones y el 80% del ellas no son nada buenas: un falso negativo , un falso positivo, sobre diagnóstico conducente a tratamientos no necesarios, asignación de recursos que se podrían destinar a causas más útiles,[37] , Berrington y Reeves que dictamina que solo es justificable un programa de cribado mediante mamografía en las mujeres mayores de 40 años cuando la reducción de la mortalidad fuera superior al 20% [38], según David Sackett [39] no son extrañas acciones de prevención que se basan en meras especulaciones como sucedió con la profilaxis hormonal en la menopausia o por último como editorializa Gilbert [40] , el riesgo de una estadounidense de morir por cáncer de mama a 10 años es del 4 por mil. Si gracias al cribado se redujera en un 10%, por cada 10000 cribados tendríamos 40 muertes mientras que por los no cribados 44. Extrapolando los datos para evitar una muerte en los 10 años siguientes se debería hacer una criba de 2500 casos, es decir mil pasarían por la experiencia de una falsa alarma y entre 5 y 15 un sobre diagnóstico y por tanto tratadas y soportando los efectos secundarios asociados.

Se hace necesario evaluación retrospectiva , excelente entrenamiento del especialista, con el fin de minimizar errores también doble lectura y sistemas CAD que faciliten ayuda añadida que cada vez será de mayor calidad y fiabilidad teniendo en cuenta el avance de la tecnología que se produce a cada momento.

### 2.1.2 Cronología de los Sistemas CAD

Por otro lado, los avances que se han producido en el mundo de la informática, siempre han ido muy de la mano de la **evolución de los sistemas CAD**. Los programas de diseño asistido empezaron a desarrollarse en el periodo final de la primera generación de computadores pero adquiere gran relevancia a partir de la cuarta generación con los avances en lenguajes de alto nivel y la evolución de circuitos LSI, la segmentación, la memoria virtual y la multiprogramación.

Partimos de 1955 en el MIT de Massachusetts cuando realizan un desarrollo del radar para el ejército Americano procesando a través de una pantalla de tubo de rayos catódicos (CRT).

El **primer Cad** se comercializa por un precio de 500.000 US$.

En 1960 Lusted introduce el diseño asistido por ordenador pero no es hasta 1988 cuando se obtiene la aprobación de la **Food and drug Administration** (FDA) para uso de mamografía digital. Pasados tres años el 10% de los centros en Estados Unidos estaban de acuerdo en el uso de esta técnica. Hoy en día la gran parte de las que se analizan en EEUU son también sometidas al CAD para ayuda en el diagnóstico. [65]

En 1959 nace el **primer Plotter** de la mano de Computervision y a partir de este momento grandes empresas automovilísticas como Ford y Chrysler comienzan a utilizar para sus diseños sistemas Cad.

En 1962 Ivan Sutherland estudia y desarrolla los **gráficos interactivos por computador** utilizando lápiz y teclado óptico para dibujar en la pantalla y dio lugar al sistema Sketchpad. Ahora las representaciones se basaban en la topología de los objetos.

Paralelamente al sistema Sketchapad, se trabaja en Itek y General Motors con **pantallas vectoriales, tableta y lápiz electrónico**.

En el año 1963 se implanta en la universidad la metodología Sketchpad. El computador mostraba las líneas visibles de un objeto ocultando las que en su posición no deberían observarse y mostrando estas últimas cuando rotaba el objeto a una posición en la que sí era natural que se mostrase.

1967 primer sistema realizado por F. Winsberg mediante computador que se realiza específicamente para **análisis de imágenes mamográficas**.

A principios de los 70 Charles Eastman difunde el **modelo BIM** (modelo de información de edificación basado en estructuras arquitectónicas que se pueden ensamblar conformando un diseño completo de arquitectura y finalmente Jerry Laiserin lo hizo popular [27].

En 1975 aparecen la pantalla de 19" desarrollada por Textronic y el **primer sistema CAD/CAM**, en este caso su artífice fue AMD.

En 1978 Computervision desarrolla un **terminal gráfico** con tecnología raster.

En 1979 Boing, Nist y General Electric crean un formato neutral de datos que permite el que se pueda intercambiar digitalmente la información entre sistemas Cad.

En 1980 nace la empresa española **Investrónica** que realiza desarrollos CAD/CAM en el campo textil.

En 1981 aparecen los **gráficos 3D**, desarrollo de la empresa 3D Eye Inc.

En 1982 Jhon Walker funda **AutoCad** para introducir el cad en el ordenador personal y los sistemas CAD se empiezan a globalizar y popularizar.

En 1985 es presentado el desarrollo **CAD MicroStation para PC**.

En 1992 aparece Autocad para Sun y en 1995 para Microsoft Windows.

En 2008, según fuente de **Gobierno de Castilla-La Mancha** y la compañía de tecnologías **INDRA**, el Servicio de Salud de Castilla-La Mancha fue la primera comunidad española en disponer de sistemas CAD para el diagnóstico de pulmón, hígado y mama [66]

Las **empresas líderes** son Parametric Technology, DassaultSystem, Eds/Intergraph, SDRC y autodesk.

Actualmente continúa la evolución con el uso de 3D ganando en velocidad, estabilidad y prestaciones. [28]

### 2.1.3 Minería de Datos

Consiste en **descubrir patrones en enormes volúmenes de datos** y hacer que esta información sea comprensible y se pueda utilizar con la finalidad de darle alguna utilidad como por ejemplo el análisis predictivo. Para ello se aplica la estadística, la inteligencia artificial, el aprendizaje automático y los sistemas de BBDD.

En los años 60 empiezan a aparecer ideas como **data fishing**, **data mining o data archeology**. Más tarde ya en los 80 se ahonda en la idea de Data Mining y KDD ("Knowledge Discovery in Databases") con personas como Rakesh Agrawal, Gio Wiederhold, Robert Blum y Gregory Piatetsky-Shapiro.

En los últimos años de la década de los 80 tan solo había dos empresas dedicadas a este campo descubrir patrones en enormes volúmenes de datos. Para ello se aplica la estadística, la inteligencia artificial, el aprendizaje automático y los sistemas de BBDD.

A principios del segundo milenio ya existían un centenar de empresas que ofrecían más de 300 soluciones enfocadas a la minería de datos. La técnica de minería de datos se puede aplicar a prácticamente cualquier ámbito social y en el relativo a la salud está especialmente **indicado para el diagnóstico precoz del cáncer de mama**. El diagnostico se realiza ayudándonos de una base de datos mamográfica [238]. **Otras aplicaciones** en medicina son

tomas de decisiones basadas en coronario grafías sobre personas que padecen cardiopatías isquémicas [240], también se aplica por ejemplo para diagnosticar enfermedades que se transmiten por vía sexual [241].

**Fases** del proceso de tratamiento de imágenes mamográficas:




Figura 26 Una visión general del método aplicado a Imágenes de mamografía:

Los pasos que se aplican en la minería de datos aplicado a mamografías son los siguientes:

- **Construcción del modelo predictivo** de clasificación o segmentación.
- **Extracción de datos** de las imágenes, para la obtención del modelo que represente el comportamiento que se observa en el problema.
- **Obtención y agrupación de los patrones** con un denominador común que representan el comportamiento de los valores o relación de asociación de las variables del problema.

- **Maximizar el nivel de coincidencias**. Se pueden combinar distintas técnicas de manera que se adapten mejor a la exigencia del problema y así dar paso a distintos modelos y elegir el que mejor se ajuste a lo que esperamos como resultado. Esto obliga a realizar diferentes preprocesados de los datos.

- **Resultados**. Una vez que tenemos el modelo debemos validar que los resultados que nos ofrece son los que esperamos y si no es así, debemos construir nuevos modelos alterando pasos previos o comenzando de nuevo Si resulta que cumple expectativas proporcionado las salidas esperadas y el margen de error es nulo o al menos admisible, ya podemos pasar a explotación el modelo generado.

- En nuestro modelo nos basamos como lo han hecho tradicionalmente muchas empresas en BBDD creadas para proporcionar la información útil necesaria para la solución del problema pero cada vez es más frecuente que en la minería de datos se trabaje con datos desestructurados que se encuentran en ficheros de texto, internet etc.

- En cuanto a las **técnicas más representativas** de minería de datos son las **redes neuronales, la regresión lineal, los arboles de decisión, los modelos estadísticos, agrupamiento o clustering y las reglas de asociación**. [30] luego entraremos con mayor profundidad en cada una de estas técnicas.

## 2.2 Sistemas CAD aplicados a la mamografía

El cáncer de mama presenta un estado de latencia de 10 a 15 años y una fase asintomática de una a tres años en la que el cáncer se puede detectar de manera precoz.

De los trabajos que se han realizado sobre cáncer de mama, la mayor parte se han dirigido a estudios sobre detección y análisis de calcificaciones y masas (Bustamante et al.,2001)[70] (Cheng et al., 2006)[71] (Tang et al., 2009)[72] (Rangayyan y Nguyen, 2006)[73] y en menor

medida se han ocupado de la distorsión de arquitectura de la glándula mamaria (Ayres y Rangayyan)[74] (Ayres y Rangayvan)[75] (Matsubara et al.)[76] (Rangayyan et al)[77].

En las últimas décadas , se muestra un **interés especial en la aplicación de visión artificial** por computador en aplicaciones de carácter médico facilitado por el avance de la tecnología en capacidad de cálculo y almacenamiento de la información que favorecen el análisis de imágenes que han sido digitalizadas simulando visión humana y aplicando a este campo la inteligencia artificial.

El sistema CAD hace una lectura de la digitalización de una mamografía e incluye una serie de técnicas al objeto de destacar, medir contrastar o clasificar partes o elementos de la imagen de manera que resulte más sencillo su análisis posterior para una persona o para un sistema de visión artificial.

Se usa por los radiólogos como ayuda en la detección de anormalidades y actualmente en la aplicación de detección de cáncer de mama equivale a un segundo lector ya que el radiólogo primero interpreta la mamografía antes de usar el sistema de ayuda CAD. Estos sistemas operan marcando las áreas donde es posible que exista lesión con marcas especiales. El diagnostico final queda en manos del médico. [33].

Por lo general, los **sistemas CAD comprenden varias fases** para obtener los resultados que esperamos partiendo de la imagen original: **Digitalización, pre-procesamiento, Segmentación, Extracción de características y por último la clasificación**.

Aunque los pasos se pueden denominar de diferente forma dependiendo del autor, básicamente la división de los mismos conducen a un mismo resultado. Ejemplo de esta división según Fernández [68].

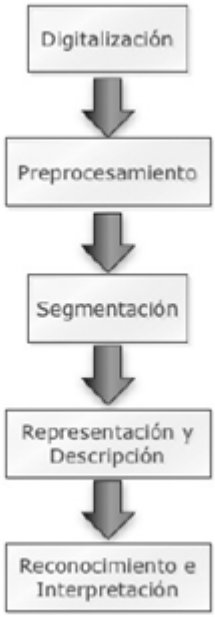



Figura 27 Fases del procesamiento digital de imágenes

Otra división realizada por Chen[238]:

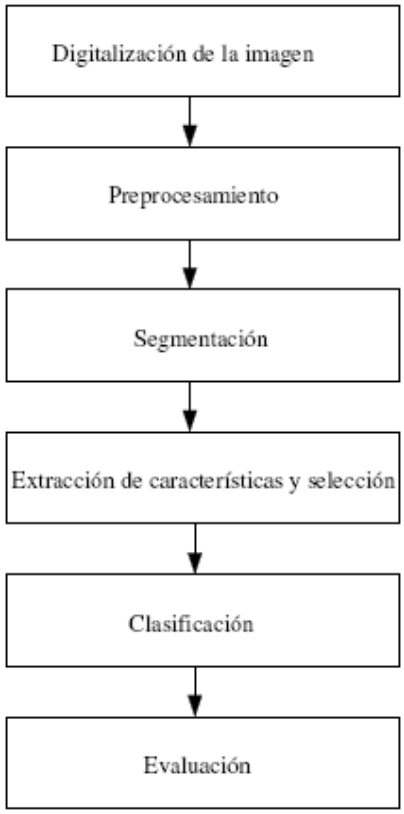



Figura 28 Etapas en el diseño de un sistema de ayuda al diagnóstico de cáncer por ordenador a partir de imágenes médicas según Chen

Para Bozek, Mustra, Delac, & Grgic, [237]; Cheng, Shi, Min, Hu, Cai, & Du, [238] las fases de procesamiento de imágenes son: **Preprocesamiento, Segmentación, Extracción de características y selección y por ultimo clasificación.**

En el momento de captura de la imagen y sobre todo en las de tipo de aplicación en medicina se producen interferencias en la información no deseadas debido al polvo, ruido aleatorio e implícito a los equipos, la propia digitalización que en definitiva hace que sea necesaria una etapa de pre-proceso [57].

Aunque vamos a centrarnos en este trabajo en el tratamiento CAD aplicado exclusivamente a micro calcificaciones, describiremos las distintas etapas por las que pasa una imagen en un sistema de ayuda por computador.

Hay diferentes tipos de clasificación dependiendo del autor como ya hemos visto. Vamos a explicar cada una de estas etapas:

### 2.2.1 Digitalización

Cuando se radia al paciente, el haz de rayos X que lo atraviesa se atenúa debido a los distintos tejidos y esto produce una **imagen radiológica primaria** que hay que transformar en una imagen visible al ojo humano utilizando **intensificadores de imagen**, pantallas fluorescentes etc. (CHEV95) [80]. Solo el 1 % de la totalidad de rayos X que atraviesan la película radiográfica es absorbida por esta y para evitar grandes dosis de radiación en la formación de la imagen, se utilizan elementos de refuerzo como las **cartulinas de refuerzo** que ayudan a capturar los fotones de rayos x evitando grandes dosis de radiación. Mediante la fluorescencia, la energía que ha absorbido se remite formando fotones de luz visible y con una ganancia importante en la relación entre los fotones que se han emitido en forma de luz y los absorbidos en rayos x.

La imagen se forma porque los fotones luminosos se transmiten a la película que se transforma en imagen al ser absorbidos.

En una mamografía las **densidades ópticas** o diferentes niveles de oscurecimiento no varían linealmente con las exposiciones. Para obtener su **curva característica** representamos el logaritmo de su exposición relativa en función de la densidad óptica. La curva generalmente tiene forma de "S"

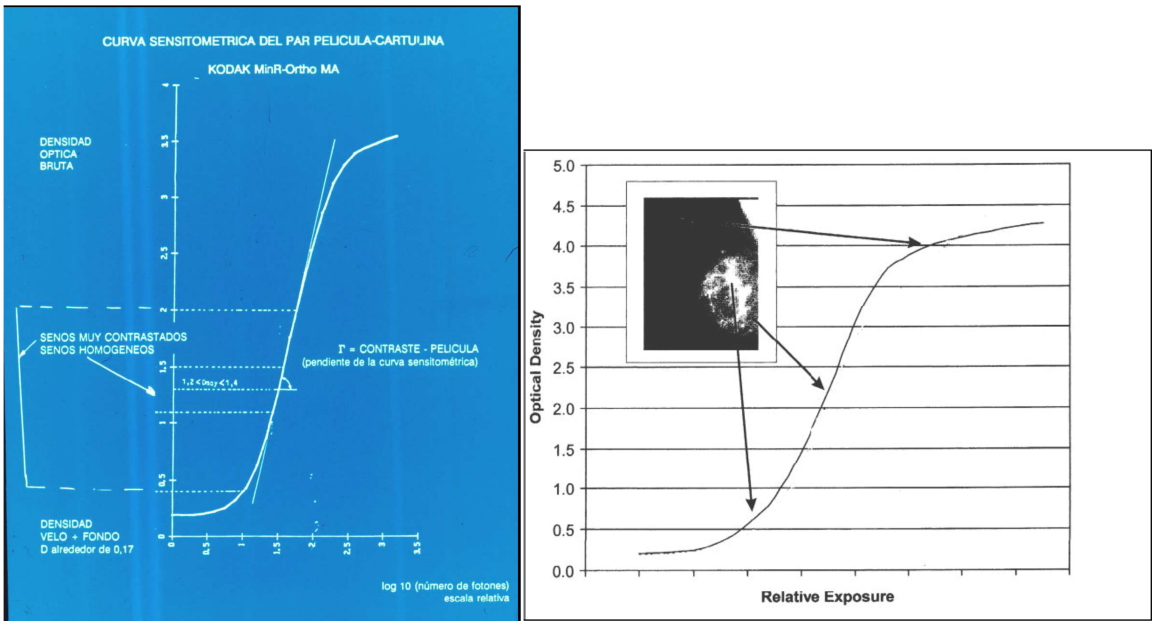



Figura 29 Imagen de la curva característica de una película Maidment et al. Sem in Roent 2003 [81] [82]

Las películas mamográficas tienen un alto intervalo de densidades y su curva superior comprende valores entre 2,8 y 3,7 [83].

La película utilizada será más **sensible** que otra si es necesaria una menor exposición para obtener la misma densidad y si es mayor hablamos de **lentitud**. Otros términos importantes son el **gradiente medio** para medir el contraste, la **latitud** para determinar donde se produce un contraste adecuado, las **densidades ópticas mínimas y máximas** ya que nos permite la comparación y distinción entre las microcalcificaciones y el tejido de la glándula mamaria.

Un contraste con densidad óptica mayor o igual a 0,8 es el mínimo necesario para que las zonas de la mama que muestran mayor opacidad a la radiación puedan ser objeto de estudio

[84] y en películas lentas , hay que aumentar la radiación para minorar ruido y el efecto que produce el grano de la película [85].

Para conocer la **calidad de una imagen digital** se utiliza la **eficiencia de detección cuántica (DQE)**. En esta medida se tienen en cuenta los parámetros necesarios más importantes como son la **resolución espacial**, la función de **transferencia de la modulación (MTF) y el ruido**.

$$DQE(f) = \frac{SNR^2(f)salida}{SNR^2(f)entrada} \quad (1)$$

la DQE es la facilidad para trasformar fotones de rayos X que recibe el detector en señal eléctrica proporcionando la eficacia del sistema de imagen de rayos x para comunicar y proporcionar una imagen con una alta **relación señal-ruido** (SNR) en relación con un detector ideal, que transmite la radiación desde la mama hasta que se produce la imagen final Si la DQE es mayor entonces se traslada en mayor calidad. Se puede considerar como una medida sustituta de la eficiencia de la dosis de un detector ya que la exposición a la radiación requerida para un paciente es menor a medida que aumenta el DQE para la misma imagen SNR y las condiciones de exposición. La **radiación a los pacientes es mínima** si el DQE se hace tan cercano a la unidad como sea posible. Permite la capacidad de detección de objetos de bajo contraste.

La imagen se convierte en caso de ser analógica a formato digital para su tratamiento. Se realiza un **mapeado de la imagen** mediante cuadricula de puntos o pixeles. Cada uno de los pixeles tiene asignado un valor de tonalidad negro, blanco, algunos matices de grises o color. Estos son almacenados en una secuencia determinada generalmente mediante una representación matemática que permite la **compresión de la imagen** de manera el ordenador pueda interpretar y leer los bits con el fin de producir una versión analógica que se pueda visualizar o imprimir.

En la imagen en general, hay **características fundamentales** como son la resolución, la **dimensión del pixel, la profundidad de los bits, el rango dinámico, el tamaño de archivo, la compresión, el formato** etc. [79]

Y en el caso de las **imágenes radiológicas**, la calidad de imagen se rige por el contraste radiológico, la resolución espacial o borrosidad, el ruido y los artefactos.

Significar que la **resolución espacial**, definida como la cantidad que se representa de cada objeto en cada pixel, **en películas mamografías** requieren de pixeles menores o iguales a 25 micras, a diferencia de otras técnicas como huesos o tórax que usan pixeles de mayor tamaño 78 – 100- 200. La resolución en la mamografía digital requiere tamaños de pixel de 50 para poder detectar microcalcificaciones.

A continuación se muestra una **comparativa entre mamografía convencional y digital** [86]

| | **Mx convencional** | **Mx digital** |
|---|---|---|
| Latitud | Moderado | Amplia |
| Contraste | Fijo | Variable |
| SNR | Limitado a la película | Limitado por radiación |
| Desplazamiento | Fácil | Difícil |
| Archivo seguridad | Moderado-pobre | Moderada-Buena |
| Facilidad de copia | Pobre | Fácil |
| CAD | Requiere digitalización | Fácil |
| Teleradiología | Difícil | Fácil |
| Procesado de imagen | Imposible | Fácil |
| Imagen en tres-D | Imposible | Posible |

Figura 30 comparativa entre mamografía convencional y digital

## 2.2.2 Preprocesamiento.

Para Álvarez la clasificación sobre el **preprocesamiento de imágenes** atiende a los siguientes pasos [78]:

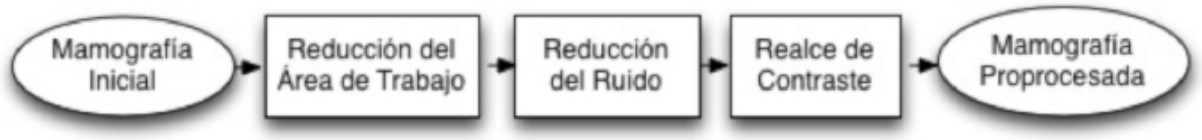



Figura 31 Preprocesamiento digital de imágenes. Fuente: Álvarez et al., 2006

## Reducción del area de trabajo:

Se eliminan zonas carentes de información o con información irrelevante

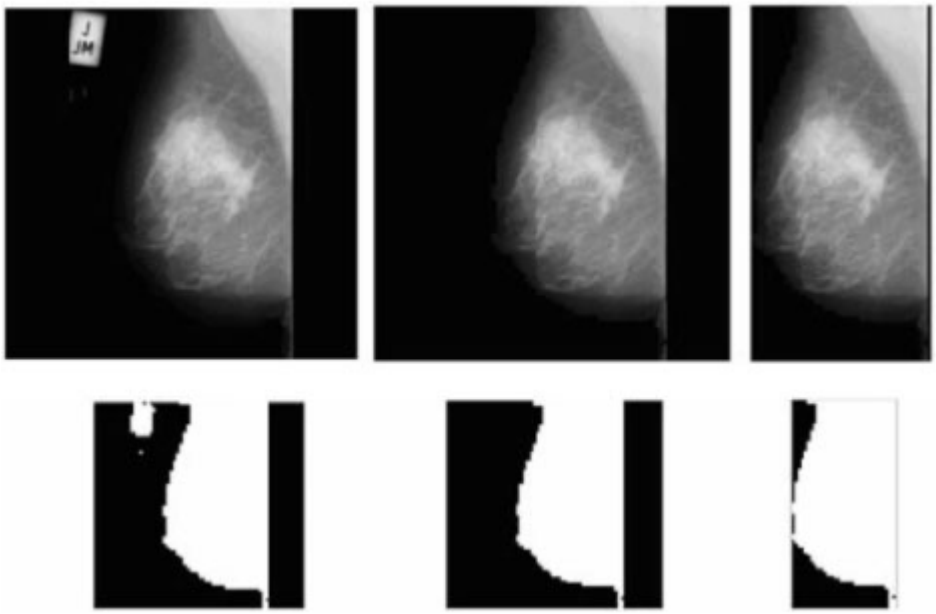


Figura 32 Proceso de reducción del  área de trabajo. Imagen tomada de Álvarez et al., 2006

## Redcción del ruido:

El **ruido** es la información indeseable que afecta los resultados de otras etapas posteriores. Se utilizan filtros y podemos citar entre otras tecnicas las que utilizan **filtros gaussianos o el filtro de wiener** para su eliminación.

Entre los distintos **algoritmos que se han probado en mamografías** podemos citar el filtrado de mediana , el filtrado de Wiener y la reducción normal de wavelets (Alvarez et al) [78].

El **filtro de mediana** tiene en cuenta la intensidad de los pixeles vecinos para calcular la del que se está operando. Permite eliminar los que tienen la intensidad muy alta o muy baja

El **fitro de Wiener** considera que la imagen y el ruido se producen aleatoriamente sin ningún tipo de correlación.

La ecuación:

$$\hat{f} = \overline{w} \times \overline{g} \qquad (2)$$

proporcionará la imagen restaurada a partir de la imagen con ruido donde g es la imagen degradada y W es el filtro.

Los **filtros Wavelets** se utilizan tambien para la reducción del ruido y podemos describir tres pasos importantes en el proceso: descomposición en **subbandas wavelets**, se modifican los coeficientes con tecnicas como la **reducción de umbral** y la imagen se reconstruye a partir de estos.

**Realce del contraste:**

Como las mamografías no tienen un buen nivel de contraste, se dificulta el diagnostico de los especialistas y el sistema CAD debe corregirlo antes de que se produzca la segmentación.

Es de vital importancia que el realce de contraste de imagen no produzca incremento de ruido a la nueva imagen [58]. Se pueden aplicar técnicas directas a través de filtros dando mejores resultados los que aplican **técnicas no lineales o multirresolución** e **indirectas de realce de imagen** con medios como el **escalado del histograma**.

El preprocesamiento es especialmente importante en el caso de las mamografías ya que hay información que no resulta para nada relevante en las áreas que se van a analizar y que generan un excesivo costo en el cómputo y esto además sin mencionar el pésimo contraste de las imágenes radiológicas y la falta de claridad de las estructuras mamarias.

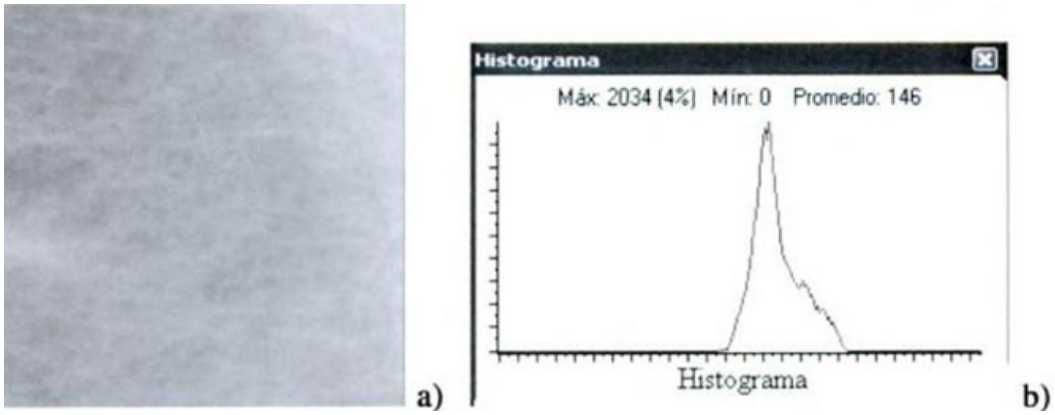
En esta etapa se pretende aumentar las diferencia entre tejido anómalo y sano.

Para la mejora en el tratamiento de las imágenes, se realiza el pre-procesado a partir de técnicas como **Modificación global del histograma ecualizando el histograma**, (imagen que contiene el número de pixeles igual al mismo nivel de gris), **Procesamiento multiescala con realce por modificación del contraste**, **técnicas de colores falsos** (pseudocolor), **procesamiento local basado en una región de la imagen. [59]**

**Modificación global del histograma:**

Se aplica la ecualización del histograma, la EH multiplico particionando el histograma de manera independiente según los picos contenidos y realizando la ecualización para expandir el histograma de manera que alcance el mayor rango posible

Con esta técnica conseguimos un realce del contraste que permite al ojo humano percibir mejor las zonas de menor contraste **aumentando las diferencias** entre el tejido que es sano y tejido anómalo. Esto se consigue ecualizando de tal manera que para todos los valores de gris tengan un valor en proporción de pixeles. En ciertos sistemas, en esta etapa se elimina el fondo de la imagen y el músculo pectoral.



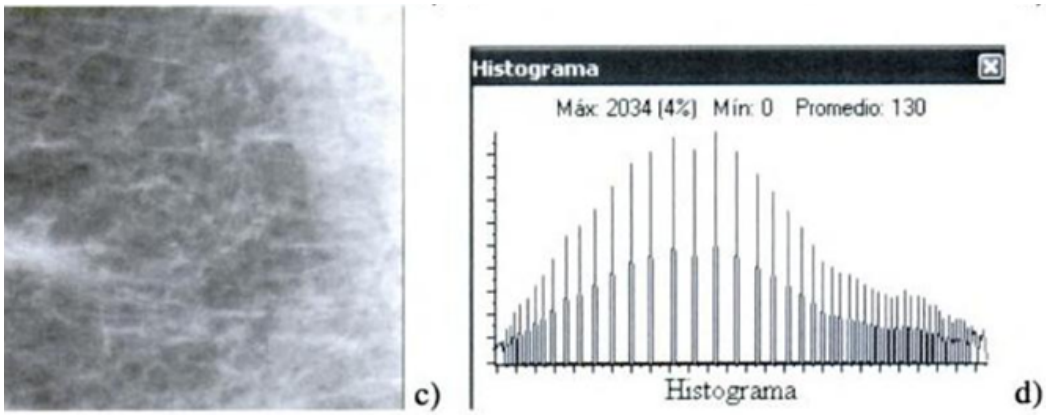



Figura 33 a) Sección extraida de una imagen original de una mamografía con tejido graso. b) Histograma correspondiente a la sección "a)". c) Ecualización aplicada a la sección "a)" d) Histograma de la sección " c)"

**Procesamiento multiescala. Realce por modificación del contraste**:

El contraste puede mejorarse mediante el re escalamiento de la intensidad de cada pixel.

El nivel de gris que corresponde a un pixel en la imagen que tenemos en la entrada se transforma relacionando la intensidad de entrada ($i$) con la intensidad de salida ($g$) obteniendo $g=T(i)$.

El procesamiento multiescala Transformación de la imagen mediante Wavelets para el realce de las lesiones e inversión para devolver el domino original a la imagen.

Por ejemplo, a continuación se muestra como mediante técnicas aplicadas como la **transformada de tercer orden de Bsplines** se realiza el contraste y su resultado. Para ello se selecciona la zona mamaria y se convierte a coeficientes de orden tres y a la salida se le aplica el operador derivada de primer orden:

$$f' = 1 - z^{-1} \qquad (3)$$

En el eje x o eje y, dependiendo de la dirección que se desee y a los nuevos operadores obtenidos, se vuelve a re escalar de manera lineal a grises con niveles entre 0 y 255 para su visual.

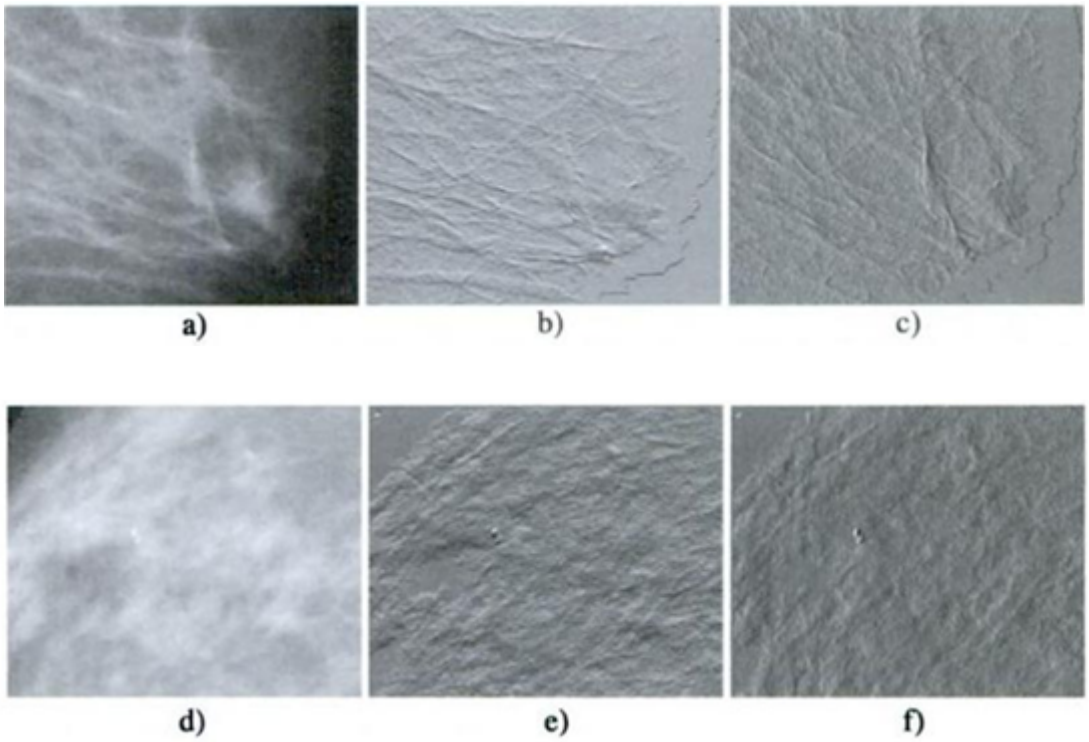



Figura 34 a,d). Sección extraida de la imagen original de dos mamografías con tejido graso "a)" y denso "d)". b,e). Realce del contraste por Bsplines en eje X. c,f). Realce del contraste por Bsplines en eje Y

## Técnicas de colores falsos y pseudocolor

**Colores falsos**: Se utilizan para asociar a un conjunto de datos otro de colores para apreciar diferentes atributos.

**Pseudocolo**r: técnica para convertir una imagen de grises en otra a color asignando a cada pixel en base a una característica como por ejemplo la intensidad un color diferente. Se aplican filtros como **paso-bajo, paso-banda y paso alto** para asignar los colores. Ejemplo de esta aplicación es dotar de color a las películas en blanco y negro.

## Procesamiento basado en una región de la imagen.

**El procesamiento local**: En **ROI´s** (regiones de Interés) se aplican algoritmos orientados al resalte de anomalías que se hallan localizadas.

Su **aplicación en contornos** es importante ya que permite extraer y realzarlos. También suaviza la imagen, atenúa ruido e introduce borrosidad. Utiliza ventana de pixel que no es ni más ni menos que una matriz en dos dimensiones de pixeles y es apropiada para el tratamiento de información de tipo regional de la imagen. Esto es más conocido como la **frecuencia espacial** cuya definición es la tasa de cambio de intensidad de los pixeles dividido por la distancia sobre la que ocurre este cambio. Muchas técnicas basadas en la región de la imagen, actúan como filtros que atenúan o realzan componentes de la frecuencia espacial. Estas técnicas usan métodos lineales como la **convolución** o no lineales como el **filtraje de la mediana**.

El **procedimiento** es el siguiente:

- Se barre pixel por pixel la imagen de entrada según las filas y columnas.
- Cada pixel es tratado teniendo en cuenta una ventana del mismo y aplicando el algoritmo más apropiado.
- El valor de salida de cada pixel se ubica en la imagen de salida en la misma posición que ocupaba en la de entrada.

Los sistemas Cad suelen tener un **módulo dedicado al pre-proceso** también llamado de tratamiento de la señal y las operaciones que se realizan a menudo dicho modulo son: el **escalado cuantificación, filtrado, modificaciones del histograma, transformaciones geométricas, operaciones morfológicas, eliminación de ruido y segmentación**. Este módulo cuando trabaja en modo de entrenamiento, también etiquetará el patrón para luego poder ser identificado [60] [61].

### 2.2.3 Segmentación.

El siguiente paso es **identificar las regiones** que contienen microcalcificaciones o regiones de Interés (ROI). Determina como es de sensible el sistema. Debería determinar el mayor número de anomalías posibles.

La clasificación atendiendo a su naturaleza: **Técnicas clásicas, Substracción de imágenes bilaterales y Técnicas multiescala.**

**La técnica clásica** más sencilla es la **umbralización global**. No es muy precisa en las ROI por lo que su resultado se usa como entrada de algún otro proceso. La **umbralización local** es más precisa y se aplican técnicas como **región Growing** en la que los tumores presentan brillo superior al tejido sano circundante y tomando una muestra como semilla se compara con los pixeles que tiene alrededor para ver si tiene las mismas características o similares y si es así se hace crecer la región. Cuando para de crecer, se compara con las regiones vecinas para comprobar si es tejido sospechoso o no y región Clustering y que a diferencia de la anterior no se toman semillas iniciales, se buscan las zonas directamente y en la que el **algoritmo de K-medias** se usa popularmente. Hay otros **algoritmos para la detección de bordes** como son **filtros de sobel [62], Prewit, Laplaciano** etc. También se utiliza ampliamente el emparejamiento de patrones comparando imágenes con muestras obtenidas previamente en el entrenamiento. En general consiste la umbralización global en asignar valores localmente a las ventanas de tamaño menor en función de niveles de grises en sus pixels.

**Substracción de imágenes bilaterales:** se somete a seguimiento al paciente y se identifican las ROI´s mediante el alineamiento elástico por comparación de las imágenes.

**Técnicas multiescala**: basado en que el pico que se genera en las zonas sanas es distinto del pico de las zonas anómalas. Se estudia si existe variación en el tamaño del tumor en tiempos diferentes [63].

La segmentación es una fase en la que a diferencia de la fase de pre-procesamiento, los trabajos que se han realizado sobre este tema no lo acotan perfectamente y por tanto **para detectar las calcificaciones** pueden proponerse métodos que utilicen etapas intermedias de localización.

Para no limitar su descripción, evitaremos ser puristas ya que lo importante no son las fases en sí sino la descripción de las técnicas y herramientas.

Los **wavelets y el análisis multirresolución** son una herramienta muy potente para resolver problemas de tratamiento de imágenes y son muy utilizadas en biomedicina, teledetección y en muchas otras disciplinas. Algunos **problemas a resolver** son la reducción del ruido, la detección de objetos, realce de contraste análisis de textura y la compresión de los datos. Se ha mostrado más eficaz que otras técnicas utilizadas para ello, como es por ejemplo sobre la transformada de Fourier ya que presentan una mejor localización en tiempo y en frecuencia, e incluso podemos encontrar bases de wavelets con soporte compacto. Existen **librerías destinadas a la práctica** y resolución como por ejemplo Wavelet Toolbox de Matlab.

El funcionamiento básico consiste en transformar la imagen de manera que se subdivide en dos imágenes de menor tamaño una de menor resolución,"**tendencia**" y otra que almacena la información que se refiere a cambios locales en la imagen original, "**las fluctuaciones**".

Con ello se descartan aquellas informaciones que son irrelevantes y además eliminamos el ruido consiguiendo comprimir la imagen.

Hay varios tipos de **familias de Wavelets** (Haar, Daubechies, Coiflets,...) y para la resolución del problema hay que encontrar aquella que mejor se adapte ya que no hay una familia ideal que resuelva todos los casos.

El análisis multirresolución o **codificación multipiramidal** se relaciona directamente con las wavelets ya que si queremos obtener un mayor detalle de resolución necesitaremos mayor número de funciones por unidad de longitud en la base de Wavelets. Burt [64]

Dependiendo de la aplicación a la que se vaya a destinar, podemos utilizar uno de los dos tipos que existen o incluso ambos: **Wavelets continuas** o **Wavelets discretas**.

Las primeras son más complejas por su integración y por tanto dificultad en el tratamiento pero a la vez son de gran flexibilidad y a veces es necesario utilizarlas allí donde las discretas dan resultados que no son los esperados. Por ejemplo, en imágenes digitales se analiza la imagen desde diferentes ángulos con las continuas y con estos parámetros se aplican wavelets discretas para su análisis computacional (Mallat). [56]

Las segundas cuyo origen se remonta al año 1976 cuando Croiser, Esteban y Galand idearon una manera de descomponer de manera discreta señales en el tiempo. Para señales de audio fueron Crochiere, Weber Y Flanagan los que desarrollaron su trabajo para señales de audio.

Aplicaciones son la cartografía, la medicina, ingeniería automovilística etc. [55]

Uso de **operadores morfológicos**:

En la segmentación, cuando se delimitan los objetos, no salen bien delimitados, la clasificación de los pixeles es errónea, regiones solapadas etc. Se requiere una etapa de post-procesamiento en la que los operadores morfológicos cobran importancia. Esta técnica es no lineal y realza la forma y geometría de los objetos. Matemáticamente se fundamenta en **teoría de conjuntos**. Se aplica a primero a imágenes binarizadas y posteriormente a los niveles de grises. Se puede utilizar por tanto al **procesado de la señal**. Su **aplicación** es de restauración de imágenes, detección de bordes, análisis de texturas, aumento y compresión de imágenes

### 2.2.4 Extracción de características y selección

Concluida la segmentación, el siguiente paso es extraer las características de las regiones de interés o útiles para estudiar con más conocimiento el nivel de sospecha. El CAD depende de esta etapa clave para detectar los tumores.

**Buscamos** resaltar las formas, el color, la textura, el tamaño, los bordes, es decir, regiones que puedan en potencia mostrar lesiones que en nuestro caso se traducen en microcalcificaciones.

Trabajando con imágenes mamográficas en la detección de lesiones, lo más normal y habitual será diferenciar entre tejido lesionado y sano. Puede parecer que el problema se reduce considerablemente pero la realidad es que por la dificultad que presenta el problema debido a las múltiples formas que pueden aparecer en los tejidos y especialmente en el sano, esto no es así.

El resultado obtenido por la aplicación de la técnica elegida nos proporcionará un conjunto de prototipos de cada una de las divisiones o clases de las que queramos obtener modelos.

Las diferentes técnicas que se aplican son **de intensidad**, **geométricas o morfológicas y de texturas.**

**Características de intensidad:** la más sencilla basada en el uso de niveles de gris de cada zona para dar la característica de la imagen.

**Características geométricas o morfológicas:** los cálculos se realizan a partir de su área y borde tienen en cuenta la forma de la región de interés. Se trabaja sobre grupos de tejidos que muestran mismos niveles de intensidad en la escala de grises. Las regiones que aparecen como sospechosas se analizan para detectar las microcalcificaciones de manera que dado un nivel de intensidad mayor a un nivel establecido, se realiza la umbralización y se aíslan los grupos buscando agrupaciones con alguna técnica técnicas como la de **Convex Hull**.

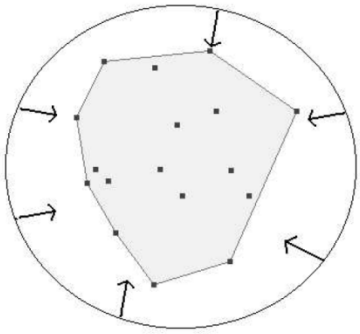


Figura 35 Idea intuitiva de envolvente convexa de 17 puntos en el plano.

La envolvente convexa, envoltura convexa o cápsula convexa de un conjunto X, de puntos en el plano euclidiano o espacio euclidiano es el conjunto convexo más pequeño contenido en X. Por ejemplo, cuando X es un subconjunto acotado del plano, la envolvente convexa puede ser visualizada como la forma encerrada por una banda elástica estirada alrededor de X. [91].

De manera formal la podemos definir como la intersección de todos los conjuntos convexos que contienen X o como el conjunto de todas las combinaciones convexas de puntos en X. Con esta última definición, cierres convexos pueden extenderse desde los espacios euclidianos a espacios vectoriales reales arbitrarios; también pueden ser generalizados más allá, a matroides orientadas. [92]

Dados k puntos x 1, x 2, . . . , x k su envolvente convexa C viene dada por la expresión:

$$C(x) = \{\sum_{i=1}^{k} \alpha_i x_i \mid x_i \in X, \alpha_i \in \mathbb{R}, \alpha_i \geq 0, \sum_{i=1}^{K} \alpha_i = 1\} \qquad (4)$$

**QuickHull** (S)
{
// Encuentra envolvente convexa del conjunto S de n puntos

Convexa del casco: = {}
Encuentra más puntos izquierdo y derecho, por ejemplo A y B, y añadir A y B de casco convexo
Segmento AB divide a los (N-2) puntos restantes en 2 grupos S1 y S2
donde S1 son puntos en S que están en el lado derecho de la línea orientada de A a B,
y S2 son puntos en S que están en el lado derecho de la línea orientada de B a A
FindHull (S1, A, B)
FindHull (S2, B, A)
}

**FindHull** (Sk, P, Q)
{
// Encuentra puntos de envolvente convexa del conjunto de puntos de Sk
// Que están en el lado derecho de la línea orientada de P a Q

Si Sk no tiene ningún punto,
luego regresar.
Del conjunto dado de puntos en Sk, encontrar punto más lejano, digamos C, desde el segmento PQ
Añadir el punto C al casco convexo en el lugar entre P y Q
Tres puntos P, Q, y C partición los puntos restantes de Sk en 3 subgrupos: S0, S1, y S2
donde S0 son puntos dentro del triángulo PCQ, S1are puntos en el lado derecho de la orientada
la línea de P a C, y S2 son los puntos en el lado derecho de la línea orientada de C a P.
FindHull (S1, P, C)
FindHull (S2, C, Q)
}

**Salida** = casco convexo

Figura 36 Un algoritmo para crear la envolvente convexa es por ejemplo Quick Hull [93]:

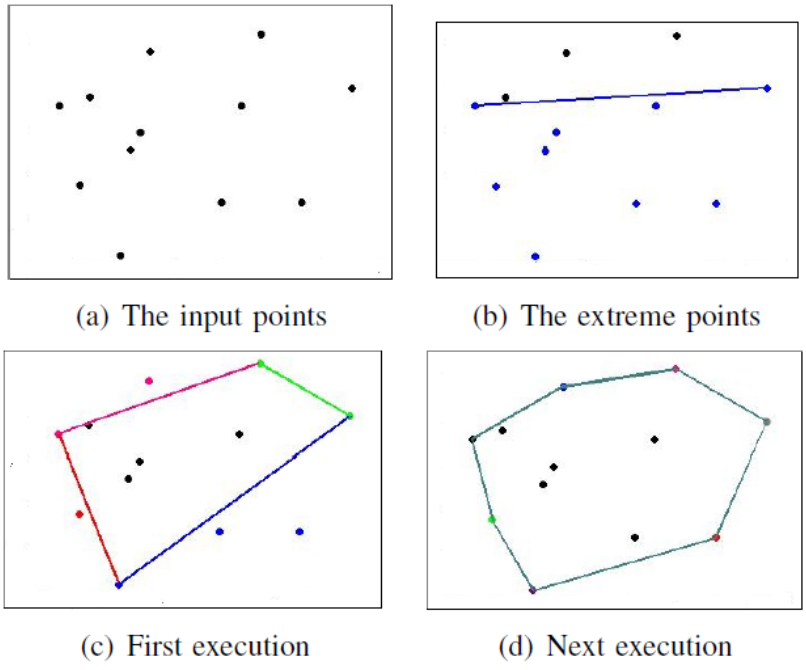



Figura 37 Resultado de la ejecución de algoritmo Quick Hull [94]:

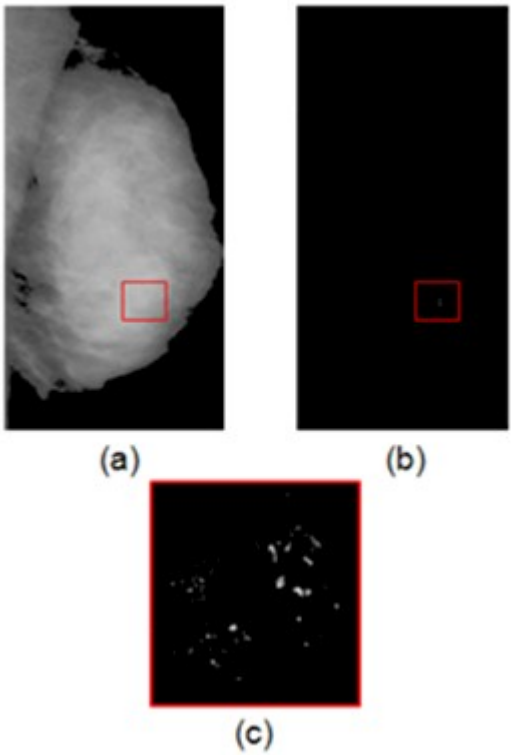



Figura 38 Ilustración del proceso para detectar microcalcificaciones, (a) muestra la imagen después del paso I, (b) se aplica el umbral a la imagen previa y (c) se muestra un acercamiento a la región segmentada.

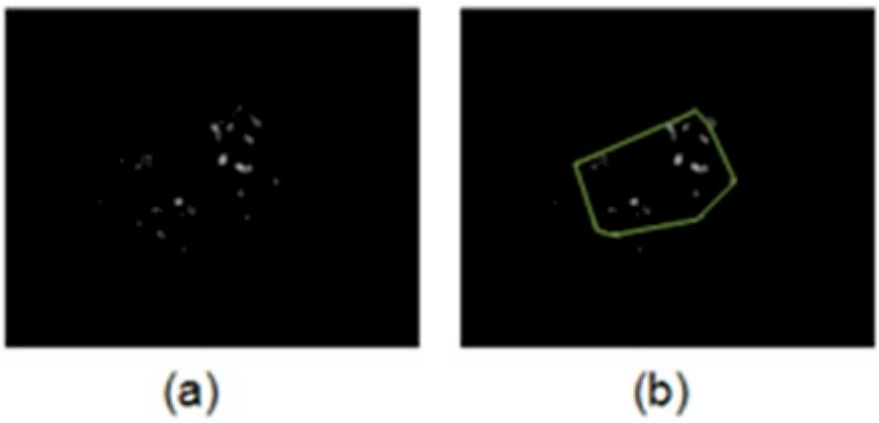



Figura 39 En estas imágenes se ilustran como se delimita un agrupamiento de microcalcificaciones, en (a) determinación de la región de interés y en (b) la región queda encerrada aplicando el método Quick Hull

**Características de texturas.** Se utiliza distribución estadística en la que se pueden observar distintas intensidades en situaciones relativas de la imagen. **Los estadísticos pueden ser** de **primer orden** (media, desviación estándar, varianza), **segundo orden** basadas en la **matriz**

**de coocurrencia** o por ultimo de **orden superior** atendiendo a la cantidad de pixeles que se consideran en cada combinación [88].

Para la característica de textura se utilizan matrices que permiten el estudio de la relación espacial entre grupos de datos con técnicas estadísticas. Estas matrices son de coocurrencia o también llamadas de dependencia de niveles de gris o su denominación en ingles **space gray level dependence matrices** (SGLDM). Con estas matrices se consiguen texturas de segundo orden.

En su funcionamiento elemental , una matriz M de este tipo se puede describir como una matriz cuadrada cuya característica es que el número de filas y de columnas coincide con el número de niveles de grises (G) de la imagen que queremos analizar y donde cada uno de los elementos M (i;j) tiene su la frecuencia relativa con la que dos pixeles en la imagen i y j respectivamente y distanciados por un espacio d y ángulo x, ocurren en una determinada vecindad.[89].

El analista requerirá de **cinco variables** a definir para generar **imágenes de textura** : **Tamaño de la ventana, banda espectral de entrada, las texturas derivadas, cuantización del canal de salida, la componente espacial** (la distancia interpixel y el ángulo para el computo de la co-ocurrencia).Como resultado obtendremos un numero único que representará la ventana completa que se coloca en el lugar del pixel que ocupa el centro. Esta ventana se mueve un pixel y el proceso se repite calculando de nuevo una matriz de coocurrencia. La imagen se construye con valores de texturas. Además si tiene dimensión NxN , alrededor de la imagen en el espacio (N-1)/2 píxeles permanecerá sin resultados. Como habitualmente son una franja menor del total no presentará mayor problema aunque habrá que tenerlo presente para el caso en que sea pequeño o demasiado grande la ventana.

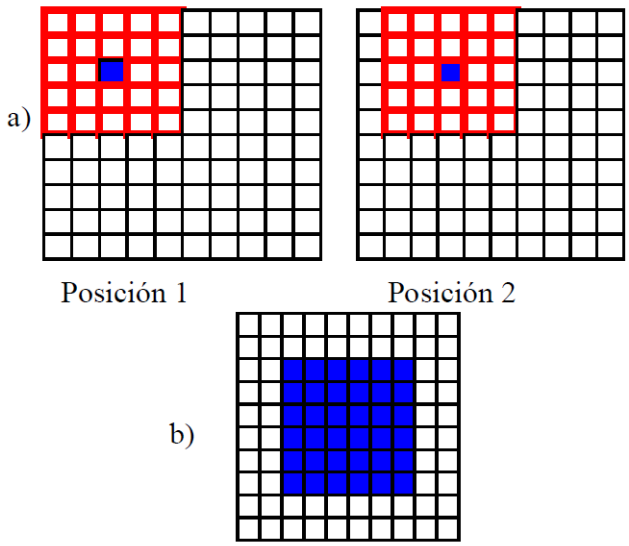



**Figura 40 En a) se muestra sobre una imagen de dimensiones 10x10 pixeles, una ventana móvil de 5x5 en dos posiciones y el pixel central que recibe el resultado. En b) los pixeles que reciben el resultado y las 2 filas y las 2 columnas del borde de la imagen que reciben el valor de textura calculados en los pixel cercanos.**

Se puede dar la paradoja de que si hay una cantidad elevada de niveles de grises en la imagen, las matrices de coocurrencia podrían ser mayores que la imagen origen y además podrían ser muy sensibles al tamaño de la ventana de vecinos. Es deseable que la matriz no tenga valores dispersos es decir que se mantenga un elevado valor de ocupación que dependerá del número de niveles de grises tenidos en cuenta. Además es importante intentar reducir la dimensión de los puntos que se van a clasificar porque será más rápido el clasificador y necesitará un menor requerimiento de la memoria [90] .

En cuanto al tamaño de la ventana debe ser menor que el objeto pero con lo suficiente para que se capture la variabilidad del mismo. La relación espacial del pixel con su vecino se puede dar en 8 direcciones , los cuatro puntos cardinales y las diagonales. Para una sola aplicación se pueden tratar más de 5000 imágenes de textura, sobrepasando a la gran mayoría de clasificadores.

### 2.2.5 Clasificación

Se aplican distintos **tipos de clasificadores** para determinar el grado de la sospecha bien por separado o de manera más habitual de forma combinada en **paralelo, cascada o de manera jerárquica** y los resultados se obtienen con marcas de clasificación como asteriscos o marcas ovaladas en las masas y triángulos o rectángulos en las microcalcificaciones de manera superpuesta en la imagen.

Ahora, con imágenes que no han sido utilizadas para el entrenamiento se obtendrán sus características y aplicando reglas de decisión se clasificará en función de la probabilidad de pertenecer a una determinada clase.

### 2.2.6 Técnicas CAD aplicadas al problema de las microcalcificaciones

Diversas técnicas CAD aplicadas al problema de las microcalcificaciones son: **Análisis multirresolución mediante wavelets 2D, Técnicas de diferenciación de imágenes, Técnicas basadas en el uso de operadores morfológicos, Lógica borrosa, Teoría de fractales, Aprendizaje profundo.**

Con la técnica de procesado de la imagen se intenta detectar las diferentes tipos de lesiones ya mencionadas en apartados anteriores como masas o nódulos, calcificaciones, distorsiones. Para el **procesado** se puede utilizar **clusterización por lógica difusa, filtros adaptativos y transformada wavelets**.

Al objeto de ayudar al especialista, N. Al SAiyd realiza una descripción muy completa de **técnicas sobre dominio espacial** (procesamiento de pixel o de ventana) con técnica de lógica borrosa, de dominio de frecuencia con transformada de wavelets basadas en filtrado de coeficientes o de selección basadas en procesamiento de ventana o mixtas que combinan wavelets bien con análisis de histograma de los coeficientes o bien con operadores morfológicos (Al-Saiyd)[242].

Investigadores que realizan **técnicas mixtas** son A. Mencattini que utiliza operadores morfológicos a las masas y wavelets a las microcalcificaciones (Mencattini)[41] y N. H. kumar que filtra en primer lugar mediante coeficientes Wavelets y a posteriori realiza análisis morfológico en la zona de baja frecuencia de la imagen y ya por último realiza la combinación para llegar al resultado final (Kumar)[243].

Los diferentes métodos que están basados en **umbrales de los coeficientes** aplicados a microcalcificaciones tienen mayor distorsión pero ganan contraste. Por el contrario los que utilizan **correlación** son superiores pero ignoran microcalcificaciones.

En la **segmentación** se aíslan las zonas sospechosas analizando los pixeles o las regiones, en una etapa posterior se extraen las características de estas regiones determinando si son sospechosas o normales y en la **clasificación** se valida con métodos como textura o discriminación que sea lesión y no tejido propio de la mama.

Al analizar la mamografía, se toman en cuenta distintas variables tales como si existen asimetrías en áreas de tejido de densidad fibroglandular más extensas en una mama que en otra (Sickles)[244]. La asimetría nos puede dilucidar que hay **presencia de masa** o que varía la normal distribución de un tejido (Malagelada)[245].

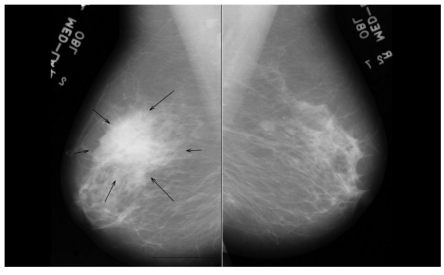


Figura 41 Ejemplos de asimetría que se puede apreciar en la proyección MLO izquierda

Si hay microcalcificaciones, entonces si son de tipo irregular, polimórficas y con variaciones de tamaño se suelen asociar a **procesos cancerígenos** (Bozek, Mustra, Delac, & Grgic, 2009; Shinde, 2003). [237]

A continuación describimos algunos **sistemas comerciales CAD** aprobados por la FDA (Food and Drug Administration, USA) que se aplican para auxiliar al radiólogo son (Sampat, Markey, & Bovik, 2005):

- R2 Technology´s Image Checker. Primer Sistema aprobado por la FDA y utilizado en la detección de masas y calcificaciones. Fue el primer sistema que se centraba en la mamografía digital es del 1998. En 2006 Hologic. Inc. Adquirió la tecnología
- MamnoReader. De mayor ámbito de detección de lesiones, aplicado a masas , calcificaciones, distorsión de la arquitectura y asimetría.
- Second Look. Destinado a la detección de masas y calcificaciones. Desarrollado por CADx Medical System y aprobado en 2006 por FDA

Otros sistemas destacados:

- Mammex TR ( Scanis , INC, Fostre city, ca , USA) : desarrollado en conjunto con Center for Medical Image Analysis of TRW Healtcare Solutions (Pleasanton, CA, Usa ; busines unit of TRW, Inc., Cleveland , OH, USA).
- Kodak Mammography computer-Aided Detection System (Kodak health group now Carestream health, INC. Rochester, NY , USA)
- ERLM Fuji
- IssiCAD (Intelligent system, Inc.)
- Image Clear (Titan Systems, Inc.)

- El 18.02.2011 La FDA aprueba el primer dispositivo para mamografías en 3D. Se trata del sistema Selenia Dimensions, una actualización del sistema de dos dimensiones aprobado por la FDA de Hologic Inc., y puede ayudar a los médicos a detectar y a diagnosticar con más precisión el cáncer de mama [95][96]

La **diferencia ente CADx Y CAD´s** es que los primeros son de mayor complejidad ya que además de la localización de las lesiones, realizan otras funciones como graduar su malignidad o benignidad y caracterizarlas.

Las **métricas que se utilizan en un sistema CAD** son (Bozek, Mustra, Delac, & Grgic,)[237] son:

- **TP** es la marca que realiza el sistema y que se corresponde con la localización de la lesión.

- **FP** o falso positivo. Es la marca que realiza el sistema CAD y que se no corresponde con la localización de una lesión.

- **Matriz de confusión** formada por verdaderos negativos (TN), verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN).

- También la **Exactitud, Precisión, Sensitividad e Índice de FP**

$$C = \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix} \tag{5}$$

$$Exactitud = \frac{TP + TN}{TN + FP + FN + TP} \tag{6}$$

$$Precisión = \frac{TP + TN}{FP + TP} \tag{7}$$

$$Sensitividad = \frac{TP}{FN + TP} \tag{8}$$

$$FPI = \frac{FP}{TN + FP} \tag{9}$$

Zhen y cols comprobaron que usando sistemas CAD podían obtener **mejores resultados** en la detección reproduciendo verdaderos positivos (TPF) sin falsos negativos (FN) que falsos positivos.

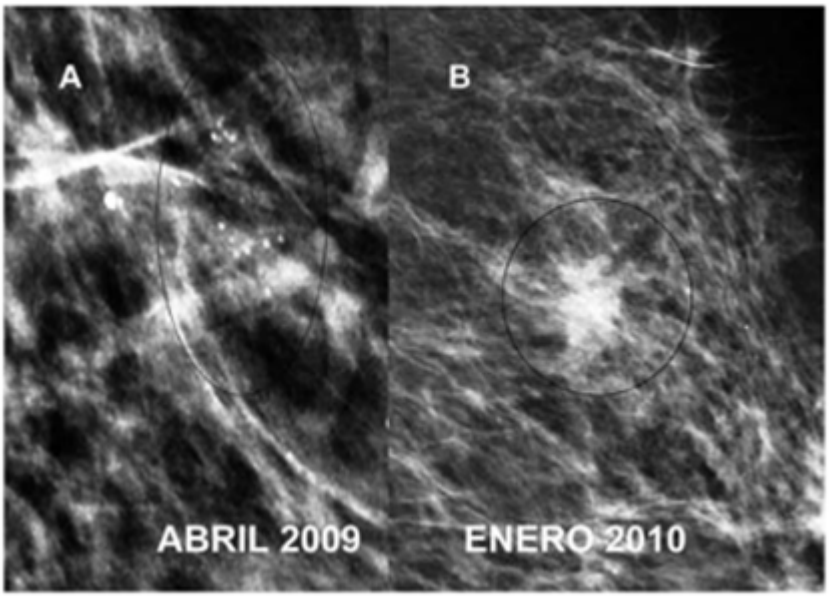



**Figura 42 Falso negativo, mamografías. a) Calcificaciones centrales previas a punción en mama izquierda. b) Control nueve meses después del procedimiento, se agregan calcificaciones y densidad del área biopsiada.**

Baker y cols demostraron que la **variabilidad del resultado** está en función del número de análisis que se realizan. A pesar de existir variabilidad en la reproducibilidad del CAD, Warren y Cols comprobaron que utilizando estos sistemas **se detectan el 75% de los sistemas que no detecta el radiólogo**.

Taylor y cols vieron que la variabilidad también depende de los pequeños cambios de posición de la placa entre una y otra digitalización que se efectúa de manera secuencial en el proceso de digitalización.

Baum y cols describe que la **mamografía digital de campo completo tiene como ventaja** que al no tener que digitalizar las placas, no se añade ruido y además las marcas aparecen una vez se ha adquirido la imagen por lo que ahorra también tiempo de lectura. Otros factores que influyen son el número de marcas o lesiones que se toman en cuenta en el estudio así como la **sensibilidad que en mamografía digital** es del 89% y en masas del 81% ; Un sistema poco sensible y con alta tasa de FP en una BBDD proporciona poca ayuda al radiólogo.

Los sistemas han ido evolucionando y progresando en mayor medida en cuanto lo ha hecho también la tecnología permitiendo un aumento en la potencia de cálculo y mejor tratamiento de la imagen e incluso algunos autores creen que puede hacer que los radiólogos bajen el porcentaje de aciertos debido a una influencia negativa del sistema y a una disminución en la agudeza del diagnóstico.

Hay trabajos que certifican que **la suma de la lectura de cribado del radiólogo en combinación con el CAD** aumenta en un 21% la sensibilidad y por tanto aumenta la detección y se reducen los canceres que el radiólogo no era capaz de diagnosticar. Equivaldría a una doble lectura, es decir que dos radiólogos leyesen la misma mamografía solo que es económicamente más rentable.

Taylor y cols sin embargo postulan que el **sistema CAD no es de gran ayuda** ya que no mejoran los porcentajes de detección. Además se publicó un estudio con dos grupos de lectura, uno con CAD y otro sin su ayuda. El resultado fue que el que la lectura se apoyaba con CAD dio menos sensibilidad que el grupo que leía sin él por lo que en cualquier caso lo que es evidente es que queda **mucho trabajo por hacer**.

Cheng ofrece otra técnica de preprocesamiento para las microcalcificaciones, la **lógica difusa** (fuzzy logic). Se utiliza un algoritmo que se fraccione en **cinco fases** de las cuales las tres primeras son en preproceso; Fuzzycation, mejora, supresión de estructuras insignificantes mediante operadores morfológicos aplicados al tamaño, forma y luminosidad, segmentación y por último reconstrucción. Con este método se consigue **realce de microcalcificaciones**. (Cheng) [53]

Highnam nos proporciona preprocesamiento orientado a facilitar la **localización de artefacto**s que son fácilmente confundidos con las microcalcificaciones. Se sustenta en la idea de que en la imagen se conforma un pequeño emborronamiento de todos los elementos anteriores a la impresión de los artefactos. Detectando estos elementos permiten eliminar los artefactos y se generan menor cantidad de falsos positivos (Highman 1999) [54]

Repasamos **distintos estudios realizados sobre procesado**:

Hay realizados estudios sobre la **mejora de imágenes mamográficas** los métodos clásicos de Unsharp, masking, filtrado espacial, multiescala etc. Que resaltan el fondo , las microcalcificaciones , la estructura del propio tejido de la mama y el ruido. También describe el uso de métodos de aproximaciones globales realizados en la imagen completa no teniendo presente las singularidades de cada región. Otra técnica descrita es el **uso de la estadística** para el pre procesamiento en el entorno de cada pixel para estimar el fondo, eliminarlo y aumentar el contraste.[43]

Otra propuesta se basa en **entornos de tamaño variable para cada pixel** de tal forma que se exige adaptación al fondo. [44]

Un método ampliamente utilizado es el de **expansión de luminosidad o de contraste** ajustando el histograma de la imagen mamográfica con el fin de que se creen mayores separaciones de las distribuciones entre el fondo de la imagen y la microcalcificación. Se reducen el contraste en las zonas más oscuras y se realzan las más luminosas.

Otros estudios revelan **diferentes formas de abordar el problema**:

La modificación de contraste lineal por **tramos de pendiente variable**. [45]

La modificación de contraste no lineal con una **función continua de curvatura variable**.[46]

La **redistribución de los niveles de grises** con la finalidad de hacer máxima la información de la imagen.

La **supresión del fondo** que se realiza detectando los niveles de variaciones suaves y eliminándolas respetando las transiciones bruscas. Con esta técnica logramos un aumento de la variación de niveles de gris realzando las microcalcificaciones.

**Unsharp masking**, que es la técnica que suprime el fondo de la imagen. Esto se consigue restando la imagen aproximada de la imagen original. Métodos que usan Unsharp son: El filtrado espacial y métodos no lineales. [47]

Gurcan utiliza el **filtrado adaptativo** estimando el fondo mamográfico de manera que las microcalcificaciones no se estiman por lo que pertenece a la **imagen de sustracción**. (Gurcan) [48].

Dengler lo hace mediante dos fases sustracción de la imagen original usando **un filtro gaussiano de ancho determinado con aproximación paso bajo** y por otro lado se procesa la imagen para suprimir todo aquello que no sean microcalcificaciones. Aparece el término de

umbralización para luminosidad muy superior a la media. Se basa en el **conocimiento de los elementos buscados**. (Dengler). [50]

Morrow trabaja un método adaptativo que tiene en cuenta **tamaño y forma** para realizar el contraste. Con el valor de la semilla usando cada pixel para calcular el crecimiento de cada región, contraste y fondo se mejora la imagen y realza las microcalcificaciones [49] (Morrow).

Li en 1997 con un **algoritmo complejo basado en teoría de fractales** obtendrá la imagen que usará para sustraer a la imagen original. Para detectar las microcalcificaciones se utiliza un umbral (li) [51].

Kook propone **dos fases** para lograr el objetivo: una eliminación con un algoritmo que utiliza la **mediana** y un **cálculo de gradientes con peso y suma de información** a la imagen original usando distintos operadores de la derivada de primer orden para hallar las microcalcificaciones.

También se usan **técnicas de wavelets en preprocesamiento** como ya se ha indicado anteriormente. Un ejemplo es la propuesta de Aydin (Aydin 1999). [52]

## 2.3 Técnicas utilizadas en inteligencia computacional

El aprendizaje se relaciona con el conocimiento. Podemos definirlo como el proceso mediante el cual un ente adquiere conocimiento. Este conocimiento puede ser transmitido por otro ente profesor o puede adquirirse sin la ayuda del mismo. El aprendizaje se traduce en cambios de adaptación que permiten realizar la tarea de una manera más eficiente.

Es fundamental para la resolución de problemas de una manera ágil y en definitiva se sirve de la experiencia propia para resolverlos con mejor resultado a futuro.

Ada Augusta, filósofa de la computación (1961) aseguraba que la máquina analítica no pretende crear nada, puede hacer lo que se le programa si se le ha indicado como hacerlo.

Existen **diversas formas de aprendizaje** [159]:

- Aprendizaje memorístico
- Aprendizaje a través de consejos
- Aprendizaje mediante experiencia en la resolución de problemas
- Aprendizaje a partir de ejemplos o Inducción
- Aprendizaje basado en explicaciones
- Descubrimiento
- Analogía
- Redes Neuronales.

A nosotros nos interesa sobre todo el **Aprendizaje automático** o Machine Learning es una especialidad del soft computing desarrollado a partir de estudios en reconocimiento de patrones y teoría del aprendizaje en inteligencia artificial (IA) y que pretende que el computador aprenda de forma autónoma. [111]

En 1959 Arthur Samuel define este aprendizaje como un "campo de estudio que da a las computadoras la capacidad de aprender sin ser programada de forma explícita". [112].

Es un proceso que es capaz de elegir una sola función para dar respuesta al problema que se plantea de aprendizaje partiendo para ello del conjunto de entrenamiento.

Los algoritmos operan construyendo un modelo basado en un **conjunto de entrenamiento** con observaciones de entrada cuya finalidad es poder hacer predicciones basadas en estos datos y tomar las decisiones adecuadas expresadas como salidas, en lugar de seguir estrictamente las instrucciones del programa estáticas.

La **inferencia inductiva** que en definitiva es otra forma de nombrar al problema del aprendizaje, se ha abordado a lo largo de la historia por filósofos pero fue a partir del siglo XVIII cuando se formuló gracias al Filósofo escocés **David Hume**. Hasta el siglo XX se piensa que es imposible la inducción pura salvo que se ayude de conocimiento a priori. Ya en el S XX, **Karl Poper**, filósofo austriaco nacionalizado más tarde británico, supera la polémica entre empirismo y racionalismo defendiendo que la teoría antecede al hecho y que además la teoría necesita refutarse con la experiencia para conocer que teoría es correcta y cual no lo es.[144].

En el aprendizaje visto desde la óptica de la estadística, **Gauss** en el SVIII propone la regresión de mínimos cuadrados y **Ronald Fisher** [145], estadístico y biólogo, es considerado el padre de la estadística moderna y diseño experimental. A partir de 1919 desarrolla el análisis de la varianza y hace grandes aportaciones en clasificación que sirve hoy en día de numerosos estudios de métodos y análisis.

En Inteligencia Artificial , **Alan Turing** ,matemático, lógico, científico de la computación, criptógrafo y filósofo , mundialmente conocido sobre todo por la concepción del test de Turing (1950), que permite juzgar la inteligencia de una máquina si sus respuestas en la prueba no se pueden distinguir de las que daría un ser humano confundiendo a la maquina con este y mostrando una cualidad relevante en máquinas de aprendizaje "el maestro desconocerá lo que sucede en el interior de su alumno pero será capaz de predecir en alguna medida su comportamiento".[146].

Los problemas de aprendizaje modelados desde el punto de vista de hallar problemas de búsqueda en espacios hipotéticos se abordan desde la **Inteligencia Artificial**. Dentro de este campo, se desarrollaron algoritmos para el aprendizaje automático dando lugar con el tiempo a un área especializada denominada máquinas de aprendizaje.

El aprendizaje automático aborda la cuestión de cómo construir equipos y **programas que mejoran automáticamente a través de la experiencia**. Es uno de los campos de estudio con

gran proyección de futuro y en el que hoy en día cuenta con un mayor desarrollo técnico; es una disciplina que va a caballo entre la informática y la estadística y que además tiene fuertes vínculos con la optimización matemática, métodos, teoría y dominios de aplicación en el campo. Por ello es una disciplina enfocada en dos cuestiones relacionadas entre sí: ¿Cómo pueden los sistemas informáticos automáticamente mejorar a través de la experiencia? y ¿Cuáles son las leyes fundamentales de la estadística que gobiernan cualquier sistemas de aprendizaje, incluyendo a las maquinas, los seres humanos, y las organizaciones?

Este campo ocupa un lugar muy importante en el campo de la inteligencia artificial y del tratamiento de grandes bases de datos (Big Data).

Los recientes progresos en el aprendizaje automático han sido impulsados tanto por el desarrollo de nuevos algoritmos de aprendizaje , como por la gran accesibilidad y disponibilidad de búsqueda de datos en línea así como la computación de bajo costo al alcance de muchos investigadores. El uso de los datos de manera intensiva ha influido notablemente en el auge y desarrollo de los métodos de aprendizaje automático que se aplican de manera habitual a la ciencia, la tecnología y el comercio, basados en toma de decisiones a través de muchos ámbitos de la vida, incluyendo el cuidado de la salud, manufactura, educación, modelos financieros, modelos de inteligencia militar y policial, comercial etc.

**Ejemplos de aplicación** son [113] [120]:

- Diagnósticos médicos
- Detección de fraude en el uso de tarjetas de crédito
- Análisis del mercado de valores
- Clasificación de secuencias de ADN
- Reconocimiento del habla y del lenguaje escrito

- Robótica

- Conducción de vehículos. Ej. ALVINN 100 Km/hora durante 150 Kms en autopista, RALPH, más de 450 kms desde Pittsburgh hasta San Diego
- Reconocimiento del habla. Ej. SPHINX Reconocimiento de fonemas y de palabras
- Jugador de backgammon de nivel mundial Ej. TD-Gammon
- Clasificación de estructuras astronómicas
- Problemas donde no hay expertos humanos
- Importancia económica de qtls en secuencias genéticas
- Tareas que los humanos hacemos pero no podemos describir cómo las hacemos
- Reconocimiento de caracteres
- Problemas donde la función objetivo cambia frecuentemente
- Predicciones bursátiles
- Situaciones donde cada usuario tiene su propia función
- Adaptación de interfaces, spam, gestión de correo electrónico
- Problemas donde el volumen de datos hace imposible a los humanos realizar cualquier análisis
- Técnicas de Data-Mining capaces de encontrar relaciones dentro de grandes BDs intratables
- Carciogénesis de compuestos químicos
- Perfil de clientes
- Concesión de créditos hipotecarios
- Sistemas automáticos de filtrado (correo, noticias)
- Entornos personalizados (buscadores, ventas,...)
- Construcción automática de portales web
- Análisis de textos
- Sistemas de reconocimiento de texto y voz
- Robots (conductores, carteros, jugadores de fútbol,...)

- Juegos Ej. DEEP-BLUE

El aprendizaje automático es sumamente importante tanto para hacer frente cuestiones que son hoy en día, fundamentales para disciplinas científicas y de ingeniería.

Hemos progresado espectacularmente en las últimas dos décadas. Dentro de la especialidad en inteligencia artificial (IA), el aprendizaje automático se ha convertido en el **método de elección principal** para el desarrollo de software en la práctica de visión artificial, reconocimiento de voz, lenguaje natural procesamiento, robótica, y muchas otras aplicaciones.

Desde la década de los 80 del siglo pasado, **los médicos cuentan con programas que les ayudan a tomar decisiones** y la mayoría basados en inteligencia artificial y el aprendizaje automático (redes neuronales artificiales, razonamiento basado en casos y clasificadores bayesianos).

Ejemplo de ello en España es el **proyecto IMED** en él que han colaborado el Centro Extremeño de Tecnologías Avanzadas del Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CETA-CIEMAT), la Facultad de Medicina de la Universidad de Oporto - Hospital de São João (FMUP-HSJ) y el Instituto de Ingeniería Mecánica y Gestión Industrial (INEGI) de esta universidad. La financiación del proyecto IMED se ha realizado conjuntamente por las tres instituciones, siendo la parte correspondiente a la aportación del CETA-CIEMAT cofinanciada por el Fondo FEDER. [108].

Este proyecto cuenta con una base datos de 1740 imágenes médicas en la actualidad [109], la aplicación CAD Mammography Image Workstation for Analysis and Diagnosis (MIWAD) con visor de imágenes y clasificadores automáticos, Machine Learning Classifiers (MLC), que se usan para diagnostico como apoyo al especialista medico en segunda opinión.

Grandes empresas tecnológicas como **Google, Microsoft o Facebook** usan aprendizaje profundo para mejorar sus sistemas de búsqueda, reconocimiento de voz, visión por computador.

En septiembre de 2014, Facebook creó un grupo de expertos para trabajar en esta disciplina y Google cuenta, desde 2010, con un departamento que es conocido como '**Google Brain'** y tienen entre sus empresas la starptup Deepmind Tecnologies, en la que trabajaban algunos de los mayores expertos en 'deep learning'.

La evolución de los **procesadores inspirados en el cerebro** (Chips neuromórficos) tuvieron su comienzo a principios de 1980, con el profesor Carver Mead del Instituto Tecnológico de California (Caltech, en EEUU). En aquellos momentos la tecnología existente para la fabricación era muy pobre y su esfuerzo no obtuvo recompensa en este sentido hasta que años más tarde ha sido posible por la evolución de la tecnología.

En Agosto de 2014 **IBM** diseño un chip neuromórfico que emula los principios de funcionamiento del cerebro y sigue trabajando en una versión para hacer más inteligentes a los dispositivos móviles en tareas fáciles para el cerebro pero difíciles para un computador como son reconocimiento de voz y la interpretación de imágenes. El **chip TrueNorth** tiene el tamaño aproximado de un sello y un millón de "neuronas" de silicio con 256 millones de conexiones entre ellas que son similares a las sinapsis que conectan las neuronas en un cerebro real. El consumo es mucho más bajo , alrededor de **mil veces menos energía** que otro convencional de parecido tamaño. IBM ha demostrado su funcionamiento en tiempo real en el reconocimiento de vehículos a través de una grabación de vídeo.[110]

Muchos desarrolladores de sistemas dedicados a la inteligencia artificial reconocen que puede ser mucho más fácil entrenar a un sistema con ejemplos del comportamiento basados en la entrada-salida deseada que programarlo de forma manual teniendo en cuenta y previendo todas las posibles respuestas para todas las entradas posibles de manera que se obtenga la respuesta deseada.

Existen gran variedad de algoritmos de aprendizaje automático que han sido desarrollados para resolver la gran variedad de problemas existentes.

La **eficiencia de un algoritmo de aprendizaje** en su aplicación para la resolución de los problemas que se le presentan varían en gran medida dependiendo del tipo de problema, del tamaño, la calidad y la naturaleza de los datos. [105]. y observar que en muchas ocasiones solo se podrá comprobar cuál es más eficiente solo mediante comparación empírica.

Hay sistemas que pretenden eliminar cualquier dependencia de intuición o conocimiento del experto en el proceso de análisis y otros por el contrario lo que tratan es de llegar a un acuerdo de colaboración entre humano y el computador. En cualquier caso, siempre es necesario que el diseñador del sistema especifique como se representan los datos, como se caracterizan y los métodos de manipulación.

### 2.3.1 Cronología e Historia

Aunque esta disciplina tiene una corta historia, podemos distinguir **tres periodos importantes** centrados en paradigmas distintos [116]:

- Técnicas de modelado neuronal y de decisión
- Aprendizaje orientado a conceptos simbólicos
- Sistemas de aprendizaje de conocimiento con exploración de varias tareas de aprendizaje

La misión de la técnica en aprendizaje de **modelado neuronal**, consiste en la construcción de sistemas de propósito general cuyo conocimiento inicial de la estructura es prácticamente nulo o con una estructura inicial aleatoria o parcialmente aleatoria.

Los sistemas se basan en redes neuronales artificiales y a menudo se les denomina **sistemas Auto-organizativos**. Aprenden mediante cambios incrementales con probabilidad de que las neuronas sean capaces de transmitir una determinada señal.

En los comienzos, la tecnología era muy básica y las investigaciones estaban limitadas a teorizar o construir sistemas de HW específicas como los **perceptrones** [Rosenblatt, 1958], pandemonium [Selfridge, 1959] y adelaine [Widrow, 1962].

Los trabajos se fundamentaron **en los años 40** del pasado siglo por Rashe**vsky y sus seguidores del área de biofísica McCulloch y Pitts al descubrir que la** lógica simbólica era útil en **modelados del sistema nervioso**.

Otros investigadores que contribuyeron en la décadas de los 50 y 60 fueron Ashby Rosenblatt, Minsky y Papert, BlockYovits, Widrow, Culberson, Kazmierczak.

Friedberg en 1958 y décadas más tarde Holland en 1980 contribuyeron en otra variante de **modelos**, como los **evolutivos**, implementados a través de operaciones aleatorias de mutación y de selección natural.

El trabajo de estos investigadores sirvió en la década de los 60 para que en base a la experiencia, naciera una **nueva disciplina** basada en el reconocimiento de patrones mediante funciones de tipo polinomial, lineal o formas que se relacionan con funciones discriminantes a partir del conjunto de ejemplos utilizados para la etapa de entrenamiento Nilsson, Koford; Uhr; Highleyman.

En 1959 Samuel [118], propuso un **sistema para jugar a las damas** que estaba programado para aprender jugando adquiriendo un mayor nivel en el juego.

En la década de los 60 y 70 se investigan técnicas que usan **métodos estadísticos** para tomar la decisión en el aprendizaje de las **reglas de reconocimiento de patrones** Sebestyen; Fu; Watanabe; Arkadev, Fukananga, Ruspini, Duda & Hart, Bezdek, Kanal.

En paralelo al modelo de las redes neuronales y sistemas de decisión se investigaron **teorías y sistemas de control que se adaptaban ajustando de manera automática sus parámetros** para mantener configuraciones que se comportaban de manera estable ante posibles perturbaciones. Trabajaron en este modelo los investigadores Truzal en 1955 y Davies, Mendel, Tsypkin, Fu en los años 70.

Los resultados basados en redes neuronales ofrecieron algunas limitaciones como revelaron trabajos como el de **Minski y Paper** en 1969 y las expectativas que ofrecían trabajos teóricos eran pobres por lo que se dejó de investigar en este campo.

Hunt en los años 60 junto a otros investigadores como Hovland, Feigenbaum, Hunt, Hilgard, Simon & Lea, Bongard, Uhr, Karpinsky y Michalski, y psicólogos de la época potenciaron el modelado basado en el aprendizaje humano basados en trabajos que usaban **estructuras de grafos y lógicas** en lugar de métodos numéricos o estadísticos.

En 1975 Winston [119], describe influenciando de modo importante en este área de trabajo con uno de los primeros **programas de aprendizaje estructural**. Su trabajo consistió en construir conceptos del dominio del mundo de los bloques, por ej.: casa, arco, etc. aprendiendo a partir de estructuras más sencillos como: ladrillo, cilindro, etc.

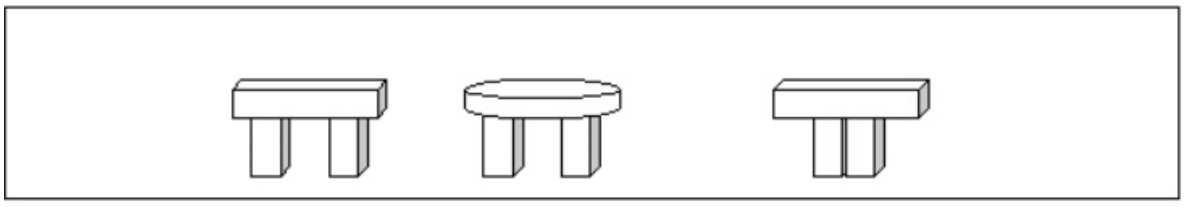


Figura 43 Estructuras de Wiston

También fueron influyentes los programas de los 70 sobre aprendizaje que se fundamentan en **lógica inductiva** Michalski, Hayes-Roth, Hayes-Roth y McDermott, Mitchell.

A partir de los años 70, el trabajo de investigación se orienta al aprendizaje de conceptos a partir de ejemplos y también de instrucciones destinados a **sistemas que se basan en el**

**conocimiento** y surgen una buena cantidad de técnicas incorporando **heurística y ejemplos para el entrenamiento**.

### 2.3.2 Clasificación:

Algunos sistemas de aprendizaje automático intentan suprimir cualquier necesidad de conocimiento del experto en el análisis de datos, otros tratan de establecer un marco de colaboración entre el experto y la computadora. Hoy por hoy, la base de **la intuición humana no se puede sustituir totalmente** porque el diseñador del sistema ha de concretar la forma en que se representan los datos así como los métodos de manipulación y caracterización de los mismos.

#### 2.3.2.1 Tipos de aprendizaje

Podemos realizar varios tipos de clasificaciones. Atendiendo a los **tipos de aprendizaje** pueden clasificarse en:

**Inductivo**: en el que modelamos conceptos generalizando ejemplos sencillos buscando patrones comunes que den explicaciones a los ejemplos.

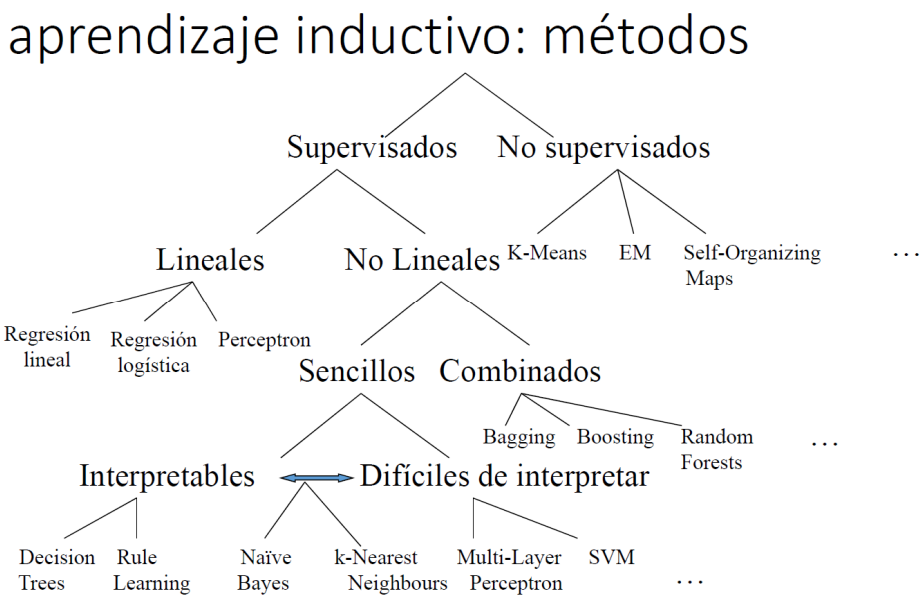



Figura 44 Aprendizaje inductivo: métodos

**Deductivo**: Se aplica la deducción para modelos generalistas partiendo de ejemplos conceptuales y su explicación.

**Genético**: Se basa en la teoría de la evolución para encontrar modelos.

**Conexionista**: Se basa en la técnica de las redes neuronales artificiales para buscar sus modelos.

### 2.3.2.2 Tipos de modelos

Como ya hemos visto, el aprendizaje automático obtiene un modelo para resolver un problema dado. **Los modelos pueden ser de los siguientes tipos** [114]:

**Modelos geométricos**:

Se construyen en el espacio de instancias y pueden tener varias dimensiones. Pueden ser de datos linealmente separables si existe un límite de decisión linear

$$w \times x = t \tag{10}$$

donde w es un vector perpendicular al borde de decisión y x es un punto cualquiera en dicho límite. el resultado t es el umbral de la decisión

**Modelos probabilísticos** :

Estos modelos se basan en la estadística de Bayes y pretenden averiguar las distribuciones de probabilidad descriptora de la función que enlaza a los valores de las características con valores determinados para tomar decisiones óptimas. El razonamiento Bayesiano nos ofrece un acercamiento probabilístico a la inferencia.

En un caso clínico para determinar cáncer se puede buscar la relación entre causa y efecto mediante la regla de Bayes:

$$P(Causa \mid Efecto) = \alpha \times P(Efecto \mid Causa) \times P(causa) \qquad (11)$$

lo que nos permite el realizar el diagnostico en función del conocimiento causal ya que es más sencillo disponer de las probabilidades causales que las de diagnóstico.

A modo de ejemplo si consideramos la siguiente información sobre el cáncer de mama aplicando Bayes [115]:

Un 1% de las mujeres de más de 40 años que se hacen un chequeo tienen cáncer de mama

Un 80% de las que tienen cáncer de mama se detectan con una mamografía

El 9.6% de las que no tienen cáncer de mama, al realizarse una mamografía se le diagnostica cáncer erróneamente.

¿Cuál es la probabilidad de tener cáncer si la mamografía así lo diagnostica?

Variables aleatorias: C (tener cáncer de mama) y M (mamografía positiva)

$$P(C \mid m) = \alpha \times P(m \mid C) \times P(C)$$

dando un resultado de que el 7,8% de las mujeres diagnosticadas positivamente con mamografía tendrán realmente cáncer de mama.

¿Cuál es la probabilidad de tener cáncer si tras dos mamografías consecutivas, en ambas se diagnostica cáncer?

Variables aleatorias: M1 (primera mamografía positiva) y M2 (segunda mamografía positiva)

Obviamente, no podemos asumir independencia incondicional entre M1 y M2 pero es plausible asumir independencia condicional de M1 y M2 dada C, por tanto,

$$P(C|m1, m2) = _P(C, m1, m2) = _P(m1, m2|C)P(C)$$

Luego aproximadamente el 41% de las mujeres doblemente diagnosticadas positivamente con mamografía tendrán realmente cáncer de mama.

**Modelos lógicos:**

Transforman las probabilidades mediante arboles de decisión en reglas organizadas.

**Modelos de agrupamiento**:

Dividen el espacio en instancias grupales.

**Modelos de gradiente:**

Representan un gradiente en el que se puede diferenciar entre cada instancia. Modelos de gradiente son los clasificadores geométricos como las maquinas vectores de soporte.

### 2.3.2.3 Algoritmos y técnicas.

### Algoritmos:

Los **pasos** que seguiremos para elegir un algoritmo serán:

- Lo primero que tenemos que tener en cuenta es la elección de la experiencia que utilizaremos en el entrenamiento de nuestro sistema de aprendizaje
- Definimos que función objetivo pretendemos que aprenda.
- ¿Cuál va a ser la representación que usaremos para nuestra función objetivo.
- Seleccionaremos el algoritmo de aprendizaje automático que vamos a utilizar.
- Aplicar el algoritmo de aprendizaje
- Evaluación del sistema

Se agrupan en función de la salida y de cómo se aborde el tratamiento de los ejemplos, los diferentes algoritmos se pueden agrupar en [117]: **Aprendizaje supervisado, Aprendizaje no**

supervisado, Aprendizaje semisupervisado, Aprendizaje por refuerzo, Aprendizaje por refuerzo, Transducción, Aprendizaje multi-tarea.

Aprendizaje supervisado

Los algoritmos realizan sus predicciones **basándose en un conjunto de ejemplos** y se establece una correspondencia entre las entradas y las salidas que se esperan del sistema. Un ejemplo de este tipo es como los precios históricos de las acciones se pueden utilizar para hacer estimar los precios que tendrán en el futuro. El algoritmo busca patrones y para ello utiliza etiquetas con valores que le sirven de interés como el precio, día de la semana, situación empresarial y financiera de la empresa, el sector de negocio, situación geopolítica. Con estos patrones , una vez halla el mejor predice datos de prueba esta vez sin etiquetar para predecir valores futuros en este caso.

La **clasificación, la regresión y la detección** de anomalías **son tipos de aprendizaje supervisado.**

- **Clasificación**. Se le denomina así si el algoritmo pretende predecir una categoría entre varias; ocurre por ejemplo en el caso de querer clasificar por ejemplo una fotografía de un 'asno' o una 'cebra'. la clasificación puede ser **binomial** para solo dos clases o clasificación **multiclase** si son varias. Muy útil en investigación biológica, biología computacional y bioinformática.

- **Regresión**. Ahora se predice un valor, por ejemplo el precio de las acciones.
- **Detección de anomalías**. En ocasiones es necesario encontrar anormalidades o datos no habituales, En los fraudes de tipo bancario , una tarjeta de crédito robada genera patrones de gasto que no suelen ser habituales

**Las técnicas** de aprendizaje supervisado son [120]::

**Árboles de Decisión:**

- Clasificación: ID3, C4.5, CART
- Regresión: CUBIST

**Sistemas basados en reglas:**

- C4-5rules, CN2, ABANICO, INNER, ...

**Aprendizaje basado en instancias:**

- k-NN (Nearest Neighbor)
- IB
- Múltiples sistemas de reducción de instancias: DROP, CNN, BETS, ...

**Híbridos:**

- RISE

**Redes de Neuronas Artificiales (ANN):**

- Perceptron
- Redes Multicapa (backpropagation)

**Clasificadores Estadísticos**

- Naive Bayes

**Basados en kernels:**

- SVM (Máquina de vectores soporte)

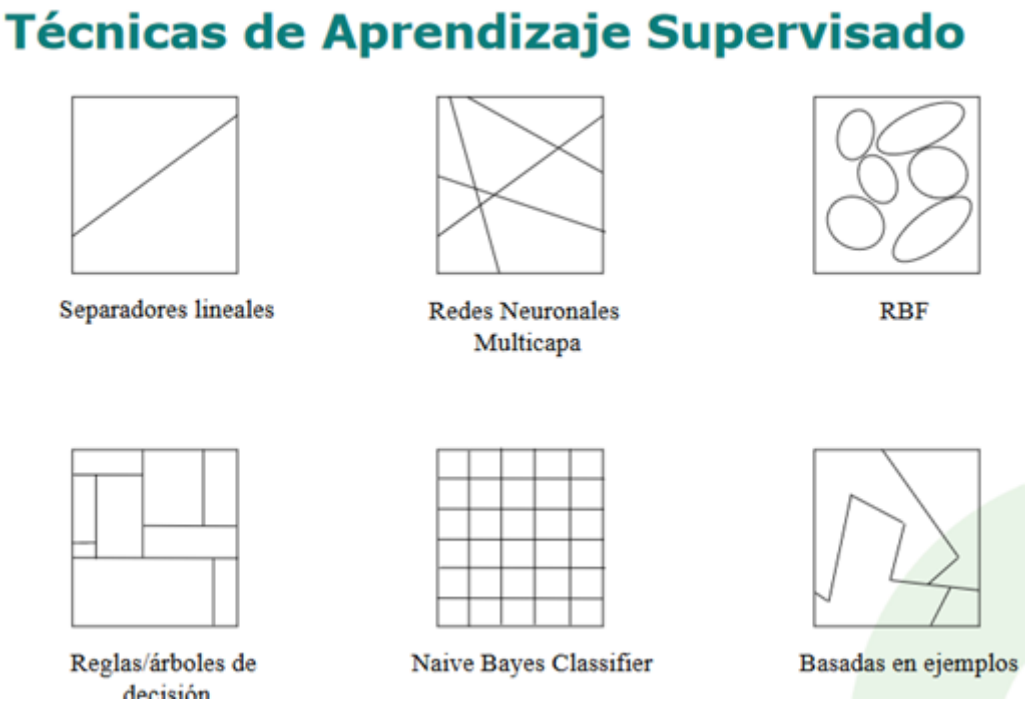



Figura 45 Técnicas de aprendizaje supervisado

## Aprendizaje no supervisado

Ya no se trabaja con etiquetas asociadas a los datos de entrada , se trabaja con conjuntos de ejemplos como entradas al sistema. Ahora el algoritmo los organiza mediante algún tipo de estructura como agrupación de clústeres o encuentra distintas maneras de examinar datos complejos para simplificarlos o dejarlos más organizados y reconoce patrones para poder etiquetar las entradas.

**Las técnicas** de aprendizaje no supervisado son:

**Técnicas de clustering**:

- Clustering jerárquicos
- k-means
- Mapas auto-organizados (SOM) [Kohonen, 95]
- EM [Dempster et al. 77]

- Cobweb [Fisher, 87]

Análisis exploratorios:

- Estudios de correlaciones
- Detección datos anómalos
- Asociaciones y dependencias
- Análisis de dispersión

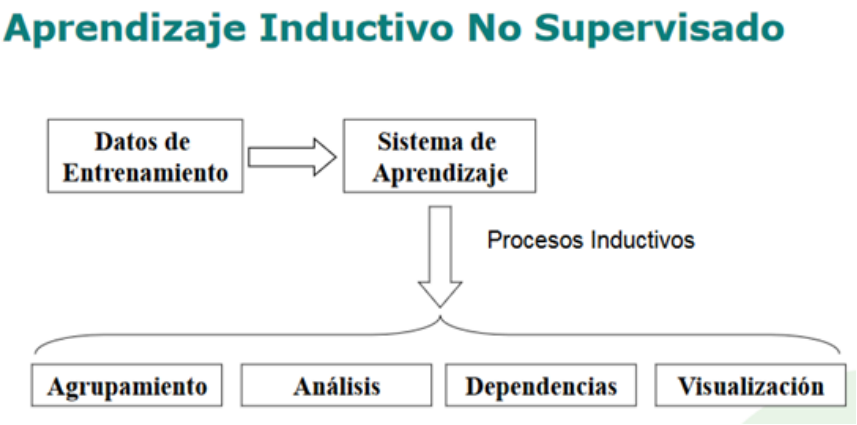



Figura 46 Aprendizaje inductivo no supervisado

Aprendizaje semisupervisado:

Es una combinación de los dos anteriores. Trabaja con datos etiquetados y no etiquetados.

Aprendizaje por refuerzo:

El algoritmo trabaja para obtener la mejor recompensa en función de la bondad de la decisión que ha tomado con los datos de entrada y modifica la decisión cada vez en función de la respuesta a sus acciones. Se retroalimenta del mundo exterior. El aprendizaje es a base de ensayo y error. es común en Robótica y es muy apropiado para el internet de las cosas.

**Transducción**:

Es muy parecido al supervisado. La función no se elabora explícitamente. Intenta predecir las categorías de los ejemplos de prueba futuros basándose en los de entrada, sus categorías respectivas y los nuevos ejemplos que se incorporan al sistema.

**Aprendizaje multi-tarea**

Usan el conocimiento que ha aprendido anteriormente el sistema con el objetivo de poder resolver problemas parecidos a los ya conocidos anteriormente.

A la hora de elegir un algoritmo se tiene en ha de tener cuenta la precisión el tiempo de entrenamiento y la linealidad.

Para evaluar si el algoritmo elegido funciona correctamente en la práctica y da buen resultado con situaciones no vistas en el entrenamiento, analizamos si la distribución refleja lo aprendido por el sistema. En el aprendizaje supervisado evaluamos la hipótesis inducida con el juego de ejemplos que no se han utilizado en el entrenamiento.

Existen diferentes técnicas de evaluación como el **Entrenamiento-test**: dos conjuntos separados, Hold-out: 66% de ejemplos para entrenamiento y 33% para realizar el test , Leave-one-out: se realiza el entrenamiento con todos los ejemplos menos uno y posteriormente se evalúa sobre ese ejemplo.

Si disponemos de un conjunto de posibles ejemplos X, cada uno de ellos se puede hallar en función de una determinada frecuencia, por tanto disponemos de una distribución de probabilidad D que define la probabilidad de encontrar cada elemento de X. El aprendizaje consistirá en hallar la función objetivo f teniendo en cuenta un espacio de hipótesis H. El sistema de aprendizaje empleado sintetizará una hipótesis de entre todas las del espacio de hipótesis H.

Disponemos de una muestra de ejemplos S generado con elementos de X siguiendo la distribución D, a los que un maestro añade la clase correspondiente $<x \in X, f(x)>$ .

Podemos tener ciertas probabilidades de que la hipótesis clasifique mal un ejemplo cualquiera y para conocer que error se está cometiendo, necesitamos conocer el **mejor estimador** que se puede usar para la medición de la precisión de h sobre ejemplos futuros, generados con la misma distribución D ¿Cuál va a ser el error probable de ese estimador?

$$error_d(h) = Pr_{x \in D}\,[f(x) \neq h(x)] \tag{12}$$

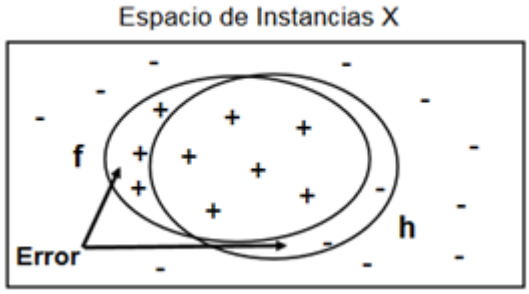



Figura 47 Error de clasificación en espacio de instancias X

No podemos calcular el error verdadero, pero sí que podremos medirlo en una muestra:

$$error_s(h) = \frac{1}{|S|} \times \sum_{x \in S} \partial\,(f(x) \neq h(x)) \tag{13}$$

Tendremos que valorar si es suficientemente bueno el error que se produce en una determinada muestra, como estimador del verdadero error y si además depende de algún factor la estimación del error real partiendo del error en una muestra.

Será posible acotar el error verdadero de una hipótesis h conociendo su error en una muestra, ayudándonos de la estadística.

Hay varias **maneras como estimar el valor del error real** durante o al final del aprendizaje. Si lo hace durante el aprendizaje, se selecciona la hipótesis adecuada y si lo hace al final se indicará el nivel de confianza que ofrece la hipótesis elegida.

En la estimación podemos generar sobre o subajuste en la hipótesis $h_1 \in H$ sobre la muestra S, si la alternativa $h_2 \in H$ cumple que:

Sobre Ajuste:

$error_s(h_1) < error_s(h_2)$ y $error_D(h_1) < error_D(h_2)$

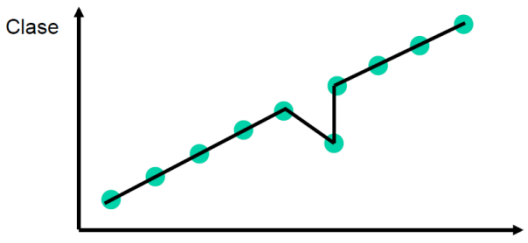



Figura 48 Sobre Ajuste

Sub Ajuste:

$error_S(h_1) > error_S(h_2)$ y $error_D(h_1) > error_D(h_2)$

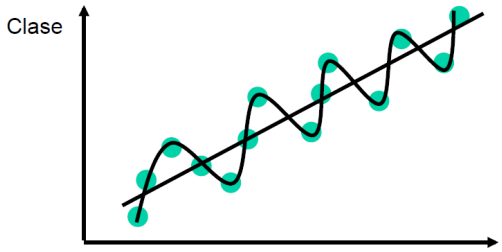



Figura 49 Sub Ajuste

**Para evitar el sobreajuste**, existen métodos empíricos como la validación cruzada , K-fold CV,Leave-One-Out CV y datos no etiquetados o métodos teóricos como técnicas bayesianas, PAC, MDL(mínimum description length) y SRM (minimización del riesgo estructural).

K-fold CV es
    Divide los n ejemplos en k conjuntos disjuntos (folds)
    Para i=1 hasta k
        Entrenar usando todos los folds menos i
        Emplear el fold i para estimar el error
    Devolver la media de los errores de los k folds
Fin.

Figura 50 Metodo K-fold CV. Una medida razonable es usar K=10 repitiendo el proceso

Para darnos cuenta de la **complejidad del problema al aplicar técnicas de aprendizaje automático al cáncer de mama**, mencionar como ejemplo que en 2007 se publicó en un artículo sobre métricas de complejidad para la transformación del problema de la detección del cáncer basado en mamografías [127]. El **grupo de investigación en sistemas inteligentes en La Salle**, investiga diferentes caminos para hallar solución al problema sobre la detección del cáncer de mama abordando el procesado de las imágenes mamográficas, tratamiento de

los datos para su extracción, configuración de los principales atributos y clasificación con métodos de aprendizaje automático. La base de datos del conjunto original, parte de la cesión por parte del **hospital universitario de Girona**, Doctor Josep Trueta y se compone de 216 casos con 121 benignos y 95 malignos.

Cada paciente dispone de un conjunto variable de microcalcificaciones al que le realizan un aplanamiento de los datos mediante una media de todas las descripciones del total de las microcalcificaciones para conseguir un único caso de estudio. El estudio evalúa las distintas transformaciones y busca determinar la mejor para sintetizar un caso. Primero transforman las microcalcificaciones para posteriormente aplicar la transformación que mejor se comporta, en la clasificación. Para la complejidad de la clasificación se tienen en muy cuenta la complejidad inherente al problema porque se concluye en este sentido, que obviarlo, es frecuentemente la causa del error del clasificación.

La dificultad del problema radica en la ambigüedad de la clasificación, la **complejidad de la frontera entre clases** y el conjunto de ejemplos reducido que puede provocar que aunque en la fase de clasificación en el entrenamiento se obtengan resultados correctos, la fase de test de pobres e inestables resultados en los aciertos.

**Este estudio aboga claramente por la evaluación de la geometría de los datos** y no solo por los aspectos estadísticos y la descripción teórica de esos datos.

Tin Kam Ho [128] [129], representa los **problemas reales como puntos en el espacio**. El agrupamiento de estos puntos permite que se asocie un conjunto de problemas a una determinada complejidad lo que permite aplicar por tanto un clasificador adecuado. En los problemas de discriminación en aprendizaje supervisado o no, las estructuras agrupadas son muy importantes para en los problemas de discriminación.

Además aplicando una cierta transformación, podemos desplazar la complejidad de un problema hacia el dominio de otro clasificador.

Para la experimentación utilizan métodos de identificación mixta Bundary e intra/inter class.

| | IB1 | | IB2 | | IB3 | |
|---|---|---|---|---|---|---|
| | Acierto[4] | Error[5] | Acierto | Error | Acierto | Error |
| Promedio' | 62.963 | 0.608 | 65.277 | 0.504 | 62.500 | 0.507 |
| Número de agujeros | 60.185 | 0.631 | 60.648 | 0.523 | 62.037 | 0.520 |
| Todas | 58.941 | 0.640 | 60.595 | 0.529 | 65.083 | 0.479 |
| Aleatoria | 57.870 | 0.649 | 60.648 | 0.521 | 54.629 | 0.519 |
| Compacidad | 52.777 | 0.687 | 55.092 | 0.568 | 55.555 | 0.566 |
| Promedio | 59.722 | 0.634 | 54.629 | 0.563 | 62.500 | 0.533 |
| Aspereza | 57.870 | 0.649 | 54.166 | 0.553 | 64.351 | 0.491 |
| Centroide | 54.166 | 0.677 | 54.166 | 0.555 | 53.240 | 0.533 |
| Elongación del feret | 53.240 | 0.683 | 57.870 | 0.551 | 71.296 | 0.452 |
| Elongación | 52.777 | 0.687 | 55.092 | 0.568 | 55.555 | 0.566 |
| Área | 52.777 | 0.687 | 54.629 | 0.554 | 57.407 | 0.519 |

Figura 51 Resultados de los clasificadores para el conjunto de 23 atributos

Sobre un conjunto de dos juegos de pruebas de 23 y de 6 atributos se obtienen resultados que reflejan poca amplitud en ambos métodos lo que nos da la idea de que son de complejidad similar. Los valores obtenidos por los clasificadores se mueven en un rango estrecho en el intervalo del 52 a 65% con una única excepción en uno de los clasificadores el árbol de decisión C4.5 que alcanza un acierto del 71,29%.

**Las trasformaciones aplicadas no ofrecen resultados destacables** lo que nos muestra una vez más la complejidad del problema.

El **porcentaje de aciertos no es elevado** por lo que suponen que parte de la información no se ha trasladado desde el experto.

La **métrica de complejidad** más favorable obtiene el mejor resultado de clasificación La métrica de complejidad que fija la dificultad de la clasificación desde las estructuras

geométricas de la separación entre clases y se demuestra que no se garantizan mejores resultados del clasificador por tener buena métrica aunque la premisa contraria si se cumple. si la medida de la complejidad no es buena, tampoco lo serán sus resultados en clasificación.

El estudio revela que las métricas de complejidad valen para la predicción del error cometido por el clasificador.

El árbol de decisión C4.5 por lo general tiene mayor eficiencia comparado con el KNN ($IB_x$)debido a su clasificación por atributos discriminantes aunque presenta distribución irregular.

| | 23 atributos | | 6 atributos | |
|---|---|---|---|---|
| | Boundary | Intra/Inter | Boundary | Intra/Inter |
| Promedio' | 0.370 | 0.898 | - | - |
| Número de agujeros | 0.412 | 0.927 | 0.458 | 0.922 |
| Todas | 0.417 | 0.928 | 0.442 | 0.931 |
| Aleatoria | 0.421 | 0.960 | 0.467 | 0.952 |
| Compacidad | 0.423 | 0.966 | 0.472 | 0.914 |
| Promedio | 0.425 | 0.955 | 0.425 | 0.914 |
| Aspereza | 0.435 | 0.954 | 0.458 | 0.913 |
| Centroide | 0.439 | 0.971 | 0.481 | 0.953 |
| Elongación del feret | 0.458 | 1.005 | 0.425 | 0.927 |
| Elongación | 0.462 | 0.966 | 0.458 | 0.913 |
| Área | 0.523 | 0.979 | 0.513 | 1.007 |

Figura 52 Resultados de métricas de complejidad presentadas por el estudio realizado de la Salle.

Por otro lado, empresas como **Microsoft** tienen en su porfolio diversos algoritmos alojados en librerías como en este caso **Azure**. Para enfocar un algoritmo correctamente a la solución del problema que se plantea, Microsoft pone a disposición como ayuda una **hoja de referencia de algoritmos de aprendizaje** automático que es una sábana que nos muestra los distintos algoritmos que ofrece Microsoft y cual sería hipotéticamente el más adecuado para ello.

Está dirigida a científicos con conocimiento de aprendizaje a nivel universitario que necesitan cierta orientación a la hora de elegir un determinado algoritmo. Por supuesto solo es una

ayuda y hay que tener en cuenta que algunos algoritmos no están representados. Actualmente está en proceso de crecimiento y de actualización.

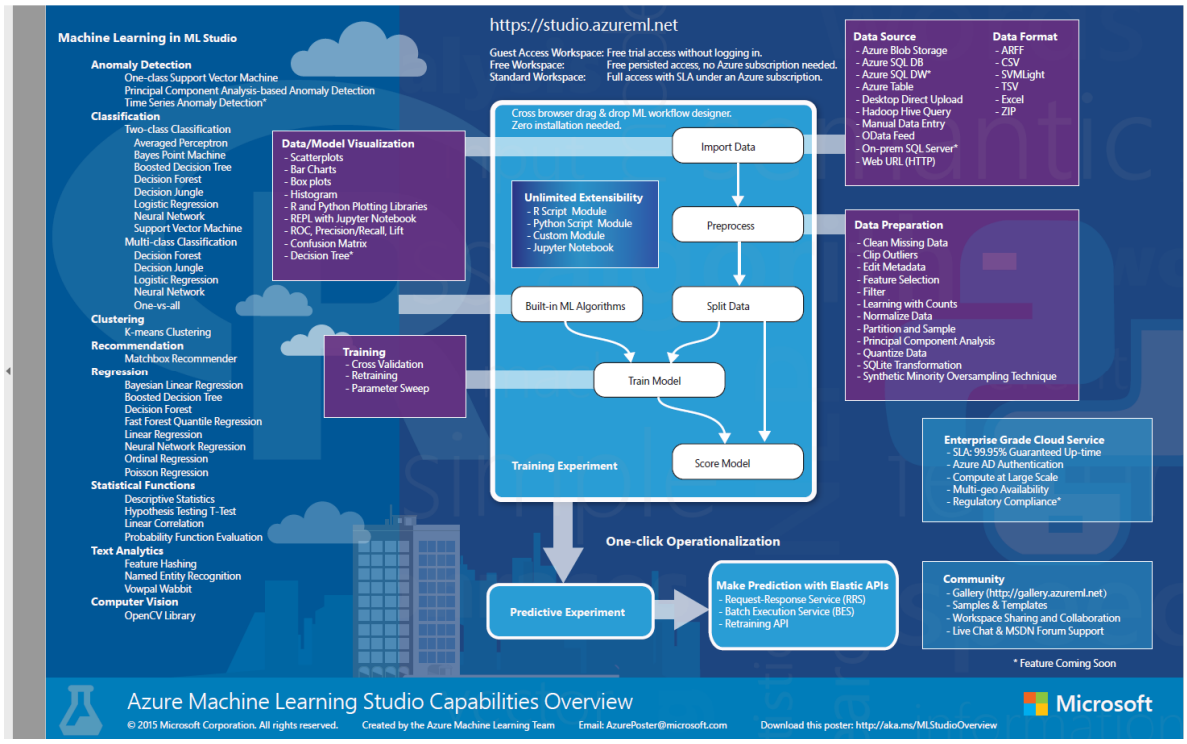



Figura 53 Diagrama con información general de las funcionalidades de Estudio de aprendizaje automático[106]

Para el uso de la hoja hay que observar las etiquetas de ruta de acceso y el algoritmo del gráfico con el siguiente formato: "Para <etiqueta de ruta de acceso>, use <algoritmo>". Por ejemplo, "Para velocidad, use la regresión logística de dos clases".

Habrá ocasiones que se pueda aplicar más de una rama y en otras ninguna.

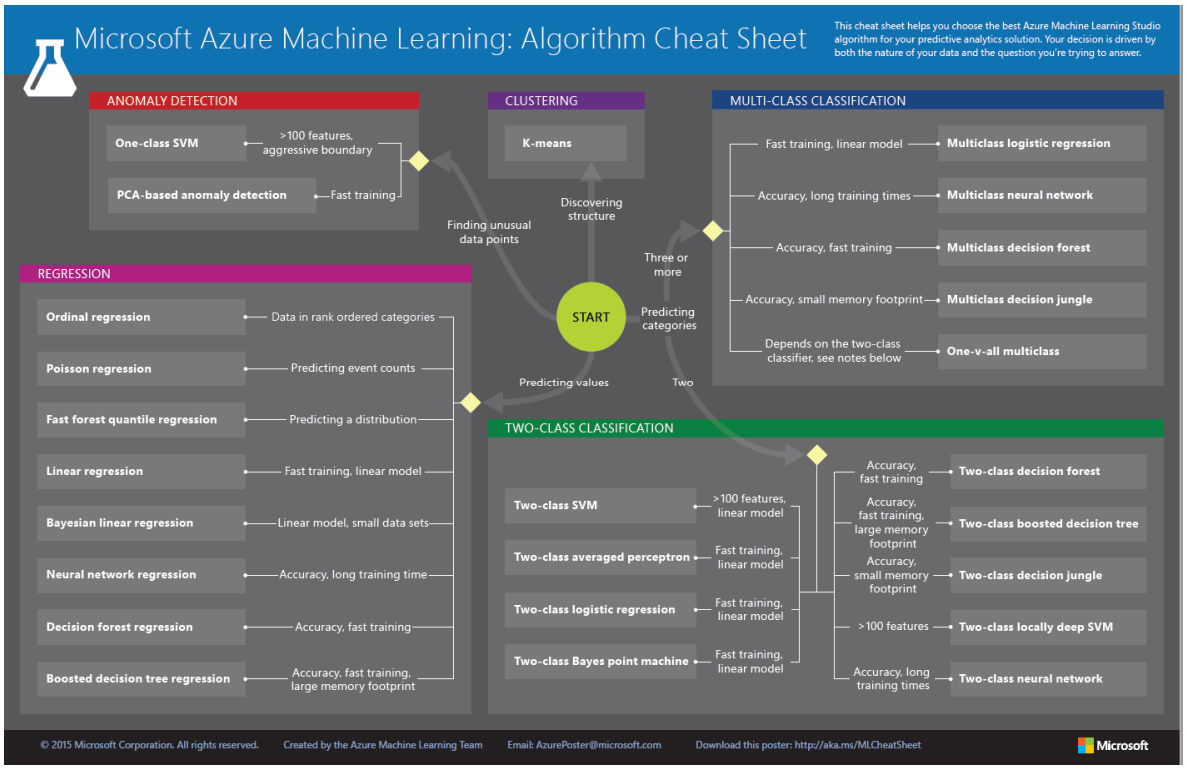



**Figura 54 Clasificación hoja de referencia rápida de algoritmos de aprendizaje automático de Estudio de aprendizaje automático de Microsoft Azure.**

Existe a disposición del usuario un ejemplo de la Galería de Cortana Intelligence http://gallery.cortanaintelligence.com/ sobre un experimento de reconocimiento de letras en el que se prueban varios algoritmos de aprendizaje con datos idénticos y se **comparan los resultados**:

http://gallery.cortanaintelligence.com/Details/a635502fc98b402a890efe21cec65b92

A continuación se muestra una **comparativa de los distintos algoritmos**:

| Algoritmo | Precisión | Tiempo de entrenamiento | Linealidad | Parámetros | Notas |
|---|---|---|---|---|---|
| **Clasificación multiclase** | | | | | |
| regresión logística | | ● | ● | 5 | |
| bosque de decisión | ● | ○ | | 6 | |
| selva de decisión | ● | ○ | | 6 | Uso de memoria bajo |
| árbol de decisión impulsado | ● | ○ | | 6 | Uso de memoria grande |
| red neuronal | ● | | | 9 | La personalización adicional es posible |
| perceptrón promedio | ○ | ○ | ● | 4 | |
| máquina de vectores de soporte | | ○ | ● | 5 | Útil para conjuntos de características de gran tamaño |

| | | | | | |
|---|---|---|---|---|---|
| máquina de vectores de soporte localmente profunda | ○ | | | 8 | Útil para conjuntos de características de gran tamaño |
| máquina del punto de Bayes | | ○ | ● | 3 | |
| **Clasificación multiclase** | | | | | |
| regresión logística | | ● | ● | 5 | |
| bosque de decisión | ● | ○ | | 6 | |
| selva de decisión | ● | ○ | | 6 | Uso de memoria bajo |
| red neuronal | ● | | | 9 | La personalización adicional es posible |
| uno contra todos | - | - | - | - | Vea las propiedades del método de dos clases seleccionado |
| **Regresión** | | | | | |
| lineal | | ● | ● | 4 | |

| | | | | | |
|---|---|---|---|---|---|
| lineal bayesiana | | ○ | ● | 2 | |
| bosque de decisión | ● | ○ | | 6 | |
| árbol de decisión impulsado | ● | ○ | | 5 | Uso de memoria grande |
| rápida de bosque por cuantiles | ● | ○ | | 9 | Distribuciones en lugar de predicciones por punto |
| red neuronal | ● | | | 9 | La personalización adicional es posible |
| Poisson | | | ● | 5 | Técnicamente logarítmica lineal. Para predecir totales |
| ordinal | | | | 0 | Para predecir el ordenamiento por rango |
| **Detección de anomalías** | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| máquina de vectores de soporte | ○ | ○ | | 2 | Especialmente adecuada para conjuntos de características grandes |
| Detección de anomalías basada en PCA | | ○ | ● | 3 | |
| K-Means | | ○ | ● | 4 | Un algoritmo de agrupación en clústeres |

Figura 55 comparativa algoritmos de aprendizaje automático

Propiedades de algoritmo:

●: muestra una precisión excelente, tiempos de entrenamiento breves y uso de linealidad

○: muestra buena precisión y tiempos de entrenamiento moderados.

Otra clasificación de **Aprendizaje automático con MATLAB** según página de MathWorks [138]:




Figura 56 Clasificación de Aprendizaje automático con MATLAB

A continuación se listan las **ANN más comunes y frecuentes** en la mayoría de los simuladores software de sistemas de computación neuronal, clasificadas en base a las aplicaciones de Predicción, Clasificación, Asociación, Conceptualización, Filtrado y Optimización [193]:

### Adaline y Madaline

- Diseñador: Bernard Widrow 1960 *Predicción*
- Características: Técnicas de Adaptación para el Reconocimiento de Patrones.

### Adaptive Resonance Theory Networks (ART)

- Diseñador: Carpenter, Grossberg 1960-86 *Conceptualización*
- Características: Reconocimiento de Patrones y Modelo del Sistema Neuronal. Concepto de Resonancia Adaptativa.

### Back-Propagation

- Diseñador: Rumelhart y Parker 1985 *Clasificación*
- Características: Solución a las limitaciones de su red predecesora el Perceptron.

### Bi-Directional Associative Memory (BAM) Networks

- Diseñador: Bart Kosko 1987 *Asociación*
- Características: Inspirada en la red ART.

### The Boltzmann Machine

- Diseñador: Ackley, Hinton y Sejnowski 1985 *Asociación*
- Características: Similar a la red Hopfield.

### Brain-State-in a Box

- Diseñador: James Anderson 1970-86 *Asociación*

- Características: Red Asociativa Lineal.

### Cascade-Correlation-Networks

- Diseñador: Fahhman y Lebiere 1990 *Asociación*
- Características: Adición de nuevas capas ocultas en cascada.

### Counter-Propagation

- Diseñador: Hecht-Nielsen 1987 *Clasificación*
- Características: Clasificación Adaptativa de Patrones.

### Delta-Bar-Delta (DBD) Networks

- Diseñador: Jacobb 1988 *Clasificación*
- Características: Métodos Heurísticos para Acelerar la Convergencia.

### Digital Neural Network Architecture (DNNA) Networks

- Diseñador: Neural Semiconductor Inc. 1990 *Predicción*
- Características: Implementación Hardware de la función Sigmoid.

### Directed Random Search (DRS) Networks

- Diseñador: Maytas y Solis 1965-81 *Clasificación*
- Características: Técnica de valores Random en el mecanismo de Ajuste de Pesos.

### Functional-link Networks (FLN)

- Diseñador: Pao 1989 *Clasificación*
- Características: Versión mejorada de la red Backpropagation.

### Hamming Networks

- Diseñador: Lippman 1987 *Asociación*

- Características: Clasificador de vectores binarios utilizando la Distancia Hamming.

### Hopfield Networks

- Diseñador: Hopfield 1982 *Optimización*
- Características: Concepto de la red en términos de energía.

### Learning Vector Quantization (LVQ) Networks

- Diseñador: Kohonen 1988 *Clasificación*
- Características: Red Clasificadora.

### Perceptron Networks

- Diseñador: Rosenblatt 1950 *Predicción*
- Características: Primer modelo de sistema Neuronal Artificial.

### Probabilistic Neural Network (PNN)

- Diseñador: Spetcht 1988 *Asociación*
- Características: Clasificación de Patrones utilizando métodos estadísticos.

### Recirculation Networks

- Diseñador: Hinton y McClelland 1988 *Filtrado*
- Características: Alternativa a la red Backpropagation.

### Self-Organizing Maps (SOM)

- Diseñador: Kohonen 1979-82 *Conceptualización*
- Características: Aprendizaje sin supervisión.

### Spatio-Temporal-Pattern Recognition (SPR)

- Diseñador: Grossberg 1960-70 *Asociación*

- Características: Red clasificadora Invariante en el espacio y tiempo.

En este punto, vamos a realizar un **breve repaso de las principales técnicas de aprendizaje** desde la utilización de **métodos tradicionales** como son los Arboles de decisión, k-vecinos, Métodos estadísticos.

Los **métodos de ensamble** o modelos combinados: los **modelos promediados** Bagging y Randon Forest. **Modelos aditivos**: Boosting. **Modelos jerárquicos**: Cascadas.

Y nos centraremos un poco más sobre métodos muy actuales como:

**Métodos de kernel**: Modelos lineales. Kernels. Support Vector Machines. Extensiones.

Redes Neuronales. Historia. **Primera ola**: el perceptrón.

**Segunda ola**: Redes de dos capas, back-propagation.

**Tercera Ola**: Redes profundas, redes convolucionales. Pre-entrenamiento no supervisado. Dropout. Aplicaciones. [103][104].

### 2.3.2.3.1 Modelos tradicionales

**Arbol de decisión.**

Es uno de los modelos inductivos de predicción en el que cualquier hipótesis hallada que sea capaz de clasificar un número suficiente de ejemplos para entrenamiento pares <x,f(x)>, siendo x la entrada y f(x) el valor de la función aplicada a dicha entrada de manera que será capaz de clasificar otros ejemplos no observados anteriormente mediante una función de hipótesis h que aproxime a f y es **frecuentemente usado en Inteligencia Artificial**, en el que sirviéndonos de una BBDD, se construyen diagramas con constructivos lógicos similares a las reglas de sistemas predictivos que nos valdrán para representar y hacer categorías de algunas condiciones que se presentan con frecuencia para la resolución de un problema[121].

Sus entradas son objetos o conjunto de atributos y su salida son valores discretos o continuos de decisiones que se toman en base a la entrada. Cuando se usan valores discretos nos referimos a la clasificación y cuando son continuos a regresión.

La clasificación se basa en las siguientes medidas: **Entropía y Efectividad de un atributo y Ganancia de información**. [143]

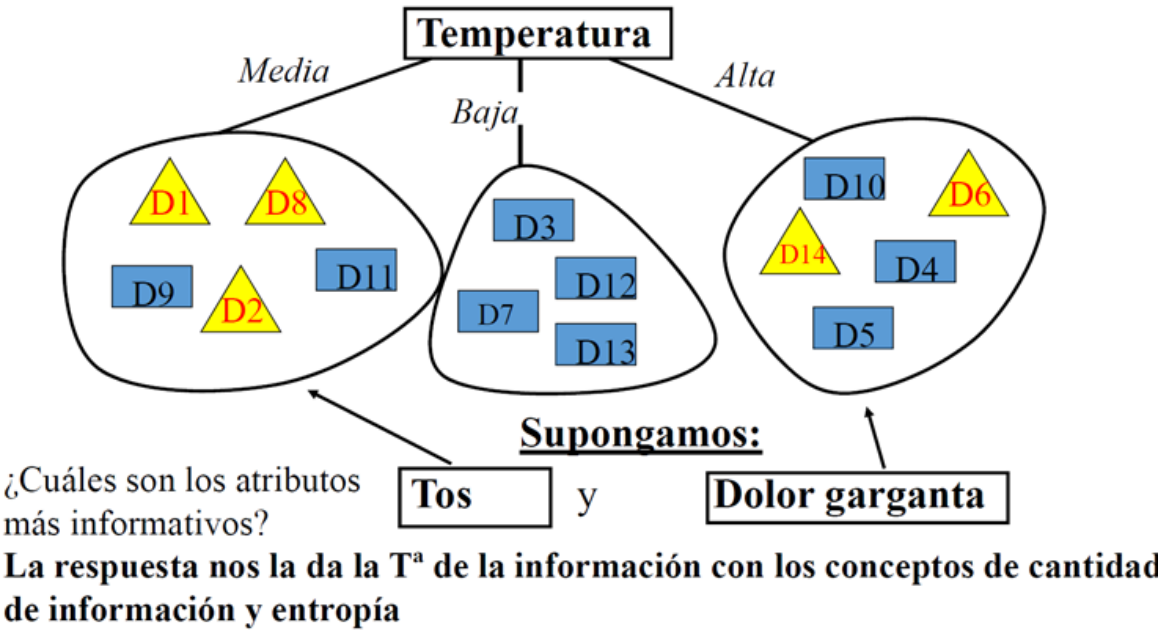



Figura 57 Entropía, efectividad y ganancia de información de un atributo

$$Entropía = S(E) = info(E) = -\sum_{j=1}^{k} p_j \, log_2 p_j \tag{14}$$

siendo $p_j$ la proporción de ejemplos de clase $C_j$ en el conjunto E .

Efectividad de un atributo para subdividir un conjunto de ejemplos en N subconjuntos:

$$Efectividad = S(E,X) = info(E,X) = \sum_{i=1}^{n} \frac{|E_i|}{|E|} \times info(E_i) \tag{15}$$

Ganancia de información que mide como clasifica el atributo a los ejemplos.

$$Ganacia(E,X) = info(E) - info_{atrib}(E,X) \tag{16}$$

El árbol contiene nodos internos, de probabilidad, hojas y arcos. El test se realiza a medida que recorre las hojas del árbol hasta lograr una decisión.

**Ejemplos reales de aplicaciones** construidas con árboles de decisión son [122]:

El diseño que se realizó en 1986 de sistemas de separación gas-petróleo en plataformas petrolíferas marinas de BP. Construido con más de 2.500 reglas,. Ahorró a BP millones de dólares o la Configuración de un equipo de protección de incendios para edificios con de 30.000 reglas.

### K-Vecinos

K-nn (k nearest neighbors fix y hodges 1951) [126] , Es otro método supervisado mediante el cual podemos estimar la **función de densidad**. Es de **clasificación no paramétrico** y reconoce patrones clasificando nuevas instancias con ejemplos que en el espacio de los elementos se encuentran más cercanos haciendo que sea la clase mayoritaria de entre los K vecinos, de los datos que se dispone para entrenamiento. El algoritmo es de tipo Lazy learning y generalmente utiliza la distancia euclediana.[123]

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{p} (x_{ri-} x_{rj})^2} \tag{17}$$

El algoritmo es el siguiente:[125]

COMIENZO
  Entrada: $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$
    $\mathbf{x} = (x_1, \ldots, x_n)$ nuevo caso a clasificar
  PARA todo objeto ya clasificado $(x_i, c_i)$
    calcular $d_i = d(\mathbf{x}_i, \mathbf{x})$
  Ordenar $d_i (i = 1, \ldots, N)$ en orden ascendente
  Quedarnos con los $K$ casos $D_{\mathbf{x}}^K$ ya clasificados más cercanos a $\mathbf{x}$
  Asignar a $\mathbf{x}$ la clase más frecuente en $D_{\mathbf{x}}^K$
FIN

Figura 58 Algoritmo K-Vecinos

### Métodos Estadísticos [130].

**Ejemplos de estas técnicas** son la regresión lineal, el Análisis Factorial, el Análisis de Componentes Principales. Existen **herramientas de software muy completas como DB Intelligent Miner for Data**. Esta un compendio de técnicas estadísticas y resulta muy apropiada si queremos realizar un análisis comparativo.

Hay métodos muy complejos y se deben elegir en función de la naturaleza de los datos y de su uso. Para la selección de la función estadística más apropiada, hay que tener en cuenta el nº de variables, la escala de medida y el número de casos.

Permite tomar mejores decisiones aplicados a la minería de datos. Se trata de encontrar patrones ocultos aplicando cálculos estadísticos a los datos de entrada. La "**Regresión lineal**" es apropiada para la predicción de valores aplicando un modelo de ajuste lineal. Para ver los atributos que predominan en los datos se aplica "**Análisis de componentes principales**". Para reducir las variables de entrada se aplica "**Análisis de factores**".

### 2.3.2.3.2 Método de ensamble: modelos promediados, modelos aditivos, modelos jerárquicos

**Los métodos de ensamble o modelos combinados** [131]

Este método considera **varias hipótesis a la vez** para conformar una hipótesis que tenga un comportamiento mejor. Si la combinación de hipótesis es de la **misma familia**, decimos que son métodos de ensemble y si son provenientes de **varias familias**, sistemas de aprendizaje múltiple

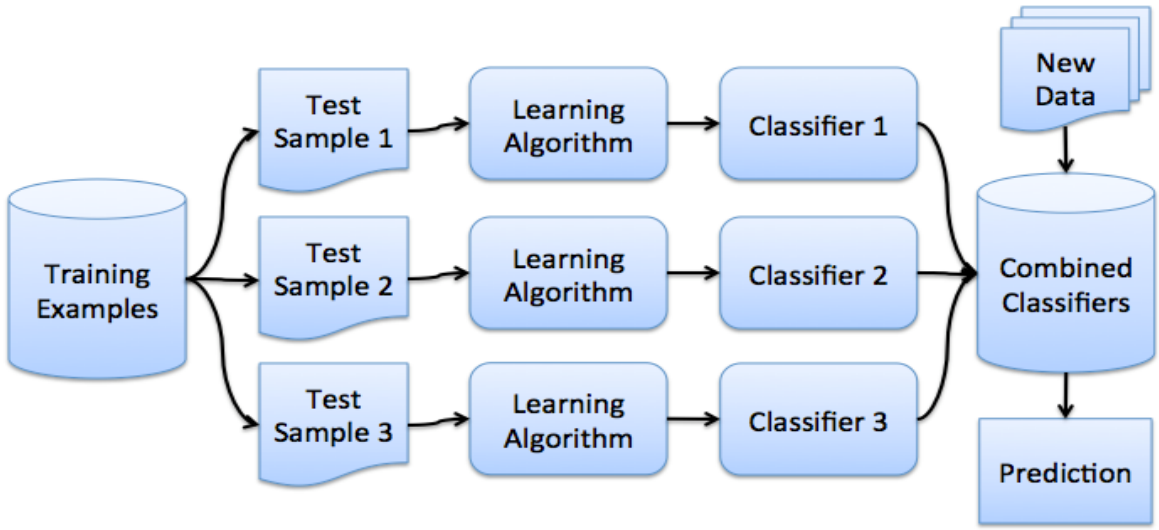



Figura 59 Método de enamble

A consecuencia de que los métodos combinados utilizan combinaciones de hipótesis de forma simultánea, se elevan los **costes computacionales** y es habitual el uso de **algoritmos rápidos** como los árboles de decisión para el espacio de hipótesis base.

Lo cierto es que un **mix de algoritmos de una misma familia** no tiene porqué ofrecer una hipótesis de las mismas características de la familia y es posible la obtención de mejores resultados resultado de la mezcla que derivado de resultados individuales.

Es una **técnica muy indicada** en la aplicación en RNA´s que ofrecen un error bajo en el conjunto de entrenamiento pero errores muy altos en la prueba con datos nuevos. Al combinar se aminora la dispersión.

**Métodos combinados comunes son**:

**Agregación Bootstrap o Bagging**:

La técnica Bagging, por Bootstrap aggregating, fue **introducido por Breiman en 1996** y es un método de agregación de modelos homogéneos basado el voto mayoritario o en el promedio, dependiendo del caso. Es un meta-algoritmo que se diseña para evitar el sobreajuste y obtener una menor varianza, es decir menor dispersión de las estimaciones ofrecidas por cada clasificador.

Se puede usar con cualquier familia aunque predominan en su uso los arboles de decisión. **Puede mejorar** en modelos de tipo individual que son inestables, redes neuronales y árboles de decisión **y empeorar** en otros métodos como el de K-vecinos. Crea nuevos conjuntos de entrenamiento con el conjunto inicial tomando al azar elementos uniformemente y con reemplazo. Pueden aparecer elementos repetidos en los nuevos juegos de entrenamiento.

Tenemos una muestra de aprendizaje $\mathcal{L} = \{(X_1, Y_1), (X_2, Y_2), \ldots . (X_n, Y_n)\}$ formada por n variables de carácter aleatorio (X,Y). Se construye la función $f: X \to Y$ tal que dado un vector entrada X, se prediga con cierta probabilidad de acertar que la variable $Y = f(x)$. Definimos $x_i \in X$ como la variable explicativa o de entrada. $Y_i$ es la variable de salida.

Este algoritmo se basa en que el error promedio que se obtiene sobre el conjunto de entrenamiento $\mathcal{L}$ es mayor o igual al error obtenido por el estimador Bagging. Se puede comprobar que si f es un predictor obtenido sobre $\mathcal{L}$ y $\mathrm{f_{bag}}$ el obtenido por Bagging entonces [139]:

$$E_{\mathcal{L}}(E_{(X,Y)}(Y - f(x))^2) \geq E_{(X,Y)}\left(Y - f_{bag}(x)\right)^2 \tag{18}$$

Un resumen del algoritmo es el siguiente:

Sea $\mathcal{L} = \{(X_1, Y_1), (X_2, Y_2), \dots (X_n, Y_n)\}$ con $\mathrm{X}_i \epsilon X, Y_i \epsilon \{1, \dots, K\}$ o $Y_i \epsilon \mathbb{R}$ .

1. Se construyen M muestras bootstrap $\mathcal{L}_1^*, \mathcal{L}_2^*, \dots \mathcal{L}_M^*$ de tamaño n
2. Para cada una de ellas se calculan estimadores $g_1^*, g_2^*, \dots g_M^*$
3. El estimador Bagging es:

$$f_{bag}(x) = Argmax_y(\#\{m: g_m^*(x) = Y\}) \quad \text{(clasificación)} \tag{19}$$

O

$$f_{bag}(x) = \frac{1}{M}\sum_{m=1}^{M} g_m^*(x) \text{ (regresión)} \tag{20}$$

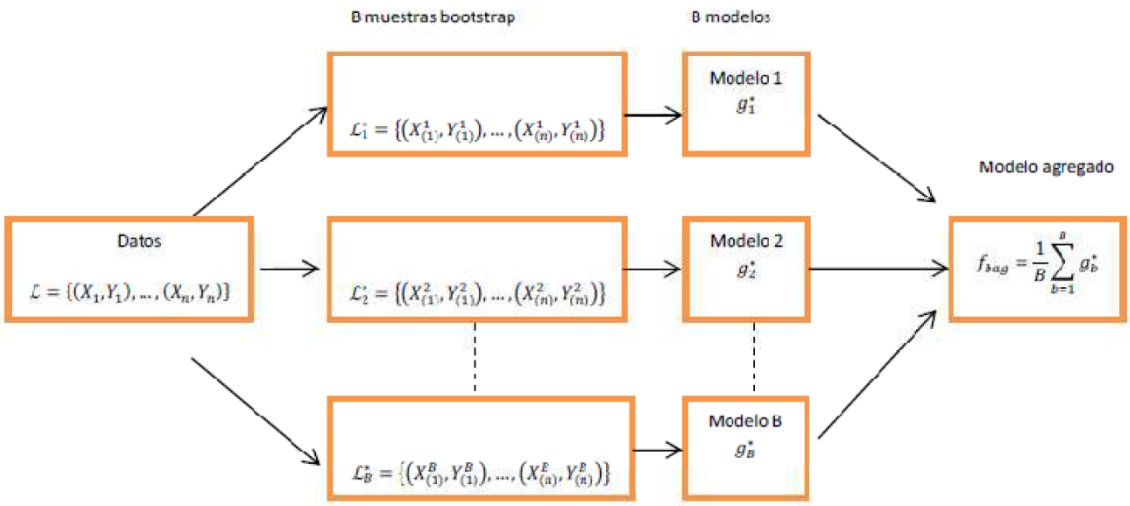



Figura 60 Método Bagging regresión

Los métodos de agregación en problemas binarios y en aprendizaje supervisado dan buenos resultados y cuentan con suficientes avales teóricos.

**Random Forest**: En 2001 Breiman [140] propone el algoritmo Random Forest (arboles aleatorios) que combinan las técnicas de árboles de clasificación o de regresión CART (Classification And Regression Trees) **propuesta por Breiman en 1984** [141] y técnicas Bagging.

Utiliza aleatoriedad en las diferentes fases de construcción del CART: Como el Bagging se construyen M muestras Bootstrap del conjunto original y sobre estas se construyen M árboles y en cada uno de ellos, de cada nodo se escoge la mejor subdivisión realizada por un subconjunto de variables aleatorias explicativas que se seleccionan de manera aleatoria. Los arboles no se podan, son maximales.

La predicción es la resultante de la **clase que es votada** mayoritariamente por los M árboles. En regresión se promedian los valores asignados.

**Este algoritmo funciona** muy bien para problemas que tienen gran cantidad de variables explicativas, si no es así es más eficaz el Boosting.

## Modelos aditivos:

**Boosting**: Trabaja siempre con el conjunto de entrada, no genera nuevas versiones del conjunto de entrenamiento. Se tienen en cuenta los pesos de los datos para crear distintos modelos. En cada repetición se incrementa en los objetos que resultan mal clasificados su peso; así, en la construcción del siguiente predictor estos objetos tendrán mayor importancia y mayor probabilidad de clasificarlos correctamente. **Podemos encontrar aplicaciones** de Boosting en reconocimiento de texto o detección de rostros humanos con probabilidades de detección muy elevadas[134]. **AdaBoost** es el algoritmo más importante, creado por Yoav Freund y Robert Schapire [133] que ganó el Premio Gödel en 2003, Investigadores de AT&T y este algoritmo consta de los siguientes pasos [132]:

T ={$t_i$} i=1, ...N conjunto de entrenamiento, cada instancia con una probabilidad de ser escogido para entrenar el clasificador p(i)=1/N

1. En la iteración k se muestrea con reemplazo el conjunto T para obtener datos de entrenamiento $T^k$, con una distribución de probabilidad p(i).
2. Se crea en clasificador $f_k$ y se prueba con TODOS los datos de entrenamiento disponibles.
   Sea d(i)=1 si el ejemplo i se ha clasificado erróneamente, 0 en caso contrario.
3. Definimos $\varepsilon_k=\sum d(i)p(i)$, y $\beta_k=(1-\varepsilon_k)/\varepsilon_k$
4. Actualizar los pesos para la iteración k+1:
   $p(i) = p(i)\beta_k^{d(i)} / \sum p(i)\beta_k^{d(i)}$

La salida del sistema será la votación ponderada de todos los $f_k$ con pesos log ($\beta_k$)

Figura 61 Algoritmo AdaBoost

Hay **diversos textos y estudios** realizados con Adaboost para aplicación en mamografías:

El estudio de **detección de masas y asimetrías** en mamografías realizado por S. L. de la Fuente-Bermúdez y otros [135] concluye que es recomendable utilizar otro clasificador, pues aunque Adaboost es más resistente a overfitting que otros algoritmos de Aprendizaje automático, también es sensible a outliers y datos ruidosos.

Hay clasificadores de Adaboost modificados para mejorar la tasa de aciertos, reducir la tasa de falsos positivos y reducir la complejidad como la **propuesta de Viola y Jones** cuyo clasificador está basado en una cascada de varios clasificadores Adaboost que en cada etapa se entrena para encontrar todos los artefactos, muestras positivas mientras se desechan las muestras negativas.

Viola y Jones proponen en su algoritmo para dividir la imagen en subventanas y escanea cada una de estas en diferentes posiciones y también escalas distintas. En cada etapa el clasificador ha de detectar si son positivas las sub ventanas en cuyo caso estas pasan a la siguiente etapa del clasificador y en caso contrario, si detecta alguna que es negativa es desechada automáticamente [136]. Es positiva solo en el caso de que supere todas las etapas de detección sin descarte alguno.

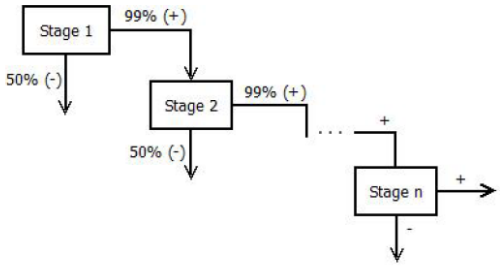



Figura 62 Figura de Cascada de clasificadores [137]

### 2.3.2.3.3 Métodos de kernel

Familia de algoritmos que combinan sencillez y eficiencia de algoritmos, como el **perceptrón y ridge regression** con otras características como puede ser la flexibilidad de sistemas no lineales y la seriedad de los métodos de regularización. Permiten realizar visiones no lineales de algoritmos lineales mediante maquinas que eliminan o minimizan al máximo los inconvenientes de los métodos clásicos no lineales y aportan la **ventaja** de ofrecer menor coste computacional y también menos heurística.

En definitiva las funciones Kernel son funciones matemáticas empleadas en las **Máquinas de Soporte Vectorial (SVM)** que permiten convertir un problema de clasificación no lineal en el espacio dimensional original, en un problema sencillo de clasificación lineal en un espacio de dimensión superior.

Se basan en el **Teorema de Mercer** que es anterior a 1909; este teorema es una potente herramienta en la teoría de ecuaciones integrales y se usa también en la teoría de Hilbert de procesos estocásticos y karhunen-Loeve.

El teorema de Mercer nos dice si un kernel cumple la propiedad de producto escalar y por tanto útil para una SVM. No nos dice sin embargo como construir las funciones.

Una **función kernel** $k(x,y): X \times X \rightarrow \mathbb{R}$ asigna a cada par de objetos de entrada, x e y, un valor real que se corresponde con el producto escalar de sus respectivas imágenes en el espacio de características $F$[158]

El estudio de Kernels del **espacio de Hilber** fue desarrollada por Aronszajn en los años 40 del siglo pasado.

Kuhn y Tucker en 1951 introdujeron restricciones de desigualdad en los problemas de optimización en el que se usa el producto del punto con funciones en el **espacio de características** llamadas Kernels. [147]

En 1964 Aizermann, Bravermann y Rozoener utilizaron el teorema de Mercer para interpretar los kernel como el producto punto en un espacio característico.

La primera utilización del Kernel polinomial se realizó por parte de **Poggio** en el año 1975y a partir del 1990, Poggio y Girosi aplicaron las técnicas kernel a las máquinas de aprendizaje y a las redes neuronales.

No fue hasta que **Boser Guyon y Vapnik** aplicaron Kernel en la máquina de vector de soporte cuando sus posibilidades se comprendieron completamente.

La dimensión de Vapnik-Chervonenkis y la **minimización del riesgo empírico** son fundamentales para lograr el híper plano separador a través de la solución óptima. Los conjuntos de entrenamiento pueden ser linealmente separables y si no lo fueran , se aplican funciones Kernel para lograrlo llevando las muestras a otro plano que tiene **mayor dimensionalidad**. Entre los más utilizados podemos nombrar el **Kernel polinomial y el de Base Radial**.

Hay requerimientos que no se pueden pasar por alto por ser fundamentales ya que las funciones deben expresarse como un producto escalar en un espacio dimensional elevado y por lo tanto deben cumplir tres **prerrequisitos**, a saber: Continuas, Simétricas y Positivas.

Cada característica de los datos es tratada como una dimensión que será definida mediante funciones Kernel y transformada en el espacio dimensional. Esto permite relacionar los datos de manera mucho más sencilla.

Existen diferentes **opciones de Kernels**: Gaussiano, Fisher, String y Hubert de anchura modificada.

### Tipos de Kernel

Un kernel [149] es una función K, tal que $K(X,Z) = \{\emptyset(x).\emptyset(z)\}$ , siendo $\emptyset$ un mapeo (eventualmente no lineal) a otro espacio de atributos.

Las **funciones hipótesis** serán de la siguiente manera:

$$f(x) = \sum_{i=1}^{n} W_i \, \emptyset(x) + b \tag{21}$$

Si tenemos una representación dual de **la función de decisión** será:

$$f(x) = \sum_{i=1}^{n} \alpha_i \; y_i \{\phi_i(x_i).\phi(x)\} + b \tag{22}$$

No se necesitan mapear los atributos para aprender y se evita por tanto la maldición de la dimensionalidad.

Entre los algoritmos más importantes aplicados en la investigación de aprendizaje automático, se definen los de aprendizaje ensemble, como representantes de toda una corriente de pensamiento en la creación de kernels de procesamiento. Tienen generalmente como característica señalada en ellos, que eligen la hipótesis mediante la llamada a otro algoritmo de aprendizaje base y además son estables con buena inmunidad al ruido, precisos gracias a la eliminación o reducción del error y eficaces porque son capaces de dividir el problema en otros más reducidos con algoritmos base más sencillos.

### Stump kernel

Este tipo de kernel supone un número infinito de decisiones Stump o decisiones de árbol de un solo nivel. Este modelo de decisión es dentro de los que existen en aprendizaje, uno de los más simples en aplicación. Como resalte podemos decir que el rendimiento es aceptable y es eficiente y fácil en su implementación lo que en definitiva lo convierte en un algoritmo ensemble muy popular

### Perceptron kernel

Se puede extender Stump como solución a problemas para obtener el kernel Perceptron. El perceptrón es importantísimo en aprendizaje con redes neuronales como veremos más adelante. Trabaja en dimensión finita. Lo veremos en profundidad más adelante

### Laplacian kernel

Se aplican transformaciones matemáticas sobre el kernel Stump para obtener un árbol infinito. Combinando varias funciones Stump y con la utilización de la serie de Taylor, se obtiene una equivalencia del tradicional kernel de base radial en la teoría de ensemble infinita, Laplace expresado con la siguiente formula:

$$k(x, x') = \exp\left(-\gamma \left| |x - x'| \right| 1\right), \gamma > 0 \qquad (23)$$

### Exponencial kernel

De la misma manera aplicando un grupo de transformaciones, en este caso a partir del Kernel Perceptron, obtenemos el equivalente en el popular Kernel exponencial expresado como árbol de decisión , Perceptron Infinitas .La fórmula es la siguiente:

$$k(x, x') = \exp\left(-\gamma \left| |x - x'| \right| 2\right), \gamma > 0 \qquad (24)$$

## SVM

Dentro de los métodos de Kernel destacamos Kernel **la Máquina de soporte Virtual** (SVM).

Como hemos mencionado anteriormente, aunque sus propiedades ya se habían usado anteriormente en máquinas de aprendizaje, **Vladimir Vapnik** y colaboradores la presentaron por vez primera en 1992. En este año se forma el clasificador de margen fuerte con la máquina de soporte vectorial básica y en 1995 la versión de margen débil introducida por Cortés y Vapnik. [151].

Shawe - Taylor y Barlett aportaron el primer limite estadístico riguroso en la generalizaron de SVM de **margen fuert**e (hard margen) que posteriormente Shawe - Taylor y Cristianini hacen lo propio con limites parecidos en SVM de margen débil (soft margen).

Weston y Watkins , Platt, Shawe - Taylor y Cristianini y Vapnik continuaron trabajando sobre la generalización y su extensión al caso de multiclasificación.

Jaakkola y Haussler , Vapnik y Chapelle, Weston y Hebrich , Wahba, Lin, Zahang, Opper y Whinther trabajan sobre el análisis del error esperado en la validación cruzada y Vapnik además, presenta el límite del error esperado en términos del margen y el radio de la esfera más pequeña contenida en un vector de soporte[146].

| Algunos de los grandes acontecimientos | |
|---|---|
| Antes del siglo XVIII. | Filósofos. Problema de Inferencia Inductiva |
| Siglo XVIII. | Carl Gauss. Mínimos cuadrados |
| 1788. | Joseph Louis Lagrange. Optimización con restricciones (igualdades) |
| 1909. | J. Mercer. Teorema de Mercer (producto punto en un espacio característico) |
| 1930. | Ronald Fisher. Procedimiento para clasificación Proporciona puntos de partida (Análisis y métodos). |
| 1940. | N. Aronszajn. Espacios Hilbert |
| 1950. | Alan Turin. Propuso la idea de las máquinas de aprendizaje (Las máquinas podrían pensar) |
| 1951. | Kuhn y Tucker. Optimización con restricciones (desigualdades) |
| 1957. | Frank Rosenblatt. Perceptrón (primera máquina de aprendizaje) |
| 1960. | Vapnik y Chervonenkis. Teoría de generalización |
| 1963. | Vapnik, V. y Lerner, A. Máximo margen, generalización |
| 1964. | M. Aizerman, E. Braverman, y L. Rozonoer. Interpretación geométrica del Kernel. Introdujeron las funciones Kernel en las máquinas de aprendizaje. |
| 1968. | F. W. Smith. Problemas no separables (con ruido) Introdujo variables slack (de relajación) |
| 1975. | T. Poggio. Kernel polinomial |
| 1986. | Rumelhart, Hilton y Williams. Back - propagation. (Segundo nacimiento del Perceptrón) Método para resolver el problema Perceptrón, hallando el peso de todas las neuronas simultáneamente |
| 1989. | J. K. Anlauf y M. Biehl. El adatrón: Adaptación del perceptrón maximizando el margen (máximos locales) |
| 1990. | T. Poggio y F. Girosi. Utilizar funciones Kernel en las redes neuronales. |
| 1991. | Bennett y Mangasarian. Perfeccionaron la técnica de variables slack y margen débil |
| 1992. | Boser, Guyon y Vapnik. Introdujo las SVM |
| 1995. | Cortes Vapnik. Se introdujo en las SVM el margen débil |
| 1996. | Shawe-Taylor y Cristianini. Margen débil |
| 1998. | Weston y Watkins; Platt, Cristianini y Shawe-Taylor; Vapnik. Multiclasificación para las SVMs |
| 1998. | Vapnik, Generalización de funciones Kernel (buen uso). |
| Después de 1998. | Un creciente número de investigadores trabaja en la evolución de las SVMs. Jaakkola y Haussler, Vapnik y Chapelle, Weston y Herbrich, Wahba, Lin, Zhang, Opper y Winther. Proporcionan el análisis de validación cruzada para el error esperado. |

Figura 63 Acontecimientos que han marcado la historia de las SVM

En los últimos años, las **líneas de investigación fusionan** disciplinas distantes como la estadística, el análisis funcional, la optimización y las máquinas de aprendizaje.

SVM es un **técnica que en clasificación** se muestra muy potente y teóricamente se basa en modelos lineales aunque puede controlar también los no lineales. **Es un algoritmo** para

encontrar clasificadores lineales en espacios transformados. Inicialmente nacieron para solucionar problemas de clasificación binaria pero hoy en día **resuelven problemas** como la regresión, agrupamiento y clasificación.

Hay estudios realizados sobre sistemas CAD aplicados a la detección de cáncer mediante análisis de imágenes digitales que aplican la técnica SVM para su tratamiento.




Figura 64 análisis de imágenes digitales que aplican la técnica SVM para su tratamiento

En este trabajo hacemos referencia a varios de ellos.

Comprenden una clase concreta de algoritmos que se caracteriza por el uso de funciones kernel, la ausencia de mínimos locales y su capacidad de control obtenida al manejar el margen.

SVM opera de forma que[156]:

Transforma los datos de un espacio de dimensión muy alta mediante una función kernel reformulando el problema de tal manera que los datos se mapean implícitamente en este espacio.

Calcula eficientemente el hiperplano óptimo encontrando un hiperplano que maximiza el margen entre dos clases.

**Si los datos no se pueden separar linealmente**, halla el hiperplano que maximiza el margen y minimiza la función del nº de clasificaciones que no son correctas , termino de penalización de la función.

Una **ventaja añadida** de SVM es que evita problemas de curse of dimensionality o **maldición de la dimensionalidad**, esto sucede cuando hay muchos atributos respecto al número de instancias. Clasificadores como SVM antes mencionado y Voted Perceptron son robustos al sobreajuste [148].

**Las entradas son** de tipo numérico mientras que las salidas son binarias. El algoritmo trabaja en un espacio de dimensionalidad máxima posible, tratado por Duda y Venado, Cubra, Vapnik y otros, construyendo uno o varios conjuntos de hiperplanos. Al buscar una frontera de decisión específica es un algoritmo resistente al sobreajuste.

El espacio contendrá separando las clases una serie de puntos de muestra. **Es importante para la clasificación** la separación de las clases que cuanto mayor sea esta, mejor se realizará. Se hallará el **Hiperplano de margen máximo** que proporcione la mayor separación entre las clases que son linealmente separables y para ello se calcula la envolvente convexa de cada clase que garantiza una solución única, no ocurre así en la red neuronal artificial entrenada con el mismo conjunto de entrada .

Analizando el problema de optimización del margen geométrico de forma algorítmica, presenta problema de optimización cuadrática con restricciones lineales.

Se determina el segmento de unión entre las envolventes que ha de ser el más corto posible. Una vez que se halla el segmento máximo de separación, se determina el Híper plano máximo con la perpendicular que divide al segmento en su punto medio [150].

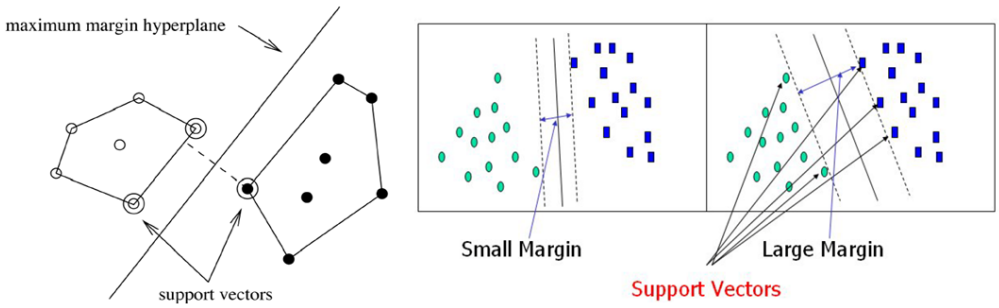



Figura 65 Vectores de soporte, determinación del hiperplano de margen máximo

Los **vectores de soporte** son las instancias o dicho de otra manera los ejemplos de entrenamiento, generadoras de una función discriminatoria lineal, que resultan más cercanas de cada una de las clases al hiperplano de margen máximo. Contendrá como mínimo un vector por clase pero es posible que haya más. Al final, el conjunto de vectores de soporte deja definido de forma única el hiperplano de margen máximo. Las otras instancias no son relevantes. Al realizar nuevas clasificaciones de instancias que no sabemos a qué clase pertenece, se tomará la proximidad espacial para determinar a cuál de ellas se le asignará.

Las SVM´s **se pueden clasificar** en:

- SVM de margen duro
- SVM de margen blando para ejemplo cuasi separables linealmente
- SVM kernelizadas dedicadas a la clasificación binaria de ejemplos no separables linealmente
- SVR (Support Vector Regression Machines) resuelven el problema de regresión lineal y no lineal.

Estos algoritmos se aplican principalmente en análisis de datos, reconocimiento de patrones y categorización de textos e hipertextos, reconocimiento de caracteres, visión artificial, análisis de series temporales, reconocimiento de imágenes y un largo etc.

Su **éxito** se basa en su fundamento teórico reconocido.

Sea X el espacio de entrada e Y el dominio de salida $X \subseteq \mathbb{R}^n, Y = \{-1,1\}$ clasificación, $Y \subseteq \mathbb{R}$ regresión. En adelante tomaremos como ejemplo la función de aprendizaje binaria $f: X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ de manera que la entrada $x = (x_{1,\dots,}x_n)^T$se asigna a la clase positiva s $f(x) \geq 0$ y en caso contrario se asigna a la clase negativa.

Nos interesaremos cuando $f(x)$ sea una función de tipo no lineal de $x \in X$ y buscaremos una de tipo lineal $f(x)$ cuya imagen corresponda a un mapeo no lineal.

Sea S el conjunto de ejemplos para entrenamiento tal que $S = \{(x_1, y_1), \dots\dots, (x_m, y_m) \subseteq (X \times Y)^m\}$ siendo m el número de ejemplos donde $x_i$ es el ejemplo o instancia e $y_i$ es su etiqueta.

Denominamos producto interno entre x y w, a la siguiente igualdad $\langle x, w \rangle = x^T w = \sum_i x_i w_i$ .

Consideramos el mapeo $\Phi: x \epsilon X \rightarrow \Phi(\mathrm{x}) \epsilon F$ siendo X: espacio de entrada y F el espacio característico.

Definimos w como el vector de pesos , b el sesgo (bias) y -b=$\Phi$ es el umbral

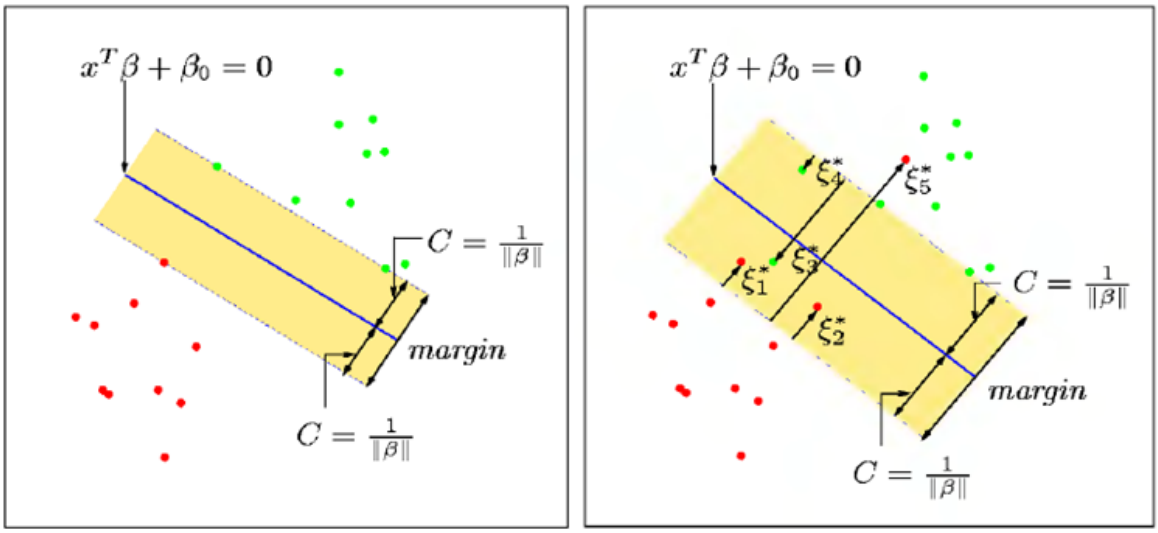



Figura 66 Clasificación Vectores Soporte. La imagen de la izda muestra un caso separable linealmente . A la derecha se muestra otro caso no separable.

Debemos encontrar el **hiperplano (subespacio afín) óptimo** de dimensión n-1 que separe los ejemplos etiquetado con -1 de los etiquetados con +1 con un margen máximo

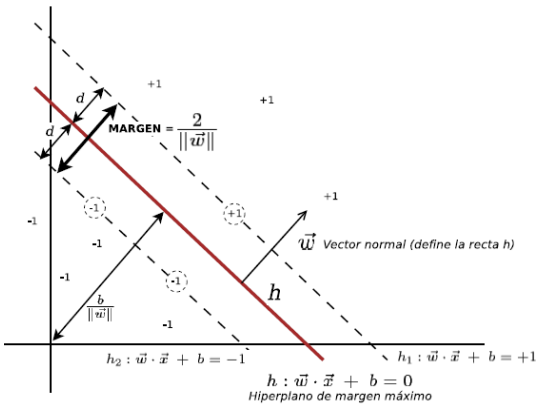



Figura 67 Hiperplano de Margen máximo

que separa dos clases y para ello debemos resolver el problema de optimización $max_{w,b}C$ de manera que:

s.a. $\frac{y_{i(\langle x,w\rangle+b)}}{||w||} \geq C. i = 1, \dots, m$

Si $C = \frac{1}{||w||}$ el problema anterior es equivalente a $min_{w,b}||w||^2$ ;

s.a $\frac{y_{i(\langle x,w\rangle+b)}}{||w||} \geq 1, i = 1, \ldots, m$

Las restricciones definen un margen de decisión de : $\gamma = \frac{1}{||w||^2}$ y por tanto el **Lagrangiano primal** es:

$$Lp(w, b, \alpha) = \frac{1}{2}\langle w, w\rangle - \sum_{i=1}^{m} \alpha_i [y_i (\langle w, x_i\rangle + b) - 1] \qquad (25)$$

siendo $\alpha_i \geq 0$ los multiplicadores de Lagrange

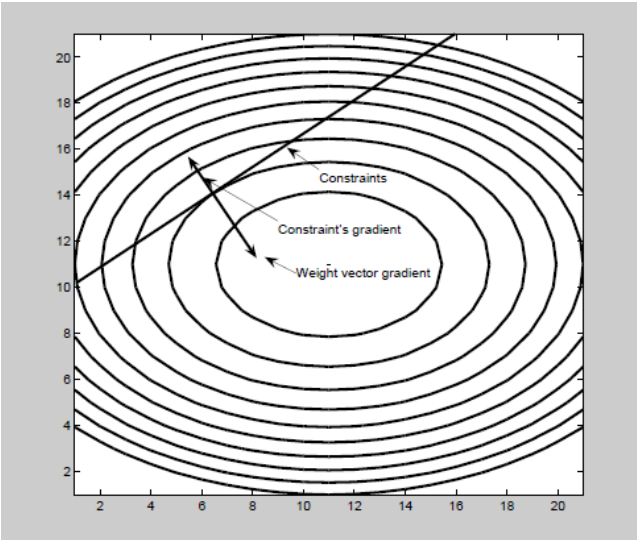



Figura 68 Restriccion de gradiente

Como se **trata de un problema de programación no lineal convexo** cuya función objetivo si es convexa y los puntos que satisfacen las restricciones forman un conjunto convexo y teniendo en cuenta que cualquier restricción lineal define un conjunto convexo y que un conjunto de K restricciones lineales simultaneas define la intersección de los K conjuntos convexos, implica que el resultado es también de un conjunto convexo.

Realizamos la **derivaciones** respecto de w y b

$$\frac{\delta Lp(w,b,\alpha)}{\delta w} = w - \sum_{i=1}^{m} y_i \alpha_i x_i = 0 \tag{26}$$

$$\frac{\delta Lp(w,b,\alpha)}{\delta b} = \sum_{i=1}^{m} y_i \alpha_i = 0 \tag{27}$$

Para hallar el lagrangiano Dual, utilizando estos resultados en el primal :

$$Lp(w,b,\alpha) = \frac{1}{2}\langle w,w\rangle - \sum_{i=1}^{m} \alpha_i [y_i (\langle w,x_i\rangle + b) - 1]$$

$$= \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle - \sum_{i,j} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle + \sum_{i=1}^{m} \alpha_i \tag{28}$$

$$LD(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle x_i, x_j\rangle \tag{29}$$

Conocida también como **Wolfe dual**, es decir un tipo de problema dual en el que la función objetivo y las restricciones son funciones diferenciables.

No sólo **no es necesario conocer** el conjunto de funciones base de transformación, sino que tampoco lo es conocer las coordenadas de los ejemplos transformados en el espacio de características. Es suficiente con conocer la forma funcional del Kernel correspondiente independientemente que este estuviera asociado a un conjunto infinito de funciones base [157].

El uso de este concepto para un problema de minimización, se puede encontrar una cota inferior debido a la **dualidad principio débil**. [152] Obtenemos una solución en la que

pueden usarse Kernels. Estos **Kernel pueden ser de varios tipos** como lineales, gaussianos, o polinómicos de grado P, sigmoidal.

**Proposición 1**

Sea $S = \{(x_1 y_1), \ldots., (x_m y_m)\}$ una muestra que es linealmente separable y ahora además suponemos que los parámetros $\alpha$ van a resolver el problema de optimización cuadrática.

$$\begin{cases} Maximizar\ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \\ \text{s.a} \sum_{i=1}^{m} y_i \alpha_i = 0 \qquad\qquad i = 1, \ldots, m \\ \alpha_i \geq 0 \end{cases} \tag{30}$$

Ahora el vector de pesos:

$$w^* = \sum_{i=1}^{m} y_i \alpha_i^* x_i \tag{31}$$

está formando parte del hiperplano de máximo margen $\gamma = \frac{1}{||w||^2}$

Como b no está en el Wolfe dual, usaremos las restricciones del primal para encontrar $b^*$

$$b^* = \frac{max_{y_i} = -1(\langle w^*, x_i \rangle) + max_{y_i} = 1(\langle w^*, x_i \rangle)}{2} \tag{32}$$

Aplicando las **condiciones karush-Kunh-Tucker** (KKT), cuyas condiciones son además de necesarias, suficientes para que la solución de un problema de programación matemática, sea óptima.

El optimo $\alpha^*\omega^*b^*$ deben cumplir que

$$\alpha_i^*[y_i(\langle w^*, x_i\rangle + b^*) - 1] = 0, \qquad i = 1, \dots, m \tag{33}$$

Es en definitiva una **generalización del método de los Multiplicadores de Lagrange**.

Únicamente las entradas $\mathrm{x_i}$ donde el margen funcional es uno, se encuentran muy cerca del hiperplano clasificador y podemos expresar la representación dual en términos de este subconjunto de parámetros

$$f(x, \alpha^*, b^*) = \sum_{i=1}^{m} y_i \alpha_i^* \langle x_i, x\rangle + b^* = \sum_{i \epsilon SV}^{m} y_i \alpha_i^* \langle x_i, x\rangle + b^* \tag{34}$$

Los multiplicadores de Lagrange **calculan** como de importantes son los ejemplos de entrenamiento en la solución definitiva . El perceptrón $\alpha_i$ media el número de equivocaciones del i-esimo ejemplo.

Se deriva otra consecuencia a tener en cuenta de las KKT para $j \in SV$

$$y_j f\left(x_j, \alpha^*, b^*\right) = y_j \left( \sum_{i \epsilon SV}^{m} y_i \alpha_i^* \langle x_i, x\rangle + b^* \right) = 1 \tag{35}$$

Ahora podemos expresar $\left|\left|w\right|\right|$ y el margen como función de los multiplicadores $\alpha$

$$\begin{aligned} \langle w^*, w^*\rangle = \sum_{i,j}^{m} y_i y_j \alpha_i^* \, \alpha_j^* \langle x_i, x_j\rangle = \sum_{j \in SV} \alpha_j^* \, y_j \sum_{i \in SV} y_i \alpha_i^* \langle x_i, x_j\rangle \\ = \sum_{j \in SV} \alpha_j^* \left(1 - \, y_j b^*\right) = \sum_{i \in SV} \alpha_i^* \end{aligned} \tag{36}$$

**Proposición 1 con Kernels**

Sea $\mathrm{S} = \{(\mathrm{x_1 y_1}), \dots., (\mathrm{x_m y_m})\}$ una muestra linealmente separable por el Kernel $K(x, z)$ y suponemos que los parámetros $\alpha$ resuelven el problema de optimización cuadrática

$$\begin{cases} Maximizar\ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j} y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle \\ \text{s. a} \sum_{i=1}^{m} y_i \alpha_i = 0 \qquad\qquad i = 1, \dots, m \\ \alpha_i \geq 0 \end{cases}$$

Entonces la regla de decisión dada por $sign(f(x))$, donde $f(x) = \sum_{i=1}^{m} y_i \alpha_i^* K(x_i, x) + b$ equivale al hiperplano de máximo margen en el espacio característico implícito definido por el Kernel $K(x, z)$ y que el hiperplano tiene margen geométrico

$$\gamma = \left( \sum_{i \in SV} \alpha_i^* \right)^{-\frac{1}{2}} \tag{37}$$

Un Kernel que satisface las condiciones de Mercer es equivalente a que la matriz $(K(x_i, x_j))_{i,j}^m$ sea definida positiva, luego ya tenemos el problema transformado a convexo porque la matriz $(y_i y_j K(x_i, x_j))_{i,j}^m$ también se define positiva e implica una solución que es única. El problema se puede comprimir usando solo los SV

Sean tomadas en consideración las funciones $\mathcal{L}$ de tipo real con un umbrla de pesos unitarios sobre un espacio X con producto interno. Para cualquier distribución de probabilidad $\text{D sobre X} \times \{-1,1\}$ , con probabilidad $1 - \delta$ sobre m ejemplos de $S$ , el hiperplano maximal tiene un error menor que:

$$err_d(f) \leq \frac{1}{m - d}\left( d \ln \frac{em}{d} + \ln \frac{m}{\delta} \right) \tag{38}$$

d es el número de SV

Proposición 2

Sea $S = \{(x_1 y_1), \ldots., (x_m y_m)\}$ una muestra y supongamos que $\alpha^* y\, b^*$ resuelven el problema de optimización dual.

Entonces $w = \sum_{i=1}^{m} y_i \alpha_i^* x_i$ forma parte del hiperplano con margen geometrico

$$\gamma = \frac{1}{||w||^2} = \left( \sum_{i \in SV} \alpha_i^* \right)^2 \tag{39}$$

SMO (Soft Margin Optimization)

Es un algoritmo para la **solución del problema de la programación cuadrática** (QP). Fue ideado por John Platt en 1998 en Microsoft Research . [154] SMO es ampliamente utilizado para entrenamiento las máquinas de vectores soporte y es implementado por la popular **herramienta LIBSVM** [155].

La publicación del algoritmo SMO en 1998 generó mucha expectación en la comunidad SVM, porque los métodos previamente disponibles para la formación de SVM eran mucho más complejos y requerían soluciones costosas. SMO

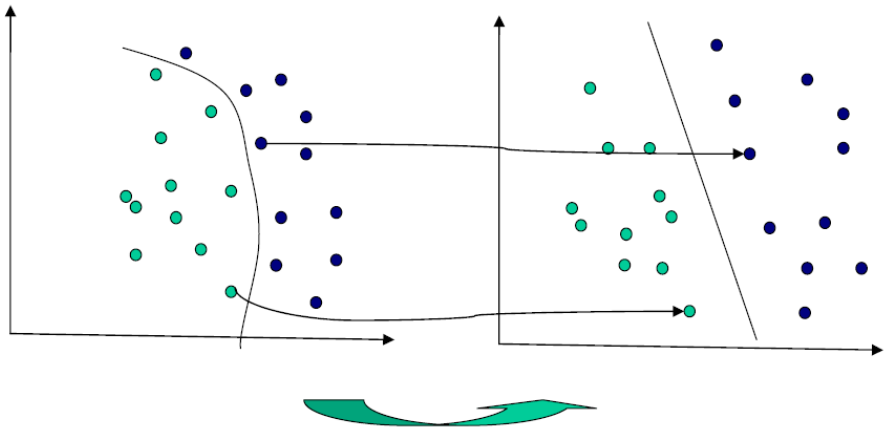


Figura 69 Encontrar la función que transforma a un espacio de dimensión muy superior.

Extiende SVM a los casos en los que los datos no son separables linealmente. Se Intenta mejorar el problema de generalización de Hard Margin cuyo problema de maximización es:

$$Minimizar_{w,b}\langle w, w\rangle \tag{40}$$

$$\text{s. a: } y_i(\langle w, x_i\rangle + b) \geq 1\,, i = 1, \dots m \tag{41}$$

Para evitar las restricciones se introducen las variables de Holgura que permiten burlar las restricciones:

$$\text{s. a: } y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \text{i} = 1 \dots \text{m} \tag{42}$$

$$\xi_i \geq 0, i = 1 \dots m \tag{43}$$

Y para tener el control de las violaciones $\xi_i$

## Soft Margin norma 2

El problema de optimización es:

$$\begin{cases} Minimizar_{w,b}\langle w, w\rangle + C\sum_{i=1}^{m}\xi^2 \\ \text{s. a: } y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \quad i = 1, \dots, m \tag{44}$$

C controla el equilibrio entre los errores de clasificación y el margen.

El lagrangiano primal es el siguiente:

$$L(w, b, \xi, \alpha) = \frac{1}{2}\langle w, w\rangle + \frac{C}{2} + \sum_{i=1}^{m}{\xi_i}^2 - \sum_{i=1}^{m}\alpha_i[y_i\,(\langle w, x_i\rangle + b) - 1 + \xi_i] \tag{45}$$

$\alpha_i \geq 0$ son los multiplicadores de Lagrange

Hallamos la forma dual derivando respecto a $w, \xi \; y \; b$:

$$\frac{\delta L(w,b,\xi,\alpha)}{\delta w} = w - \sum_{i=1}^{m} y_i \alpha_i x_i = 0 \tag{46}$$

$$\frac{\delta L(w,b,\xi,\alpha)}{\delta \xi} = C\xi - \alpha = 0 \tag{47}$$

$$\frac{\delta L(w,b,\xi,\alpha)}{\delta b} = \sum_{i=1}^{m} y_i \alpha_i = 0 \tag{48}$$

Sustituyendo en el primal, se obtiene la siguiente función objetivo:

$$\mathrm{L(w,b,\xi,\alpha)} = \sum_{\mathrm{i=1}}^{\mathrm{m}} \alpha_\mathrm{i} - \frac{1}{2}\sum_{\mathrm{i,j}} \mathrm{y_i y_j \alpha_i \alpha_j \langle x_i, x_j \rangle} - \frac{1}{2\mathrm{C}}\langle \alpha, \alpha \rangle \tag{49}$$

Como maximizar sobre $\alpha$ es lo mismo que maximizar :

$$w(\alpha) = \sum_{\mathrm{i=1}}^{\mathrm{m}} \alpha_\mathrm{i} - \frac{1}{2}\sum_{\mathrm{i,j}} \mathrm{y_i y_j \alpha_i \alpha_j} \left( \langle \mathrm{x_i, x_j} \rangle + \frac{1}{C}\partial_{i,j} \right) \tag{50}$$

donde $\partial_{i,j}$ es la $\partial$ de Kronrcker [153].

En matemática, la **delta de Kronecker** es una función de dos variables, que vale 1 si son iguales, y 0 si son diferentes. Se escribe con el símbolo $\partial_{i,j}$ y se usa como una taquigrafía notacional más que como la función definida a trozos:

Se la nombra así por el matemático Leopold Kronecker (1823-1891). Es una función muy usada en álgebra lineal, la matriz identidad puede ser escrita como:

$$\partial_{i,j} = \begin{cases} 1, & si\ i = j \\ 0, & si\ i \neq j \end{cases} \tag{51}$$

mediante la utilización de Kernels:

Proposición 1

basandonos en la muestra de entrenamiento $S = \{(x_1 y_1), \ldots, (x_m y_m)\}$ y utilizando el espacio caracteristico inducido por el Kernel $k(x, z)$ y dando por hecho que $\alpha^*$ resuelve el problema de optimización cuadratico:

$$w(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j \left( \langle x_i, x_j \rangle + \frac{1}{C} \partial_{i,j} \right) \tag{50}$$

$$s.a: \sum_{i=1}^{m} y_i \alpha_i = 0 \tag{48}$$

$$\alpha_i \geq 0, i = 1, \ldots, m \tag{52}$$

Proposición 2

Sea $f(x) = \sum_{i=1}^{m} y_i \alpha_i^* K(x_i, x) + b^*$ siendo $b^*$ se elige de forma que $f(x_i) = 1 - \frac{\alpha_i^*}{C}\ \ \forall i$ con $\neq 0$.

entonces la regla de decisión que se define por $(f(x))$ tiene su equivalencia en el hiperplano en el espacio caracteristico que viene definido implicitamente por $K(x, z)$ que nos resuelve el problema de optimización cuyas variables de holgura se definen en relación al margen geometrico. Ahora el margen optimo es:

$$\gamma = \left( \sum_{i \in SV} \alpha_i^* - \frac{1}{C} \langle \alpha^*, \alpha^* \rangle \right)^{\frac{-1}{2}} \tag{53}$$

Soft Margin norma 1

El problema a resolver de optimización es :

$$\begin{cases} Minimizar_{w,b}\langle w,w\rangle + C\sum_{i=1}^{m}\xi_i \\ \text{s. a: } y_i(\langle w,x_i\rangle + b) \geq 1-\xi_i \\ \xi_i \geq 0 \end{cases} \quad i=1,\dots,m \tag{54}$$

. El Lagrangiano primal de este problema es :

$$L(w,b,\xi,\alpha) = \frac{1}{2}\langle w,w\rangle + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y_i\,(\langle w,x_i\rangle + b) - 1 + \xi_i - \sum_{i=1}^{m} r_i\xi_i \tag{55}$$

donde $\alpha_i \geq 0$ y $r_i \geq 0$

Hallamos la forma dual derivando respecto a $w, \xi\ y\ b$ y sustituyendo las relaciones resultantes en el primal al igual que ya hicimos en norma 2 obteniendo:

$$\mathrm{L(w,b,\xi,\alpha,r)} = \sum_{\mathrm{i=1}}^{\mathrm{m}}\alpha_\mathrm{i} - \frac{1}{2}\sum_{\mathrm{i,j}}\mathrm{y_i y_j \alpha_i \alpha_j}\,\langle \mathrm{x_i, x_j}\rangle \tag{56}$$

Se observa que es similar al problema de máximo margen salvo por la excepción de la restricción $\mathrm{C} - \alpha_\mathrm{i} - \mathrm{r_i} = 0$ siendo $\mathrm{r_i} \geq 0$ lo que obliga a que $\alpha_\mathrm{i} \leq \mathrm{C}$.

Ahora las condiciones de KKT son :

$$\alpha_\mathrm{i}^*[\mathrm{y_i}(\langle \mathrm{x_i, w}\rangle + \mathrm{b}) - 1 + \xi_\mathrm{i}] = 0, \qquad \mathrm{i} = 1,\dots,\mathrm{m} \tag{57}$$

$$\xi_i(\alpha_i - C) = 0 \;\; i = 1,\dots,m \tag{58}$$

Es obvio que si $\xi_\mathrm{i} \neq 0$ entonces $\alpha_\mathrm{i} = \mathrm{C}$. Por otro lado los puntos $\xi_\mathrm{i} \neq 0$ tienen un margen que es menor a $\frac{1}{||\mathrm{w}||}$

Proposición 1

de nuevo usamos la premisa que basandonos en la muestra de entrenamiento $S = \{(x_1 y_1), \ldots, (x_m y_m)\}$ , utilizando el espacio caracteristico inducido por el Kernel $k(x, z)$ y dando por hecho que $\alpha^*$ resuelve el problema de optimización cuadratico:

$$Maximizar\ W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \tag{59}$$

$$s.a: \sum_{i=1}^{m} y_i \alpha_i = 0 \tag{48}$$

$$C \geq \alpha_i \geq 0, i = 1, \ldots, m$$

Proposición 2

Sea $f(x) = \sum_{i=1}^{m} y_i \alpha_i^* K(x_i, x) + b^*$ siendo $b^*$ se elige de forma que $y_i f(x_i) = 1 \quad \forall i$ con $C > \alpha_i^* > 0$.

entonces la regla de decisión que se define por $(f(x))$ tiene su equivalencia en el hiperplano en el espacio caracteristico que viene definido implicitamente por $K(x, z)$ que nos resuelve el problema de optimización cuyas variables de holgura se definen en relación al margen geometrico. Ahora el margen optimo es:

$$\gamma = \left( \sum_{i,j \in SV} y_i y_j \alpha_i^* \alpha_j^* K(x_i x_j) \right)^{\frac{-1}{2}} \tag{60}$$

$b^*$ se obtiene a partir de las KKT, en las que si $\xi_i^* = 0$ e $(y_i(\langle x_i, w^* \rangle + b^*) - 1 + \xi_i^*) = 0$

Proposición 3

La expresión de la norma $w^*$:

$$\langle W^* W^* \rangle = \sum_{i,j} y_i y_j \alpha_i^* \alpha_j^* \, K(x_i x_j) = \sum_{j \in SV} \sum_{i \in SV} y_i y_j \alpha_i^* \alpha_j^* K(x_i x_j) \tag{61}$$

La restricción para $\alpha_i$implica **Box Constaint**, limitar la influencia de outliers. El problema de encontrar un vector que sea un mínimo local.

Las soluciones de ambas formas de Soft Margin están basadas en Hard Margin.

## SRV

Cuando las SVM pueden resolver **problemas de regresión**, se las denomina comúnmente SRV (Support Vector Regression Machines). En la realidad es muy complicado y nada fácil que los ejemplos que se usan para entrenar se ajusten al modelo lineal con errores iguales a cero en la predicción y por esto se recurre al concepto visto anteriormente para clasificación de margen blando. Mediante dos variables de holgura $\xi$se cuantifica la magnitud de dicho error. La suma de todas las variables de holgura permitirá, de alguna forma, medir el coste que se asocia al número de ejemplos con un error de predicción no nulo.

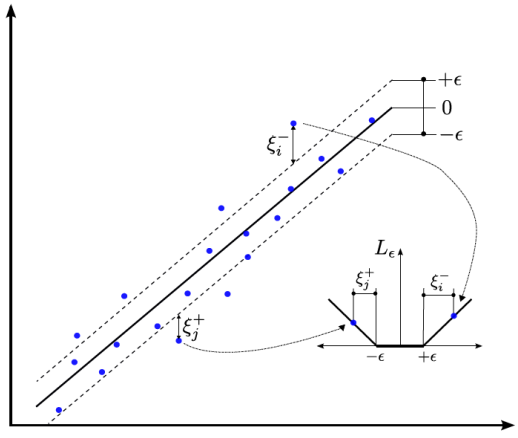



Figura 70 SVR con margen blando se muestra la relación entre las variables de holgura,

asociadas a ejemplos que quedan fuera de la zona tubular, ξ insensible y la función de pérdida L

### 2.3.2.3.4 Redes Neuronales (RN).

**Podemos definir** una red neuronal como un grupo de neuronas aisladas o una agrupación de neuronas que se interconectan y que reciben señales que son procesadas en un proceso similar a un circuito reconocible. Las neuronas se comunican a través de procesos electroquímicos. Las neuronas se excitan una vez superan un determinado umbral y se despolariza transmitiendo la señal que a su vez puede excitar a otras aledañas.

**El aprendizaje se da** cuando de los patrones de activación son reiterados. Las conexiones neuronales se reestructuran de manera que ciertas conexiones sinápticas se refuerzan mientras otras conexiones sinápticas se debilitan.

El número y el tipo de conexiones son determinantes para el número de patrones diferentes que dicho circuito puede identificar. De la misma manera, **el olvido y la pérdida** de capacidad tienen también una base fisiológica y se debe al debilitamiento de sinapsis poco usadas. Si un circuito neuronal concreto se activa poco, sus sinapsis disminuyen y el reconocimiento de cierto patrón puede llegar a perderse.

#### 2.3.2.3.4.1 Historia y conceptos básicos

En 1936 Alan Turing, aunó el estudio del cerebro con el de la computación, pero no fue hasta 1943 cuando Warren McCulloch, un neurofisiologo, y Walter Pitts, un matemático teorizaron sobre la manera en que trabajaban las neuronas y modelaron mediante circuitos eléctricos una red neuronal sencilla creando los fundamentos de la computación neuronal [160].

En 1943 **McCulloch and Pitts** idearon un modelo simple de una **neurona artificial**, esta es la unidad básica de procesamiento en una red neuronal artificial. En la figura se muestra su esquema:

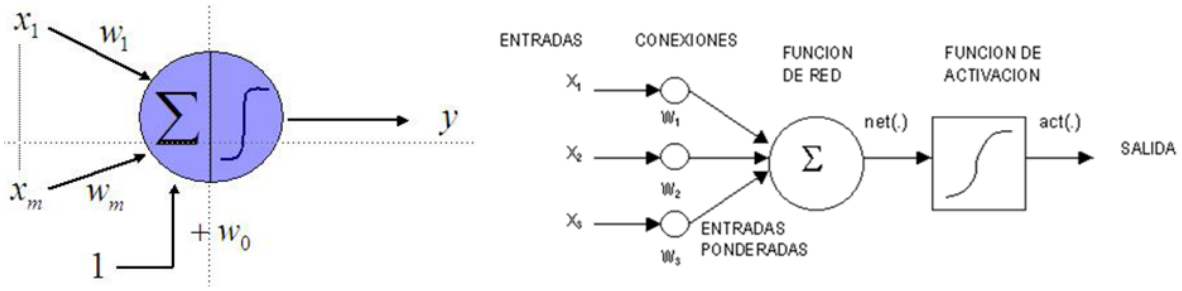



Figura 71 Esquema básico de una neurona artificial

$w = (w_1, \ldots, w_d)^T$ es el **vector de pesos** que equivale a las conexiones psinapticas de una neurona real, $w_0$ es el umbral de activación y $x = (x_1, \ldots, x_d)^T$ es el vector de entrada, $y$ es la salida de la unidad. A partir de la aplicación de la función de activación $\gamma$, la suma ponderada del vector de entrada y el vector de pesos más un sesgo $w_0$se genera una única salida $y = \gamma(\sum_{i=1}^{m} w_i x_i + w_0)$ siendo $\gamma$ una función no lineal que posee una salida binaria $\pm 1$ [167].

Se pueden **simular neuronas con Initff y Simuff** y las funciones de entrenamiento: Trainbp (normal), Trainbpx (rápida), Trainlm (más rápida, mucha memoria de PC) y funcionamiento de transferencia logsig (n) que genera salidas 0 y 1 para entradas a la red neuronal que van desde un valor negativo a positivo, infinito. Se pueden usar otras funciones como tansig(n) y purelin(n) [188].

### ¿Cómo funciona una neurona biológica?

El número de neuronas en el cerebro, varía drásticamente según la especie que estudiemos desde unas pocas hasta miles de millones. Por ejemplo la especie Caenorhabditis elegans, un gusano nematodo de 1 mm de longitud, muy utilizado como animal modelo, posee solo 302 neuronas; y la mosca de la fruta, unas 300.000, que bastan para permitirle exhibir conductas complejas. Se estima que cada cerebro humano unos cien mil millones de neuronas y cada una tiene 1.000 conexiones que la vinculan con sus vecinas [171].

Si el **hombre es hoy en día 10 mil veces más lento** en la velocidad de proceso que las máquinas, en teoría se requerirían 10 mil millones de conexiones para lograr una estructura que se aproxime de modo remoto a la del cerebro. Teniendo en cuenta que aún no se han logrado establecer 20 mil conexiones apropiadas, nos podemos hacer a la idea de la lejanía a la que está la meta última de la cibernética.

Para el sicólogo Carlos León de Bogotá que trabaja en el Laboratorio de Inteligencia Artificial y Epistemología de la Universidad de Ginebra "aún faltan unos dos mil años para que se pueda pensar en un robot senso-motriz capaz de imitar realmente el comportamiento humano" [180].

| Características | Cerebro humano | Computador |
|---|---|---|
| Velocidad de proceso | Entre $10^{-3}$ y $10^{-2}$ seg. | Entre $10^{-8}$ y $10^{-9}$ seg. |
| Estilo de procesamiento | Paralelo | Secuencial (en serie) |
| Número de procesadores | Entre $10^{11}$ y $10^{14}$ | Pocos |
| Conexiones | 10.000 por procesador | Pocas |
| Almacenamiento del conocimiento | Distribuido | En direcciones fijas (posiciones precisas) |
| Tolerancia a fallos | Amplia | Poca o nula |
| Tipo de control del proceso | Autoorganizado (democrático) | Centralizado (dictatorial) |
| Consumo de energía para ejecutar una operación/sg. | $10^{-16}$ Julios | $10^{-6}$ Julios |

Figura 72 Cerebro humano Vs computador [164]

Fue a finales del siglo XIX cuando **Santiago Ramón y Cajal** fijó las neuronas como elementos cuya función se engloba dentro del sistema nervioso. En su tesis, describía a las neuronas como entes discretos que se intercomunicaban entre sí estableciendo una red con conexiones especializadas, idea reconocida como la doctrina de la neurona y es uno de las ideas fundamentales en la disciplina de neurociencia moderna. [168]

La neurona es la **unidad estructural y funcional** del sistema nervioso. Trata estímulos que provienen del medio ambiente, los transforma en impulsos nerviosos y los transmite con precisión, rapidez y a larga distancia. Las señales transmitidas son impulsos nerviosos que

viajan comenzando en las dendritas y llegan a los botones terminales y se conectan con otra neurona, fibras musculares o glándulas. Las conexiones neuronales se denominan **sinapsis**.

Los tres componentes del sistema nervioso: **sensitivo, motor e integrador o mixto** están formados por neuronas que además interconectan .Un estímulo que es captado en alguna región sensorial entrega cierta información que enviada través de las neuronas , se analiza por el componente integrador, el cual puede elaborar una respuesta, que de nuevo es conducida a través de las neuronas. Esta respuesta se ejecuta a través de una **acción motora**, como puede ser la contracción muscular o la secreción glandular.

En la morfología de las neuronas, decir que una neurona típica está compuesta de un **núcleo voluminoso central, pericarion y neuritas (axón y dendritas).**

**El núcleo** se sitúa en el cuerpo celular ocupando generalmente la posición central de manera visible. Tiene **alta actividad** transcripcional y está **formado** por uno o un par de nucléolos de forma prominente y por cromatina dispersa. La **envoltura** del núcleo dispone de gran cantidad de poros nucleares y posee una lámina nuclear altamente desarrollada. Contiene además el llamado **cuerpo accesorio de Cajal** cuya estructura esférica con un diámetro aproximado de 1 µm se traduce en una acumulación de proteínas ricas en los aminoácidos arginina y tirosina.

**Pericarion.** El citoplasma rodea al núcleo y está **lleno de** diversos orgánulos cuyo miembro más notable es la sustancia de Nissl. Esta abundancia de orgánulos que están relacionados con la síntesis proteica se debe a la alta tasa biosintética del pericarion. Los **cuerpos de Nissl** se hallan también en las dendritas, aunque no en el axón, y esto nos permite diferenciar de dendritas y axones en el neurópilo. Son notables en neuronas motoras somáticas

La región del **aparato de Golgi**, posibilita a la sustancia de Nissl una síntesis adicional. Hay presencia de lisosomas primarios y secundarios. Las mitocondrias, son diminutas y redondeadas y poseen generalmente crestas longitudinales.

Si hablamos del citoesqueleto, el pericarion es rico en microtúbulos llamados neurotubulos cuya función es el transporte rápido de las moléculas de proteínas que se sintetizan en el cuerpo celular y filamentos intermedios llamados neurofilamentos y que viajan a través de dendritas y el axón.

**Las Dendritas** son ramificaciones procedentes del **soma neuronal** formadas por proyecciones citoplasmáticas que se envuelven en una membrana plasmática sin mielina. Compuesta de orgánulos y otros componentes, muchos microtúbulos, mitocondrias y pocos neurofilamentos, grumos de Nissl en mayor concentración en la zona que colinda con el soma, retículo endoplasmático liso relacionado con la sinapsis.

**El Axón** se prolonga a través del soma neuronal y está recubierto por células de Schwann en el sistema nervioso de los vertebrados y pueden producir o no mielina. **Puede dividirse, en**: cono axónico, se observa la progresiva desaparición de los grumos de Nissl y la abundancia de microtúbulos y neurofilamentos, segmento inicial donde comienza la mielinización externa y rico en material electrodenso a esta altura en el citoplasma , resto del axón [169] donde ya aparecen los nódulos de Ranvier y las sinapsis

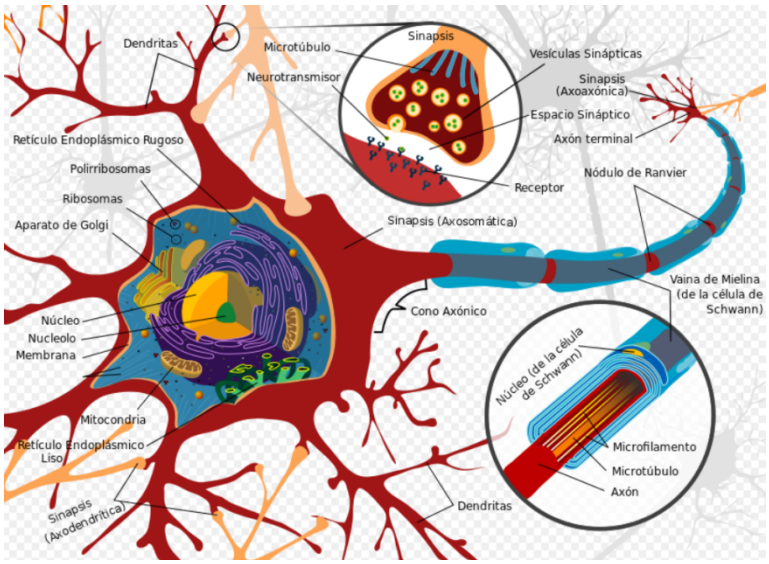



Figura 73 Morfología de una neurona [172]

Un sistema nervioso procesa la información siguiendo un circuito más o menos estándar. Cuando una neurona sensorial recibe un estímulo externo, lo trasmite a otra cercana y de esta forma llega al centro de integración del sistema nervioso.

**Las interneuronas**, transportan la señal a través de sinapsis Se produce una transformación del impulso eléctrico en un mensaje neuroquímico mediante la liberación de unas sustancias llamadas **neurotransmisores** y si deben generar respuestas, se excitan neuronas eferentes que tienen el control de los músculos, glándulas y demás estructuras anatómicas. **El circuito neuronal está compuesto de** neuronas aferentes, eferentes e inter-neuronas[170].

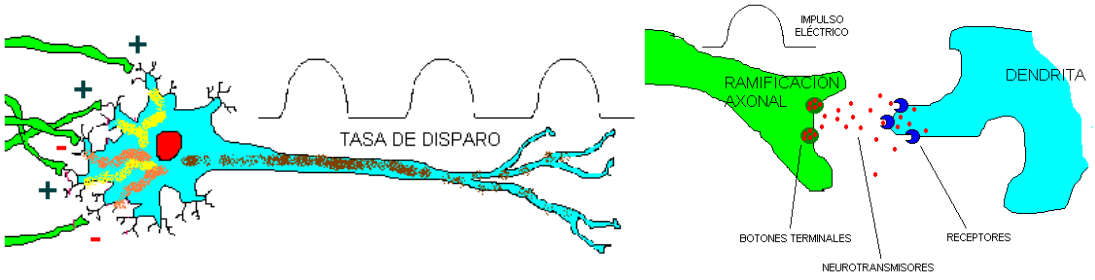



Figura 74 tasa de disparo e impulso eléctrico en la neurona

En el circuito neuronal, la **neurona sufre cambios de estado**, que en su activación vienen caracterizados por las señales eléctricas (potenciales eléctricos) y que a través de la influencia de las neuronas excitadoras, las neuronas inhibidoras de dicho circuito y además por los potenciales e. La función de un determinado grupo de neuronas es llegar a un determinado estado final que va en función de los estímulos externos que provienen de las neuronas sensoriales.

La teoría **de Hopfiled y la regla de Hebb** estiman la relación entre el número de neuronas N que intervienen en reconocer p patrones y la probabilidad de error Pe en el reconocimiento de patrones:

$$Pe \sim \frac{1}{2}\left(1 - \operatorname{erf}\left(\sqrt{2p/N}\right)\right) \tag{62}$$

donde $erf\ ()$ es la función error asociada a la curva de Gauss. [174]

Esta ecuación refleja que cualquier **actividad que realice un profesional entrenado** como por ejemplo un violinista o un deportista de élite , ejecuta una determinada tarea relacionada con su entrenamiento con una probabilidad de error muy pequeña porque este hace que un mayor número de neuronas N se involucre en dicha tarea y eso minimiza mucho la probabilidad de error .

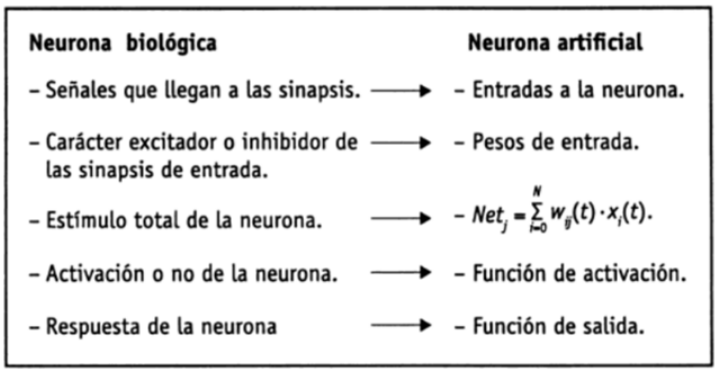



**Figura 75 Analogía entre neurona biológica y artificial [164]** El número, la longitud cuyo rango oscila desde 5 hasta 135 micrómetros, la función, la polaridad o el medidor químico brindan un método morfológico para la clasificación de las neuronas.

| Polaridad | Característica de las neuritas | Mediadoras Químico | Función |
|---|---|---|---|
| Unipolares | Axón muy largo | Colinérgicas | Motoras |
| Bipolares | Axón Corto | Noradrenérgica | Sensoriales |
| Multipolares | Sin Axón | Dopaminégicas | Interneuronas |
| Pseudounipolares | Isodentríticas | Serotoninérgicas | |
| Anaxónicas | Alodentríticas | GABAérgicas | |

Figura 76 características y clasificación de las neuronas

En 1988, Rodolfo Llinás , médico neurofisiólogo colombiano de reconocida trayectoria a nivel mundial por sus aportes al campo de la Neurociencia, presentó un nuevo punto de vista funcional sobre la neurona en su artículo «**The Intrinsic Electrophysiological Properties of Mammalian Neurons: Insights into Central Nervous System Function**» y este articulo marcó el cambio de mentalidad en neurociencia, corrigiendo el dogma sobre que las conexiones y los neurotransmisores liberados por las neuronas, determinaban la función de una neurona [173]

En 1949 Donald Hebb escribió el libro "**La organización del comportamiento**" estableciendo el vínculo entre sicología y fisiología y sentando las bases de la teoría de las redes neuronales. Desarrolló reglas y describió los procesos llevados a cabo en el aprendizaje y lo hizo desde la óptica psicológica. Vinculó la actividad nerviosa con el aprendizaje.

En 1957 **Frank Rosenblatt** desarrolla la red neuronal más antigua, el **Perceptrón**. La característica de este modelo es que es capaz de generalizar aprendiendo nuevos patrones

después de aprender otros parecidos. Este modelo tenia **serias limitaciones** como la incapacidad en la resolución de problemas OR- exclusiva y de clases no separables linealmente.

En 1959 publica el **Teorema de Convergencia del Perceptron** en el libro Principios de Neurodinámica en el que describe que en ciertas condiciones el perceptrón converge hacia un estado finito en el aprendizaje.

En 1960 Bernard Widrow/Marcial Hoff desarrollan el **modelo Adaline**(ADAptative LINear Elements). Esta **fue la primera vez que** una red neuronal se aplica al problema real de utilizar filtros adaptativos para eliminar ecos en las líneas telefónicas.

En 1969 Marvin Minsky/Seymour Papert publican el libro "**Perceptrones: una introducción a la geometría computacional**". Rosenblatt y Minsky se conocían desde la adolescencia, después de haber estudiado con una diferencia de un año en la Escuela Secundaria de Ciencias del Bronx . Los autores son **responsables de un cambio erróneo** en la dirección de la investigación en IA, concentrando los esfuerzos en los sistemas llamados "simbólicos", y contribuyendo a la **llamada de invierno de inteligencia artificial**, surgieron críticas que frenaron, hasta 1982, el crecimiento que estaban experimentando las investigaciones sobre redes neuronales Esta decisión, resultó aparentemente desafortunada, puesto que en la década de 1980, los nuevos descubrimientos mostraron que los pronósticos en el libro estaban equivocados. En su libro Perceptrons, probaron matemáticamente que el perceptrón era incapaz de resolver problemas relativamente fáciles como el aprendizaje de una función no-lineal. [161].

En 1977 James Anderson creó un modelo basado en la idea de que las conexiones neuronales **se refuerzan** cada vez que se activan. Este modelo lineal se llama "**Asociador Lineal**" formado por integradores lineales que suman sus entradas. Además diseño una extensión del Asociador llamada **BSB** (Brain Styate in a Box)[162][163].

- BSB es generalmente una red recurrente auto asociativa.
- Es una red monocapa de n unidades (neuronas) totalmente conectadas.
- Es prácticamente una red de Hopfield con conexiones autorecurrentes que pueden tomar valores diferentes de cero.
- BSB aprende de manera auto organizada usando el postulado de Hebb.
- Es capaz de efectuar cálculos basados en la dinámica de atractores.
- El estado del sistema es atrapado dentro de una caja cuadrada, como una hiperregión delimitada en un hipercubo [+1, -1].
- Los estados estables de la red son representados por las aristas de un hipercubo n dimensional
- Los valores de entrada de cada neurona son un valor analógico.
- El estado de salida de cada neurona es un valor real continuo entre +1 y -1. Tiende a saturarse a estos últimos valores.
- La región de una red con n nodos es un hipercubo n dimensional.
- El BSB posee una función de energía (Lyapunov).
- El modelo BSB es una red que minimiza su estado de energía.
- La función de energía E del modelo BSB decrece con el número de iteraciones hasta alcanzar un punto de equilibrio.

1974 Paul Werbos, gestó la idea del algoritmo **Backpropagatión** o propagación hacia atrás. El significado de esta idea queda perfectamente claro en 1985.

En 1977 lanza la teoría de la resonancia adaptada (ART). Es una arquitectura de red diferente de otras teorías previas que **simula habilidades cerebrales** en cuanto a la memoria a largo y corto plazo.

En 1985 John Hopfield lanza y **revoluciona con su libro** "Computación neuronal de decisiones en problemas de optimización" el renacimiento de las redes neuronales. John Joseph Hopfield (nacido el 15 de julio de, 1933) es un científico americano, ampliamente conocido por su invención de la red neuronal asociativa en 1982. En la actualidad se la conoce comúnmente como la **Red de Hopfield** [165] . Las redes de Hopfield se usan como sistemas de Memoria asociativa con unidades binarias. Las redes son similares al modelo del Perceptrón, presentando una característica adicional: las neuronas en la capa media presentan conexiones de salida hacia otras neuronas de la capa media. Además, cada una de las neuronas de la capa de entrada estan conectadas con una de las neuronas de la capa media, y cada neurona de la capa media se conecta mediante una sola conexión hacia la capa de salidas y estas conexiones (capa de entradas - capa media, y capa media - capa de salidas) no implican calculo de pesos sinápticos ni de valores umbral.

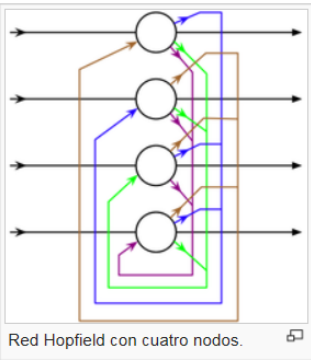



Figura 77 Imagen de Red Hopfield de cuatro nodos[166]

Desde 1985, el compromiso con la idea se extendió a todo el mundo, y surgió una línea de investigación interdisciplinaria, que algunos han llamado "**ciencia cognoscitiva**", en la que aportan con su investigación neurobiólogos, ingenieros de sistemas, matemáticos, sicólogos, lógicos.

Un ejemplo del logro es **Net Talk** [181], una red neuronal resultado de la investigación llevada a cabo a mediados de la década de 1980 por Terrence Sejnowski y Charles Rosenberg [180] que **aprendió por sí misma** mediante retropropagación en un corto plazo de ocho horas a leer un diccionario inglés de 20.000 palabras a través de un captador electrónico. Posteriormente, armada con los conocimientos aprendidos la red fue capaz de leer un texto completo y en voz alta con las pausas de la puntuación en el momento preciso, diferenciaba consonantes y vocales y aplicaba un esquema diferente de codificación para unas y otras. Algo totalmente inimaginable para la computación tradicional. Net Talk cuenta con aproximadamente 300 nodos de neuronas (siendo 80 en la capa escondida) y 20,000 conexiones individuales.

NetTalk establecía primero una sistemática para el tratamiento de los casos más sencillos y posteriormente se servía de patrones complejos para la codificación de los casos excepcionales. Solo el 15%de las unidades ocultas eran activadas por entradas concretas y el resto se basaba en la representación distribuida. [182]

1986 David Rumelhart y G. Hinton redescubren el algoritmo backpropagation y se relanzan las investigaciones así como el desarrollo en redes neuronales.

### 2.3.2.3.4.2 Redes neuronales artificiales (RNA)

**Técnica de aprendizaje automático** que se inspira en el comportamiento del cerebro humano emulando procesos lineales más o menos complejos. Sus unidades, las neuronas, están capacitadas para almacenar y usar posteriormente el conocimiento que han experimentado mediante aprendizaje.

De la misma manera que el cerebro, una RNA se compone de un **conjunto paralelo masivo** de unidades de proceso muy sencillas y es en las conexiones entre estas unidades donde reside la inteligencia de la red. En términos de escala, **un cerebro es muchísimo mayor** que cualquier RNA creada hasta la actualidad.

Aunque las neuronas artificiales no son tan complejas como las biológicas desempeñan su función en interconexión con otras que determinan la función de red. La red neuronal puede estar formada por una o más capas de neuronas. Como vimos anteriormente , cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. Las interconexiones son los pesos sinápticos. [31]. **La función de red** es típicamente el sumatorio ponderado, mientras que **la función de activación** suele ser alguna función de umbral o una función sigmoidal.

Tres funciones en la salida: Una **función de propagación**, **función de activación** y la **función de transferencia**.[176]

La de **propagación** o excitación nos proporciona su valor neto. Es el sumatorio de cada entrada multiplicada por el peso de su interconexión. Las señales excitatorias e inhibitorias recibidas por una neurona se combinan, y en función del estímulo total recibido, la neurona toma un cierto nivel de activación, generando breves impulsos nerviosos con una frecuencia determinada o **tasa de disparo**, y se propaga a lo largo del axón hacia las neuronas con las cuales "conecta". Equivale a la combinación de las señales excitatorias e inhibitorias de las neuronas biológicas. Si es positivo es una conexión excitatoria y si es negativo es inhibitoria.

- **Función lineal de base(LBF)** es el sumatorio de las entradas ponderadas. Se trata de una función de primer orden de tipo hiperplano. Si $j\ y\ n$ son dos unidades conectadas a esta , $X$ es el vector de entrada y $W_j$ el vector de pesos , la función queda de la siguiente manera:

$$net_j(X, W_j) = \sum_{i=1}^{n} x_i w_{ij} \quad (63)$$

  Al representar los pesos utilizamos dos subíndices para indicar que conectan dos unidades, *i* y *j*, dónde *j* se refiere la unidad actual.

- **Función radial de base (RBF):** de tipo hiperesférico, de segundo orden, no lineal. El valor de red representa la distancia a un determinado patrón de referencia.

$$net_j(X, W_j) = \sqrt{\sum_{i=1}^{n}(x_i w_{ij})^2} \quad (64)$$

La de **activación** puede existir o no. Si existe modifica a la anterior función y en caso de no existir la salida es la función de propagación. Es la que mejor define al comportamiento de las neuronas. Se usan diferentes tipos de funciones, desde simples funciones de umbral de escalón a funciones no lineales. Calcula el nivel o estado de activación de la neurona en función de la entrada total.

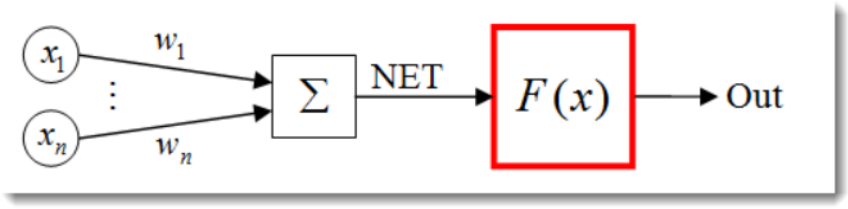



Figura 78 Neurona artificial con la descripción de la función de activación.

$F(x)$ es la función de activación.

$NET$ es la suma ponderada de la primera etapa del cálculo de la información de salida de una neurona.

$\theta$ es el valor del umbral de la función de activación.

- **Función Umbral:** se puede modelar con una función escalón por ejemplo escalón unitario en la que devuelve cero por debajo del umbral y uno si lo supera.

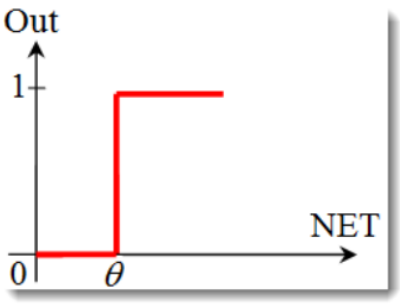



Figura 79 Función Umbral [175]

$$\boldsymbol{Out} = \begin{cases} \mathbf{0}, & \boldsymbol{NET} < 0 \\ \mathbf{1}, & NET \geq 0 \end{cases} \qquad (65)$$

- **Función sigmoidal o logística.** La realidad es que las neuronas emiten impulsos eléctricos con frecuencia variable e incluso en reposo tienen cierta actividad lo que obligó a crear funciones de tipo no lineal como la sigmoidal que en la actualidad es la más empleada.

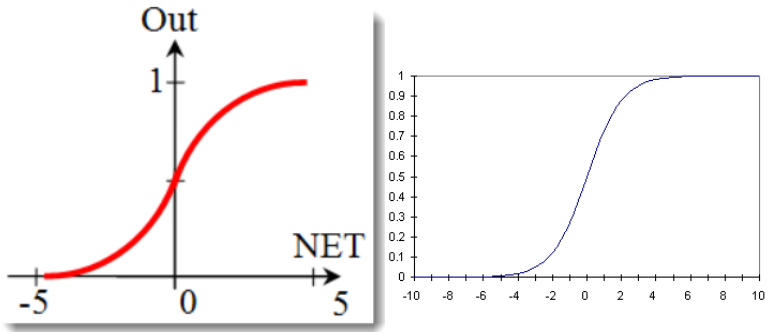



Figura 80 Función Sigmoidea

$$Out = \frac{1}{1 + e^{-NET}} \qquad (66)$$

Es muy utilizada en con señales continuas y en redes neuronales multicapa. Se mueve en el rango entre $0\ y\ 1$ .En la representación grafica observamos que la salida de una unidad vale 0.5 cuando la entrada es nula, lo que quiere decir que tiene cierta actividad aún en ausencia de estimulación. El comportamiento es similar a una neurona real.

Presenta las siguientes características deseables :[177]

- Acomodación de señales muy intensas sin producir saturación.
- Admite señales débiles sin mucha atenuación
- Fácilmente derivable,

- **Tangente Hiperbólica:** Su uso también está indicado en funciones continuas pero tiene la ventaja de que puede devolver valores negativos

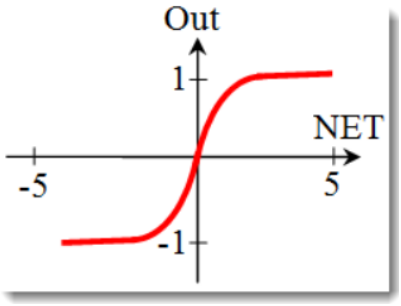



Figura 81 Tangente Hiperbólica

$$Out = th(net)\ ^\circ\ Out = \frac{e^{NET} - e^{NET}}{e^{NET} + e^{-NET}} \tag{67}$$

La de **transferencia**, actúa sobre el valor de la función anterior. limita la salida y se afecta por la interpretación que se pretenda de dichas salidas. Algunas de las funciones más usadas son la sigmoidea para valores de intervalo entre $0\ y\ 1$ o la tangente hiperbólica para obtener valores entre $-1\ y\ 1$

Para diseñar una red definimos la conexión de unas unidades con otras. Lo habitual es formar las unidades en forma de capas, pudiéndose ser la configuración de redes de una, de dos o también de más de dos capas, las llamadas **redes multicapa**.

La arquitectura de una red neuronal se puede describir con una **capa de entradas** que se nutre de la información que se le suministra desde el exterior del sistema, almacenando la información bruta suministrada a la red o realizando un sencillo pre-proceso de la misma; **capas intermedias** ocultas cuya misión principal es hacer el trabajo de red encargadas de extraer, procesar y memorizar la información, las denominamos capas ocultas y una **capa de salida** que almacena la respuesta de la red para que pueda ser leída, además de determinar adecuadamente los pesos de las conexiones; este coctel nos entrega los resultados obtenidos.

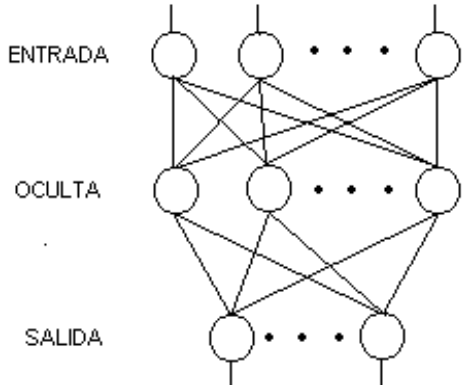



Figura 82 Capas entrada, oculta y salida

Dependiendo de la interconexión de unas capas con otras, podemos clasificarlas en **redes recurrentes** (feed-back) y **redes no recurrentes** o redes en cascada (feed-forward).

En las **redes en cascada** la información viaja de manera unidireccional de unas capas a otras y no se admiten conexiones intracapa.

En las redes recurrentes **la información si puede volver** a lugares ya visitados, por lo que se admiten los bucles y las conexiones intracapa (laterales), incluso intraunidad.

Las conexiones entre capas pueden ser **conexiones totales**, o **parciales** es decir, que cada unidad se conecta con todas las de la siguiente capa, o una unidad se conecta con sólo algunas de las capas de la unidad siguiente.

Las redes neuronales aprenden mediante el mecanismo de la variación en la efectividad de la sinapsis que se traduce en la influencia que se produce de unas neuronas a otras. La memoria o el estado del conocimiento de la red esta representada por la arquitectura, el tipo y la efectividad de las conexiones.

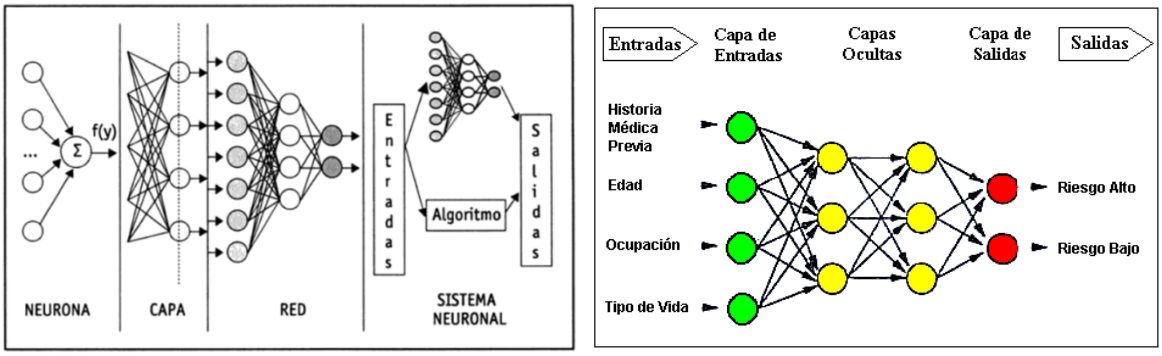



Figura 83 Estructura jerárquica basado en una RNA [164]

| ***Redes Neuronales Biológicas*** | ***Redes Neuronales Artificiales*** |
|---|---|
| Neuronas | Unidades de proceso |
| Conexiones sinápticas | Conexiones ponderadas |
| Efectividad de las sinapsis | Peso de las conexiones |
| Efecto excitatorio o inhibitorio de una conexión | Signo del peso de una conexión |
| Efecto combinado de las sinapsis | Función de propagación o de red |
| Activación -> tasa de disparo | Función de activación -> Salida |

Figura 84 Comparativa redes neuronales biológicas y artificiales.

Una **red** puede ser **síncrona** o **asíncrona** en función del tiempo y el computo de la activación. Si en cada ciclo de aprendizaje se calcula la activación de todas las unidades de una capa, hablamos de redes síncronas, en caso contrario se habla de redes asíncronas o probabilísticas; estas últimas son **menos propensas a caer** en mínimos locales.

Las redes síncronas **suelen emplear** reglas de aprendizaje deterministas mientras que las asíncronas suelen emplear métodos estadísticos.

Las RNA ofrecen una serie de ventajas frente a otro tipo de programación como:

- En la etapa de **aprendizaje** disponen de la habilidad de aprender suministrándole datos de entrada e indicándole cuál es la salida esperada.

- **Auto organización**: El usuario ya no tienen que representar la información internamente en la red, la propia red lo realiza por él.

- La RNA dispone de una mayor **Tolerancia a fallos** y esto se debe a aunque se dañe de forma parcial, como es redundante en el almacenamiento de la información ésta puede seguir trabajando de manera aceptable.

- Si se producen cambios que no sean muy drásticos en la información de entrada, como puedan ser señales con ruido, ligeros cambios en la imagen etc. la red los puede manejar gracias a su mayor **Flexibilidad.**

- Trabajan en **Tiempo real** porque su estructura es paralela.

Las RNA pueden ser **redes analógicas** o **redes discretas** en función del proceso de datos de entrada.

Son analógicas si procesan valores continuos y generalmente acotados. **Ejemplos** de estas redes son Hopfield, Kohonen y las redes de aprendizaje competitivo.

Son redes discretas si procesan datos de entrada de naturaleza discreta como los valores lógicos booleanos. Las máquinas de Boltzmann y Cauchy, y la red discreta de Hopfield trabajan de este modo.

Las RNA **almacenan información aprendida** y esta se registra de manera distribuida en los pesos asociados a las conexiones entre neuronas. Tenemos que definir la asociación entre la información de entrada a la red y la salida que esta nos ofrece. Es lo que se conoce como **memoria asociativa**. Se generan dos tipos distintos de redes basándonos en esta asociación, las **redes heteroasociativas** y las **autoasociativas** [179]

Las heteroasociativas aprenden por parejas de datos $(A_i, B_i)$ de manera que al presentarle información de entrada $A_i$ responda con su salida correspondiente $B_i$. Necesitan como mínimo 2 capas, una para captar y retener la información de entrada ya que puede ser

necesario el acceso a ella varias veces y otra para mantener la salida con la información asociada. El aprendizaje de este tipo de redes puede ser supervisado.

| Redes heteroasociativas | |
|---|---|
| Perceptrón | Linear Associative Memory |
| Adaline / Madaline | Optimal Linear Associative Memory |
| Backpropagation | Drive-Reinforcement |
| Linear Reward Penalty | Fuzzy Associative Memory |
| Associative Reward Penalty | Counterpropagation |
| Adaptive Heuristic Critic | Bidirectional Associate Memory (BAM) |
| Boltzmann Machine | Adaptive BAM |
| Cauchy Machine | Cognitrón / Neocognitrón |
| Learning Matrix | Learning Vector Quantizer |
| Temporal Associative Memory | Adaptive Resonance Theory |

Figura 85 Tipos de redes heterasociativas

Redes autoasociativas: La red aprende determinadas informaciones $A_i$ de manera que al presentarle nueva información de entrada, realizará asociación, respondiendo con el dato almacenado más similar al de entrada. Este tipo de redes se pueden construir con solo una capa de neuronas. El tipo de aprendizaje habitual es no supervisado. Son utilizadas en tareas de filtrado de información para la reconstrucción de datos, eliminar distorsiones o ruido, explorar relaciones entre informaciones similares para facilitar la búsqueda por contenido en bases de datos y para resolver problemas de optimización.

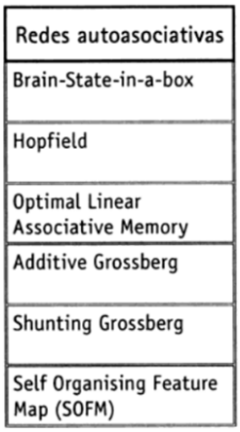



Figura 86 Tipos de redes autoasociativas

Hay muchos **tipos de modelos diferentes** de RNA:

**Perceptrón, Adaline, Perceptrón multicapa, Memorias asociativas, Máquina de Boltzmann, Máquina de Cauchy, Propagación hacia atrás (backpropagation), Redes de Elman, Redes de Hopfield, Red de contrapropagación, Redes de neuronas de base radial, Redes de neuronas de aprendizaje competitivo, Mapas Autoorganizados (RNA) (Redes de Kohonen), Crecimiento dinámico de células, Gas Neuronal CrecienteRedes ART (Adaptative Resonance Theory)** [ 178]

**¿Cómo construimos los modelos?**

Se utiliza la llamada **top-down** que comienza en la aplicación, se pasa al algoritmo y de aquí a la arquitectura:

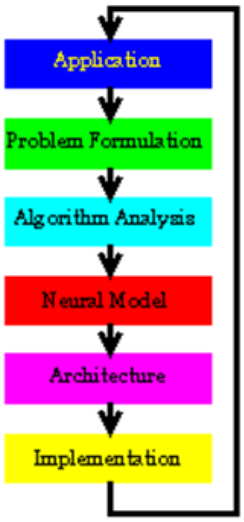



Figura 87 Llamada top-down

Las **aplicaciones más importantes** son para el procesado de la señal y el reconocimiento de patrones.

La **primera etapa** algorítmica es resultado de combinar la teoría matemática y el fundamento heurístico a modelos neuronales. El objetivo final es la nueva revolución informática mediante la construcción de **neurocomputadores** digitales, la imitación de la mente humana copiando su estructura neuronal con la ayuda de las tecnologías VLSI y el procesado adaptativo, digital y paralelo. La nueva computación "neuronal" **es rupturista** con el esquema lineal, secuencial y "cartesiano" de su predecesora, buscando la imitación del cúmulo de procesos que anteceden a una decisión en el cerebro [180].

Observando desde la visión de la **etapa de aplicación**, las redes neuronales **ofrecen** procesado en paralelo adaptativo y no lineal y están teniendo un éxito notable en visión artificial, en el procesado de señales e imágenes, reconocimiento del habla y de caracteres, sistemas expertos, análisis de imágenes médicas, control remoto, control de robots, inspección industrial y exploración científica.

Podemos obtener una clasificación basándonos en el dominio de aplicación de las RNA: **asociación y clasificación, regeneración de patrones, regresión y generalización, y optimización**.

Cuando los patrones de entrada estáticos o de señales temporales, se clasifican, se dice que es una aplicación de **Asociación y Clasificación.** Ya hemos hablado anteriormente de la asociación autoasociación y heteroasociación, de la clasificación supervisada en redes de retropropagación, apropiadas para las aplicaciones que manejan gran cantidad de clases no supervisadas con límites de separación complejos, basándose únicamente en los patrones de entrada, con el tipo de aprendizaje no supervisado.

La regeneración de los patrones se produce cuando se nos presentan problemas de tipo temporal o estático en los que disponemos únicamente de información parcial en el patrón original y se trata la información de manera contextual.

Una red se considera que se **ha entrenado de forma exit**osa si es capaz de aproximar los valores de los patrones de entrenamiento y puede dar interpolaciones "suaves" para el espacio de datos no entrenado [176]. El objetivo de la Generalización es obtener una respuesta acertada a la salida a un estímulo de entrada que la maquina desconoce porque no ha sido entrenado con anterioridad.

La búsqueda del mínimo absoluto de una **función de energía** para lo cual se realiza primero la definición de esta función y después se puede definir rápidamente los pesos sinápticos, que a veces tiene un coste importante además de limitaciones y esto es misión de las RNA´s. Se suelen utilizar propuestas estadísticas para afrontar el mayor problema en la optimización a saber, converger a un mínimo local en vez del mínimo absoluto.

Ejemplos de **problemas que se han afrontado con RNA´s son**: Procesamiento de lenguaje natural, Aprendizaje de gramáticas, Compresión de imágenes, Reconocimiento de caracteres,

Reconocimiento de patrones en imágenes, Visión artificial en robots industriales, Filtro de ruido, Problemas de combinatoria, Modelado y predicción de indicadores económicos.

Se puede representar la arquitectura de una red neuronal con aplicaciones populares como:

- MATLAB (Neural Network Toolbox)
- NeuroSolution
- SNNS

### 2.3.2.3.4.2.1 Primera ola en redes neuronales artificiales. Perceptron y Adaline.

#### Perceptrón

El perceptrón dentro del campo de las redes neuronales tiene **dos acepciones**, nos podemos referir a él como un tipo de **red neuronal** que como ya hemos visto a lo largo de este trabajo, fue desarrollado por Frank Rosenblatt o como una **simple neurona artificial** en forma de discriminador lineal dentro este modelo de red cuyo algoritmo está capacitado para generar un criterio de selección para tomar componentes que forman parte de un grupo mayor. Tiene como **limitación** la posibilidad de poder separar elementos que deseamos de los que no deseamos en una representación en un hiperplano (objeto de dimensión N-1 contenida en un espacio de dimensión N).**No puede resolver** por ejemplo la función XOR

Como unidad básica, podemos ponerlos a trabajar con otros perceptrones formando **redes complejas**.

Ya vimos anteriormente en que consiste y como trabaja una neurona artificial intentando modelar el comportamiento de la neurona biológica.

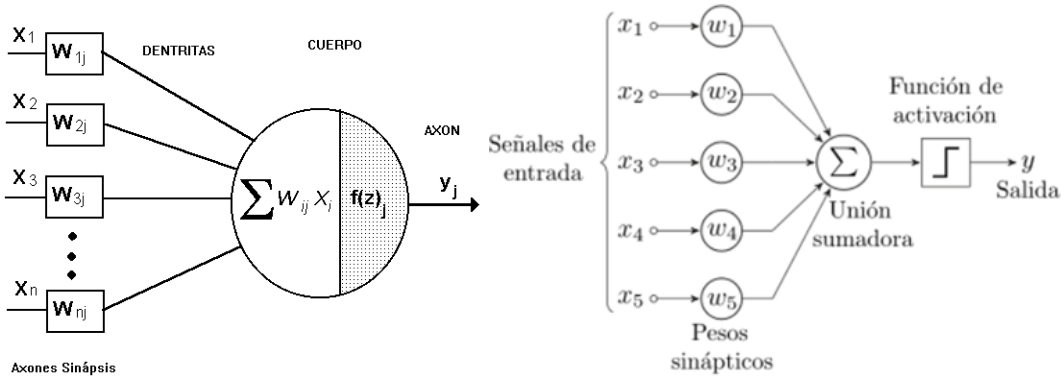



Figura 88 Representaciones del Perceptron de McCulloch-Pitts

Buscando similitudes entre la biológica y la artificial, el cuerpo de la neurona artificial, está representado por un sumador lineal de los estímulos externos $z_j$, seguida de una función no lineal $y_j = f(z_j)$. La función $f(z_j)$ es la función de activación, y como ya sabemos es la función que suma los estímulos para cuantificar la actividad de salida de la neurona.

Este modelo es la unidad básica de la gran parte de las arquitecturas RNA.

Hemos hablado de las diferentes funciones de activación que en el caso de la aplicación del perceptrón, pueden ser lineales, sigmoidales, umbral de disparo. La eficiencia sináptica se representa por factores de peso de interconexión $w_{ij}$, desde la neurona i, hasta la neurona j.

Los pesos y las funciones f(z) marcan la excitación o inhibición dictando la operación de la red. Como las funciones no se suelen modificar, recae la mayor responsabilidad en los pesos de la sinapsis.

En el perceptrón, el sumatorio de las entradas multiplicadas por sus pesos W correspondientes son evaluados por la función que marca finalmente la activación. El perceptrón usa una matriz para representar las redes neuronales. Es un discriminador terciario que traza un vector binario como entrada x a un valor de salida único f(x) por medio de dicha matriz

$$f(x) = \begin{cases} 1, & si\ w \times x - u > 0 \\ 0, & en\ otro\ caso \end{cases} \quad (68)$$

$w \times x$ es el producto escalar, u es el umbral y f(x) marca la clasificación. La suma ponderada de las entradas debe ser mayor que el umbral para cambiar el estado de la neurona de cero a uno.

Según las capas el perceptrón puede ser:

**Básico** (o simple) que consta de **dos capas** de neuronas con N celdas de entrada y M neuronas de salida. El **Perceptrón simple** de dos capas cuyas entradas se construyen con neuronas lineales, analógicas, y la de salida con la función de activación de tipo umbral escalón, digital), Este perceptron, nada más que puede establecer dos zonas separadas por una frontera de tipo lineal en el espacio de patrones de entrada, donde se tendría un hiperplano.

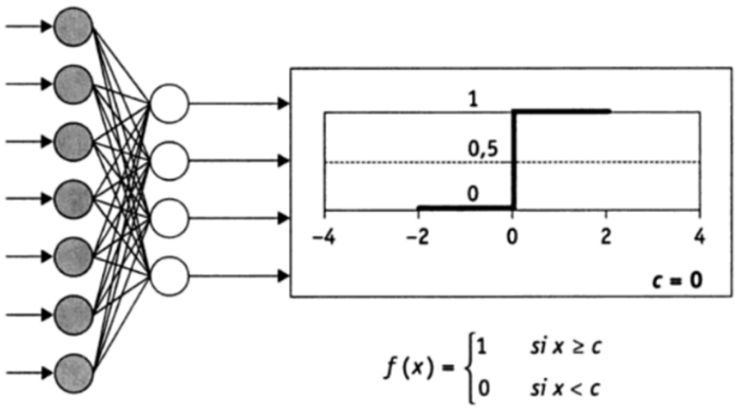



Figura 89 Perceptrón simple

Cada salida de Perceptrón simple se calcula de la siguiente manera:

$$y_j = f\left(\sum_{i=0}^{n} w_{ij}\, x_j\right), \forall j, 1 \leq j \ \leq M \tag{69}$$

la **función de activación de tipo e**scalón:

$$y_j = H\left(\sum_{i=0}^{n} w_{i,j} x_j\right), \forall j, 1 \leq j \leq M \tag{70}$$

$$\text{f(x)} = \begin{cases} 1, & \text{si } \sum_{i=0}^{n} w_{ij} x_j \geq 0 \\ & \forall\, j, 1 \leq j \leq M \\ 0, & \text{si } \sum_{i=0}^{n} w_{ij} x_j < 0 \end{cases} \tag{71}$$

y la expresión de la regla de aprendizaje es por medio de la **regla de aprendizaje delta**:

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{72}$$

en cada peso $\Delta w_{ij}(t)$ se realiza un ajuste donde $\alpha$ es la razón del aprendizaje, $t_j$ es el valor deseado, $y_j$ es el valor obtenido y $x_i$ es la entrada del perceptrón.

$$\Delta w_{ij}(t) = \alpha\big(t_j - y_j\big)x_i \tag{73}$$

Las aplicaciones del perceptrón simple se encomiendan a tareas de clasificación, representación de funciones booleanas que permitan discriminar entre dos clases linealmente separable. **No posibilita** las que son no separables como el problema OR exclusivo o XOR.

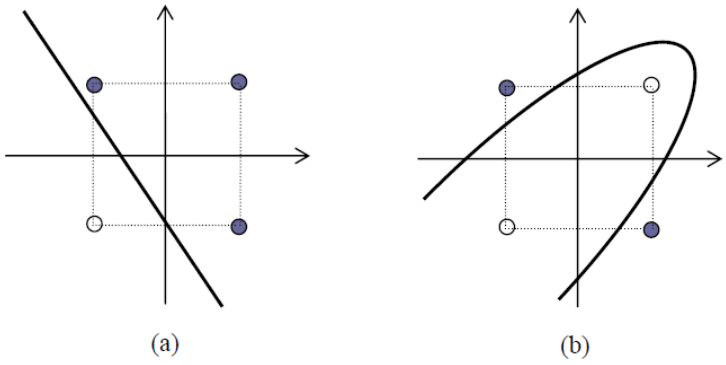



Figura 90 a) Patrones separables linealmente b) patrones no separables linealmente.

Para su **aprendizaje supervisado del perceptron** es necesario conocer que :[183]:

- **Los datos** son puntos en un espacio multidimensional.
- Se tendrá un **conjunto de observaciones** u ejemplos, puntos representados en el hiperplano, de los que se sabe su categoría o clase.
- Hay que determinar la **ecuación del hiperplano** que separa los ejemplos de un tipo de los de otro.
- La ecuación del hiperplano **se deduce** a partir de los ejemplos.
- El aprendizaje es **proceso iterativo** supervisado.
- Hasta que se encuentra el hiperplano discriminante, **se deben ir modificando** los parámetros de la red (pesos y umbral).
- Este proceso de aprendizaje supervisado tiene un **número finito** de iteraciones.

Los **pasos a seguir para el aprendizaje** son:

- se realiza la inicialización aleatoria de los pesos y umbral.
- Se toma un patrón de entrada-salida.
- Se calcula la salida de la red.
- Si y = d(x) (clasificación correcta).
- Si y ≠ d(x) (clasificación incorrecta) se modifican los parámetros.
- Se vuelve al paso 2 hasta completar el conjunto de patrones de entrenamiento.

- Se repiten los pasos anteriores hasta alcanzar el criterio de parada.

El algoritmo del perceptrón es el siguiente:

**PASO 1: Inicialización de variables**

**PASO 2: Bucle de iteraciones (hasta condición de parada)**

**Paso 2.1: Bucle de paso por todos los ejemplos**

**2.1.0 Leer valores del ejemplo**

**2.1.1 Calcular error en ese ejemplo**

**2.1.2 Actualizar los pesos según el error de ese ejemplo**

**2.1.2.1 Actualizar los pesos de las entradas**

**2.1.2.2 Actualizar el bias (= -umbral)**

**2.1.3 Incrementar contador de ejemplos**

**Paso 2.2.: Ver si el vector de pesos es correcto**

**Paso 2.3.: Incrementar el contador de iteraciones**

**PASO 3: Salida**

Figura 91 Algoritmo del perceptrón simple

## Seudocódigo del Algoritmo del perceptrón Discreto:

VARIABLES

$I$ $(nº\ de\ iteraciones)$

$K$ $(contador\ de\ ejemplos)$

$W[i{:}\,1..n+1]$ $(pesos\ del\ perceptron)$

$X[i{:}\,1..n]$ $(valores\ de\ la\ entrada\ en\ el\ ejemplo\ actual)$

$Y$ $(valor\ de\ la\ salida\ en\ el\ ejemplo\ actual)$

$SP$ $(salida\ del\ perceptrón\ con\ la\ entrada\ del\ ejemplo\ actual)$

$Error$ $(error\ en\ el\ ejemplo\ actual)$

$T$ $(Variable\ para\ indicar\ si\ el\ vector\ es\ correcto)$

**PASO 1: Inicialización de variables**

*I=0;*

*Para i=1..n+1*

*W[i]=<nº aleatorio* (normalmente entre –1 y 1*)>* (muy importante que el nº sea aleatorio,

distinto en cada ejecución)

*T=false*

**PASO 2: Bucle de iteraciones**

*Mientras (I < NºMaxIteraciones) y (no T)*

**Paso 2.1: Bucle de paso por todos los ejemplos**

*Abrir(<fichero de ejemplos>)*

*K=0*

*Mientras no este en el final del fichero*

**2.1.0 Leer valores del ejemplo**

*Leer(x[i] (i=1..n), y)*

**2.1.1 Calcular error en ese ejemplo**

*SP = W[n+1] + Σ x[i] * W[i]* (salida del perceptrón con la entrada del ejemplo h)

*Si SP < 0, entonces SP= 0, en otro caso SP=1*

*Error = y - SP* (diferencia entre el valor de y en el ejemplo h y SP)

**2.1.2 Actualizar los pesos según el error de ese ejemplo**

2.1.2.1 Actualizar los pesos de las entradas

*Para cada i=1..n*

*W[i] = W[i] + x[i]*Error*

2.1.2.2 Actualizar el bias (= -umbral)

*W[n+1]=W[n+1] + Error*

2.1.3 Incrementar contador de ejemplos

*k=k+1*

*Cerrar(<fichero de ejemplos>)*

Paso 2.2.: Ver si el vector de pesos es correcto

*T= True*

*Abrir(<fichero de ejemplos>)*

*Mientras ( no este en el final del fichero y T)*

*Leer(x[i] (i=1..n), y)*

*SP = W[n+1] + Σ x[i] * W[i]*

*Si SP < 0, entonces SP= 0, en otro caso SP=1*

*Error = y – SP*

*Si Error ≠ 0, T = False*

*Cerrar(<fichero de ejemplos>)*

Paso 2.3.: Incrementar el contador de iteraciones

*I=I+1*

PASO 3: Salida

*Escribe( "El perceptrón aprendido es el de pesos:" )*

*Para i=1..n*

*Escribe(" W",i,"=",W[i])*

*Escribe("Con bias =", W[n+1])*

Un **Perceptrón multicapa** es una red con alimentación hacia delante, se compone de distintas capas de neuronas entre la entrada y la salida de la misma, esta red establece zonas de decisión de mayor complejidad que las de dos semiplanos, como lo hace el Perceptrón de un solo nivel.

Un Perceptrón con tres niveles de neuronas ya puede formar cualquier región convexa.

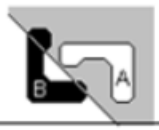
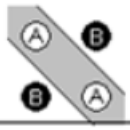
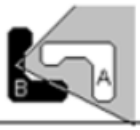
| Estructura | Regiones de Desición | Problema de la XOR | Clases con Regiones Mezcladas | Formas de Regiones más Generales |
|---|---|---|---|---|
| 1 Capa | Medio Plano Limitado por un Hiperplano |  |  |  |
| 2 Capas | Regiones Cerradas o Convexas |  |  |  |
| 3 Capas | Complejidad Arbitraria Limitada por el Número de Neuronas |  |  |  |

Figura 92 Capacidades del Perceptrón multicapa con dos y tres capas y con una única neurona en la capa de salida [185]

Un Perceptrón con cuatro capas puede generar regiones de decisión arbitrariamente complejas separando las clases en hipercubos que requieren 2n neuronas en la segunda capa siendo n el número de las entradas a la red, una por cada lado del hipercubo y otra que se encarga de llevar a cabo el AND lógico en la tercera capa de la salida de los nodos del anterior nivel.

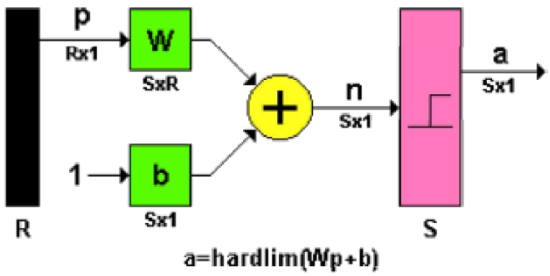



Figura 93 Representaciones del Perceptrón multicapa [187]

Las salidas de los nodos del tercer nivel solo se activaran para las entradas del hipercubo y estos se asignan a la región de decisión adecuada a través de la conexión de salida de cada nodo del tercer nivel solo con la neurona de salida, cuarta etapa. En cada nodo de salida se lleva a cabo la operación OR. Se puede generalizar para que la forma de las regiones convexas sea arbitraria. Este perceptrón puede tener implicaciones en la generación de ruido pero es más tolerante a fallos por el sistema redundante.

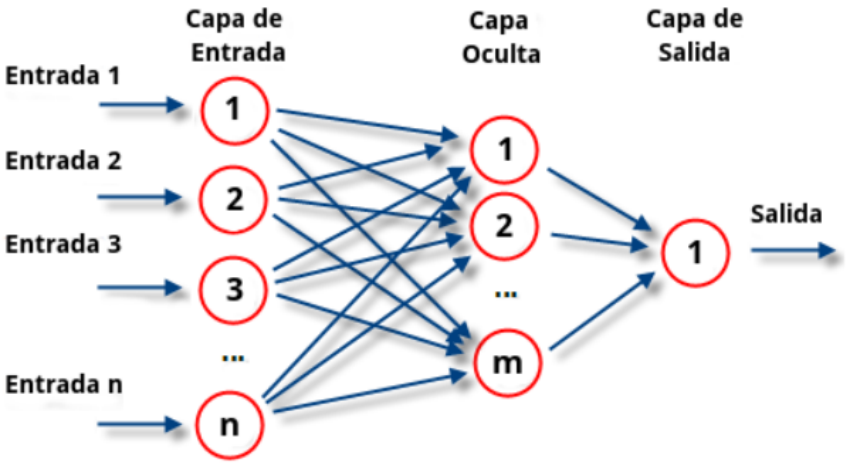



Figura 94 Imagen de una red neuronal artificial (RNA) perceptrón multicapa (MLP)

Las capas pueden clasificarse en tres tipos[184]:

- Capa de entrada: Formada por las neuronas que tratan los patrones de entrada en la red. no se produce procesamiento

- **Capas ocultas**: Aquellas neuronas cuyas entradas provienen de capas anteriores y cuyas salidas pasan a neuronas de capas posteriores.

- **Capa de salida**: Neuronas cuyos valores de salida se corresponden con las salidas de toda la red.

El perceptrón funciona en problemas que escapan de ser decisiones simples y sirve como predictor de datos como por ejemplo financieros además de mediciones de gas y electricidad y de muchas otras aplicaciones como codificación, diagnósticos médicos.

**Adaline (Adaptative linear Neuron).**

Desarrollada por el profesor Bernie Widrow y su alumno Ted Hoff en la Universidad de Stanford en 1960. Junto con el perceptrón es **una de las estructuras neuronales más sencillas** y permite realizar aprendizaje. Utiliza funciones de transferencia lineales pudiendo conectarse de forma sucesiva entre sí (red Madaline). Su estructura es de una capa neuronal con n neuronas o salidas, estas salidas son directamente la salida real de la red y m entradas que representan un vector de $x$ entradas pertenecientes a $\mathbb{R}^m$ . **En el entrenamiento**, el Adaline tiene en cuenta el grado de corrección de la salida estimada respecto a la deseada es decir para un patrón de entrada se tiene en cuenta el error que se produce observando el valor real que espera y lo **compara con la salida** producida en la red; se **diferencia con el perceptrón** en como utiliza la regla de aprendizaje, se utiliza la regla de **Windrow-Hoff** y posee la ventaja de que su gráfica de error es un hiperparaboloide que dispone de un único mínimo global, o una recta de infinitos mínimos, que son todos ellos globales. Esto evita problemas en el entrenamiento.

Por cada neurona hay un vector $w$ de pesos indicando la ponderación entre la neurona y los valores de entrada, $\theta$ es una constante y la función de activación esta definida como

$$y = \sum_{i=1}^{n} x_i w_i + \theta \tag{74}$$

La salida de la red con su proceso de actualización de pesos se corresponde con la siguiente formulación [164]:

$$y_j = \sum_{i=0}^{n} w_{ij} x_i \tag{75}$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{76}$$

$$\Delta w_{ij}(t) = \alpha\big(t_j - y_j\big)x_i \text{ o bien } \Delta w_{ij} = \alpha \sum_{\mu=1}^{p} (t_j^{\mu} - y_j^{\mu}) x_i^{\mu} \tag{77}$$

La función de error es cuadrática con relación a los pesos y describe una superficie paraboloide alcanzando siempre un mínimo global. Se utiliza como filtros adaptativos como por ejemplo en la cancelación de ruidos en transmisión de señales

### 2.3.2.3.4.2.2 Segunda ola: Redes de dos capas, back-propagation (BPN).

En 1986, Rumelhart, Hinton y Williams, popularizaron un método ideado en los 70 por Werbos que lamentablemente se quedó en el olvido y más tarde, a principios de los 80, fue redescubierto por Parker y Rumelhart. Hablamos del algoritmo **Red back-propagation** (propagación del error hacia atrás) que consiste en una red formada por neuronas artificiales que aprende el vínculo que existe entre los patrones de entrada y las clases correspondientes utilizando para ello varios niveles neuronales.

Parker y Rumelhart **generalizaron la regla delta de aprendizaje**, ampliando de manera importante el rango de uso de las RNA´s. Desde 1986 han surgido versiones nuevas que han tratado de aumentar la velocidad de convergencia del algoritmo y de superar algunos

inconvenientes, como la **tendencia a alcanzar mínimos locales y no globales**. Esta red trabaja bajo aprendizaje supervisado y aprovecha su naturaleza paralela para reducir los tiempos necesarios para determinar la correspondencia entre patrones.

En una red Backpropagation existe una capa de entrada con $n$ neuronas y una capa de salida con $m$ neuronas y al menos una capa oculta de neuronas internas.

**No existe una técnica** para determinar el número de capas ocultas necesarias, ni el número de neuronas que debe contener cada una de ellas para un problema determinado, es el programador quien bajo su experiencia determina la elección, pero eso sí, debe cumplir con las limitaciones que marcan el limite computacional.




Figura 95 ¿cómo elegir el número de capas?

Su funcionamiento se basa en el aprendizaje de un conjunto ejemplo de pares de entrada-salida predefinidos. Cada neurona de una capa (excepto las de entrada) recibe entradas de todas las neuronas de la capa anterior y envía su salida a todas las neuronas de la capa posterior (excepto las de salida). La técnica **reajusta los pesos de conexión** en cada neurona de cada capa y para esto el algoritmo se trabaja en el error que producen las neuronas en el resultado a su salida. Esto se logra minimizando **el error medio cuadrático** en cada iteración del proceso de aprendizaje.

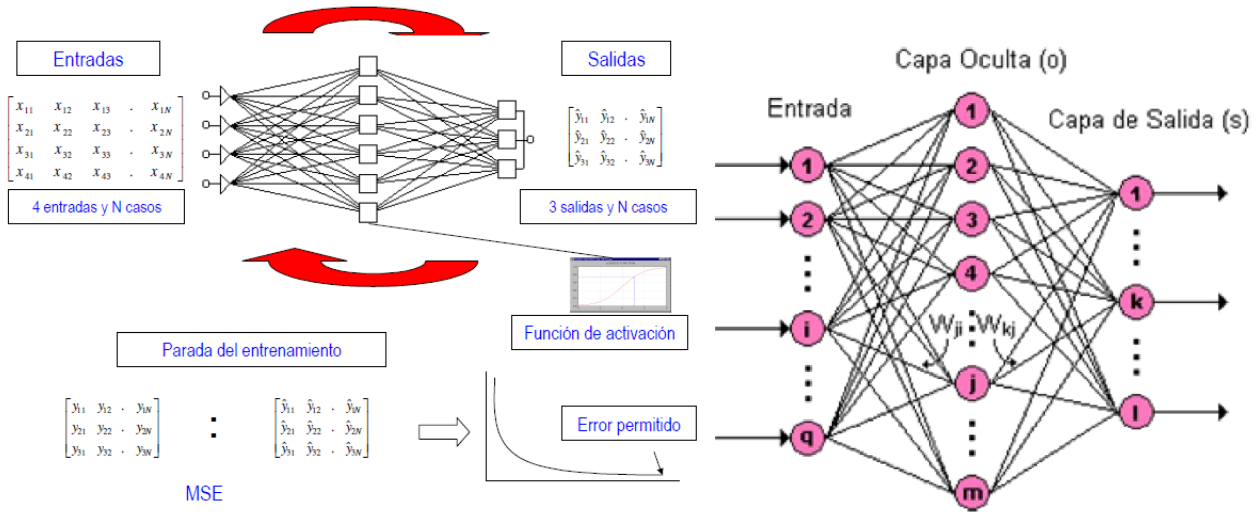



Figura 96 Diagrama de la Red back-propagation

¿Cómo lo hace?, En una **primera fase** se aplica un patrón de entrada para estimular las neuronas de la primera capa de la red y se propaga por las capas superiores hasta generar un resultado de salida; ahora se compara este con el que hubiéramos deseado obtener, calculando el error producido en cada neurona de la capa de salida. En la **segunda fase**, estos errores se transmiten hacia atrás capa por capa desde la última hacia todas las neuronas de la capa anterior que han contribuido a generar dicha salida. No hay conexiones hacia atrás feedback ni laterales entre las neuronas de la misma capa. A las neuronas se les proporciona el error que describe su relativa aportación al error total y se produce el ajuste.

Se ha demostrado que durante el entrenamiento se producen relaciones internas entre neuronas cuya finalidad es organizar los datos de entrenamiento en las distintas clases. Extrapolando, todas las unidades de la capa oculta se asocian de alguna forma a las peculiaridades del patrón de entrada a consecuencia del entrenamiento para encontrar la representación interna. Esto **a menudo no resulta evidente** en el observador humano

Ya **entrenada la red se podrá** aplicar la misma relación a los nuevos vectores de entrada dando una salida activa si presentan similitudes con los ejemplos. Las unidades de las capas

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

La función de activación **debe ser derivable y creciente**. Las funciones más habituales son la sigmoide, logística y la tangente hiperbólica.

El algoritmo Backpropagation para redes multicapa es una **generalización del algoritmo LMS**; los dos algoritmos actualizan los pesos y ganancias con base en el error medio cuadrático.

**Pasos para aplicar el Algoritmo de Entrenamiento** [186]

- Paso 1. Se inicializan los pesos de la red con pequeños valores aleatorios.
- Paso 2. Presentar un patrón de entrada y especificar la salida deseada que debe generar la red.
- Paso 3. Calcular la salida actual de la red mediante la presentación de las entradas a la red y calculando la salida de cada capa hasta llegar a la capa de salida, ésta será la salida de la red.

Para describir matemáticamente el proceso de aprendizaje definimos:

W: Matriz de pesos asignada a cada una de las entradas de la red de dimensiones

$SxR$, con S igual al número de neuronas, y R la dimensión del vector de entrada

$p$: Vector de entradas a la red de dimensiones Rx1

$b$: Vector de ganancias de la red de dimensiones Sx1

$y$: Calculo de la RNA

$d$: valor esperado.

Vamos a obtener primero la regla delta para varias neuronas de salida dando por hecho que la activación es lineal [187].

Para **totalizar el error** se realiza el sumatorio para todos los ejemplos de entrenamiento. El error cuadrático que se produce cuando se presenta el ejemplo $p$ es el siguiente:

$$E_p = \frac{1}{2}\sum_j \left(d_{pj} - y_{pj}\right)^2 \tag{78}$$

Para ver como cambia el error con respecto a la salida y como cambia esta respecto al cambio de los parámetros de la red aplicamos la **regla de la cadena** y derivamos respecto a los pesos:

$$\begin{cases} \dfrac{\delta E_p}{\delta w_{ji}} = \dfrac{\delta E_p}{\delta y_{pj}} \dfrac{\delta y_{pj}}{\delta w_{ji}} \\ \dfrac{\delta E_p}{\delta y_{pj}} = -d_{pj-} y_{pj} = -\delta_{pj} \end{cases} \tag{79}$$

El aporte de la neurona j al error es proprcional a $\delta_{pj}$.

$$y_{pj} = \sum w_{ji} x_{pi} \Rightarrow \frac{\delta y_{pj}}{\delta w_{ji}} = x_{pi} \tag{80}$$

por tanto:

$$\frac{-\delta E_{Ep}}{\delta w_{ji}} = \delta_{pj} x_{pi} \tag{81}$$

Resumiendo, ajustamos pesos mediante la regla delta:

$$\Delta_p w_{ij} = \eta\left(d_{pj} - y_{pj}\right) x_{pi} = \eta \delta_{pj} x_{pi}\,;\ \delta_{pj} = d_{pj-} y_{pj} \tag{82}$$

La regla delta trabaja de forma acorde al **gradiente descendente**. La variación del error global con los pesos después de procesar el conjunto completo de entrenamiento se muestra en proporción a la derivada total que suma las derivadas sobre todos los ejemplos de entrenamiento. Esto ocurre cuando los pesos o bien no se modifican o si lo hacen, es de manera suave después de procesar cada ejemplo. Como en este caso no hay neuronas ocultas **se garantiza que el error es único y mínimo**.

Esto no ocurre así si existen capas ocultas porque es realmente complicado obtener el valor mínimo en valores discretos. En el caso de valores continuos hay métodos satisfactorios mediante cálculos de derivadas.

Si consideramos diferenciable y no decreciente la función de activación $f$, con $z$ como representación de la excitación de la neurona:

$$z_{pj} = \sum w_{ji} x_{pi} \Rightarrow y_{pj} = f_j(z_{pj}) \tag{83}$$

Aplicando la regala de la cadena de nuevo y Suponiendo que la función de error es la misma que hemos utilizado anteriormente:

$$\frac{\delta \mathrm{E_p}}{\delta \mathrm{w_{ji}}} = \frac{\delta \mathrm{E_p}}{\delta \mathrm{z_{pj}}} \frac{\delta \mathrm{z_{pj}}}{\delta \mathrm{w_{ji}}} \tag{84}$$

$$\frac{\delta \mathrm{z_{pj}}}{\delta \mathrm{w_{ji}}} = \frac{\delta}{\delta \mathrm{w_{ji}}} \sum \mathrm{w_{ji} x_{pi}} = \mathrm{x_{pi}} \tag{85}$$

Obtenemos ahora las expresiones siguientes:

$$-\frac{\delta E_p}{\delta w_{ji}} = \delta_{pj} x_{pi} \tag{86}$$

Definiendo $\delta_{pj} = -\frac{\delta E_p}{\delta z_{pj}}$ $\Delta_p w_{ij} = \eta \delta_{pj} x_{pi}$

Podemos obtener las $\delta_{pj}$ retropropagando de manera recursiva a partir de los valores de error que obtenemos en la capa de salida para las capas ocultas. Aplicando la regla de la cadena y realizando derivadas parciales una del error respecto a la salida de la neurona y otra de la salida de la neurona con respecto a su excitación (derivada de la función de activación)

$$\delta_{pj} = -\frac{\delta E_p}{\delta z_{pj}} = \frac{\delta E_p}{\delta y_{pj}} \frac{\delta y_{pj}}{\delta z_{pj}} = -\frac{\delta E_p}{\delta y_{pj}} f_j'(z_{pj}) \tag{87}$$

Para el cálculo del primer factor, distinguimos si la neurona pertenece a la capa de salida o de una capa anterior. Si es de la capa de salida, el cálculo es directo. Así la expresión para los $\delta_{pj}$ es inmediata:

$$\frac{\delta E_p}{\delta y_{pj}} = -(d_{pj} - y_{pj}) \Rightarrow\Rightarrow \delta_{pj} = (d_{pj} - y_{pj}) f'_j(z_{pj}) \tag{88}$$

Para las neuronas de las capas ocultas como no disponemos de expresión alguna para el error, el cálculo es más complejo:

$$\sum_p \frac{\delta E_p}{\delta z_{pk}} \frac{\delta z_{pk}}{\delta y_{pj}} = \sum_p \frac{\delta E_p}{\delta z_{pk}} \frac{\delta}{\delta y_{pj}} \sum w_{ij} x_{pi} = \sum_p \frac{\delta E_p}{\delta z_{pk}} w_{kj} = -\sum_k \delta_{pk} w_{kj} \tag{89}$$

El resultado definitivo para las neuronas de capas ocultas:

$$\delta_{pj} = f'_j(z_{pj}) \sum_k \delta_{pk} w_{kj} \tag{90}$$

Resumiendo, tenemos **tres ecuaciones que nos indican como se adaptan los pesos y los valores** de $\delta_{pj}$ en función de la capa donde se ubica la neurona, si es o no de salida:

$$\Delta_p w_{ij} = \eta \delta_{pj} x_{pi}$$

con

$$\delta_{pj} = (d_{pj} - y_{pj}) f'_j(z_{pj})$$

o

$$\delta_{pj} = f'_j(z_{pj}) \sum_k \delta_{pk} w_{kj}$$

Si la función de activación es sigmoide, mostramos la función y su derivada:

$$y_{pj} = \frac{1}{1 + e^{z_{pj}}} \Rightarrow f'_j(z_{pj}) = y_{pj}\left(1 - y_{pj}\right) \tag{91}$$

los valores de $\delta_{pj}$ ahora:

$= (d_{pj} - y_{pj}) y_{pj}(1 - y_{pj})$ si j es de Salida (92)

$\delta_{pj} = y_{pj}(1 - y_{pj}) \sum_k \delta_{pk} w_{kj}$ si no lo es (93)

en la expresión anterior, para las neuronas ocultas, las $\delta_{pk}$ se refieren a las calculadas en la iteración anterior y corresponden a la capa siguiente.

Además se suele utilizar un término de inercia llamado momentun para ajustar los pesos que hace que el algoritmo converja con mayor rapidez. Ajusta la iteración $n+1$ teniendo en cuenta el ajuste de la iteración anterior. Obvia los saltos bruscos en las direcciones de optimización que marca el gradiente

La ecuación que marca el ajuste de los pesos es la siguiente:

$$\Delta_p w_{ij}(n+1) = \eta \delta_{pj} x_{pi} + \Delta_p w_{ij}(n) \tag{94}$$

la superficie del error para una red multicapa no es una función cuadrática, presenta diferencias para diferentes zonas del espacio, la velocidad de convergencia puede incrementarse debido a la distintas tasas de aprendizaje en cada parte de la superficie del error, sin sobrepasar el valor máximo para aprendizaje estable que se ha definido

anteriormente. El elegir un incremento adecuado desde luego tiene gran influencia en la velocidad de convergencia del algoritmo, esta velocidad se controla mediante la tasa de aprendizaje, la que por lo general se escoge como un número pequeño, para asegurar que la red pueda encontrar una solución óptima. Se recomienda aumentar el valor a medida que disminuye el error de la red para converger más rápidamente con la precaución de no elegir valores muy grandes que hagan que la red oscile y se aleje del valor mínimo.

De todas formas **en el algoritmo Backpropagation no está asegurado que** el mínimo global sea el que se encuentre pero en cualquier caso si el error es admisible, esto no resulta demasiado importante.

Se puede **acelerar el aprendizaje** por medio de otra técnicas basadas en la función de activación por ejemplo Algoritmo alfa-LMS ; Algoritmo de Mays ; Regla II y III del Madaline ; Algoritmo mu.LMS ; Cálculo del gradiente hacia delante y hacia atrás ; Retroprogación en el tiempo ; Algoritmo FullPropagation ; Regla Delta-Bar-Delta ; Algoritmo QuickPropagation ; Método de direcciones conjugadas ; Método de métrica variable quasi-Newton) ; Método de direcciones aleatorias. Método de Levemberg-Marquardt.

Como **deficiencias del algoritmo** destacamos: la lenta velocidad de convergencia, a menudo el gradiente descendente se queda **atrapado en un mínimo local, sobreentrenamiento** entendido como que aprende muy bien los ejemplos de entrenamiento pero la respuesta a ejemplo no entrenados no es buena, la saturación debida a que no se produce apenas modificación de los pesos y por tanto del error porque las salidas esperadas en cada neurona de salida es 0 o 1 y la **estabilidad o robustez** aplicando medidas de estabilidad estadística, que se pueden usar para seleccionar la RNA entrenada de mayor estabilidad y también medidas de entrenamiento de la red.

Por último apuntar que generalmente los datos se pre-procesan adaptando los reales al modelo y las entradas se normalizan incluso en casos innecesarios. Después se realiza un post-proceso a los resultados para poder validarlos y adecuar a la realidad del problema.

Este algoritmo y sus variantes se ha aplicado con éxito a multitud de aplicaciones.

- Codificación de patrones para clasificación
- Estructura secundaria de proteínas
- Corte adecuado de palabras con guiones
- Reconocimiento de tarjetas sonar
- Conducción de vehículos
- Compresión de datos o imágenes
- Juego del backgammon
- Reconocimiento de voz
- Tratamiento de señales digitales
- Predicción y previsión
- Reconocimiento de texto
- reconocimiento de sonido
- Comparación y ordenación de números difusos
- Segmentación de clientes
- Obtención de reglas difusas para describir sistemas difusos.
-

El algoritmo implementado en Matlab es el siguiente [192]:

```
%Algoritmo de Entrenamiento de BACKPROPAGATION para Redes Neuronales

function entrenamiento = backpropagation(X,D,L,rate,alfa,dmse)

%% 1. Inicializacin y Cálculo de Parametros

%Parmetros del algoritmo

    mse = Inf;              %Asumiendo Pesos Iniciales Malos

    epoch = 1;
```

```
%Se calculan los tamaños respectivos de las matrices
    [N,P] = size(X);            %Numero de Patrones P y Entradas N
    [Q,Pd] = size(D);           %Numero de Patrones P y Salidas Q
    depth = length(L);          %Profundidad de la red - nro capas
%Inicializar Matriz de Pesos para cada capa en el rango [-1,1]
W = cell(1,depth-1);            %Pre-alocacion de la matriz de pesos
for m = 1:depth-2
    W{1,m} = [-1+2*rand(L(1,m+1),L(1,m)+1)];
end
W{end} = [-1+2*rand(L(end),L(end-1)+1)];
%Inicializar Matrices de los delta-pesos de ajuste
dW = cell(1,depth-1);               %Pre-alocacion de los delta pesos
for m = 1:depth-1
    dW{m} = zeros(size(W{m}));
end
%Inicializar los campos locales inducidos 'v'
v = cell(1,depth);                  %Pre-alocacion de los campos locales
for i=1:depth-1;
    v{i} = [zeros(L(i),1); 1];
end
v{end} = zeros(L(end),1);
```

```
%Inicializar las salidas 'y' de cada capa
y = cell(1,depth-1);              %Pre-alocacion de las salidas locales
for i=1:depth-2;
    y{i} = zeros(L(i+1),1);
end
y{end} = zeros(L(end),1);
%% 2. Cálculo Forward y Backward para cada epoch
while (mse > dmse) && (epoch <= 10000)
    e = zeros(Q,P);
    err = zeros(1,P);
    for p = 1:P
        %Calculo Forward capa-por-capa para cada patrn p
        v{1}(1:end-1) = X(:,p);
        for i = 1:depth-1
            y{i} = W{i}*v{i};
            if i < depth-1
                v{i+1}(1:end-1) = tansig(y{i}); %Cálculo de la salida
            else
                v{i+1} = tansig(y{i});          %Cálculo de la salida
            end
        end
```

```
    %Cálculo de la seal de error
    e(:,p) = D(:,p)-v{end};
    %Cálculo de la energa del error
    if size(D,1) == 1
        err(1,p)=0.5*(e(:,p).^2);
    elseif size(D,1) > 1
        err(1,p)=0.5*sum(e(:,p).^2);
    end
    %Cálculo Backward capa-por-capa para cada patrn p
    delta = e(:,p).*(tansig('dn',y{end}));
    %Ajuste de pesos
    for i = depth-1:-1:1
        dW{i} = rate * delta * v{i}' + alfa.*dW{i};
        W{i} = W{i} + dW{i};
        if i > 1
            delta = tansig('dn',y{i-1}).*(delta'*W{i}(:,1:end-1))';
        end
    end
end
%Cálculo del mean square error
mse = (1/P)*sum(err);
```

```
        epoch = epoch +1;
        hold on
        figure(2)
        semilogx(epoch,mse,'ro')
        hold off
    end
    entrenamiento.pesos = W;
    entrenamiento.epocas = epoch;
    entrenamiento.estructura = L;
    entrenamiento.error = mse
```

Además podemos encontrar **trabajos sobre microcalcificaciones** utilizando Backpropagation entre otros:

"Detección de microcalcificaciones en imágenes mamográficas usando redes neuronales" de Aníbal Guerra, Joel Rivas. La RNA se implementó en lenguaje C++, con una arquitectura de una capa oculta y el algoritmo de aprendizaje Backpropagation, combinando con técnicas basadas en el análisis estadístico sobre la textura de imágenes. La base de datos para el desarrollo de la aplicación proviene de la base de datos de la Sociedad de Análisis de Imágenes Mamográficas (MIAS, en sus siglas en inglés) [189]

"Aprendizaje Multi-Tarea Selectivo utilizando Redes Neuronales Artificiales" de Marta Fernández Camba. [190]

"Construcción de una base de datos de imágenes de mamografía para la identificación de microcalcificaciones", Santamaría Moya, Claudia Joana [191]

### 2.3.2.3.4.2.3 Tercera Ola: Redes profundas, redes convolucionales. Pre-entrenamiento no supervisado. Dropout. Aplicaciones.

**Aprendizaje profundo** o en ingles Deep Learning,

**Un poco de Historia.** Un equipo de doctorandos del profesor Geoffrey E. Hinton de Canadá **se presentó en el último momento a un concurso** propuesto por la Farmacéutica Merck. El logro fue determinar con una red neuronal creada por ellos, basada en el aprendizaje profundo, que molécula sería la más efectiva como principio activo en un medicamento partiendo de la información de la estructura química de 15 moléculas y de una cantidad de datos muy pequeña, rompiendo así el esquema de la necesidad de cargar en el sistema una matriz enorme de información.

Por supuesto **ganaron** y el uso exitoso de esta metodología se convirtió en un nuevo y gran avance en el desarrollo de la inteligencia artificial. Hablamos del año 2012. Poco después **google compró** la compañía *start-up* cofundada por Geoffrey Hinton.

Los investigadores de Google Andrew Y. Ng y Jeff Dean realizaron un proyecto para crear una **simulación del cerebro humano** e intentaron emular su comportamiento de forma que fuera capaz de pensar y aprender por sí misma. Para entrenar el sistema de reconocimiento de imágenes, se sirvieron de 16.000 procesadores, se usó la **red ImageNet**, que contiene 14 millones de imágenes de 20 mil objetos diferentes y dotaron a la red de un algoritmo de aprendizaje que trataba de imitar al de una persona. **La máquina de Google aprendió, sin que nadie le diera indicaciones**, su primer concepto abstracto, un gato. Los investigadores de este proyecto opinan que aumentando la escala de computación en la nube que lo alimenta, estarían en condiciones de crear una **red neural a escala 'humana' hacia finales de esta década** [198].

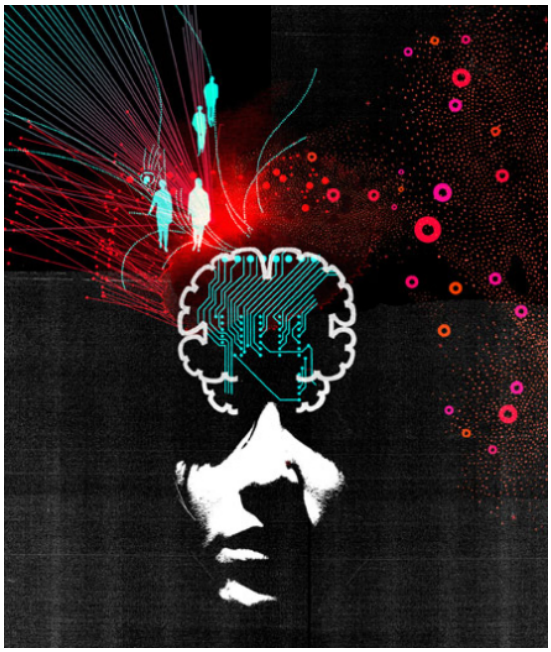


Figura 97 Dibujo creativo de red neuronal

El aprendizaje profundo se ha visto beneficiado por el método de **división de las tareas de computación** entre muchas máquinas. Dean ayudó a desarrollar esta tecnología a lo largo de su carrera de 14 años en Google.

**Google también ha utilizado** la tecnología para reducir la tasa de errores en el reconocimiento de voz en su último software Android para teléfonos móviles. El sistema puede reconocer fragmentos de sonido de forma más fiable, de manera especial en entornos ruidosos como puede ser una plataforma de metro.

En el 2012, científicos suizos crearon un sistema que reconocía con **mayor acierto que una persona las señales de tráfico** en imágenes con una precisión del 99.46 % en un conjunto de 50000 imágenes. y las personas obtuvieron el 99,22% de máxima y 98,84 de media en el grupo de 32 personas.

En octubre de 2012 Richard F. Rashid, coordinador de los programas científicos de Microsoft, **en una conferencia en Tianjin** (República Popular China) transcribió sus palabras habladas a texto en inglés con una tasa de error del 7 por ciento, después lo tradujo a texto en chino, y más tarde **no sin dejar de asombrar a los asistentes** .simuló su propia

voz para atención, pronunciar el texto en mandarín [200]. El sistema ha sido llamado "Deep Neural Net" (DNN).[199]

Según estima Dileep George, cofundador de la start-up de aprendizaje de máquinas Vicarious, como mínimo **el 80 por ciento de los avances recientes en IA** se pueden atribuir a la disponibilidad de mayor potencia informática [200]

En la teoría del aprendizaje profundo, **el profesor Hinton**, tataranieto de George Boole (padre del algebra booleana) es uno de los principales investigadores en esta área.

En casos como el ordenador **Watson de IBM**, que ha sido ganador del concurso americano de preguntas y repuestas Jeopardy!, que utiliza algunas técnicas de aprendizaje profundo y hoy en día se entrena para colaborar con los médicos en la toma de decisiones.

**Microsof**t, ha incorporado el aprendizaje profundo a su teléfono Windows y la búsqueda por voz de Bing, **Appl**e ha creado a Siri, el ayudante por voz, en base a la tecnología Nuance Communications y **Google** el reconocimiento de direcciones en el servicio de Visionado de calles.

La **ventaja** de Deep learnig es que necesita menos información que otros tipos menos evolucionados de RNA´s habituales porque estudia el objetivo mientras no encuentra suficiente información en los niveles de representación con información suficiente sobre el objeto estudiado. Es un tipo de red más precisa muy apropiada para buscar matices en las matrices de información multidimensional, bien estructuradas.

Debemos mencionar que las redes neuronales incluso las de tercera generación, como las redes convolucionales, los auto-codificadores y las máquinas de Boltzmann tienen **muy poco en común** con las neuronas biológicas, quizás el nombre.

Jeff Hawkins, fundador de Palm Computing, desarrolla en la empresa **Numenta** un sistema de aprendizaje automático que no utiliza el aprendizaje profundo aunque si está inspirado en la biología. Jeff es crítico con las expectativas creadas sobre aprendizaje profundo y la rivalidad con la inteligencia humana. Numenta ayuda a predecir los patrones en el consumo de energía y la probabilidad de que una máquina entre en fallo.

Hawkins, critica que el aprendizaje profundo no tiene en cuenta el concepto de tiempo , es el autor del libro *On Intelligence* en 2004.

Últimamente es un término de moda porque forma parte de un grupo de métodos de aprendizaje automático (Machine Learnig) utilizado para modelar abstracciones de alto nivel en datos así como reconocer patrones y para ello se sirve de arquitecturas compuestas de transformaciones no lineales múltiples. [194].

Aprende representaciones de datos y las observaciones se pueden ver representadas de varias maneras distintas pero algunas de estas representaciones pueden facilitar el aprendizaje de ciertas tareas interesantes como por ejemplo si una imagen es un perro o un gato o si lo que se observa es una botella o una cara humana etc.

**Para entender como funciona** de manera muy simple, podemos asimilarlo a un procesador de información, que recibe información entrante, codificada como números y produce como resultado información saliente, codificada como otros números.

Mediante un conjunto de ejemplos se investiga que tipo de representaciones son las apropiadas y como crear los modelos que aprendan estas representaciones.

Imaginemos una red neuronal que debe **aprender si una imagen mamográfica tiene lesiones** o no. Se podría codificar la imagen con una lista de números de tal forma que esta red imaginaria recibiría tantos números a su entrada como píxeles tiene la imagen (o tres por cada píxel si utilizamos imágenes en color). Como la información que esperamos a la salida es que nos diga si hay cáncer o no, bastaría con un solo número como resultado en la lista saliente.

Si toma un valor cercano a 1.0 significa que hay un cáncer, y si toma un valor cercano a 0.0 significa que no lo hay. Los valores intermedios se pueden interpretar como inseguridad, o probabilidad.

Existen **varias arquitecturas disponibles** de Deep Learnig pero las más evolucionadas son las redes neuronales y ya sabemos que forman parte de presente y del futuro. **Al efecto tenemos** las redes neuronales profundas, redes neuronales profundas convolucionales, y redes

de creencia profundas con resultados vanguardistas en campos como la visión por computador, reconocimiento automático del habla, reconocimiento de señales de audio y música, aplicaciones médicas etc.

Los algoritmos tienen las siguientes **características**:

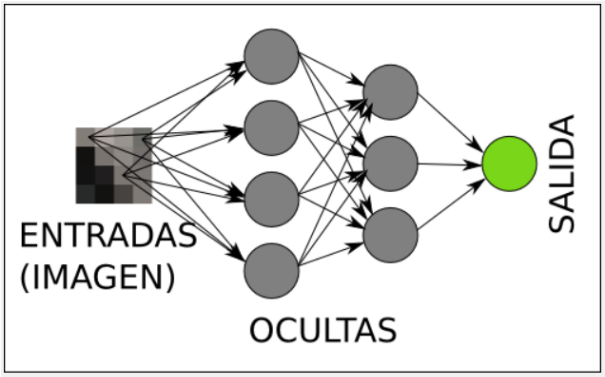
Pueden utilizar aprendizaje supervisado o aprendizaje no supervisado.

Usa capas en cascada de unidades de procesamiento no lineal para transformar variables y extraer variables. La entrada de una capa es la salida de la anterior

Se basan en el aprendizaje de múltiples niveles de características o representaciones de datos en representación jerárquica. Los niveles de representación se corresponden con distintos niveles de abstracción.

**Un algoritmo es profundo** según la mayoría de los investigadores cuando supone al menos dos transformaciones intermedias considerando que una transformación equivale a una capa.

Generalmente todas las neuronas de cada capa tienen una conexión con cada neurona de la siguiente capa, como se representa en el diagrama.




Arquitectura de la red de neuronas para detectar elementos en una imagen.

Figura 98 Arquitectura de la red de neuronas que detecta elementos de una imagen

Ya vimos anteriormente **cómo funcionan las redes neuronales**. cada neurona de la siguiente capa recibe valores de varias conexiones entrantes, y lo primero que hace es sumarlos todos. de una manera sencilla, almacenamos los pesos en matrices y si ahora guardamos los valores de todas las neuronas de una capa en un vector, el producto del vector y la matriz de pesos de salida, nos da los valores de entrada de cada neurona en la siguiente capa. Se Aplica

entonces la función de activación, por ejemplo la sigmoidal, a cada elemento de ese segundo vector, y se repite el proceso. Ya existen muchas librerías de código abierto y si disponemos una que implemente redes de neuronas, como **libfann**[195] se puede construir la red en unas pocas líneas.

Para interés del lector se listan las **características de esta librería** anunciadas es su página web:

Fast Artificial Neural Network Library ,es una biblioteca de redes neuronales, libre, de código abierto que implementa redes neuronales artificiales de múltiples capas en C, ejecución multiplataforma. Fácil manejo de conjuntos de datos de entrenamiento. Es fácil de usar, versátil, bien documentada y rápida. Tiene conectores para más de 20 lenguajes de programación. Se dispone de un artículo introductorio y un manual de referencia que acompaña a la biblioteca con ejemplos y recomendaciones sobre el uso de la misma.

- Dispone de varias interfaces gráficas de usuario.
- Biblioteca de red neuronal artificial de múltiples capas en C, formación de propagación hacia atrás (Rprop, QuickProp, por lotes, por pasos)
- Evolución dinámica de la topología, y forma a la ANN (Cascade2).
- Fácil de usar ,crear, entrenar y ejecutar una RNA, con sólo tres llamadas de función
- Rápida ejecución, hasta 150 veces más rápido que otras bibliotecas
- Versátil (es posible ajustar muchos parámetros y funciones sobre la marcha)
- Bien documentado, artículo de introducción, manual de referencia completo y un informe que describe las consideraciones de implementación
- Multiplataforma con script de configuración para Linux y UNIX, los archivos DLL para Windows, archivos de proyecto para MSVC ++ y además se facilitan los compiladores de Borland
- Varias funciones de activaciones diferentes implementadas incluyendo funciones lineales
- de fácil de guardado y carga de las RNA completas

- Varios ejemplos fáciles de usar
- Se puede utilizar tanto en coma fija y coma flotante
- Caché optimizado (para dar mayor velocidad)
- De código abierto aunque todavía se puede utilizar en aplicaciones comerciales (licenciado bajo LGPL)
- Entorno para un fácil manejo de conjuntos de datos de entrenamiento,
- Interfaces gráficas.
- Enlaces a diferentes lenguajes de programación
- Utilizado ampliamente.

**Otras arquitecturas de redes de neuronas** diferentes usadas en *Deep Learning* son las siguientes:

Las redes de neuronas **recurrentes** no se estructuran en capas propiamente dichas, sino que las conexiones entre todas las neuronas son arbitrarias e incluso se crean ciclos. Esto abre nuevos conceptos como el de **temporalidad**, y **memoria** en la red, porque datos que introducimos en un instante dado en las neuronas de entrada, son transformados y siguen circulando por la red incluso después de cambiar los números de entrada por otros totalmente distintos.

**Redes convolucionales**

Otra arquitectura muy importante e interesante es la concerniente a las **redes de neuronas convolutivas** (Convolutional neural networks). En este caso si se mantiene el concepto de capas, Tiene **capas convolucionales y de submuestreo alternas** y al final capas de conexión total como un perceptrón multicapa, pero las conexiones neuronales son selectivas es decir cada neurona de una capa determinada no recibe conexiones entrantes de todas las neuronas de la capa anterior, sino sólo de algunas. Esto favorece que existan neuronas especializadas en una región de la lista de números de la capa anterior, y **tiene la ventaja de que** se reduce fuertemente el número de pesos y de multiplicaciones necesarias. Lo normal es que dos

neuronas que se encuentran consecutivas en una capa intermedia se especialicen en zonas que se encuentren solapadas de la capa anterior. Una **convolución es un operador matemático** que transforma dos funciones $f$ y $g$ en otra función que podríamos decir, representa la magnitud en la que se superponen $f$ y una versión trasladada e invertida de $g$. Una convolución es un tipo muy general de media móvil, como se puede observar si una de las funciones se toma como la función característica de un intervalo [196].

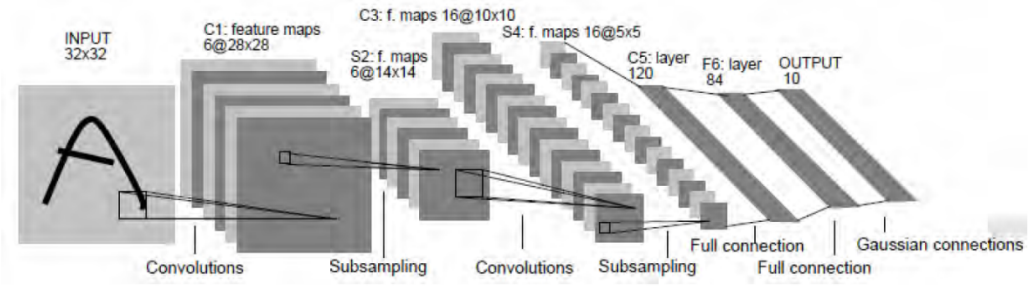



Figura 99 Arquitectura de una red convolucional [225]

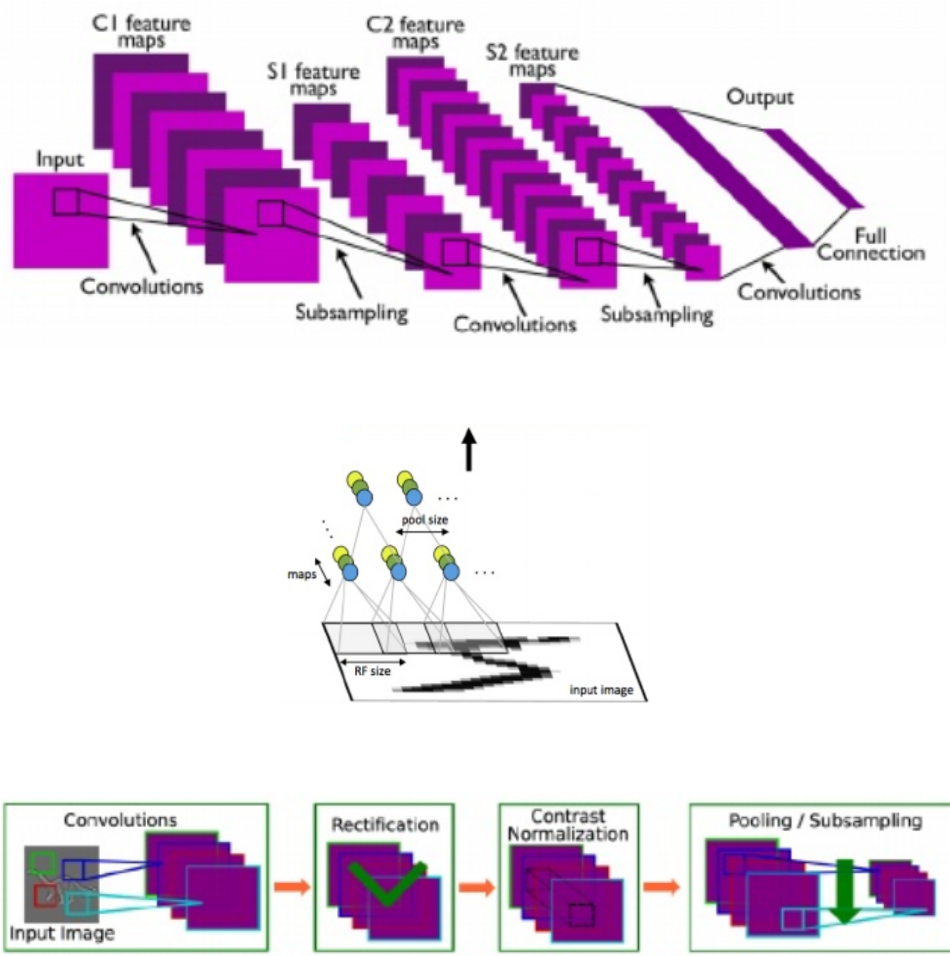



Figura 100 Arquitectura de una red convolucional [226]

La **entrada de una capa convolucional** es una imagen $m \times m \times r$ siendo $m$ el ancho y el alto y $r$ el número de canales, por ejemplo 3 en RGB. Las capas convolucionales contienen $k$ filtros $nxnxq$ definidos por el diseñador y son variables de filtro en filtro. Cada uno de estos filtros crea mediante convolución un mapa de rasgos (feature maps) cuyo tamaño es $m$-$n$-1. Posteriormente, cada mapa es submuestreada aplicando la **operación mean pooling o max pooling** sobre regiones anexas de tamaño $p\ x\ p$ con valores desde 2 para pequeñas imágenes (MNIST) hasta 5 para imágenes grandes.

**También se aplica** un función de activación sigmoidal que puede ser anterior o posterior al submuestreo y también un sesgo para cada mapa de rasgos.

Sigamos con el **ejemplo simplista de las microcalcificaciones**. Imaginemos que ya conocemos el número de neuronas de entrada y el de salida, además **hay que decidir** el número de capas ocultas, neuronas por capa oculta y los pesos concretos en las conexiones entre cada par de capas. El número de capas ocultas y las neuronas por capa se suelen elegir en función del problema de muchas variables ya explicadas pero la realidad es que es a **criterio del constructor de la red** y en ocasiones se fija a base de ensayo y error.

Como es lógico, cuantas más neuronas dispongamos en las capas ocultas, **más compleja es la red** y podrá resolver a su vez problemas más complejos pero tiene el inconveniente de que cuantas más neuronas ocultas, mayor coste computacional para realizar todos los productos y sumas.

Reseñar que si hemos elegido una función de activación lineal **no merece la pena utilizar** varias capas ocultas porque la potencia de la red será la misma . Solo aumenta con funciones de activación no lineales como la sigmoide.

En la práctica, se puede utilizar como en nuestro caso para el entrenamiento de una red de este tipo, un conjunto de ejemplos recopilados de imágenes mamográficas con lesiones y otro sin ellas. Las **etiquetaremos oportunamente** de manera que así tengamos controlado el resultado deseado a la salida. Ya sabemos por lo que hemos visto anteriormente a lo largo de este trabajo que hablamos de aprendizaje supervisado.

**Si usásemos el algoritmo backpropagation**, ya explicado anteriormente obtendríamos detalles del proceso en forma de error cuadrático ya que la red promedia el error al cuadrado que se produce en las neuronas de salida. **Los errores** siempre son positivos y no se anulan entre las distintas neuronas. Podemos limitar el error aceptable en cada neurona de salida antes de calcular cuadrados y media para decidir cuando tiene que parar el algoritmo su entrenamiento. El problema de usar este algoritmo es que el error se va diluyendo de forma exponencial a medida que atraviesa capas en su camino hasta el principio de la red; solo las capas finales se entrenan en redes profundas por lo que **si decidimos usarla es mejor** usar

pocas capas con muchas neuronas. Esto era así hasta que *Deep Learning se ha puesto de moda*.

Si hablamos de **Deep learning con aprendizaje no supervisado** podemos referirnos a la herramienta los **autoencoder, o autocodificador** que aprende a producir la misma información que recibe a la entrada a su salida. Se implementa generalmente con tres capas, una de ellas oculta que es la pieza fundamental ya que es la encargada de comprimir la información y de encontrar la representación intermedia.

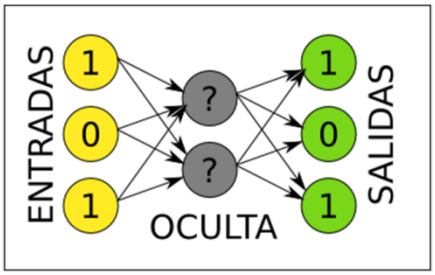



Figura 101 Autoencoder

una vez entrenada la red, podemos realizar una división para una primera red que utiliza la capa oculta como capa de salida, y una segunda le da utilidad a esta capa oculta como capa de entrada. La primera sería un compresor y la segunda red un descompresor. No es una utilidad muy frecuente en Deep Learnig.

¿Qué ocurriría si obligamos a que cada neurona de esa capa se especialice en un cierto patrón en la entrada?. Para ello tendríamos que construir una red con una capa intermedia que dispusiese de más neuronas que las capas de entrada y salida además forzarlas para que se activasen en pocas ocasiones consiguiendo con esto la red aprendería códigos dispersos y alternativos. A este algoritmo se le llama **Auto-codificadores dispersos**.

Al crear restricciones de dispersión, la capa oculta es incapaz de construir tantas representaciones de las distintas combinaciones de valores como lo podría haber hecho sin la restricción. Se le obliga a la red a generalizar y buscar patrones en los ejemplos que pueda

representar con las neuronas de la capa oculta. Estos patrones son las características frecuentes en los ejemplos.

Hay diversas formas de conseguir que la red no se limite a copiar la información de la capa de entrada a la de salida:

- Añadir, a la hora de calcular el error, un factor de dispersión
- introducir un poco de **ruido en el vector de entrada**, y dejar el de salida sin ruido. Además conseguimos ampliar el juego de ejemplos, muy conveniente en caso de tener poca cantidad de ellos.

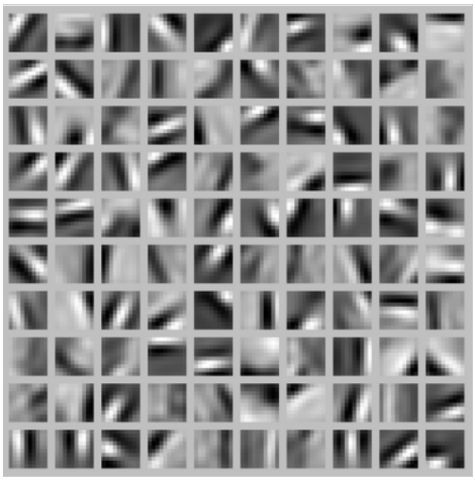


Figura 102 Unidades ocultas han aprendido a detectar bordes en diferentes posiciones

**Para entrenar un auto-codificador**, vamos a suponer que recortamos parches de 6×6 píxeles de una base de datos de mamografías, y configuremos el auto-codificador con 64 neuronas en su capa oculta. Enseñaremos a la red el mismo parche a la entrada y la salida, y la red aprenderá una codificación intermedia. Ahora, si dibujamos el parche que activa completamente cada una de las 64 neuronas de la capa oculta, obtendremos el siguiente gráfico (los 64 patrones se han distribuido en una matriz 8×8 por comodidad):

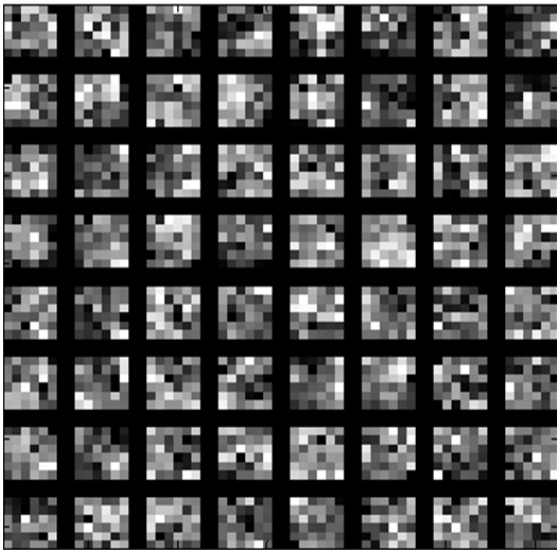


Figura 103 Gráfico de patrones de las microcalcificaciones realizado por neuronas de la capa oculta

Los parches que activan al máximo cada neurona de la capa oculta, representan características fundamentales de las microcalcificaciones.

Una vez que se ha entrenado un auto-codificador, la segunda mitad de la red se puede descartar, generalmente solo interesa la parte que codifica.

La realidad es que para el ejemplo anterior que es muy complejo harían falta varios **auto-codificadores apilados** es decir, usar varios codificadores, y entrenarlos uno a uno, usando el conocimiento de cada codificador entrenado para entrenar al siguiente. Se trata de un algoritmo Voraz. Para resolver el problema, sigue la heurística de elegir la opción óptima en cada paso local con la esperanza de llegar a una solución óptima general.

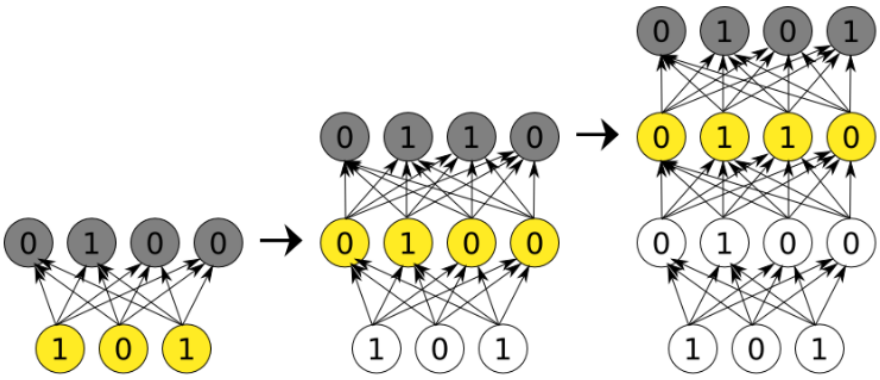



Figura 104 Auto codificadores apilados

**Una red profunda que se crea con autocodificadores apilados** se caracteriza porque aprende sin supervisión etiquetando los datos con las características que la red encuentra por sí misma y por lo tanto permite entrenar redes con el nivel de profundidad que se requiera sin problemas ni restricciones de entrenamiento en las primeras capas.

En el anterior caso de ejemplo que planteábamos sobre las microcalcificaciones, usamos parches de $6x6$ pixel para entrenamiento pero ¿que ocurriría si se introducen imágenes completas?

Si la entrada del auto-codificador es grande, hay que dedicar muchos recursos a entrenar con lo cual el proceso se hace muy costoso mientras que si usamos imágenes completas una imagen de una microcalcificación desplazada y otra igual que estuviera centrada serían distintas para el autocodificador.

Para resolver el punto anterior, usamos la **técnica de la convolución** que se basa en la estructura neuronal del sistema visual humano. La convolución es una operación de productos y sumas entre la imagen de entrada y un filtro o kernel que genera un mapa de rasgos. La extracción de estos rasgos se corresponde con todas las ubicaciones posibles del filtro o neurona en la imagen original. El filtro extrae el mismo rasgo en cualquier ubicación de la imagen por lo que el ahorro en conexiones y entrenamiento es considerable en comparación con redes multicapa de conexión total.

En una imagen de $m\,x\,n$ con $k$ filtros de $a\,x\,b$ en una capa convolucional, se generan K mapas de rasgos que forman en total una matriz de $k\,x\,(n-a+1)\,x\;\;(m-b+1)$ rasgos. A su vez cada rasgo está formado por $f=\sigma(Wx+b)$.

**La convolución es completa** si se extiende la imagen para ubicar el filtro en todos sus pixeles y **es válida** si se convoluciona exclusivamente donde cabe el filtro dentro de la imagen.

Podemos entrenar al autocodificador para que funcione de forma parecida a un escáner que busque las características. Se realiza una transformación de la matriz de pixeles en otra de características creadas por el codificador. Ahora es mayor la información que contiene el pixel sobre la región.

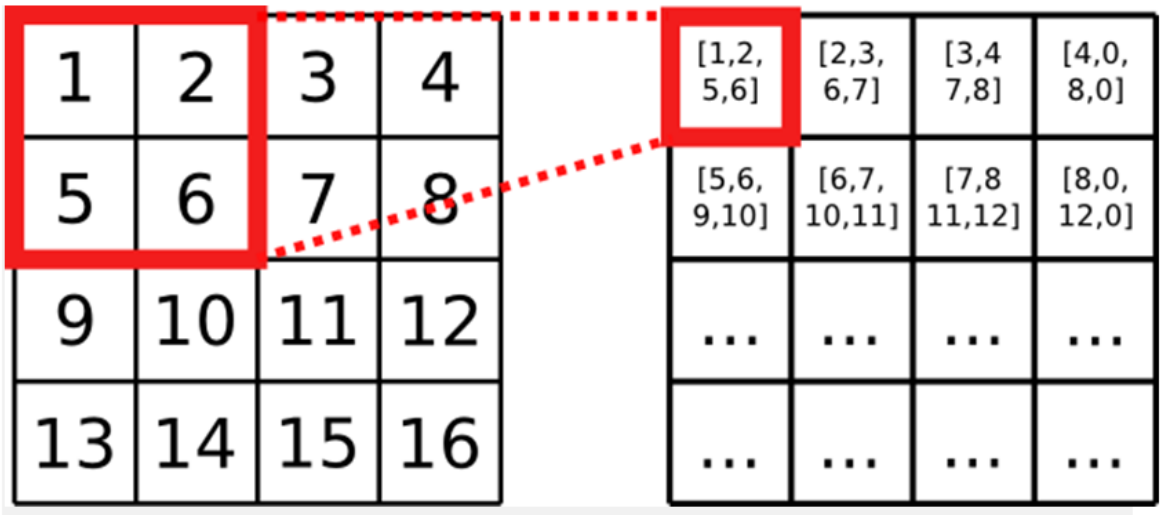



Figura 105 Convolución de una ventana de 2x2

Después de la convolución aplicamos **Pooling.** Esta técnica agrupa las listas de números de las características de coordenadas contiguas de la imagen con funciones como la media o el máximo. Esto reduce el número de rasgos usados para la clasificación y reduce el problema del sobreajuste. En esta etapa se reduce la resolución de la imagen. Varias capas de agrupamiento hacen que se pierda la información de la localización del rasgo, provocando que por ejemplo se puedan reconocer rostros, pero no a quienes pertenecen. Podemos continuar las veces necesarias aplicando nuevas convoluciones y poolings hasta que se obtenga la imagen resultante de $1x1$ pixeles con abundante información. Además se pueden introducir auto-codificadores intermedios para encontrar características de alto nivel.

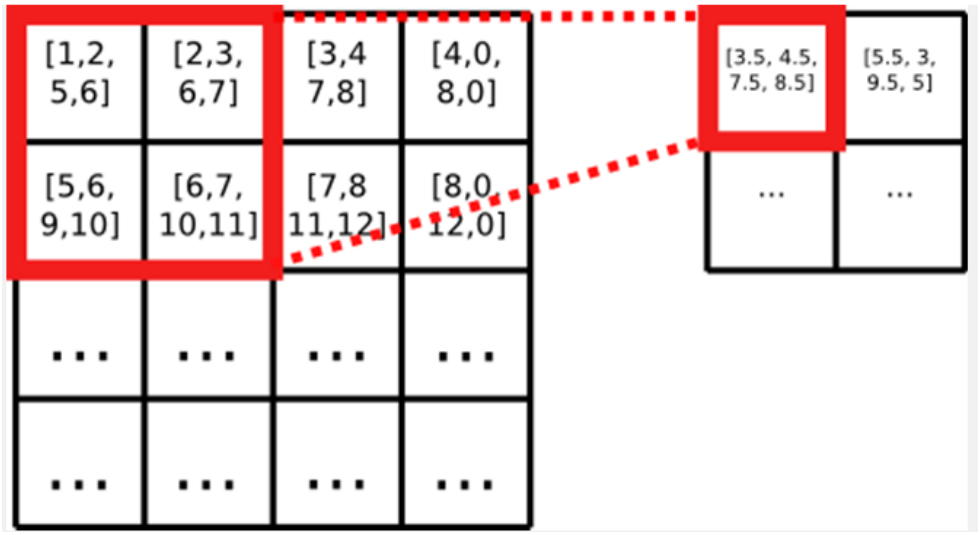



Figura 106 Pooling de una ventana de 2x2 con cálculo del promedio

**Otra técnica distinta es** entrenar sin supervisión una pila de autocodificadores y seguidamente componer los codificadores y seguir entrenando con supervisión. Se entrena comenzando con pesos útiles en lugar de pesos al azar sobre todo para las capas que ocupan los primeros lugares.

Después de la convolución se aplican a los mapas de rasgos con las **funciones de activación de tipo no lineal** como [226]:

- Sigmoide logística , de poco uso actual
- Sigmoide hiperbólica, centrada en cero, da mejores resultados que la sigmoide logística pero presenta el problema de la saturación y desvanecimiento del gradiente.
- Sigmoide rectificada, se usa en el reconocimiento de imágenes naturales.
- **Relu** (rectified linear unit): es la que vamos a usar en nuestro programa $f(x) = \max(0, x)$ muy popular en redes profundas ya que acelera el entrenamiento en un factor de 6 [227] comparado con las funciones sigmoide /Tanh y evita el problema del **desvanecimiento del gradiente** si se da el caso de que la red esté formada por muchas capas. Esto se debe a que es lineal y no hay saturación en el sentido positivo de su dominio. Es posible que debido a esto **muchas neuronas "mueran"** porque no se van a activar nunca para los ejemplos de entrenamiento debido al movimiento de

los pesos por el gradiente. El gradiente siempre será cero. Con aprendizaje alto pueden morir hasta el 40% de las neuronas. Podemos evitarlo ajustando la tasa de aprendizaje o usando variaciones de la función.

- **Función Softplus :** es una versión aproximada y continua de ReLU

$$f(x) = \ln(1 + e^x) \tag{95}$$

- **Leaky** permite tener un gradiente pequeño (no cero) cuando la neurona no se activa, se usa en reconocimiento de voz.

$$f(x) = \begin{cases} x, & si\ x > 0 \\ 0{,}01x & otro\ caso \end{cases} \tag{96}$$

- **Noisy ReLUs** usadas en máquinas de Boltzmann restringidas

$$f(x) = \max(0, x + N(0, \delta(x))) \tag{97}$$

- Neurona maxout Es una generalización de las funciones ReLU y Leaky ReLU con la ventaja de que evita la muerte de las neuronas.

En el ejemplo de las microcalcificaciones, podríamos utilizar las imágenes que de una BBDD para entrenar la pila de auto-codificadores y seguidamente utilizar unas pocas que de forma manual tengamos etiquetadas para afinar la red de manera que se fije sencillamente en si hay o no microcalcificaciones.

Además de los autocodificadores existen las **Deep Belief Networks**. Las capas se entrenan desde la más específica a la más genérica y de una en una con una Restricted Boltzmann Machine

**Aplicaciones de redes convolucionales.**

Aplicaciones comerciales [226]:

- LeNet: Lectura de cheques automático usado en U.SA. y Europa en los 90s en un DSP por AT&T (video http://ow.ly/SCpir )

- Reconocimiento de números y letras manuscritas incluyendo caracteres arábigos y chinos por Microsoft (http://ow.ly/SCtEX, http://ow.ly/SCri0)
- Detección y borrado automático de rostros y placas de vehiculos para protección de la privacidad en Google Street View.
- Identificación del género y la edad de los usuarios de máquinas expendedoras por la empresa NEC Labs en Japón (http://ow.ly/SCKhe)
- Detección y seguimiento de clientes en supermercados por NEC Labs, video: (http://ow.ly/SFqFs)
- Detección de intrusos y ladrones por la empresa AgilityVideo (http://www.agilityvideo.com/)
- Aplicaciones experimentales: Detección de rostros con record en velocidad y precisión usando GPU
- Identificación de expresiones faciales (video http://ow.ly/SCMyB)
- Evasión de obstáculos para robots móviles usando visión (video http://ow.ly/SCN0S)
- Detección de obstáculos a gran distancia.
- Segmentación de imágenes biológicas (video http://ow.ly/SFqxg)
- Restauración de imágenes (eliminación del ruido)
- Reconstrucción de circuitos neuronales a partir de imágenes transversales del cerebro con espesor nanométrico.
- Y cuando combinan las redes neuronales con algoritmos evolutivos, una computadora puede aprender a jugar Mario Bros (video http://ow.ly/)
- Robot que aprende a hacer tareas (video http://ow.ly/SP3y2)

### Dropout

Esta técnica fue presentada por Srivastava et al en 2014 [230]. Las redes neuronales profundas con un gran número de parámetros son potentes sistemas de aprendizaje pero padecen overfitting, que es un grave problema que se da en este tipo de redes consistente en la obteneción de un error de entrenamiento bajo y un error de generalización alto

comprendido como un **sobreaprendizaje** de una región particular del espacio de parámetros, especializándose solamente en una región del mismo.

Las grandes redes también tienen en su contra que son de uso lento, por lo que es difícil tratar el sobreajuste. **La idea clave es** dejar caer al azar algunas unidades (junto con sus conexiones) de la red neuronal durante el entrenamiento. Sólo se actualizan los pesos de las que quedan activas. Esto reduce significativamente el sobreajuste y da importantes mejoras con respecto a otros métodos de regularización y evita el pre-entrenamiento de las capas superiores que tiene como meta restringir a las capas inferiores poder capturar patrones y regularidades de la distribución de los patrones de entrada.

Se demuestra que **Dropout mejora el rendimiento** de las redes neuronales en las tareas de aprendizaje supervisado, en la visión, reconocimiento de voz, la clasificación de documentos, la biología computacional, la obtención de resultados del estado de la técnica en muchos conjuntos de datos de referencia.

## 2.4 Bases de datos.

Como he explicado anteriormente, vamos a analizar **las más importantes** BBDD de mamografías públicas o privadas: Digital Database for Screening Mammography, IRMA, AMDI o B-SCREEN, Nijmegen Digital Mammogram Database, Mammographic Image Analysis Society, Magic-5, MiniMIAS, Lawrence Livermore National Laboratories Database, MammoGrid.

De entre las BBDD de mamografías que existen, nos vamos interesar sobre todo en las que son de **libre acceso** como son Magic-5, miniMIas o DSSM y que nos trasladan toda la información necesaria sobre los pacientes de manera que podamos fabricar el conjunto de entrenamiento con los datos controlados y verificados por expertos radiólogos. Es muy importante para el buen desarrollo de un CAD **la elección de la BBDD** y para ello tendremos en cuenta además de la popularidad debida a la frecuencia de uso o de aparición

en los distintos trabajos realizados por otros investigadores, cuyos trabajos he consultado, los siguientes factores:

El número de casos ofrecidos por la base de datos debe tener un tamaño o número de casos adecuado ya que hay que tener en cuenta que debemos disponer suficientes ejemplos para el aprendizaje además de casos no vistos anteriormente para testear el CAD.

Además de ello, el número de casos debe contener proporciones adecuadas de casos normales, casos benignos y casos malignos.

Debe incluir lógicamente casos de microcalcificaciones pero además de otros tipos de casos para poder usarse como validación final.

Es importante que otros investigadores nos aporten referencias de aplicación a los sistemas CAD para que podamos comparar y comprobar si va a ser útil para nuestro trabajo.

Hay que tener en cuenta además los aspectos técnicos como la normalización, el formato de imagen, tamaño en memoria de la misma, resolución y niveles de grises porque suelen tener requisitos importantes de memoria dedicada y procesador.

La descripción de las imágenes está estandarizada y se basa fundamentalmente en los descriptores y categorías BI-RADS de las microcalcificaciones ampliamente comentadas en este trabajo y cuyo fin, recordemos, es unificar la manera de presentar el análisis de las mamografías. Este estándar contempla siete categorías de evaluación radiológica asignándoseles una numeración que va desde el cero al seis y cuanto mayor sea, más seguridad en la presencia de lesiones de carácter maligno. La densidad de la mama nos indica la cantidad de tejido fibroso y glandular y está representada en otro cuatro niveles aportan información añadida que proporciona información adicional a la fiabilidad del diagnóstico,

### 2.4.1 Magic-5

Se ha utilizado con éxito para el desarrollo de aplicaciones mamográficas junto con CAD específicos desarrollados con este fin. Disponible con un servidor central donde se ejecutan lo servicios comunes y clientes que se conectan a este servidor. También se ejecutan **aplicaciones de distinta naturaleza** como las destinadas a la detección de nódulos malignos en las TC de pulmón, análisis PET con miras a principios de la enfermedad de Alzheimer. Las imágenes están recogidas **desde 1999** por una comunidad de físicos en colaboración con radiólogos en hospitales Italianos Contiene **3369 mamografías** con datos de **967 pacientes** clasificadas por lesión y morfología, tipo de tejido mamario y el tipo de patología. Las vistas tomadas en las imágenes son **cráneo caudal o medio lateral** con un tamaño de **2067 x 2667** pixeles y resolución de 85 $\mu$/píxel.

Se ha desarrollado una **interfaz gráfica** de alta resolución para visualizar y procesar las radiografías como apoyo al diagnóstico médico. Su uso es óptimo para **entorno distribuido**. [201]

### 2.4.2 Mini-MIAS (Mini Mammographic Image Analysis Society)

**Promovida por** grupos de investigación organizados del Reino Unido interesados en la comprensión de las mamografías. Muy referenciada por artículos de investigación.

Las Imágenes proceden del programa nacional de **cribado mamográfico** digitalizadas con **200µ** en vez de las 50 µ por pixel de la MIAS y un dispositivo lineal en el rango de densidad óptica 0-3.2 y representando cada píxel con una palabra de **8 bits**. Las imágenes son de **1024x1024** Consta de 322 compuesta de **322 mamografías** sobre 161 pacientes con 204 casos normales, 25 con microcalcificaciones (12 benignas y 13 malignas), 21 masas espiculadas, 20 masas circunscritas, 15 masas poco definidas, 20 distorsiones arquitecturales y 17 asimetrías.[208].

Se han **digitalizado** a partir de imágenes analógicas **Disponible en** 8mm y tamaño de 2,3 GB. BBDD **gratuita** para la investigación a través (PEIPA) en la Universidad de Essex; únicamente se exige cumplir el contrato de licencia. Contiene las indicaciones del radiólogo sobre las ubicaciones de anormalidades presentes.

Las Imágenes se han **clasificado** teniendo en cuenta el grado de malignidad y fiabilidad del diagnóstico y su contenido por lesiones en microcalcificaciones y masas tumorales.

Proporciona los siguientes **detalles** [202]:

Primera columna: MIAS número de referencia de la base de datos.

Segunda columna: Carácter de tejido de fondo:

- F grasos
- G-glandulares grasos
- D-glandular denso

Tercera columna: Clase de anormalidad presente:

- masas SPIC espiculadas
- MISC Otros, masas mal definidas
- ARCH distorsión de la arquitectura
- La asimetría ASINTOM
- NORMA normal
- La calcificación CALC
- CIRC / masas circunscritas bien definidos

Cuarta columna: La severidad de la anomalía;

- B benigna
- M maligno

Quinta y sexta columnas:

- x, y coordenadas de imagen de centro de anormalidad.

Séptima columna:

- radio aproximado (en píxeles) de un círculo que encierra la anormalidad.

También hay que tener en cuenta que la lista está ordenada por pares imágenes, donde cada par está representado a la izquierda (números pares de nombre de archivo) y el derecho por mamografías (números impares nombre de archivo) de un solo paciente.

Las imágenes se han centrado en la matriz. Cuando están presentes las calcificaciones, las ubicaciones de los centros y de los radios se aplican a grupos en lugar de calcificaciones individuales. El origen de coordenadas del sistema está situado en la esquina inferior izquierda.

En algunos casos las calcificaciones están ampliamente distribuidas en toda la imagen en lugar de concentrarse en un solo sitio. En estos casos el centro ubicaciones y radios son inadecuadas y se han omitido.

### 2.4.3 DDSM (Digital Database For Screening Mammography)

El proyecto DDSM [203] ha sido un **esfuerzo de colaboración** entre las entidades siguientes: Hospital General de Massachusetts (D. Kopans, R. Moore), la Universidad del Sur de Florida (K. Bowyer), y los Laboratorios Nacionales Sandia (P. Kegelmeyer). Se suman casos adicionales de la Facultad de Medicina de la Universidad de Washington proporcionados por Peter E. Shile, Profesor Asistente de Radiología y Medicina Interna. Otras Instituciones colaboradoras que aportaron fueron Wake Forest School de la Universidad de Medicina (Departamentos de Ingeniería Médica y Radiología), Hospital del Sagrado Corazón e ISMD [205].

El proyecto DDSM. Es **mantenido por la Universidad del Sur de Florida** con el propósito de mantenerlo accesible en la web. La funcionalidad adicional para DDSM se describe en las siguientes publicaciones:

"Web services for the DDSM and digital mammography research, Chris Rose, Daniele Turi, Alan Williams, Katy Wolstencroft and Chris Taylor, IWDM 2006".

Es una de las **BBDD más extendidas** en su uso por la comunidad científica y también más interesante para uso en sistemas CAD por ser la más completa. Contiene miles de casos. Para cada uno de estos, clasificados según sistema BI-Rads y tiene la vista cráneo caudal y medio lateral de cada pecho y también información personal relativa a cada paciente edad, sexo y raza y la relativa al análisis propio de las imágenes como son el grado de fiabilidad y el diagnostico.

La base de datos digital para las mamografías exploratorias **se organiza en "casos" y "volumen"**. Un "caso" es una colección de imágenes y contiene la información correspondiente a un examen mamográfico de una paciente. Un "volumen" es simplemente una colección de casos recogidos conjuntamente a los efectos de facilitar la distribución. Todos los volúmenes están disponibles en cinta de 8 mm; hay disponibles 43 volúmenes.

Las imágenes contenidas de cada estudio **incluyen dos imágenes de cada seno**, junto con algo de información del paciente asociado y la información de la imagen. Las imágenes que contienen áreas sospechosas contienen información sobre las ubicaciones a nivel de pixel y tipos de regiones sospechosas.

Los **casos normales** se forman a partir de un examen de detección previa normal y se tiene en cuenta que al menos cuatro años más tarde el paciente sigue dando normalidad en su examen. Un examen de detección normal es aquel en el que no se requería más "trabajo de campo".

Los **casos de cáncer** se forman a partir de pruebas de detección en los que se encontró al menos una patología del cáncer probado. Los **casos benignos** se forman a partir de pruebas de detección en los que se encontró algo sospechoso, pero no maligno (por la patología, ecografía o algún otro medio).

El término benigno se utiliza para identificar los casos los que no se realizó ninguna prueba o biopsia adicional. Estos casos, sin embargo, contenían algo lo suficientemente interesante como para que el radiólogo lo marcara. Un pequeño número de casos de cáncer puede contener, además una o más regiones probadas de lesiones malignas y también una o más regiones, que se han podido probar. Estas son las regiones sospechosas para los que no hay resultados. Hay información sobre el léxico BI-RADS utilizado para codificar las descripciones de los archivos.

**Se proporciona el software** tanto para acceder a las imágenes de mamografía como para el cálculo del rendimiento de los algoritmos para análisis de imagen automatizado. El formato **".ljpeg"** puede leerse mediante software de la USF, que funciona en entorno Unix. Para visualizarlos en entorno PC hay que convertir el formato a otro más estándar.

**Las imágenes DDSM son digitales**, no son analógicas convertidas a digital.

**Se actualiza frecuentemente la BBDD** . Para una mejor organización se divide en cuatro tipos de volúmenes[204]:

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

El contenido de **cada caso se organiza en un directorio que contiene** los siguientes archivos:

Archivo ".ics". contiene la información adjunta de 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 cuatro imágenes del caso se almacenan con un **formato sin pérdidas** jpeg (lossless jpeg).

**Archivos ".overlay".** Cada caso contiene de uno a cuatro archivos ".overlay". Los ficheros contienen la siguiente información correspondiente a una imagen:

- El número de anormalidades presentes.
- El tipo de lesión según el BI-RADS
- La categoría de la lesión, de 1 a 5, según BI-RADS

  Clasificación de BI-RADS

  1. BI-RADS 1: examen normal.
  2. BI-RADS 2: hallazgos benignos.
  3. BI-RADS 3: hallazgos que muy probablemente corresponden a algo benigno, pero que requieren control en un intervalo corto, a lo seis meses.
  4. BI-RADS 4: hallazgos que hacen sospechar al radiólogo la posibilidad de un cáncer, por lo cual es necesario una biopsia.
  5. BI-RADS 5: hallazgo altamente sugerente de cáncer. Biopsia indispensable.

6. BI-RADS 0: estudio incompleto, son necesarias proyecciones adicionales y/o ecotomografía.

- 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.

Los casos normales no tienen fichero ".overlay".

**Archivos ".16_PGM"**. Son cuatro imágenes que se concatenan a baja resolución, almacenados en PGM (portable gray map) a 16 bits. Permite una visión rápida de las imágenes del caso.

**Características de las imágenes**: 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.

Esta base de datos **intenta emular una casuística completa de Screening de mama** y se puede aplicar perfectamente a estudios detección y aprendizaje.

Por último, si se utiliza esta BBDD en trabajo de investigación **hay que referenciar a**:

The Digital Database for Screening Mammography, Michael Heath, Kevin Bowyer, Daniel Kopans, Richard Moore and W. Philip Kegelmeyer, in Proceedings of the Fifth International Workshop on Digital Mammography, M.J. Yaffe, ed., 212-218, Medical Physics Publishing, 2001. ISBN 1-930524-00-5.

- Current status of the Digital Database for Screening Mammography, Michael Heath, Kevin Bowyer, Daniel Kopans, W. Philip Kegelmeyer, Richard Moore, Kyong Chang, and S. Munish Kumaran, in Digital Mammography, 457-460, Kluwer Academic Publishers, 1998; Proceedings of the Fourth International Workshop on Digital Mammography.

Y enviar una copia de la publicación al Professor Kevin Bowyer / Computer Science and Engineering / University of Notre Dame / Notre Dame, Indiana 46530.

A continuación se muestra la página Web en la que se observa la versión en miniatura de las imágenes del caso 1252 de la base de datos DDSM. El límite de la región en cada imagen está marcado por el marcado más grande y el núcleo de la masa estrellada está marcado por la región más pequeña dentro de los límites. La anormalidad es una masa estrellada maligna con calcificaciones.

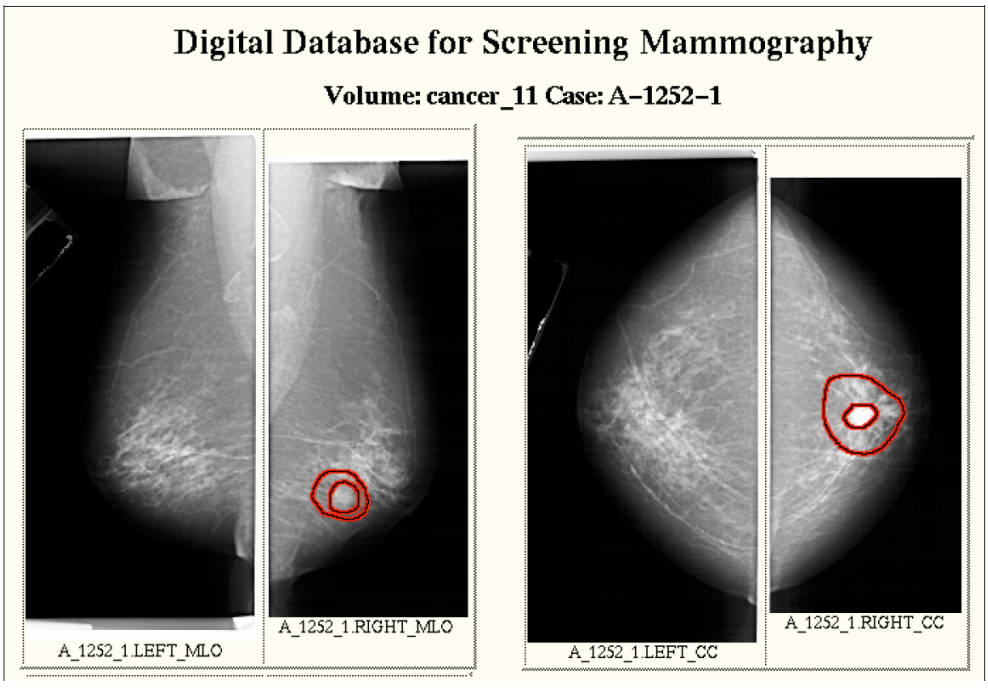



Figura 107 Ejemplo del caso 1252 de la BBDD DDSM. Imagenes

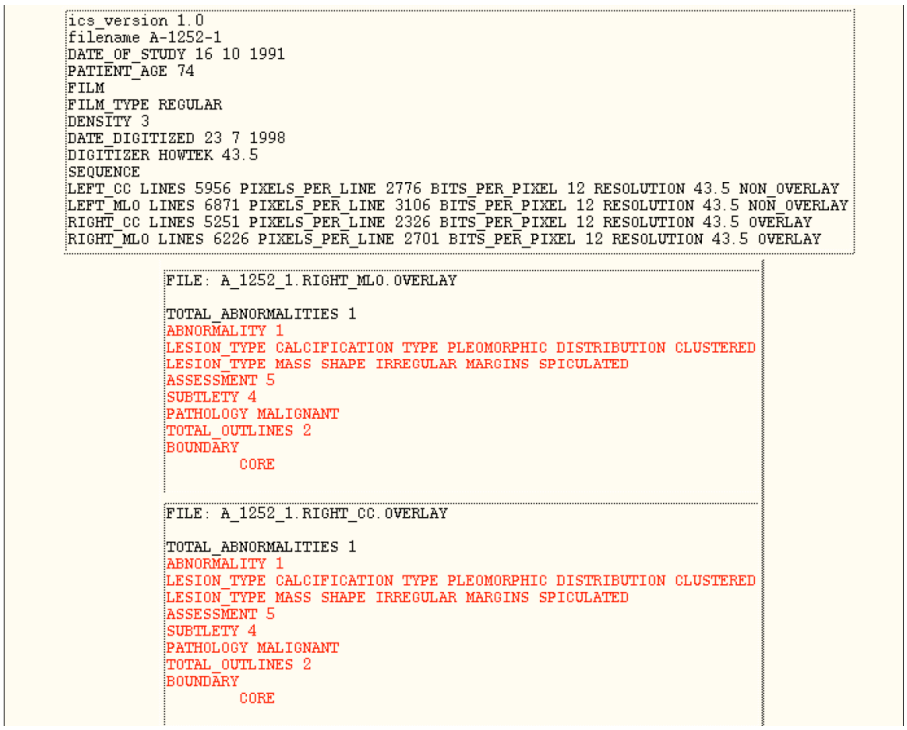



Figura 108 Ejemplo del caso 1252 de la BBDD DDSM. Información contextual

### 2.4.4 NDB (Nijmegen Digital Mammogram Database)

La base de datos de Nijmegen **se retiró del servidor, el 31 de marzo de 2000**. Formada por 40 mamogramas tomadas de 21 pacientes obtenidas del Centro Nacional de Formación de Expertos y de cribado mamográfico y el departamento de radiología de la Universidad de Nijmegen en Holanda. Los mamogramas se recogen con **película Kodak** MIN-R/S0177 y se digitalizan con una **cámara Eikonix 1412 12bits** con apertura de **50$\mu$** y resolución de **100 $\mu$** por bits. Cada Imagen tiene un tamaño de **2048 × 2048**.[206][207]. Es un referente por ser una de las primeras BBDD disponibles

### 2.4.5 LLNL (Lawrence Livermore National Laboratories Database)

Universidad de California en San Francisco (LLNL/UCSF). La BBDD solo contiene **casos de hallazgo sobre microcalcificaciones**. Contiene **198 imágenes** de 50 pacientes, con dos proyecciones de cada mama (OML y CC), excepto en un caso de una mujer mastectomizada.

**Contenido** de cada caso:

Archivo ".ics".

Cada una de las imágenes posee dos archivos:

- Extensión ".ics", que contiene descripciones como el tamaño de la imagen e información relativa al caso.
- Extensión ".ids", que contiene los datos de la imagen que se puede abrir con cualquier programa que lea archivos raw.

La nomenclatura de los archivos consta de dos letras que indican el **código del paciente**, seguido de "r" o "l" según se trate de la mama derecha o izquierda y continua con dos letras más que indican la proyección "ml" mediolateral y "cc" craneocaudal

Archivos de las **áreas sospechosas**.

Radiólogos expertos identifican **192 imágenes con áreas sospechosas**. Estas áreas son determinadas El proceso es el siguiente: Las mamografías se digitalizan y se visualizan en la pantalla del ordenador. Con el conocimiento de la imagen original y la información del caso basándose en la historia clínica, diagnóstico, resultado de la biopsia etc. Los expertos marcan las calcificaciones incluidas las que son **benignas confirmadas** tras varios años de seguimiento. Señalan la zona perimetral de la calcificación y rellenan el área de esta. Si hay agrupaciones de calcificaciones, bien sean benignas o malignas, señalan la extensión de la región que ocupan estas. Los **resultados finales** en la imagen digital archivada mediante dos archivos: el primero con las calcificaciones individuales, tanto si forman parte o no de una agrupación y en el otro archivo contiene el área solida demarcada como región de la agrupación.

**Se compone de** 5 casos normales, 5 casos también normales pero añadiendo alguna particularidad que dificulta el diagnóstico, 20 casos evidentes de microcalcificaciones que son

benignas con varios años de seguimiento, 12 casos que no parecen evidentes sobre microcalcificaciones benignas con seguimiento y 8 casos de microcalcificaciones malignas fehacientemente comprobadas

Las imágenes se digitalizan con un escáner láser, con una **resolución de** 35 micras y 12 bits de profundidad de niveles de gris.

La base de datos completa se forma por 12 volúmenes en CD-ROM.

### 2.4.6 IRMA (Image Retrieval in Medical Applications)

IRMA (recuperación de imágenes en aplicaciones médicas) [210] es un **proyecto de cooperación** de los departamentos de Radiología Diagnóstica, el Departamento de Informática Médica, División de Procesamiento de Imágenes Médicas y de la Cátedra de Informática en la Universidad Técnica de Aquisgrán (RWTH Aachen). EL Objetivo del proyecto es el desarrollo y aplicación de métodos de alto nivel para la consulta de imágenes mediante ejemplo prototípico con la aplicación de diagnóstico médico-tareas en un archivo de imágenes radiológicas.

El mantenimiento de este proyecto, se desarrolla en el Instituto de Informática Médica con el registro y evaluación de la información de contenidos geométricos. También se desarrollan los métodos de segmentación en imágenes de múltiples escalas.

En el Departamento de Radiología Diagnóstica, se lleva a cabo la clasificación con **análisis de textura**, así como la **segmentación** del modelo y se basa en el conocimiento de las radiografías. Los radiólogos evalúan los resultados con el fin de mejorar la **fiabilidad** de las consultas y la usabilidad de la interfaz gráfica de usuario (GUI). También elaboran series de datos de entrenamiento para la evaluación a gran escala.

En la Cátedra de Ciencias de la Computación, se desarrollan y se mejoran los algoritmos para el análisis de agrupamiento y clasificación de imágenes. Aquí, el foco se dibuja con el

enfoque estadístico, que permite el análisis discriminante en los grandes espacios de características y grandes conjuntos de datos de entrenamiento.

Se presenta una base de datos de mamografías desarrollado a partir de **la unión de** la base de datos MIAS, La base de datos DDSM, LLNL, y RWTH de Aachen.

IRMA, utiliza la **codificación estándar** de acuerdo con el American College of Radiology (ACR) para los códigos de tejidos y el sistema de notificación de datos de imagen de mama y ACR (BI-RADS).

La base de datos también contiene información acerca de cada algoritmo de extracción de características.

Obviando la resolución, dio lugar a un total de **10.509 imágenes de referencia**, y 6.767 imágenes asociadas con un archivo de información de función contorno IRMA. De libre disposición para los investigadores [209].

### 2.4.7 AMDI Indexed Atlas of Digital Mammograms

AMDI - Atlas indexado de las mamografías digitales, **proporciona una herramienta** que permite al usuario descargar los casos de la base de datos mamográfica a fin de que la información se encuentre disponible para investigación autorizada y para las comunidades médicas dedicados al diagnóstico del cáncer de mama. El proyecto de la base de datos mamográfica se pensó para incluir casos con **todas las vistas mamográficas** disponibles, hallazgos radiológicos, diagnósticos probados por biopsia, la historia clínica del paciente, y la información sobre el estilo de vida del paciente.

Cada examen de cada caso incluye **cuatro vistas** (dos vistas de cada mama cráneo-caudal o CC y oblicua medio lateral o MLO). Para hacer frente a los aspectos docentes y de investigación, la base de datos **enlaza** cada mamografía con el contorno de la mama, el límite del músculo pectoral (sólo MLO), los contornos de masas (si está presente), las regiones de

grupos de calcificaciones y el número de calcificaciones (si está presente), y los lugares y detalles de cualesquiera otras características de interés. La base de datos mamográfica también se apoya en la inclusión de varios exámenes mamográficos del mismo paciente, realizado en diferentes instantes de tiempo.

### 2.4.8 B-SCREEN (Bayesian Decision Support in Medical Screening)

En 2006 comienza en Holanda la digitalización de en la detección del cáncer de mama. Todas las mamografías se almacenan en un **archivo nacional**, gracias al uso de la tecnología de banda ancha. Consecuentemente, este proyecto contará con una gran base de datos de casos de cáncer de mama que estará disponible en unos pocos años. [211]

### 2.4.9 MammoGrid

La base de datos **europea** implementada en una estructura GRID, es una aplicación que tiene como fin desarrollar una base de datos a nivel europeo de mamografías y apoyar la cooperación efectiva de trabajo entre los profesionales sanitarios de la UE. El proyecto MammoGrid, ya finalizado, ha sido financiado por el 5º Programa Marco Europeo y duró desde **2002 hasta 2005**.

Los objetivos del proyecto eran de utilizar nuevas tecnologías Grid para proporcionar un mejor acceso a los datos distribuidos y el desarrollo de herramientas software para comparar automáticamente las mamografías y así extraer información de la imagen, y de los tejidos.

En el CERN, esta tecnología **se utiliza para**:

- Consulta y recuperación de imágenes.
- CAD de detección de microcalcificaciones opacidades y el análisis de imágenes.
- Formato mamografía estándar (SMF) que permite la estandarización de las imágenes y por lo tanto su comparación.

- Se utiliza para adquirir imágenes de mamografía digital o con un escáner en los hospitales y para asociar y almacenar datos.

- Análisis de imágenes médicas más allá del diagnóstico para controlar la progresión y la terapia de la enfermedad.

- Mamografías, compartiendo información y asesoramiento médico en una escala a nivel europeo utilizando las tecnologías GRID.

- La infraestructura de la arquitectura no se limita a las mamografías o diagnóstico de cáncer. Se incorpora el conocimiento en los programas sobre las masas, el tejido mamario denso y microcalcificaciones.

Tiene **características innovadoras** como son:

- Persistencia de modelos de datos y meta-datos para manejar la heterogeneidad.

- El uso práctico de la cuadrícula en los hospitales.

- Distribuida y con el sistema de gestión de datos clínicos compartido.

- Sistema seguro para ser integrado en los sistemas de información del hospital.

- Anotación de las mamografías.

La base de datos de las mamografías contiene más de 30.000 imágenes.

Permite:

- El intercambio a distancia de los diagnósticos y la segunda opinión entre los médicos.

- Epidemiología más diversa.

- SMF mejora la calidad de las imágenes de mamografía.

- Estadísticamente significativo número de exámenes y condiciones.

- Los informes del médico clínico estructurado.

### 2.4.10 Comparativa y Elección

| | DDSM | LLNL | NDB | Mini MIAS | Magic -5 |
|---|---|---|---|---|---|
| Casos | 2620 | 50 | - | 161 | 967 |
| Imágenes | 10480 | 198 | 40 | 322 | 3369 |
| Lado pixel | 50 m | 35 m | 100 m | 200 m | 85 |
| Niveles gris | 12 a 16 bits | 12 bits | 12 bits | 8 bits | |

Concluimos con la comparativa de las BBDD con el siguiente análisis:

Las bases de datos más populares en trabajos de investigación encontrados en la red son DDSM, MIAS o NDB pero la NDB está desequilibrada en cuanto a los casos malignos, normales y benignos.

**Nosotros nos decantamos por DDSM** porque la calidad del diagnóstico está garantizada con comprobaciones posteriores como biopsia u otras pruebas (cáncer) con seguimiento posterior a cuatro años (benigno). Ademas se desarrolla en un entorno de colaboración entre entidades con cierto prestigio, lo que presupone que sus datos son ciertos y de gran calidad.

Es una de las BBDD más extendidas en su uso por la comunidad científica y también más interesante para uso en sistemas CAD por ser la más completa. Las imágenes DDSM son digitales. Se actualiza frecuentemente la BBDD. Intenta emular una casuística completa de Screening de mama.

# Capitulo 3 Modelo Práctico.

## 3.1 Dificultad del Modelo

El problema de reconocimiento de imágenes mamográficas mediante entrenamiento y en concreto el saber determinar si hay lesiones cancerígenas o no, **no es un problema trivial**.

Recordemos que estudios basados en programas de screening actuales de cáncer de mama señalan con datos objetivos basados en la **retrospectividad** que hay un porcentaje elevado de tumores diagnosticados y que no fueron detectados en su momento porque fueron interpretados como casos normales, que se mueve en la banda **del 30 al 70 %** [213] [214] [215] y que el experto radiólogo no los ha advertido bien por errores de detección o bien por errores de interpretación. El CAD ayuda a la detección, pero no a la interpretación y al manejo de las lesiones, actividades que dependen únicamente del radiólogo.

Además se confirma por otros estudios, que la **sensibilidad** que se alcanza en la detección visual de la mamografía, no supera el 75%. [212] Bien es verdad que la sensibilidad varía en función del tipo de mamografía y del tipo de lesión siendo más alta en mamografía digital que en la analógica y en microcalcificaciones que en masas.

Entre otros objetivos se trata de **reducir los falsos positivos y los falsos negativos** y obtener un elevado porcentaje de verdaderos positivos.

Debemos ser capaces de discriminar un conjunto de clases procesando datos de entrada cuyas características no se definen claramente por un tipo de algoritmo concreto de la programación. Nos encontramos que tenemos para clasificar una **clase demasiado muy amplia** si la comparamos con el número de ejemplos que tenemos a disposición por ejemplo en el caso que nos ocupa, es del todo intratable un conjunto de imágenes posibles de $256x256$ con baja resolución pero si lo que ocurre es que tenemos limitado el número de ejemplos y el espacio de hipótesis es amplio, nuestro modelo puede incurrir en **overfitting o**

**sobreaprendizaje**. Esta situación es muy frecuente en los modelos de características similares al propuesto y tiene como consecuencias que el clasificador generaliza de manera muy pobre ante nuevos datos. Además de esto, los algoritmos pueden resultar poco eficaces debido a varias causas tales como encontrarse con **mínimos locales** añadiendo otra dificultad y es la gran cantidad de parámetros necesarios para el control del algoritmo ajustados a través de la heurística que hacen que los CAD no sean de gran fiabilidad o el uso se vuelva dificultoso.

En cuanto a la **capacidad de procesamiento** de Imágenes, hasta hace poco tiempo era una de las limitaciones importantes en la investigación de tipo doméstico. Es importante disponer de equipos con tarjeta gráfica dedicada GPU de 2 GB o superior aunque lo deseable por la experiencia en este trabajo hubiera sido disponer de 8GB de GPU.

Para trabajar con importantes volúmenes de datos, lo mejor es utilizar GPUs potentes antes que usar CPUs, esto es debido fundamentalmente por el ancho de banda de las GPU, **principal cuello de botella**. En nuestro caso se ha utilizado sistema operativo Windows 7, GPU GeForce GTX860M, procesador Intel Core I7-4710HQ CPU@ 2.50 GHZ y memoria RAM de 8GB.

```
>> gpuDevice(1)

ans =

  CUDADevice with properties:

                      Name: 'GeForce GTX 860M'
                     Index: 1
         ComputeCapability: '5.0'
            SupportsDouble: 1
             DriverVersion: 7.5000
            ToolkitVersion: 5
        MaxThreadsPerBlock: 1024
          MaxShmemPerBlock: 49152
        MaxThreadBlockSize: [1024 1024 64]
               MaxGridSize: [2.1475e+09 65535 65535]
                 SIMDWidth: 32
               TotalMemory: 2.1475e+09
                FreeMemory: 2.0112e+09
       MultiprocessorCount: 5
              ClockRateKHz: 1019500
               ComputeMode: 'Default'
      GPUOverlapsTransfers: 1
    KernelExecutionTimeout: 1
          CanMapHostMemory: 1
           DeviceSupported: 1
            DeviceSelected: 1

>>
```

Figura 109 Caracteristicas mostradas por Matlab de la GPU

GTX860M cuenta con todos los núcleos CUDA desbloqueados, 640 en total. Las especificaciones las completan 40 TMUs, 16 ROPs y 2 GB de memoria con un ancho de banda de 128 bits. El núcleo funciona a 540 Mhz y la memoria a 5 Ghz efectivos (GDDR5).

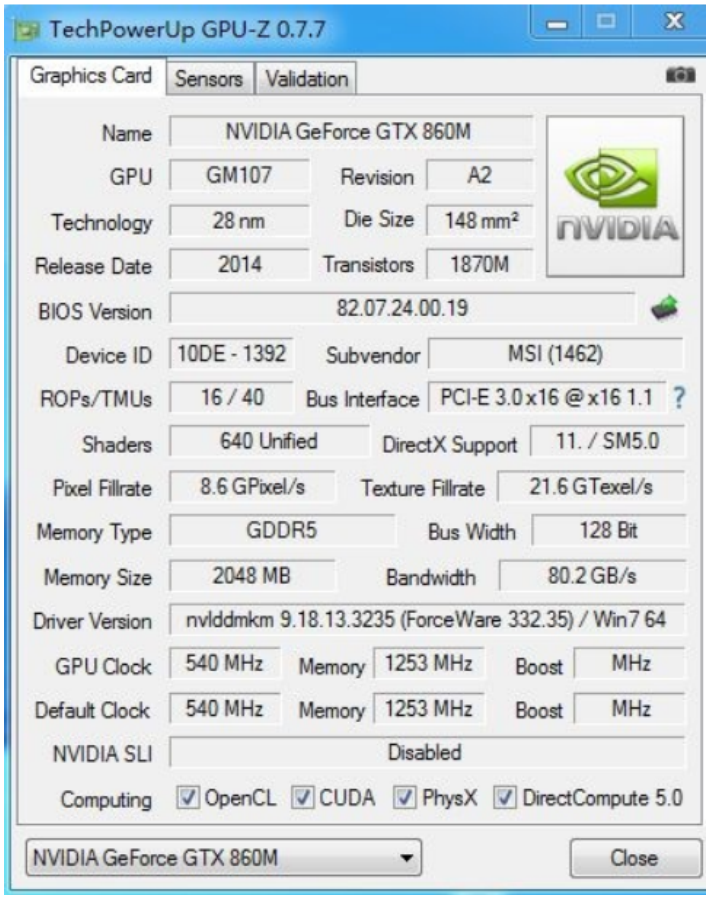



Figura 110 caracteristicas de la GPU NVIDIA GeForce GTX 860M

Si una GPU dedicase demasiado tiempo en el entrenamiento, convendría usar varias GPU en una sola máquina SLI o un clúster y en este caso podemos visitar el sitio WEP Apache. http://hama.apache.org/

## 3.2 Descripción del Modelo

En primer lugar tenemos que **seleccionar imágenes** con regiones de interés (ROI) que vamos a extraer de imágenes de la BBDD DSSM.

Para poder llevar a término esta etapa, realizamos las siguientes acciones:

Se selecciona un conjunto de mamografías candidatas que tengan casos normales, lesiones malignas y benignas. Las **lesiones que nos interesan** corresponden a microcalcificaciones; como ya viene indicado en la BBDD, la ubicación de la zona donde se encuentra la lesión en la imagen, así como las coordenadas del centro de la lesión, calcularemos los extremos la ROI en base a un número específico de píxeles. En este trabajo vamos a usar imágenes de la base de datos DDSM y después de realizar este proceso, contaremos con una **colección de imágenes ROI** que usaremos para entrenar y probar el sistema CAD propuesto. El tamaño de las imágenes ROI usadas para nuestro trabajo es de 256×256 píxeles con una resolución espacial de 50 $\mu/pixel$.

La segmentación es una de las etapas de mayor importancia en cualquier trabajo en el que se procesen imágenes digitales. Nuestro objetivo es subdividir la imagen en regiones u objetos homogéneos respecto a características como nivel de intensidad de gris, color, textura.

Aunque hay técnicas novedosas como el algoritmo de agrupamiento particional PFCM (Possibilistic Fuzzy c-Means). Propuesto por Ojeda-Magaña, Quintanilla-Domínguez, Ruelas y Andina en 2009 [220] y otras más clásicas como el caso de C-medias y C-medias difuso, nosotros realizaremos el proceso de forma manual construyendo una BBDD con ROI´s de $256x256$ con casos patológicos relativos solo a Microcalcificaciones. Nuestra BBDD consta de 390 imágenes de las cuales parte irán destinadas a entrenamiento mediante aprendizaje supervisado y otro subconjunto de imágenes desconocidas para el sistema para pruebas y evaluación.

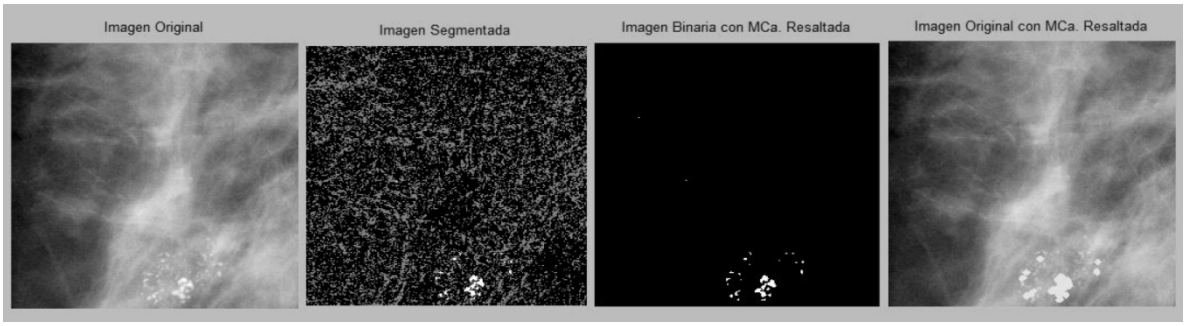



Figura 111 Aplicación de la herramienta de detección PFCM

Aunque sería deseable **un conjunto mayor**, el procesador y la memoria gráfica disponible para la realización de este trabajo limita actualmente el modelo, pero no debemos preocuparnos porque nuestro conjunto es suficiente para montar el CAD con ciertas garantías. Los modelos de otros investigadores tienen conjuntos que oscilan entre 100 y 3000 imágenes aproximadamente.

La BBDD más completa es la **DDSM** que cumple con los criterios de tener suficientes casos e imágenes y además equilibrada en casos malignos, benignos y normales. Muy utilizada por los investigadores, actualizada, muy bien informada, de alta resolución y por estos motivos he decidido que sea la BBDD elegida como la **más apropiada para el CAD objeto de nuestra investigación.**

Bien es verdad que la **descarga supone 270 GB** y solo recientemente disponemos de medios con capacidad suficiente para que a nivel doméstico sea posible su descarga. Las Imágenes tienen **formato LJPEG** y se descomprimen mediante software disponible en sitio ftp de DSSM o con otras **herramientas completas** como XMedCon o Imread, gdcmimg, Herramientas Yakami DICOM [224] y demás aplicaciones puestas a disposición de los investigadores para hacer la descompresión y conversión de la DDSM a múltiples formatos.

Durante la investigación en la que hemos podido realizar la comparación en diversos modelos de **preprocesado**, hemos podido comprobar que para obtener la separación de las microcalcificaciones del fondo propio de la imagen, así como para evitar que el perímetro de la microcalcificación sea confundido con zonas brillantes, tenemos que realizar el procesamiento a las imágenes basado en una **mejora de contraste de imagen** y en **detectar los bordes de estas**. En una imagen ROI por lo general, Las microcalcificaciones corresponden a píxeles con un nivel de intensidad mayor que los píxeles de entorno, aunque esto no siempre es así, por ejemplo, el contraste en un seno con tejido denso de las microcalcificaciones respecto a este tejido es bajo por lo que el ojo humano muchas veces es incapaz de distinguirlo. Para corregir esto utilizamos técnicas exprofeso para ello como el máximo de la transformada Wavelet y un procesado realizado a posteriori para el seguimiento y conexión de bordes nos permite cerrar el contorno de la calcificación [216] por ejemplo con la transformada contourlet [217].

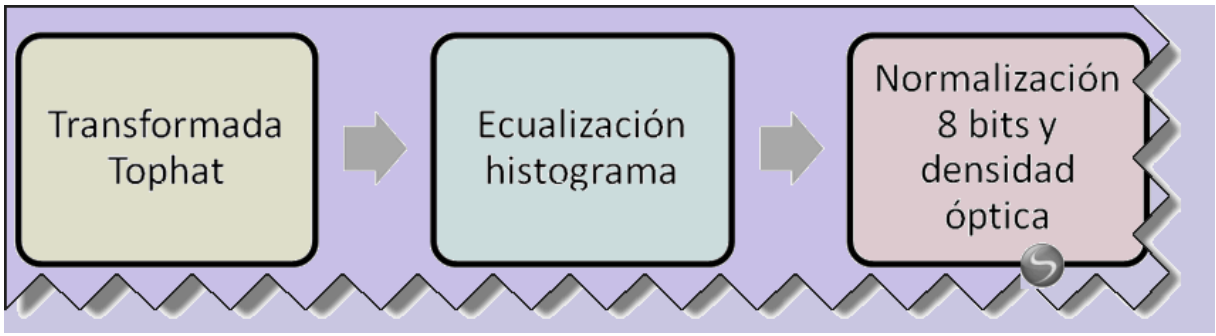



Figura 112 Etapa de preprocesado

En nuestro trabajo vamos a mejorar el contraste de las microcalcificaciones utilizando **operadores morfológicos,** las operaciones morfológicas permiten extraer estructuras geométricas de una imagen para transformarlas o mejorarlas de acuerdo con algún objetivo predefinido, **mediante transformaciones Tophat y bottomhat** propuesta por Meyer[218].

Este **filtro de tipo residual** compuesto por la diferencia entre dos transformaciones básicas que realizan el resalte de aquellas regiones de la imagen anteriormente suprimidas por la operación de apertura. Mediante la elección adecuada del tamaño y de la forma de un elemento estructurante y de la aplicación de una operación de apertura, obtenemos la imagen modificada en la que se han eliminado elementos concretos de la imagen original. Diferenciando la imagen original con la modificada se consigue aumentar el contraste de las regiones eliminadas. **Tophat nos evita** que se puedan realzar artefactos que no son interesantes y que pueden confundir al sistema CAD si la escala no es muy selectiva. Es como una lupa de diámetro variable que permite que los objetos con menor radio que el de la lupa se seleccionen.

La operación **devuelve una imagen** con los objetos menores que el elemento estructurante de 3x3 y con más brillo que sus vecinos. Si queremos usar otro tamaño diferente o aplicarlo sobre una imagen en escala de grises se debe usar **imtophat**. Muy útil para extraer objetos pequeños o detalles claros sobre fondo oscuro o corregir la iluminación desigual cuando el fondo es oscuro. el contraste se mejora además si se suma la imagen original y la imagen filtrada con top-hat.

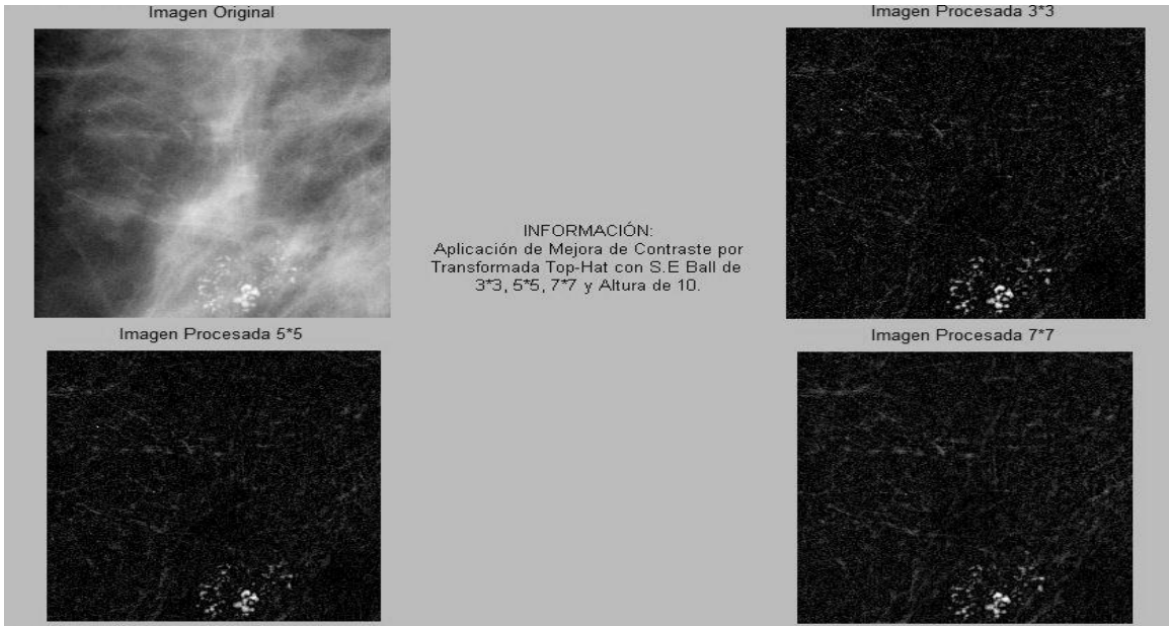



Figura 113 Mejora de contraste por acción de la transformada de Tophat [221]

En nuestro caso, el **elemento estructurante** es de forma circular, de 7 pixeles de radio por lo que suprimirá estructuras de mayores de 0,35 mm de diámetro con la resolución de nuestras imágenes de 50 micras por pixel.

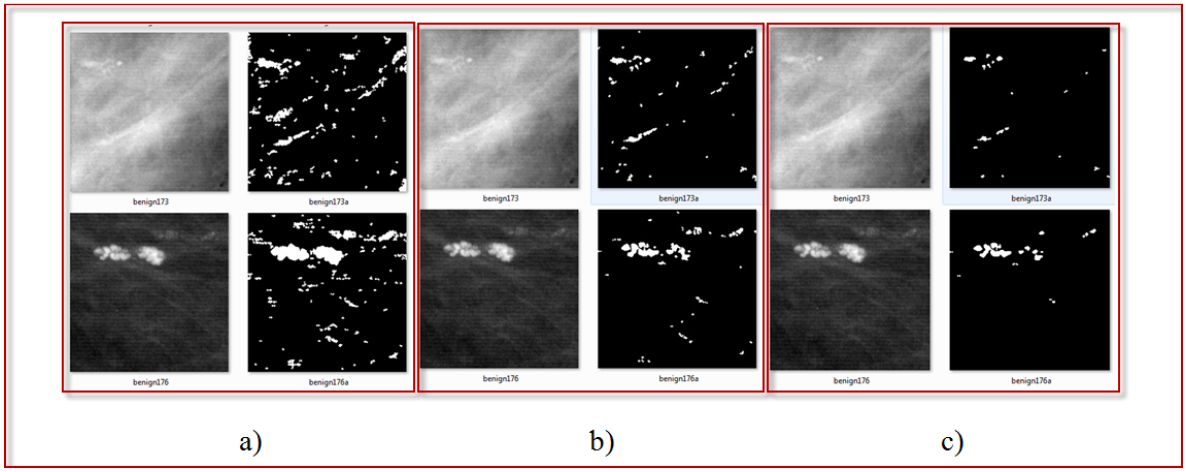



Figura 114 Efecto de tophat

Podemos observar el **efecto de tophat variando el tamaño** del elemento estructurante:

a)Imagen con elemento estructurante de 12 pixeles

b)Imagen con elemento estructurante de 8 pixeles

c) Imagen con elemento estructurante de 6 pixeles

Hay trabajos que refieren el excelente resultado de la transformada tophat para este tipo de CAD como el de Juan Luis Cerrato Vargas y Joel Quintanilla Domínguez [219]

Es importante que las microcalcificaciones estén presentes tras el preprocesado. El módulo de preprocesado generará a partir de las 390 imágenes, que disponen de una **escala de profundidad de grises de 12 de bits** otro paquete en el que se eliminan ruidos y artefactos y cuyas imágenes reducen la profundidad a 8 bits. El procesamiento es común para el conjunto de entrenamiento y para el de validación.

En la etapa de preprocesado aplicamos la **transformada Tophat**, seguidamente realizamos la **ecualización del histograma.** El histograma es un gráfico que relaciona los niveles de intensidad de gris de una imagen con el número de pixeles que hay para cada nivel de intensidad. Una imagen tendrá buen contraste cuando su histograma se extiende ocupando prácticamente todo el nivel de tonos. En la transformación de la imagen, todos los pixeles que tengan un mismo nivel de gris se transformarán a otro nivel de gris, y el histograma se distribuirá en todo el rango disponible, separando en lo posible las ocupaciones de cada nivel. El resultado de la ecualización hace máximo el contraste de una imagen sin perder la estructura de la información, conservando su información.

Se puede realizar la ecualización con el programa **ImageJ**[222] cuyo código fuente es gratuito y puede ejecutarse en un applet en línea, como aplicación ejecutable, o en cualquier computadora con Máquina virtual Java 5 o superior. ImageJ puede mostrar, editar, analizar, procesar, guardar, e imprimir imágenes de 8 bits (256 colores), 16 bits (miles de colores) y 32 bits (millones de colores). El programa es compatible con cualquier número de imágenes al mismo tiempo, limitado solamente por la memoria disponible.

También existe "**La Image Processing Toolbox™ (IPT)** de MATLAB que es un conjunto completo de algoritmos y herramientas graficas para el procesado, análisis y visualización de imágenes así como desarrollo de aplicaciones y nuevos algoritmos en el campo de procesado y análisis de imágenes digitales

En este trabajo proponemos la variante de ecualización adaptativa de histograma con limitación de contraste (CLAHE). Para ello se dispone de la función ***adapthisteq*** **.** Es la función más potente de la toolbox para mejorar el contraste de la imagen. También es posible realizarla con funciones más sencillas como *histeq* o *imadjust*

**Sintaxis**

```
J = adapthisteq(I)
J = adapthisteq(I,param1,val1,param2,val2...)
```

Figura 115 Sintaxis comando para CLAHE, adapthisteq

**CLAHE** actúa en pequeñas regiones de la imagen llamadas baldosas en lugar de hacerlo en toda la imagen. Seguidamente se combinan las baldosas vecinas mediante interpolación lineal para suprimir las fronteras artificiales. La ventaja es que se puede aumentar el contraste necesario a la carta en cada zona evitando la sobreexposición de zonas que ya se encuentran bien contrastadas [223]. **Hemos llegado a la conclusión** de manera empírica que nuestras baldosas van a ser de $8x8$ teniendo en cuenta las características de las imágenes de la base de datos de mamografías y evitando así variaciones pronunciadas. El valor del **límite de amplificación del ruido** en el algoritmo CLAHE oscila entre 0,02 y 0,06 porque valores superiores perjudican ocasionando un aumento del ruido en las imágenes.

Y por último en la fase del preproceso también vamos a **normalizar** a 8 bits para cumplir con dos características, la compatibilidad y el ahorro de memoria en el tratamiento de imágenes. Se genera un paquete de imágenes de 256 niveles de gris normalizadas en las cuales se han resaltado las microcalcificaciones y el fondo aparece en color negro.



Figura 116 Imagen ecualizada y normalizada

En las estructuras de tipo cell se almacenan las imágenes intermedias para su ulterior estudio.

En nuestro caso al ser todas las imágenes procedentes de la misma BBDD , DSSM y procedentes del mismo escáner Howtek de 12 bits no es especialmente importante corregir la densidad óptica. Si lo sería si procedieran de distintos scaner como ocurre con la BBDD MammoGrid cuyas fuentes son las distintas BBDD europeas de sus "socios", los centros médicos. La **función softmax** se suele implementar en la capa final de la red utilizada para la clasificación. Su sintaxis en Matlab es la siguiente: A = softmax(N,FP) siendo N: $SxQ$ matriz de vectores (columna) de entrada neta. y FP: Estructura de los parámetros de función.

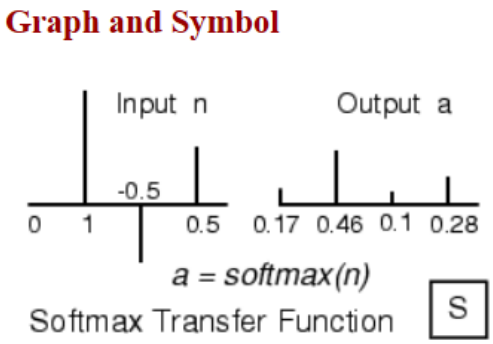



Figura 117 Softmax Transfer Function

Hay muchas **aplicaciones para trabajar en aprendizaje profundo**:

Theano con Pylearn: Es una de las herramientas que mayor popularidad tiene en la comunidad científica. Se programa en lenguaje python, y permite utilizar CPU y GPU. http://deeplearning.net/software/pylearn2/

Torch: Usado por Facebook Research, Google Deepmind y Twitter. Rivalidad y competencia con Theano, sobre todo en rendimiento. programación en Lua. Soporta CPU y GPU. Más detalles en: http://torch.ch/

Caffe: Es una herramienta con el enfoque puesto en el procesado de imágenes. La ventaja es que se programa en C++. Soporta CPU y GPU. Más detalles en: http://caffe.berkeleyvision.org/

Google acaba de liberar Tensorflow, una librería muy flexible que soporta además de deep learning, machine learning en general. http://www.tensorflow.org

OpenNN, una librería muy popular en código abierto que implementa redes neuronales profundas.

MatConvnet (Matlab) http://www.vlfeat.org/matconvnet/

Cuda-convnet (CUDA) https://code.google.com/p/cuda-convnet2/

Deepleaning4j (Java) http://deeplearning4j.org/

ConvNet: Deep Convolutional Networks http://libccv.org/doc/doc-convnet/

Deep Learning Libraries by Language http://www.teglor.com/b/deep-learning-libraries-language-cm569/

En nuestro caso nos hemos decidido por **Matcovnet (redes neuronales convolucionales para Matlab)**. es una implementación de Redes Neuronales Convolucionales (CNNs) para **MATLAB**.

Millones de usuarios ingenieros y científicos en el mundo utilizan MATLAB® como herramienta multiplataforma de software matemático con lenguaje de programación propio, para analizar y diseñar los sistemas y productos que transforman nuestro mundo. MATLAB está presente en variedad de soluciones complejas como los sistemas de seguridad activa de muchos automóviles, naves espaciales, Biotecnología, aparatos de monitorización de la salud, redes móviles LTE y eléctricas y su uso cada vez mayor de aprendizaje automático, proceso

de señales, proceso de imágenes, visión por computador, robótica, telecomunicaciones etc. [228] Es un software muy utilizado en universidades de todo el mundo y en centros I+D+I.

Matconvnet es una librería simple y flexible e incluye varias **funciones de MATLAB organizados de la siguiente manera** [229]:

**Bloques de construcción** . Estas funciones implementan los bloques de cálculo de la CNN que se pueden combinar de forma manual o usando uno de los envoltorios previstos para construir CNNs.

**Envoltorio SimpleCNN** . Son cadenas de bloques lineales computacionales.

**Envoltorio DagNN .** Es una capa orientada a objetos para el apoyo de topologías de red más complejas.

**Otras funciones** . Funciones auxiliares que se utilizan para inicializar y compilar MatConvNet.

**Bloques de construcción CNN**:

- **vl_nnbnorm** normalización de lotes.
- **vl_nnbilinearsampler** Bilineal Samppler.
- **vl_nnconv** convolución lineal por un filtro.
- **vl_nnconcat** Concatenación.
- **vl_nnconvt** Transpuesta convolución.
- **vl_nncrop** Recorte.
- **vl_nndropout** Abandonar o matar una neurona con probabilidad $p$
- **vl_nnloss** Registro de la pérdida de la clasificación.
- **vl_nnnoffset** Norma dependiente de offset.
- **vl_nnnormalize** La normalización de respuesta local (LRN).
- **vl_nnpdist** distancias por parejas.

- **vl_nnpool** Max y suma la puesta en común.
- **vl_nnrelu** Unidad lineal Rectificada.

- **vl_nnsigmoid** Sigmoidal.
- **vl_nnsoftmax** Canalizar suave-max.
- **vl_nnsoftmaxloss** Obsoleto
- **vl_nnspnorm** Normalización espacial.´

**SimpleCNN**

- **vl_simplenn** Una envoltura ligera para CNNs con una topología lineal.
- **vl_simplenn_tidy** de actualización o de otra manera fijar una CNN.
- **vi_simplenn_display** Imprimir información sobre la arquitectura de la CNN.
- **vl_simplenn_move** Para mover la CNN entre la CPU y la GPU.

**DagNN** Es un envoltorio orientado a objetos para la CNN con topologías complejas

**Otras funciones**

- **vl_argparse** una función de ayuda para analizar los argumentos opcionales.
- **vl_compilenn** Compilar los campos MEX en la caja de herramientas.
- **vl_rootnn** devuelve la ruta hasta la instalación de Toolbox MatConvNet.
- **vl_setpunn** configuración de MatConvNet para su uso en MATLAB.
- **vl_imreadjpeg** cargar rápida de un lote de imágenes JPEG.

Soporta los cálculos de proceso en la CPU y en la GPU, lo que permite entrenar modelos complejos con grandes conjuntos de datos.

Esta librería puede aprender grandes modelos CNN como tales como la red neuronal profunda AlexNet diseñada para el reconocimiento de imágenes.

Para utilizar la toolbox, lo primero que hay que hacer es **instalarla y compilarla**:

Descargar y descomprimir el código fuente de la biblioteca en un directorio de nuestra elección. En la ruta de acceso , el directorio se llamará <MatConvNet>.

### Compilar la biblioteca.

Iniciar MATLAB y escribir: > run <MatConvNet>/matlab/vl_setupnn con el fin de añadir MatConvNet a la ruta de búsqueda de MATLAB.

En este punto, la biblioteca está lista para usarse. Lo podemos comprobar utilizando el comando:

> vl_test_nnlayers

Para comprobar la GPU (si ha compilado ) utilizamos:

>vl_test_nnlayers (true)

La compilación de MatConvNet se produce utilizando la función de MATLAB

> vl_compilenn

Hay que asegurarse de que MATLAB está configurado para utilizar el compilador. Hay que abrir MATLAB y escribir los comandos:

> cd <MatConvNet> > addpath matlab > vl_compilenn

y en este punto la biblioteca se empieza a compilar. previo a la compilación hay que tener instalado en el PC Visual studio 2012.

## Programa

Vamos a describir el módulo de preprocesado. El fichero que hemos creado se llama TophatprocesadoCNNnatconvnet390.

Las imágenes de la base de datos DSSM se preparan manualmente,

```
a = im_lee_ljpeg('B_3111_1.LEFT_CC.LJPEG');
imtool(a);
A= 963;
B=2352;
C = a(B:B+255,A:A+255);
imtool(C);
```

Figura 118 Corte manual de una imagen

Se preprocesan con un script que hemos preparado denominado TophatprocesadoCNNnatconvnet390 y se almacenan en una estructura de tipo struct llamada imdb390.mat que estaría lista ya para que fuera tratada para la librería Matconvnet

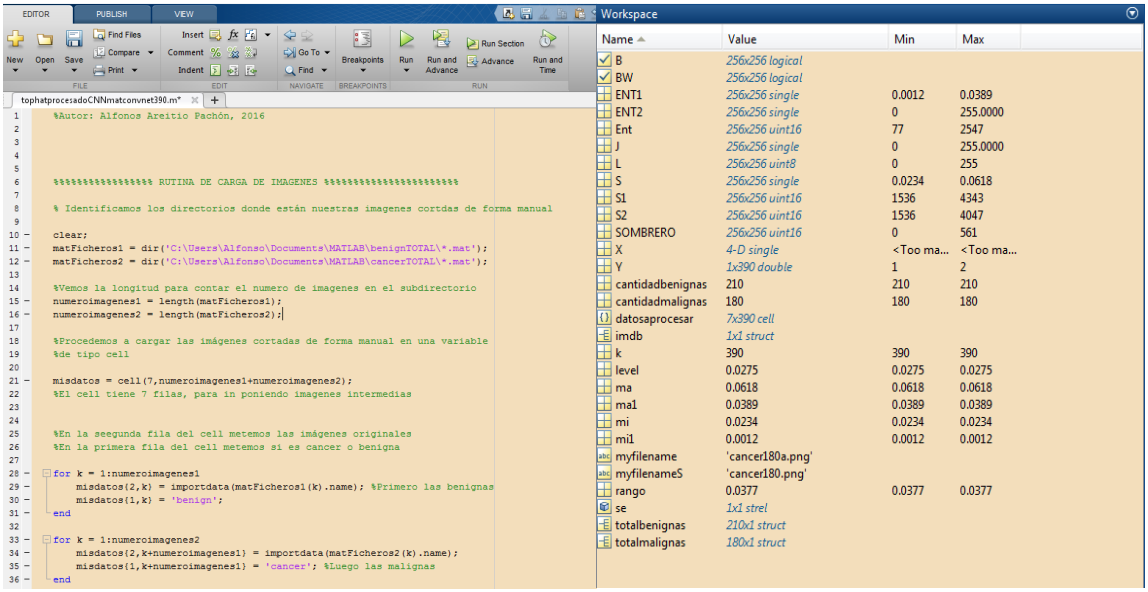



Figura 119 Extracto de TophatprocesadoCNNnatconvnet390 y vista del espacio de trabajo del script

Se trabaja con 390 imágenes de las cuales de la **1 a la 210** son benignas y **el resto** cancerígenas.

**Tomamos de la imagen** una a 160 para entrenar, de la 161 a la 261 para probar y de la 262 a la 390 para entrenar.

Creamos **estructura cell intermedia formada por 7 filas y 390 columnas**.

Una vez cargada la fila 1, contiene la información de la imagen si es benigna o maligna.

La fila 2 almacena la propia imagen original.

Con estos datos ya estamos listos para realizar el preprocesado a las imágenes.

Aplicamos tophat probando **configuraciones con distintos radios** de 3, 5, 6, 7, 8 pixeles mediante el comando "imtophat" y seguidamente para mejorar el contraste, aplicamos CLAHE a cada imagen de entrada y para eliminar artefactos, que no aportan nada al problema, los objetos menores de 1 mm son eliminados de la imagen.

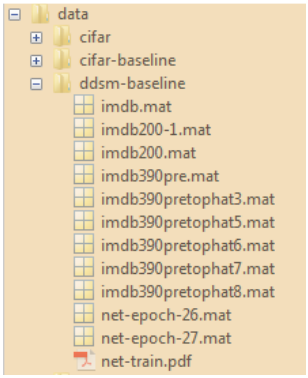



Figura 120 diferentes configuraciones de preporcesado utilizadas en el módelo

**Mapeamos** los 4096 tonos de grises reduciéndolos a tan solo 256 para ahorrar tiempo de proceso y las guardamos en la fila 3 del cell "datosaprocesar" y en la fila 7 guardamos la imagen de entrada normalizada a 255 para visualizarla en la pantalla.

La fila 4 contiene los diagnósticos $\begin{cases} 1\ cancer \\ 2\ benigno \end{cases}$

**Umbralizamos** para convertir a imagen binaria B de todo o nada 0 o 256, tomando el graythresh y lo guardamos en la fila 6.

Creamos los ficheros de salida en los directorios Obening y Ocancer extrayendo las filas 6 y 7.

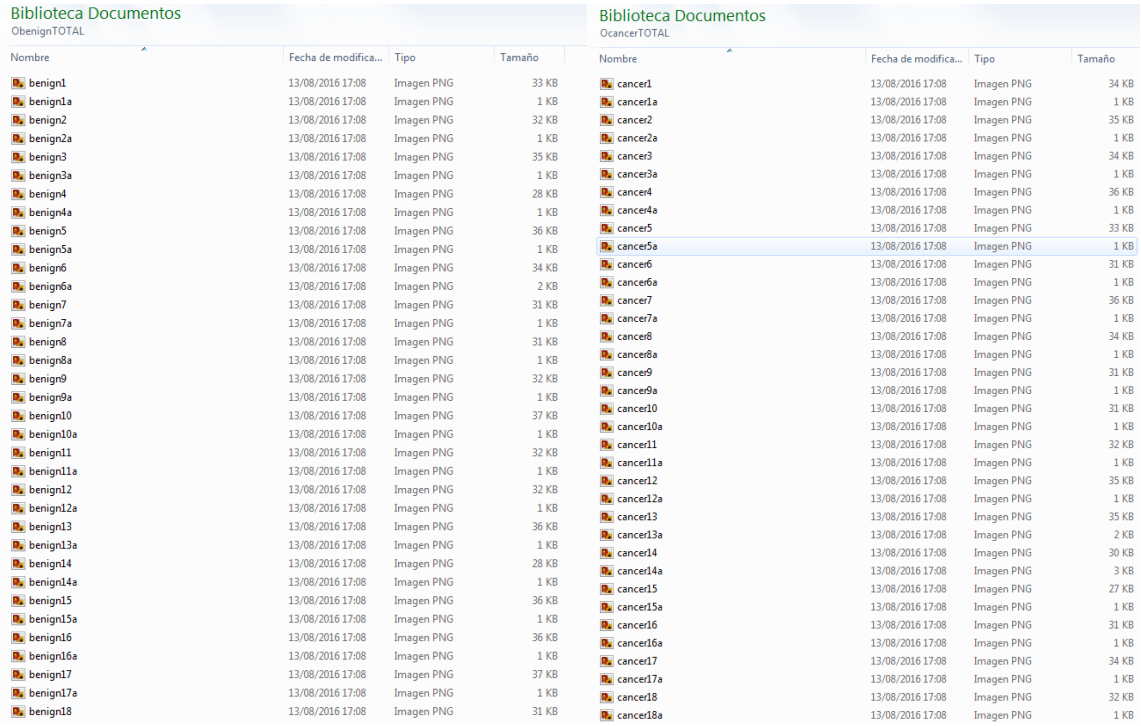



Figura 121 vista parcial de los directorios que contienen las imágenes preprocesadas con cáncer y benignas

**Creamos matrices** tetra-dimensionales X de imágenes e Y de diagnósticos

La matriz tetra-dimensional con video de 390 imágenes de tamaño 256 $x$ 256 y un nivel de gris.

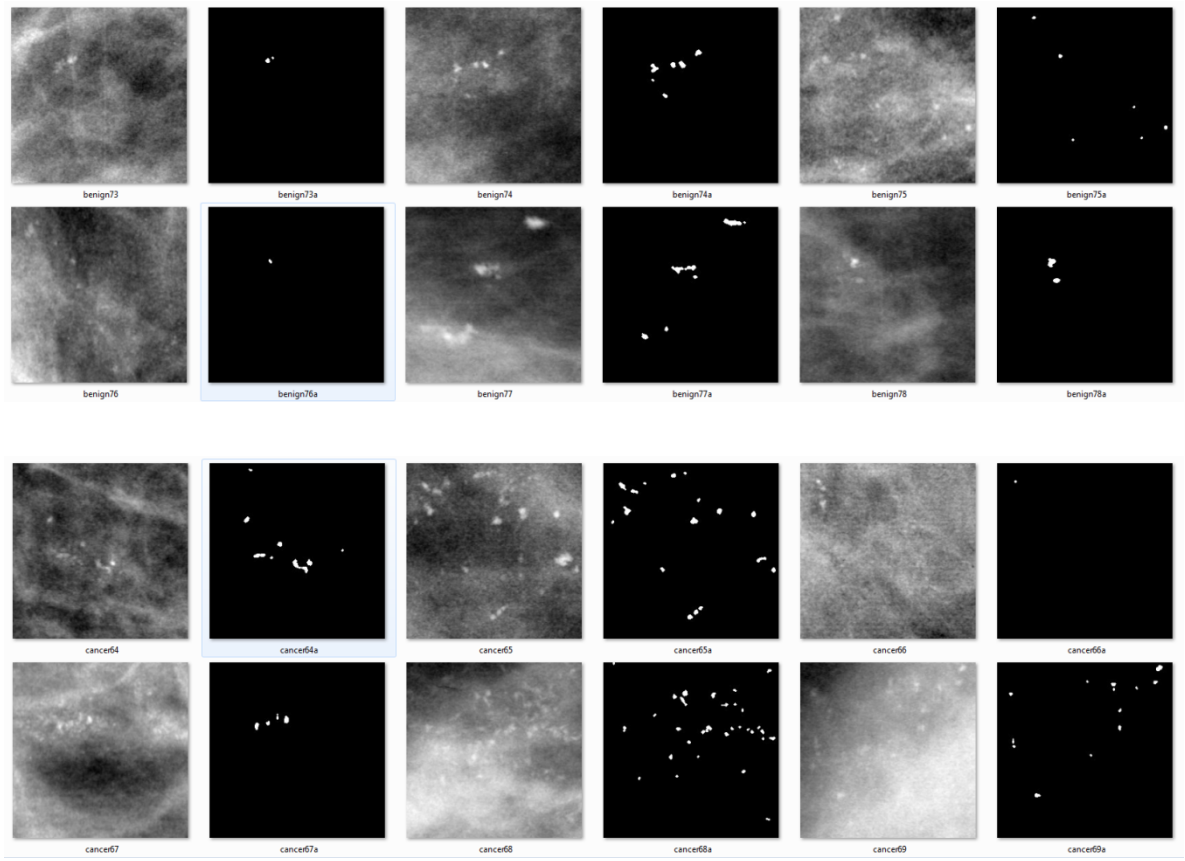



Figura 122 Imágenes preprocesadas aplicando Tophat y CLAHE

En el módulo principal hacemos la llamada a imdb390pre.mat alojado en el directorio “data”

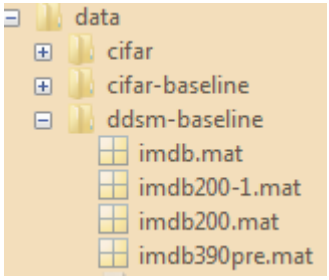



Figura 123 Imdb390pre de tipo struct

Utilizamos para su construcción el archivo ejemplo cnn_mist.m.

Probamos **varias configuraciones** cargando las imágenes finalmente de 10 en 10 y entrenamos con varias iteraciones diferentes en cada prueba aunque comprobamos que 30 es el óptimo en base a la experiencia ya que no es hasta que transcurren al menos 20 iteraciones cuando empieza a dar un resultado estable en los errores de entrenamiento.

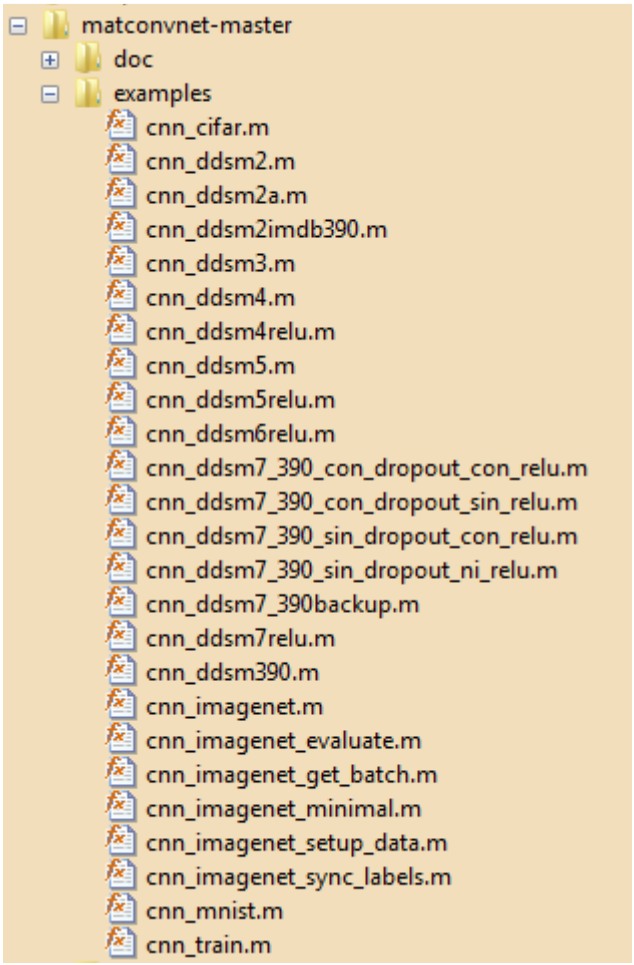



Figura 124 Varias configuraciones probadas en distintas redes de nuestro modelo

Habilitamos el procesamiento de imágenes a través de la **GPU** (graphics processing unit) lo que nos permite una mayor velocidad de proceso ya que su función es aligerar la carga de trabajo del procesador central y por ello, está unidad está muy optimizada para el cálculo en coma flotante, predominante en las funciones 3d. La GPU tiene un elemento de potencia

muy sobresaliente (**Shaders**) que en el caso de la GPU NVIDIA recibe el nombre de **núcleos CUDA.** En nuestro caso la tarjeta gráfica es dedicada.

Vamos a utilizar 4 capas de convolución, 2 de pooling y vamos a probar con varias configuraciones de ReLU y de Dropout.

Usamos las nueva funcion de activación como rectified linear units (ReLUs) que no tiene tantos problemas "disolución" del gradiente (Nielsen 2015) y el mecanismo dropout que permite evitar el overfitting,

A continuación se muestran los módulos de preprocesado y principal:

## Módulo de preprocesado

```
%Autor: Alfonso Areitio Pachón, 2016 Preprocesado.
 %%%%%%%%%%%%%%%%% RUTINA DE CARGA DE IMAGENES
%%%%%%%%%%%%%%%%%%%%%%%%

% Identificamos los directorios donde están nuestras imágenes cortadas de forma manual

clear;

totalbenignas = dir('C:\Users\Alfonso\Documents\MATLAB\benignTOTAL\*.mat');
totalmalignas = dir('C:\Users\Alfonso\Documents\MATLAB\cancerTOTAL\*.mat');

%Vemos la longitud para contar el numero de imágenes en el subdirectorio
cantidadbenignas = length(totalbenignas);
cantidadmalignas = length(totalmalignas);

%Procedemos a cargar las imágenes cortadas de forma manual en una variable
```

```
%de tipo cell

datosaprocesar = cell(7,cantidadbenignas+cantidadmalignas);

%cantidadbenignas+cantidadmalignas es la suma de cáncer+benignas que en este caso suman 390
%El cell tiene 7 filas, para ir poniendo imágenes intermedias
%En la segunda fila del cell metemos las imágenes originales
%En la primera fila del cell metemos si es cancer o benigna

for k = 1:cantidadbenignas
    datosaprocesar{2,k} = importdata(totalbenignas(k).name); %Primero las benignas
    datosaprocesar{1,k} = 'benign';
end

for k = 1:cantidadmalignas
    datosaprocesar{2,k+cantidadbenignas} = importdata(totalmalignas(k).name);
    datosaprocesar{1,k+cantidadbenignas} = 'cancer'; %Luego las malignas
end

%%%%%%%%%%%%%%%%          RUTINA          DE          PREPROCESADO
%%%%%%%%%%%%%%%%%%%%%%%%%%

for k = 1:(cantidadbenignas+cantidadmalignas) % Bucle de 1 hasta 390 imágenes

% La imagen original la llamamos Ent de entrada
Ent = datosaprocesar{2,k};

% Aplicamos tophat con radio de 7 pixeles
SOMBRERO = imtophat(Ent, strel('disk',7));

%Aplicamos CLAHE habitual de Matlab
```

```
S1 = adapthisteq(SOMBRERO,'ClipLimit',0.02);

%Eliminamos objetos de menos de 1 pixel de radio
se = strel('disk',1);
S2 = imopen(S1,se);

% Mapeamos los 4096 niveles de gris en 256 niveles
S = im2single(S2); %CNNs requieren formato single
ma = max(max(S)); %maximo del máximo de todas las filas tono de gris
mi = min(min(S)); %minimo del minimo de todas las filas tono de gris
rango = ma-mi;
J = 255*(S-mi)/rango; %La imagen J es ya de 256 niveles de gris

%Guardamos las imágenes tophat en la fila 3
datosaprocesar{3,k} = J;

%Para viasualizar en un ordenador es preciso pasar a 256 niveles de gris
ENT1 = im2single(Ent); %CNNs requieren formato single
ma1 = max(max(ENT1));
mi1 = min(min(ENT1));
rango = ma1-mi1;
ENT2 = 255*(ENT1-mi1)/rango;
datosaprocesar{7,k} = ENT2; %En la fila 7 guardamos la imagen de entrada normalizada para visualizarla en pantalla

%Umbralización para imagen binaria B, tomando el graythresh
level = graythresh(S);
BW = im2bw(S,level);

%Quitamos objetos de menos de 9 pixeles cuadrados (por ensayo)
B = bwareaopen(BW, 9);
```

```
%Guaradamos en la fila 6 la imagen binaria blanco-negro todo-nada
datosaprocesar{6,k} = B;

end

%%%%%%%%% CREACION DE FICHEROS DE SALIDA EN DIRECTORIOS OBENING Y
OCANCER

for k = 1:cantidadbenignas
myfilename = sprintf('benign%da.png', k);
myfilenameS = sprintf('benign%d.png', k);
cd ('C:\Users\Alfonso\Documents\MATLAB\ObenignTOTAL');
L = 255*uint8( datosaprocesar{6,k}); %Saco toda la fila 6
imwrite( L, myfilename );
L = uint8( datosaprocesar{7,k}); %Saco toda la fila 7
imwrite( L, myfilenameS );
end

for k = 1:cantidadmalignas
myfilename = sprintf('cancer%da.png', k);
myfilenameS = sprintf('cancer%d.png', k);
cd ('C:\Users\Alfonso\Documents\MATLAB\OcancerTOTAL');
L = 255*uint8( datosaprocesar{6,k+cantidadbenignas}); %Saco toda la fila 6
imwrite(L, myfilename);
L = uint8( datosaprocesar{7,k+cantidadbenignas}); %Saco toda la fila 7
imwrite(L, myfilenameS);
end

%%%%%%%%%% PREPARACION matrices tetradimensionales X de imagenes e Y de diagnosticos
%%%%%%%%%
```

```
X = single(zeros( 256,256,1,(cantidadbenignas+cantidadmalignas)));
% X es un “video” de 390 imagenes de tamaño 256x256 y 1 nivel de gris

Y = zeros( 1,(cantidadbenignas+cantidadmalignas) );
%aqui guardamos los diagnosticos, que seran 0 para bening y 1 para cancer

for k = 1:(cantidadbenignas+cantidadmalignas)

    if strcmp(datosaprocesar{1,k},'cancer');
        datosaprocesar{4,k} = 1; %si es cancer pongo 1 y si no pongo 2
        datosaprocesar{5,k} = 0; %si es cancer pongo 0 y si no pongo 1
    else
    datosaprocesar{4,k} = 2;
    datosaprocesar{5,k} = 1;
    end

    %Rellenamos X con las 390 imagenes de la fila 7
    X(:,:,1,k) = datosaprocesar{7,k};

    %Rellenamos Y con los diagnosticos de fila 4: 1 si cancer y 2 si benign
    Y(1,k) = datosaprocesar{4,k};
    %Y(2,k) = datosaprocesar{5,k};

end

%%%%%%%%% CREACION DEL STRUCT imdb.mat PARA PROCESAR EN MATCONVNET
%%%%%%

imdb.images.data = X;
```

```
imdb.images.labels = Y;
imdb.images.set = [ones(1,160) 3*ones(1,100) ones(1,130)] ; % =1 para entrenar, =3 para test total 390
imagenes
imdb.meta.sets = {'train', 'val', 'test'} ;
imdb.meta.classes = {'benign','cancer'} ;

save('imdb390.mat','-struct','imdb');
%save(opts.imdbPath, '-struct', 'imdb') ;
```

## Modulo principal de Matconvnet ddsm7390.m

```
function [net, info] = cnn_ddsm(varargin)
% CNN_DDSM  Demonstrated MatConNet on DDSM

run(fullfile(fileparts(mfilename('fullpath')),'..','matlab','vl_setupnn.m')) ;

opts.dataDir = fullfile('data','ddsm') ;
opts.expDir = fullfile('data','ddsm-baseline') ;
opts.imdbPath = fullfile(opts.expDir, 'imdb390pre.mat');
opts.train.batchSize = 10 ; %cargamos imágenes de 10 en 10
opts.train.numEpochs = 30; %hemos probado con 10 a 50 iteraciones
opts.train.continue = false ;
opts.train.useGpu = true ; %si no hay GPU, poner false para usar la CPU
opts.train.learningRate = 0.001 ;
opts.train.expDir = opts.expDir ;
opts = vl_argparse(opts, varargin) ;

% --------------------------------------------------------------------
%                         Prepare data
% --------------------------------------------------------------------
```

```
if exist(opts.imdbPath, 'file')
  imdb = load(opts.imdbPath) ;
else
% imdb = getMnistImdb(opts) ;
% mkdir(opts.expDir) ;
% save(opts.imdbPath, '-struct', 'imdb') ;
end

%%%%%%%% DEFINICION DE LA ARQUITECTURA DE LA RED %%%%%%%%%%

f=1/100 ;
net.layers = {} ;
%CAPA 1 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(5,5,1,32, 'single'), ...
                           'biases', zeros(1, 32, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 1 POOLING
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;
%CAPA 1 RELU PARA LA CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;

%CAPA 2 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,32,196, 'single'),...
                           'biases', zeros(1,196,'single'), ...
                           'stride', 3, ...
                           'pad', 0) ;
```

```
%CAPA 2 POOLING
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 3, ...
                           'pad', 0) ;
%CAPA 2 RELU PARA LA CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;

%CAPA 3 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(13,13,196,1536, 'single'),...
                           'biases', zeros(1,1536,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 3 RELU CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;
%CAPA 3 DROPOUT AL 50%
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;

%CAPA 4 “CONVOLUCION” (CAPA TOTALMENTE CONECTADA CON 2 SALIDAS)
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,1536,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 4 FUNCION ACTIVACION NEURONA SALIDA SOFTMAX Y LOGLOSS
net.layers{end+1} = struct('type', 'softmaxloss') ;

% --------------------------------------------------------------------
%                                                         Train
% --------------------------------------------------------------------
```

```
% Take the mean out and make GPU if needed
imdb.images.data = bsxfun(@minus, imdb.images.data, mean(imdb.images.data,4)) ;
if opts.train.useGpu
  imdb.images.data = gpuArray(imdb.images.data) ;
end

[net, info] = cnn_train(net, imdb, @getBatch, ...
    opts.train, ...
    'val', find(imdb.images.set == 3)) ;

% --------------------------------------------------------------------
function [im, labels] = getBatch(imdb, batch)
% --------------------------------------------------------------------
im = imdb.images.data(:,:,:,batch) ;
labels = imdb.images.labels(1,batch) ;

% --------------------------------------------------------------------
function imdb = getMnistImdb(opts)
% --------------------------------------------------------------------
files = {'train-images-idx3-ubyte', ...
         'train-labels-idx1-ubyte', ...
         't10k-images-idx3-ubyte', ...
         't10k-labels-idx1-ubyte'} ;

mkdir(opts.dataDir) ;
for i=1:4
  if ~exist(fullfile(opts.dataDir, files{i}), 'file')
    url = sprintf('http://yann.lecun.com/exdb/mnist/%s.gz',files{i}) ;
    fprintf('downloading %s\n', url) ;
    gunzip(url, opts.dataDir) ;
  end
end
```

```
f=fopen(fullfile(opts.dataDir, 'train-images-idx3-ubyte'),'r') ;
x1=fread(f,inf,'uint8');
fclose(f) ;
x1=permute(reshape(x1(17:end),28,28,60e3),[2 1 3]) ;

f=fopen(fullfile(opts.dataDir, 't10k-images-idx3-ubyte'),'r') ;
x2=fread(f,inf,'uint8');
fclose(f) ;
x2=permute(reshape(x2(17:end),28,28,10e3),[2 1 3]) ;

f=fopen(fullfile(opts.dataDir, 'train-labels-idx1-ubyte'),'r') ;
y1=fread(f,inf,'uint8');
fclose(f) ;
y1=double(y1(9:end)')+1 ;

f=fopen(fullfile(opts.dataDir, 't10k-labels-idx1-ubyte'),'r') ;
y2=fread(f,inf,'uint8');
fclose(f) ;
y2=double(y2(9:end)')+1 ;

imdb.images.data = single(reshape(cat(3, x1, x2),28,28,1,[])) ;
imdb.images.labels = cat(2, y1, y2) ;
imdb.images.set = [ones(1,numel(y1)) 3*ones(1,numel(y2))] ;
imdb.meta.sets = {'train', 'val', 'test'} ;
imdb.meta.classes = arrayfun(@(x)sprintf('%d',x),0:9,'uniformoutput',false) ;
```

## 3.3 Casos practicos MatConvNet

Aplicamos varias configuraciones con un componente de azar para ir acotando la configuración idónea en base a los mejores resultados. Vamos a tener en cuenta sobre todo el radio de tophat en el preprocesado, los Filtros Convolucionales, Relu y Dropout

*Caso 1*: 30 iteraciones con ReLU en todas las capas y con Dropout y con 32 filtros de partida.

```
% Define a network similar to LeNet
f=1/100 ;
net.layers = {} ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(5,5,1,32, 'single'), ...
                           'biases', zeros(1, 32, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,32,196, 'single'),...
                           'biases', zeros(1,196,'single'), ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(13,13,196,1536, 'single'),...
                           'biases', zeros(1,1536,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,1536,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'softmaxloss') ;

% --------------------------------------------------------------------
%                                                         Train
```

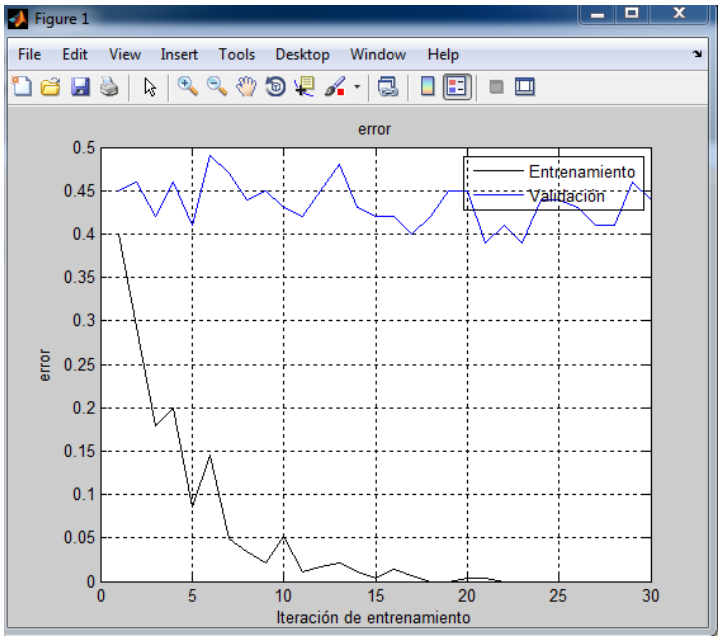



Figura 125 *Caso* 1 30 iteraciones con ReLU en todas las capas y con Dropout y con 32 filtros de partida.

*Caso 2* : 30 iteraciones con ReLU en todas las capas y con Dropout con 64 filtros:

Observamos que conseguimos alcanzar un mínimo error de validación de un 36% lo que en 25 iteraciones obtenemos porcentaje de acierto del 64% para acabar finalmente en el 55%

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|     11|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     16|     16|     16|    196|    196|    196|   1536|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|    n/a|     16|    n/a|    n/a|    196|    n/a|    n/a|   1536|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|    244|
c/g net KB|    0/7|    0/0|    0/0| 0/1767|    0/0|    0/0|0/795000|    0/0|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/778 MB
```

```
f=1/100 ;
net.layers = {} ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,1,64, 'single'), ...
                           'biases', zeros(1, 64, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [4 4], ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(10,10,64,128, 'single'),...
                           'biases', zeros(1,128,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [4 4], ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(24,24,128,512, 'single'),...
                           'biases', zeros(1,512,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,512,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'softmaxloss') ;
```

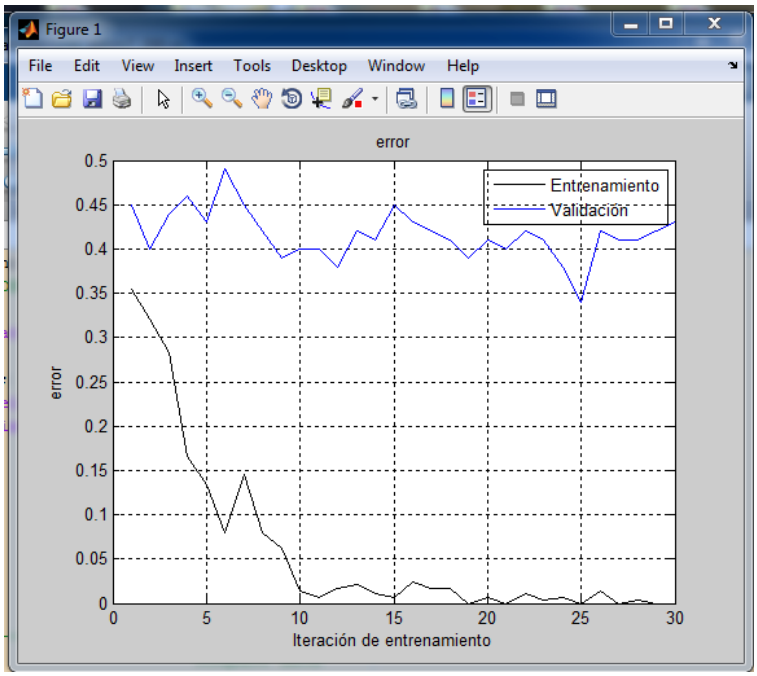



Figura 126 *Caso* 2 : 30 iteraciones con ReLU en todas las capas y con Dropout con 64 filtros

*Caso 3*: con las mimas condiciones probamos con 25 iteraciones con ReLU en todas las capas y con Dropout:

Probamos si con menos iteraciones da mejor resultado y comprobamos que no es así.

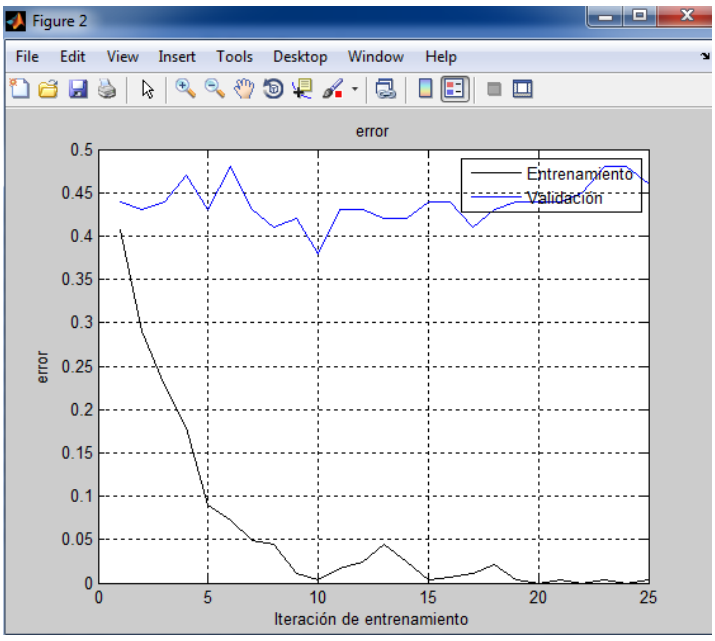



Figura 127 *Caso* 3: con las mimas condiciones probamos con 25 iteraciones con ReLU en todas las capas y con Dropout

*Caso 4*: con ReLU en todas las capas y con Dropout disminuyendo las ventanas de convolución:

```
I=1/100 ;
net.layers = {} ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(5,5,1,64, 'single'), ...
                           'biases', zeros(1, 16, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,64,196, 'single'),...
                           'biases', zeros(1,196,'single'), ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(13,13,196,1536, 'single'),..
                           'biases', zeros(1,1536,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,1536,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'softmaxloss') ;

% -------------------------------------------------------------------
%                                                              Train
```

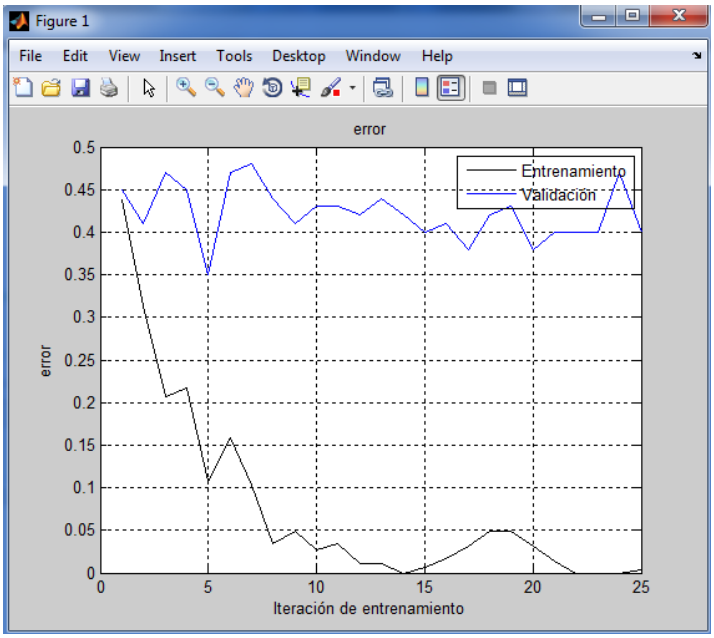



Figura 128 *Caso* 4 con ReLU en todas las capas y con Dropout disminuyendo las ventanas de convolución

*Caso 5*: con los mismos datos con ReLU en todas las capas y con Dropout:

Volvemos a repetir con 25 iteraciones:

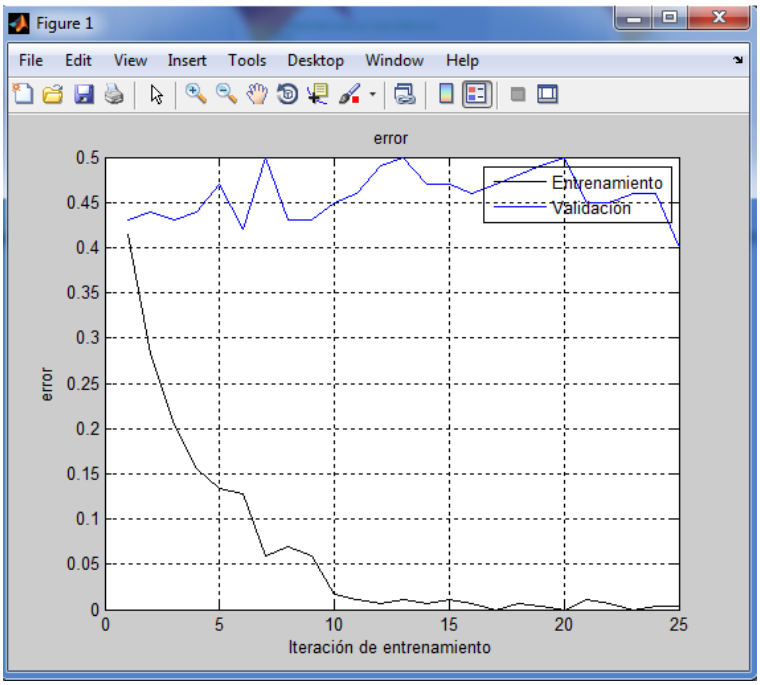



Figura 129 *Caso* 5 con los mismos datos que *caso* 4 con ReLU en todas las capas y con Dropout con 25 iteraciones

*Caso 6*: variando a 35 iteraciones con ReLU en todas las capas y con Dropout:

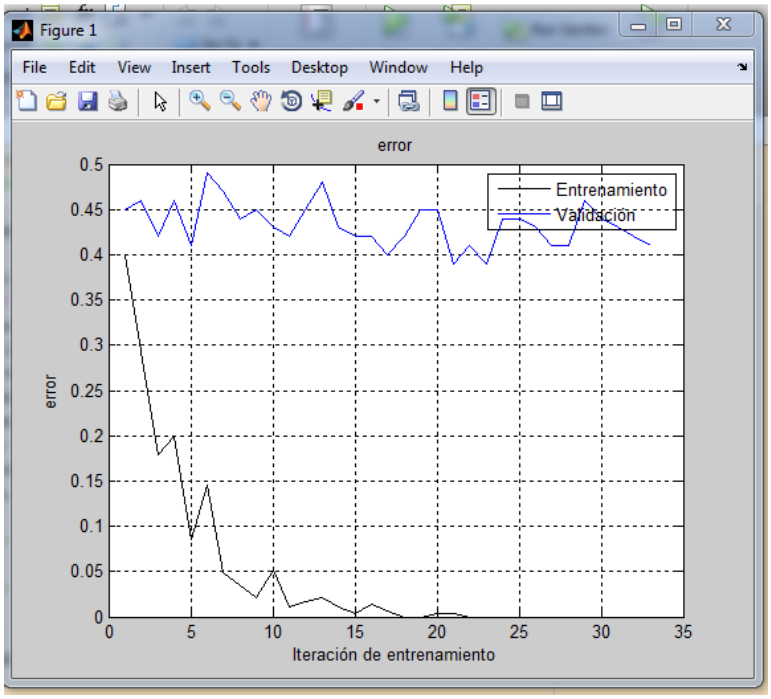



Figura 130 *Caso* 6 variando a 35 iteraciones con ReLU en todas las capas y con Dropout

*Caso 7*: 40 iteraciones con ReLU en todas las capas y con Dropout y 16 filtros de partida

```
% Define a network similar to LeNet
f=1/100 ;
net.layers = {} ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(5,5,1,16, 'single'), ...
                           'biases', zeros(1, 16, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,16,128, 'single'),...
                           'biases', zeros(1,128,'single'), ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(13,13,128,512, 'single'),.
                           'biases', zeros(1,512,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,512,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'softmaxloss') ;

% --------------------------------------------------------------------
%                                                              Train
```

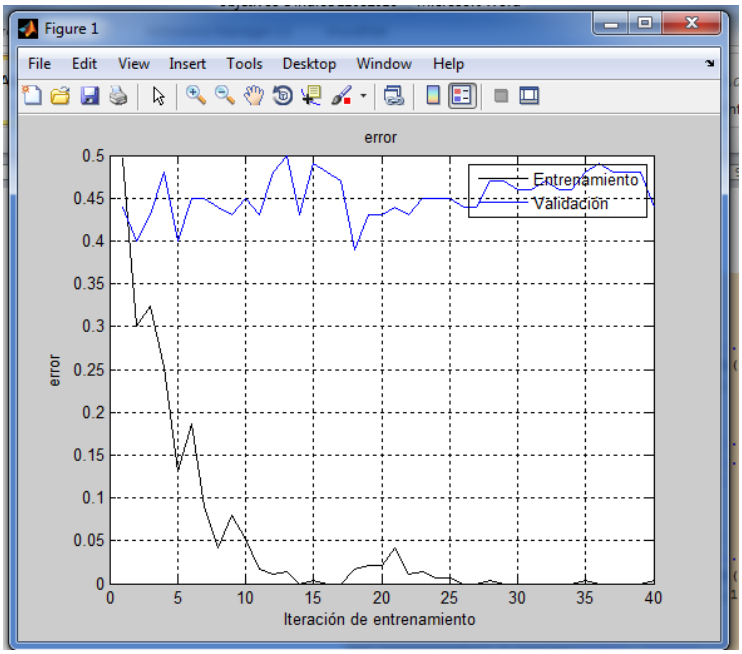



Figura 131 *Caso* 7 40 iteraciones con ReLU en todas las capas y con Dropout y 16 filtros de partida

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|     11|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     16|     16|     16|    128|    128|    128|    512|    512|    512|      2|      2|
  filt dim|      1|    n/a|    n/a|     16|    n/a|    n/a|    128|    n/a|    n/a|    512|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|    244|
c/g net KB|    0/7|    0/0|    0/0| 0/1154|    0/0|    0/0|0/173064|    0/0|    0/0|   0/16|    0/0|
total network CPU/GPU memory: 0.0/170 MB
```

*Caso 8*: con ReLU en todas las capas y con Dropout alterando tamaño de las ventanas convolucionales y filtros de salida.

```
% Define a network similar to LeNet
f=1/100 ;
net.layers = {} ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,1,128, 'single'), ...
                           'biases', zeros(1, 128, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(15,15,128,512, 'single'),...
                           'biases', zeros(1,128,'single'), ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 3, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(30,30,512,1024, 'single'),...
                           'biases', zeros(1,1024,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'relu') ;
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,1024,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
net.layers{end+1} = struct('type', 'softmaxloss') ;

% --------------------------------------------------------------------
%                                                                Train
% --------------------------------------------------------------------
```

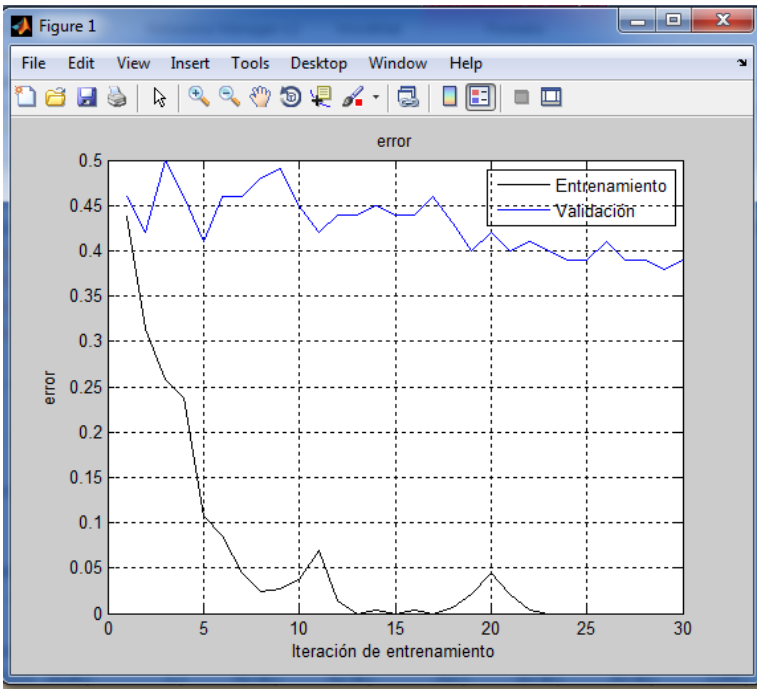



Figura 132 *Caso* 8 con ReLU en todas las capas y con Dropout alterando tamaño de las ventanas convolucionales y filtros de salida.

Vemos que la solución mejora , converge y que a partir de la iteración 27 estabiliza en el aprendizaje y ademas la validación mejora hasta llegar a la cifra del 37%

**Ahora vamos a probar las siguientes configuraciones para ver como actua ReLU y Dropout en el resultado:**

| Relu | Dropout |
|---|---|
| No | No |
| No | Si |
| Si | No |
| Si | Si |

Caso 9: Con ReLU en todas las capas, con Dropout y 32 filtros salida en primera capa

```
% Define a network similar to LeNet
f=1/100 ;
net.layers = {} ;
%CAPA 1 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(5,5,1,32, 'single'), ...
                           'biases', zeros(1, 32, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 1 POOLING
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;
%CAPA 1 RELU PARA LA CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;
%CAPA 2 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,32,196, 'single'),...
                           'biases', zeros(1,196,'single'), ...
                           'stride', 3, ...
                           'pad', 0) ;
%CAPA 2 POOLING
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 3, ...
                           'pad', 0) ;
%CAPA 2 RELU PARA LA CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;
%CAPA 3 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(13,13,196,1536, 'single'),...
                           'biases', zeros(1,1536,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 3 RELU CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;
%CAPA 3 DROPOUT AL 50%
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;
%CAPA 4 CONVOLUCION CONECTANDO CON 2 SALIDAS
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,1536,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
```

```
training: epoch 30: processing batch  15 of  29 ... 0.40 s (24.8 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  16 of  29 ... 0.28 s (36.0 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  17 of  29 ... 0.50 s (20.1 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  18 of  29 ... 0.34 s (29.4 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  19 of  29 ... 0.41 s (24.7 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  20 of  29 ... 0.29 s (35.0 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  21 of  29 ... 0.53 s (19.0 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  22 of  29 ... 0.37 s (26.9 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  23 of  29 ... 0.42 s (24.0 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  24 of  29 ... 0.28 s (35.7 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  25 of  29 ... 0.51 s (19.8 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  26 of  29 ... 0.34 s (29.6 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  27 of  29 ... 0.41 s (24.5 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  28 of  29 ... 0.28 s (36.0 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  29 of  29 ... 0.51 s (19.7 images/s) err 0.0 err5 0.0
validation: epoch 30: processing batch   1 of  10 ... 0.15 s (67.0 images/s) err 20.0 err5 0.0
validation: epoch 30: processing batch   2 of  10 ... 0.13 s (78.5 images/s) err 35.0 err5 0.0
validation: epoch 30: processing batch   3 of  10 ... 0.13 s (77.5 images/s) err 26.7 err5 0.0
validation: epoch 30: processing batch   4 of  10 ... 0.13 s (77.8 images/s) err 30.0 err5 0.0
validation: epoch 30: processing batch   5 of  10 ... 0.13 s (78.8 images/s) err 30.0 err5 0.0
validation: epoch 30: processing batch   6 of  10 ... 0.13 s (77.4 images/s) err 33.3 err5 0.0
validation: epoch 30: processing batch   7 of  10 ... 0.13 s (78.7 images/s) err 32.9 err5 0.0
validation: epoch 30: processing batch   8 of  10 ... 0.13 s (77.7 images/s) err 37.5 err5 0.0
validation: epoch 30: processing batch   9 of  10 ... 0.14 s (71.7 images/s) err 41.1 err5 0.0
validation: epoch 30: processing batch  10 of  10 ... 0.13 s (78.7 images/s) err 39.0 err5 0.0
Warning: Plot empty.
> In legend at 286
  In cnn_train at 231
  In cnn_ddsm7_390 at 90

ans =

    layers: {1x11 cell}

>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|     11|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|     32|    196|    196|    196|   1536|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|    n/a|     32|    n/a|    n/a|    196|    n/a|    n/a|   1536|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0|    0/0| 0/3531|    0/0|    0/0|0/795000|    0/0|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
>>
```

Figura 133 *Caso* 9 con ReLU en todas las capas , con Dropout y 32 filtros salida en primera capa

Caso 10: Con ReLU en todas las capas , sin Dropout y 32 filtros salida en primera capa:

Le quitamos el Dropout al 50%

```
%CAPA 3 DROPOUT AL 50%

net.layers{end+1} = struct('type', 'dropout', ...

                           'rate', 0.5) ;
```

```
validation: epoch 30: processing batch   1 of  10 ... 0.07 s (142.5 images/s) err 30.0 err5 0.0
validation: epoch 30: processing batch   2 of  10 ... 0.06 s (178.0 images/s) err 30.0 err5 0.0
validation: epoch 30: processing batch   3 of  10 ... 0.05 s (187.2 images/s) err 30.0 err5 0.0
validation: epoch 30: processing batch   4 of  10 ... 0.07 s (144.7 images/s) err 35.0 err5 0.0
validation: epoch 30: processing batch   5 of  10 ... 0.05 s (183.4 images/s) err 36.0 err5 0.0
validation: epoch 30: processing batch   6 of  10 ... 0.07 s (151.8 images/s) err 36.7 err5 0.0
validation: epoch 30: processing batch   7 of  10 ... 0.06 s (178.9 images/s) err 35.7 err5 0.0
validation: epoch 30: processing batch   8 of  10 ... 0.05 s (184.4 images/s) err 38.8 err5 0.0
validation: epoch 30: processing batch   9 of  10 ... 0.06 s (180.9 images/s) err 42.2 err5 0.0
validation: epoch 30: processing batch  10 of  10 ... 0.05 s (182.8 images/s) err 43.0 err5 0.0
```

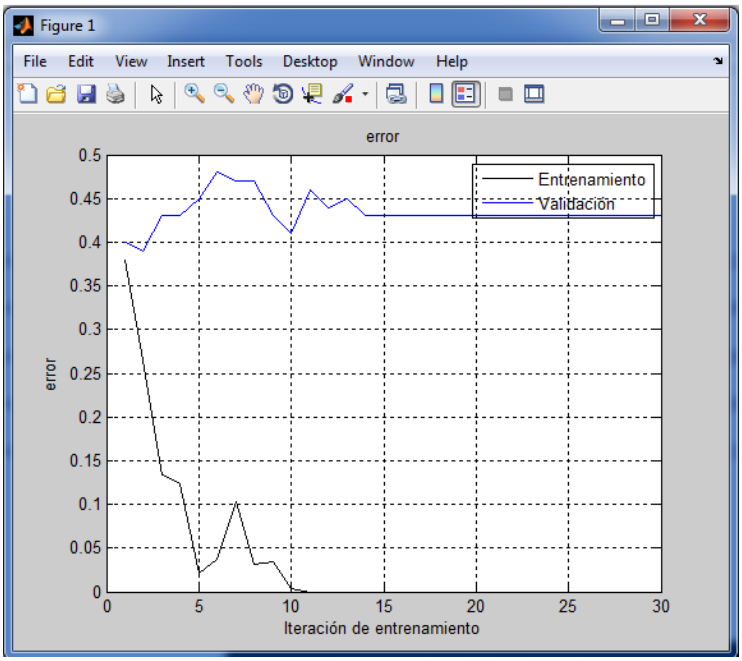



Figura 134 *Caso* 10 con ReLU en todas las capas , sin Dropout y 32 filtros salida en primera capa

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|     32|    196|    196|    196|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|    n/a|     32|    n/a|    n/a|    196|    n/a|   1536|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0|    0/0| 0/3531|    0/0|    0/0|0/795000|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
```

*Caso* 11: Sin Relu en capas intermedias, sin Dropout y 32 filtros salida en primera capa.

Quitamos las siguiente linea de codigo para las capas 1 y dos:

```
%CAPA RELU PARA LA CAPA SIGUIENTE

net.layers{end+1} = struct('type', 'relu') ;
```

```
training: epoch 30: processing batch  23 of  29 ... 0.51 s (19.7 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  24 of  29 ... 0.43 s (23.4 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  25 of  29 ... 0.54 s (18.6 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  26 of  29 ... 0.44 s (22.5 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  27 of  29 ... 0.59 s (16.9 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  28 of  29 ... 0.44 s (22.9 images/s) err 0.0 err5 0.0
training: epoch 30: processing batch  29 of  29 ... 0.53 s (18.9 images/s) err 0.0 err5 0.0
validation: epoch 30: processing batch   1 of  10 ... 0.26 s (38.9 images/s) err 50.0 err5 0.0
validation: epoch 30: processing batch   2 of  10 ... 0.27 s (37.7 images/s) err 40.0 err5 0.0
validation: epoch 30: processing batch   3 of  10 ... 0.17 s (58.3 images/s) err 43.3 err5 0.0
validation: epoch 30: processing batch   4 of  10 ... 0.17 s (58.1 images/s) err 47.5 err5 0.0
validation: epoch 30: processing batch   5 of  10 ... 0.17 s (59.6 images/s) err 42.0 err5 0.0
validation: epoch 30: processing batch   6 of  10 ... 0.17 s (59.9 images/s) err 45.0 err5 0.0
validation: epoch 30: processing batch   7 of  10 ... 0.17 s (57.7 images/s) err 41.4 err5 0.0
validation: epoch 30: processing batch   8 of  10 ... 0.18 s (54.4 images/s) err 45.0 err5 0.0
validation: epoch 30: processing batch   9 of  10 ... 0.11 s (92.9 images/s) err 48.9 err5 0.0
validation: epoch 30: processing batch  10 of  10 ... 0.11 s (91.5 images/s) err 48.0 err5 0.0
```

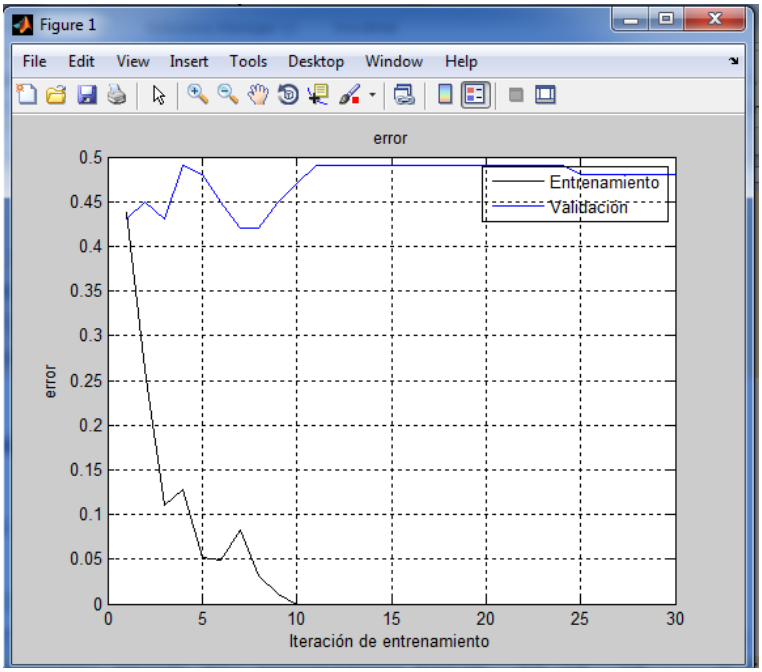



Figura 135 *Caso* 11 sin Relu en capas intermedias, sin Dropout y 32 filtros salida en primera capa

Se puede apreciar el pobre resultado en errores de validación al suprimir Relu en capa 1 y 2 y tambien dropout.

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|
      type|    cnv|  mpool|    cnv|  mpool|    cnv|   relu|    cnv|    cnv|
   support|    5x5|    2x2|    6x6|    3x3|  13x13|    1x1|    1x1|    1x1|
    stride|      1|      2|      3|      3|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|    196|    196|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|     32|    n/a|    196|    n/a|   1536|    n/a|
rec. field|      5|      6|     16|     28|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0| 0/3531|    0/0|0/795000|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
```

Caso 12: sin Relu en capas intermedias , con Dropout y 32 filtros salida en primera capa.

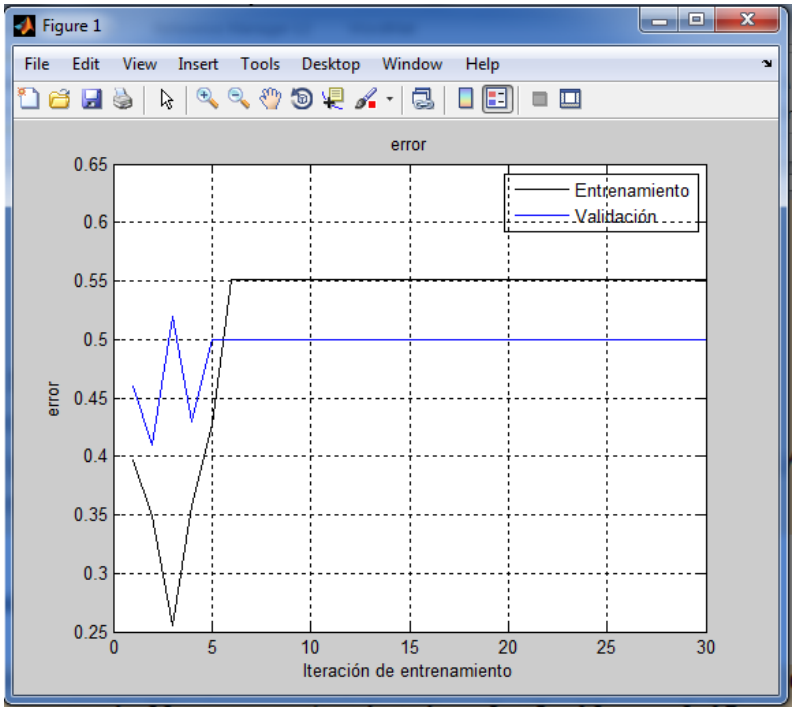



Figura 136 *Caso* 12 sin Relu en capas intermedias , con Dropout y 32 filtros salida en primera capa.

```
validation: epoch 30: processing batch   1 of  10 ... 0.18 s (55.1 images/s) err 100.0 err5 0.0
validation: epoch 30: processing batch   2 of  10 ... 0.17 s (57.8 images/s) err 100.0 err5 0.0
validation: epoch 30: processing batch   3 of  10 ... 0.18 s (54.1 images/s) err 100.0 err5 0.0
validation: epoch 30: processing batch   4 of  10 ... 0.11 s (90.1 images/s) err 100.0 err5 0.0
validation: epoch 30: processing batch   5 of  10 ... 0.11 s (89.6 images/s) err 100.0 err5 0.0
validation: epoch 30: processing batch   6 of  10 ... 0.11 s (89.8 images/s) err 83.3 err5 0.0
validation: epoch 30: processing batch   7 of  10 ... 0.11 s (88.2 images/s) err 71.4 err5 0.0
validation: epoch 30: processing batch   8 of  10 ... 0.11 s (89.2 images/s) err 62.5 err5 0.0
validation: epoch 30: processing batch   9 of  10 ... 0.11 s (89.6 images/s) err 55.6 err5 0.0
validation: epoch 30: processing batch  10 of  10 ... 0.11 s (89.1 images/s) err 50.0 err5 0.0
Warning: Plot empty.
> In legend at 286
  In cnn_train at 231
  In cnn_ddsm7_390_con_dopout_sin_relu at 90

ans =

    layers: {1x9 cell}
```

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|
      type|    cnv|  mpool|    cnv|  mpool|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    6x6|    3x3|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      3|      3|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|    196|    196|   1536|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|     32|    n/a|    196|    n/a|    n/a|   1536|    n/a|
rec. field|      5|      6|     16|     28|    244|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0| 0/3531|    0/0|0/795000|    0/0|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
```

Está claro despues de las pruebas realizadas que la **configuración optima es con ReLU y Dropout**. Por lo tanto probamos variaciones de esta con distintas iteraciones y con distinto radio de Tophat.

*Caso 13*: Con 80 iteraciones y tophat 7:

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|     11|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|     32|    196|    196|    196|   1536|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|    n/a|     32|    n/a|    n/a|    196|    n/a|    n/a|   1536|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0|    0/0| 0/3531|    0/0|    0/0|0/795000|    0/0|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
```

```
validation: epoch 80: processing batch   1 of  10 ... 0.07 s (143.8 images/s) err 30.0 err5 0.0
validation: epoch 80: processing batch   2 of  10 ... 0.05 s (183.6 images/s) err 40.0 err5 0.0
validation: epoch 80: processing batch   3 of  10 ... 0.05 s (184.8 images/s) err 40.0 err5 0.0
validation: epoch 80: processing batch   4 of  10 ... 0.05 s (185.5 images/s) err 42.5 err5 0.0
validation: epoch 80: processing batch   5 of  10 ... 0.06 s (155.9 images/s) err 42.0 err5 0.0
validation: epoch 80: processing batch   6 of  10 ... 0.05 s (182.0 images/s) err 43.3 err5 0.0
validation: epoch 80: processing batch   7 of  10 ... 0.05 s (184.8 images/s) err 38.6 err5 0.0
validation: epoch 80: processing batch   8 of  10 ... 0.05 s (185.2 images/s) err 43.8 err5 0.0
validation: epoch 80: processing batch   9 of  10 ... 0.05 s (185.1 images/s) err 45.6 err5 0.0
validation: epoch 80: processing batch  10 of  10 ... 0.05 s (185.9 images/s) err 45.0 err5 0.0
```

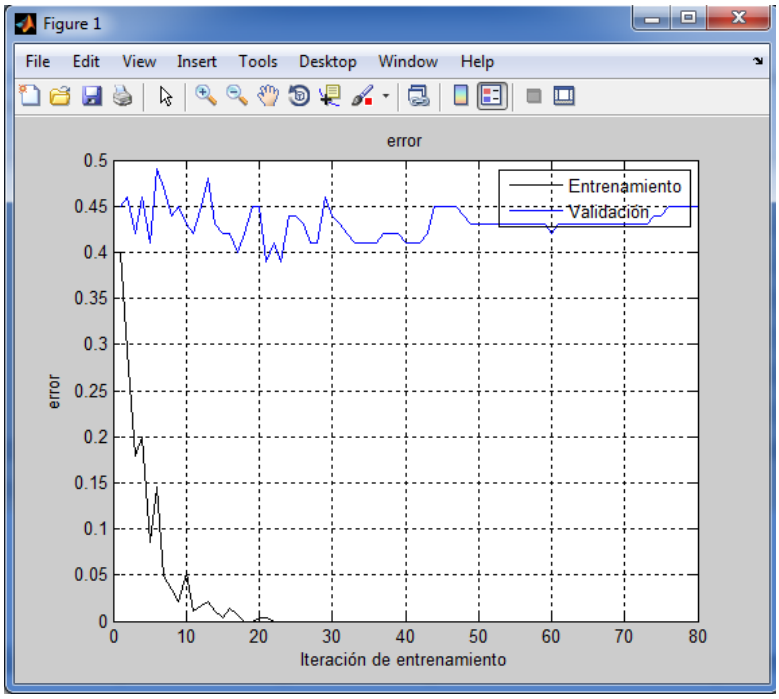



Figura 137 *Caso* 13 con 80 iteraciones y tophat 7:

Caso 14: con 27 iteraciones y tophat 7:

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|     11|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|     32|    196|    196|    196|   1536|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|    n/a|     32|    n/a|    n/a|    196|    n/a|    n/a|   1536|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0|    0/0| 0/3531|    0/0|    0/0|0/795000|    0/0|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
```

```
validation: epoch 27: processing batch   1 of  10 ... 0.07 s (143.5 images/s) err 20.0 err5 0.0
validation: epoch 27: processing batch   2 of  10 ... 0.05 s (184.4 images/s) err 30.0 err5 0.0
validation: epoch 27: processing batch   3 of  10 ... 0.05 s (186.2 images/s) err 30.0 err5 0.0
validation: epoch 27: processing batch   4 of  10 ... 0.05 s (183.8 images/s) err 32.5 err5 0.0
validation: epoch 27: processing batch   5 of  10 ... 0.05 s (186.0 images/s) err 34.0 err5 0.0
validation: epoch 27: processing batch   6 of  10 ... 0.06 s (174.4 images/s) err 36.7 err5 0.0
validation: epoch 27: processing batch   7 of  10 ... 0.06 s (175.9 images/s) err 32.9 err5 0.0
validation: epoch 27: processing batch   8 of  10 ... 0.05 s (186.4 images/s) err 37.5 err5 0.0
validation: epoch 27: processing batch   9 of  10 ... 0.05 s (182.5 images/s) err 41.1 err5 0.0
validation: epoch 27: processing batch  10 of  10 ... 0.05 s (183.7 images/s) err 41.0 err5 0.0
```

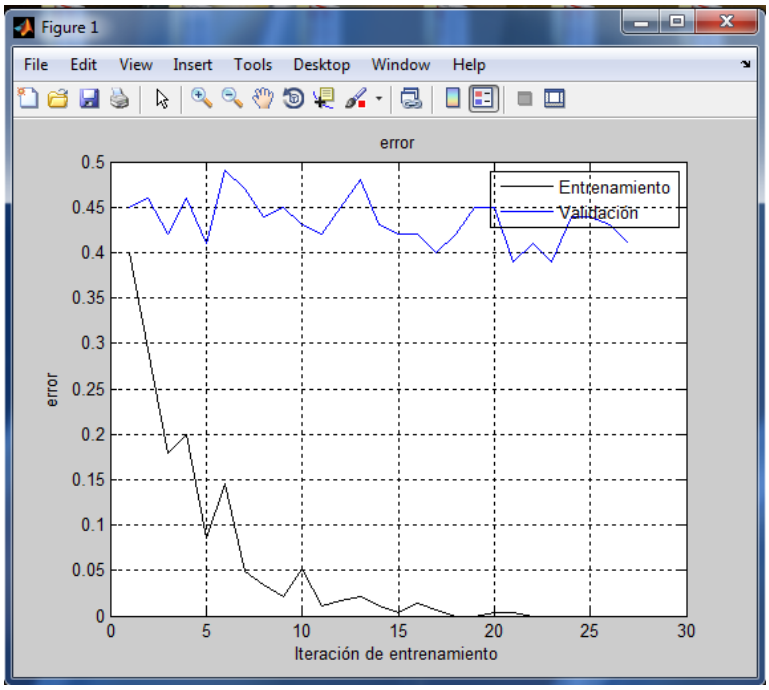



Figura 138 *Caso* 14 con 27 iteraciones y tophat 7

*Caso 15*: con 27 iteraciones y tophat radio 8 en vez de 7.

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|     11|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|     32|    196|    196|    196|   1536|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|    n/a|     32|    n/a|    n/a|    196|    n/a|    n/a|   1536|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0|    0/0| 0/3531|    0/0|    0/0|0/795000|    0/0|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
```

```
training: epoch 27: processing batch  23 of  29 ... 0.23 s (44.3 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  24 of  29 ... 0.23 s (44.2 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  25 of  29 ... 0.23 s (44.2 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  26 of  29 ... 0.23 s (44.3 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  27 of  29 ... 0.23 s (44.3 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  28 of  29 ... 0.23 s (43.8 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  29 of  29 ... 0.23 s (44.1 images/s) err 0.0 err5 0.0
validation: epoch 27: processing batch   1 of  10 ... 0.07 s (137.6 images/s) err 50.0 err5 0.0
validation: epoch 27: processing batch   2 of  10 ... 0.06 s (173.2 images/s) err 45.0 err5 0.0
validation: epoch 27: processing batch   3 of  10 ... 0.07 s (150.4 images/s) err 40.0 err5 0.0
validation: epoch 27: processing batch   4 of  10 ... 0.06 s (171.5 images/s) err 45.0 err5 0.0
validation: epoch 27: processing batch   5 of  10 ... 0.06 s (173.6 images/s) err 40.0 err5 0.0
validation: epoch 27: processing batch   6 of  10 ... 0.06 s (173.5 images/s) err 41.7 err5 0.0
validation: epoch 27: processing batch   7 of  10 ... 0.06 s (175.6 images/s) err 41.4 err5 0.0
validation: epoch 27: processing batch   8 of  10 ... 0.06 s (174.5 images/s) err 43.8 err5 0.0
validation: epoch 27: processing batch   9 of  10 ... 0.06 s (172.9 images/s) err 44.4 err5 0.0
validation: epoch 27: processing batch  10 of  10 ... 0.06 s (175.1 images/s) err 46.0 err5 0.0
```

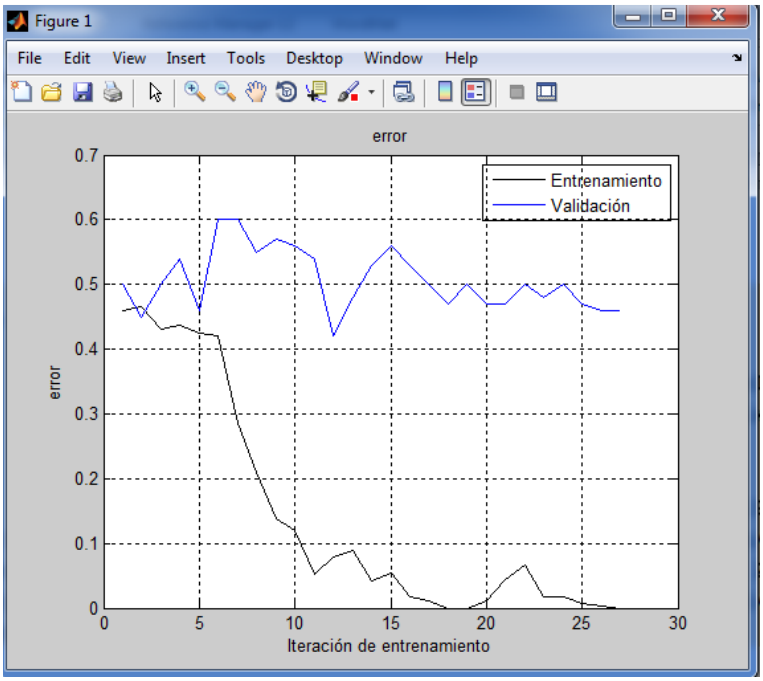



Figura 139 *Caso* 15 con 27 iteraciones y tophat radio 8 en vez de 7

Caso 16: con 27 iteraciones y tophat radio 6.

```
%CAPA 1 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(5,5,1,32, 'single'), ...
                           'biases', zeros(1, 32, 'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 1 POOLING
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;
%CAPA 1 RELU PARA LA CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;
%CAPA 2 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(6,6,32,196, 'single'),...
                           'biases', zeros(1,196,'single'), ...
                           'stride', 3, ...
                           'pad', 0) ;
%CAPA 2 POOLING
net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [3 3], ...
                           'stride', 3, ...
                           'pad', 0) ;
%CAPA 2 RELU PARA LA CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;
%CAPA 3 CONVOLUCION
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(13,13,196,1536, 'single'),...
                           'biases', zeros(1,1536,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 3 RELU CAPA SIGUIENTE
net.layers{end+1} = struct('type', 'relu') ;
%CAPA 3 DROPOUT AL 50%
net.layers{end+1} = struct('type', 'dropout', ...
                           'rate', 0.5) ;
%CAPA 4 CONVOLUCION CONECTANDO CON 2 SALIDAS
net.layers{end+1} = struct('type', 'conv', ...
                           'filters', f*randn(1,1,1536,2, 'single'),...
                           'biases', zeros(1,2,'single'), ...
                           'stride', 1, ...
                           'pad', 0) ;
%CAPA 4 FUNCION ACTIVACION NEURONA SALIDA SOFTMAX
```

```
training: epoch 27: processing batch  21 of  29 ... 0.23 s (43.5 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  22 of  29 ... 0.23 s (43.1 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  23 of  29 ... 0.23 s (43.6 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  24 of  29 ... 0.23 s (43.5 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  25 of  29 ... 0.24 s (41.5 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  26 of  29 ... 0.24 s (42.4 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  27 of  29 ... 0.23 s (43.0 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  28 of  29 ... 0.24 s (42.2 images/s) err 0.0 err5 0.0
training: epoch 27: processing batch  29 of  29 ... 0.23 s (42.9 images/s) err 0.0 err5 0.0
validation: epoch 27: processing batch   1 of  10 ... 0.07 s (134.1 images/s) err 50.0 err5 0.0
validation: epoch 27: processing batch   2 of  10 ... 0.07 s (145.8 images/s) err 45.0 err5 0.0
validation: epoch 27: processing batch   3 of  10 ... 0.10 s (97.9 images/s) err 40.0 err5 0.0
validation: epoch 27: processing batch   4 of  10 ... 0.06 s (162.3 images/s) err 45.0 err5 0.0
validation: epoch 27: processing batch   5 of  10 ... 0.06 s (175.5 images/s) err 40.0 err5 0.0
validation: epoch 27: processing batch   6 of  10 ... 0.06 s (173.8 images/s) err 41.7 err5 0.0
validation: epoch 27: processing batch   7 of  10 ... 0.06 s (175.6 images/s) err 41.4 err5 0.0
validation: epoch 27: processing batch   8 of  10 ... 0.06 s (176.2 images/s) err 43.8 err5 0.0
validation: epoch 27: processing batch   9 of  10 ... 0.06 s (164.5 images/s) err 44.4 err5 0.0
validation: epoch 27: processing batch  10 of  10 ... 0.06 s (169.6 images/s) err 46.0 err5 0.0
```

```
>> vl_simplenn_display(ans);
     layer|      1|      2|      3|      4|      5|      6|      7|      8|      9|     10|     11|
      type|    cnv|  mpool|   relu|    cnv|  mpool|   relu|    cnv|   relu|dropout|    cnv|    cnv|
   support|    5x5|    2x2|    1x1|    6x6|    3x3|    1x1|  13x13|    1x1|    1x1|    1x1|    1x1|
    stride|      1|      2|      1|      3|      3|      1|      1|      1|      1|      1|      1|
       pad|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|      0|
   out dim|     32|     32|     32|    196|    196|    196|   1536|   1536|   1536|      2|      2|
  filt dim|      1|    n/a|    n/a|     32|    n/a|    n/a|    196|    n/a|    n/a|   1536|    n/a|
rec. field|      5|      6|      6|     16|     28|     28|    244|    244|    244|    244|    244|
c/g net KB|   0/13|    0/0|    0/0| 0/3531|    0/0|    0/0|0/795000|    0/0|    0/0|   0/48|    0/0|
total network CPU/GPU memory: 0.0/780 MB
```

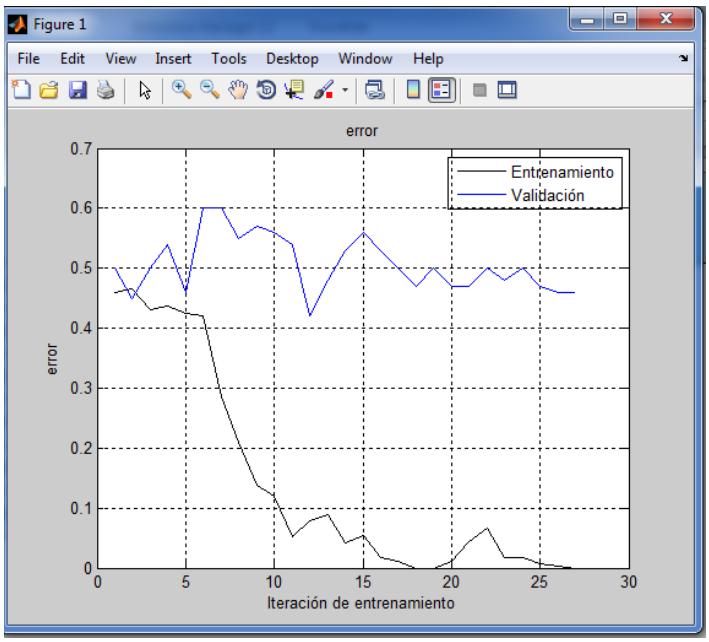



Figura 140 *Caso* 16 con 27 iteraciones y tophat radio 6

Descartamos tophat de 3 y de 5 porque el preprocesado es malo dejando imágenes en blanco como se aprecia en las imágenes siguientes:

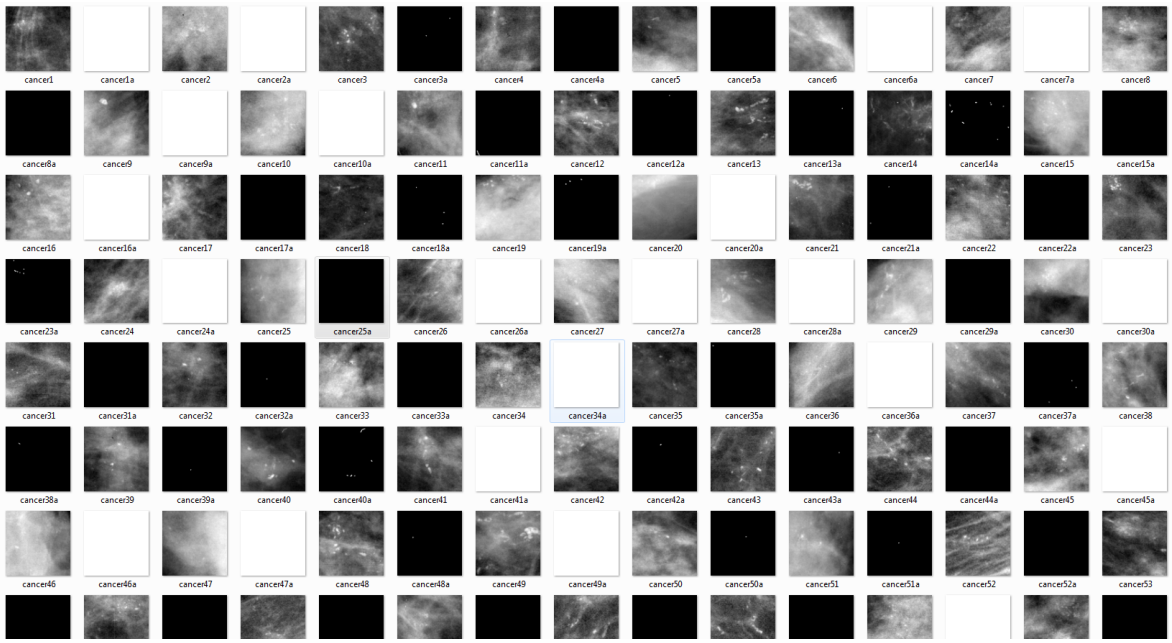


Figura 141 Imágenes tratadas con preprocesado Tophat de 3

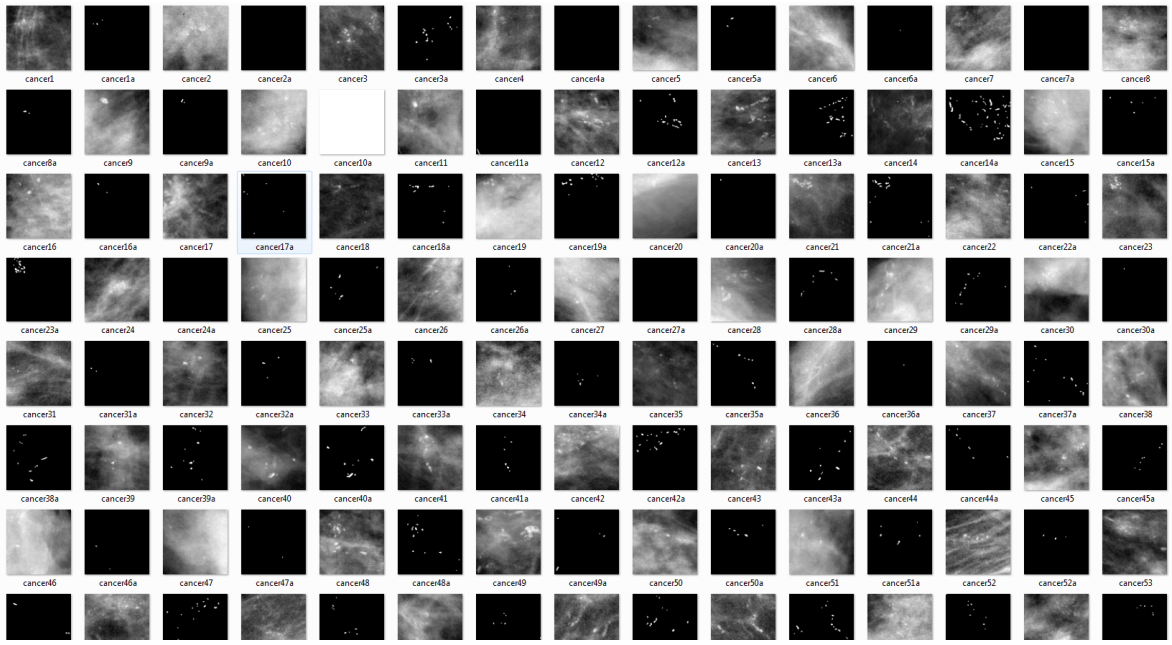


Figura 142 Imágenes tratadas con preprocesado Tophat de 5

# Capítulo 4 Conclusión modelo convolucional.

| | Capas de convolución | Capas de pooling | ReLU | Dropout | Numero de Iteraciones | Filtros primera capa | Filtros segunda capa | Filtros tercera capa | Cuarta capa salida | Error de entrenamiento | Error de validación | Error Minimo Validación | Acierto % | Acierto maximo % | Tophat |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Caso 1 | 4 | 2 | 3 | si , 0,5 | 30 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 44 | 39 | 56 | 61 | 7 |
| Caso 2 | 4 | 2 | 3 | si, 0,5 | 30 | 6,6,1,64 | 10,10,64,128 | 24,24,128,512 | 1,1,512,2 | 0 | 43 | 34 | 57 | 66 | 7 |
| Caso 3 | 4 | 2 | 3 | si,0,5 | 25 | 6,6,1,64 | 10,10,64,128 | 24,24,128,512 | 1,1,512,2 | 0 | 46 | 38 | 54 | 62 | 7 |
| Caso 4 | 4 | 2 | 3 | si,0,5 | 25 | 5,5,1,64 | 6,6,64,196 | 13,13,196,1536 | 1,1,1536,2 | 1 | 40 | 35 | 60 | 65 | 7 |
| Caso 5 | 4 | 2 | 3 | si,0,5 | 35 | 5,5,1,64 | 6,6,64,196 | 13,13,196,1536 | 1,1,1536,2 | 1 | 40 | 40 | 60 | 60 | 7 |
| Caso 6 | 4 | 2 | 3 | si,0,5 | 35 | 5,5,1,64 | 6,6,64,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 42 | 38 | 58 | 62 | 7 |
| Caso 7 | 4 | 2 | 3 | si,0,5 | 40 | 5,5,1,16 | 6,6,16,128 | 13,13,128,512 | 1,1,512,2 | 1 | 44 | 39 | 56 | 61 | 7 |
| Caso 8 | 4 | 2 | 3 | si,0,5 | 30 | 6,6,1,128 | 15,15,128,512 | 30,30,512,1024 | 1,1,1024,2 | 0 | 37 | 37 | 63 | 63 | 7 |
| Caso 9 | 4 | 2 | 3 | si,0,5 | 30 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 39 | 26 | 61 | 61 | 7 |
| Caso 10 | 4 | 2 | 3 | no | 30 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 43 | 42 | 57 | 58 | 7 |
| Caso 11 | 4 | 2 | 1 | no | 30 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 48 | 43 | 52 | 57 | 7 |
| Caso 12 | 4 | 2 | 2 | si,0,5 | 30 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 55 | 50 | 42 | 50 | 58 | 7 |
| Caso 13 | 4 | 2 | 2 | si,0,5 | 80 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 45 | 38 | 55 | 62 | 7 |
| Caso 14 | 4 | 2 | 3 | si,0,5 | 27 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 41 | 38 | 59 | 62 | 7 |
| Caso 15 | 4 | 2 | 3 | si,0,5 | 27 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 46 | 43 | 54 | 57 | 8 |
| Caso 16 | 4 | 2 | 3 | si,0,5 | 27 | 5,5,1,32 | 6,6,32,196 | 13,13,196,1536 | 1,1,1536,2 | 0 | 46 | 43 | 54 | 57 | 6 |

Figura 143 Resultado de red Matconvnet

Se prueban muchas configuraciones diferentes con el objetivo de encontrar el número ideal de capas, cuantas neuronas incluimos por capa, si es beneficiosa la aplicación de Relu en todas las capas o tan solo en la última, probamos también la aplicación de la técnica de dropout con distintos porcentajes para evitar sobre-entrenamiento y en base a estas pruebas, una vez obtenemos los resultados debemos analizar con que parámetros se comporta mejor nuestra red para ir descartando ciertas configuraciones y afinarla lo máximo posible con variaciones de los parámetros adecuados.

Hasta ahora, hemos comprobado que el mejor comportamiento en cuanto a rendimiento y resultado se produce en la prueba numero 8 ya que vemos que la solución converge. Partimos de la configuración de aplicar Relu en todas las capas intermedias, dropout al 50%, pooling,

ventana de convolución de 6x6 y 128 filtros de salida en la primera capa obteniendo un resultado final de un 63% de acierto. Hay que tener en cuenta que nos movemos con 390 imágenes que ya es una cantidad a tener en cuenta y que no podemos probar configuraciones deseables por falta de memoria CPU y GPU.

En el caso numero 2 obtenemos un acierto del 66% en la iteración de entrenamiento nº 25 con un error de entrenamiento del 0% pero sobre-entrena en las 5 iteraciones restantes estropeando el resultado final.

Estos resultados son alentadores y animan a probar distintas combinaciones a partir de la configuración que ha resultado mejor. Recordemos en las referencias analizadas en los capítulos de teoría que los expertos radiólogos humanos presentan tasas de acierto medidas de forma retrospectiva de un 75%. En este sentido, una red convolucional relativamente simple si la comparamos con el conocimiento de un radiólogo de 20 años de experiencia produce resultados notablemente interesantes para servir de ayuda en programas de cribado masivo a la población. Las 390 imágenes de nuestra base de datos DDSM son casos reales, con ruido y alto de nivel de dificultad y no es una base de datos preparada para un experimento particular, sino para representar la casuística de un programa de cribado real.

# Capítulo 5 Nuevas líneas de investigación y aplicaciones practicas.

El aprendizaje profundo, representado en nuestro trabajo practico por la técnica de redes neuronales convolucionales tiene **un largo desarrollo** por delante ya que esta técnica se encuentra en sus comienzos y la **mayor limitación** en la que se halla este campo es la tecnología con la que cuenta el investigador; pero solo es cuestión de tiempo sortear estos límites ya que inexorablemente la evolución tecnológica avanza a pasos agigantados y va a ser inevitable que dentro de poco tiempo se popularice como ocurrió con otros sistemas. Cuando esto ocurra y se ponga al alcance de muchos, la probabilidad de que ocurran genialidades se hará mucho mayor y **aparecerán desarrollos disruptivos**, ya se están notando en campos como la conducción autónoma de vehículos. Ya sabemos porque lo hemos comentado anteriormente que se trabaja actualmente en el **desarrollo de neurochips** que facilitará enormemente los avances en redes difíciles de implementar hoy en día.

Por ello **proponemos como nuevas líneas de investigación** el estudio con **una BBDD más amplia** porque el resultado de cualquier proyecto que utilice redes CNN no es trivial. Si ha funcionado para *x* imágenes no significa que el resultado deba ser igual de óptimo para *x+1,* ya que es necesario un refinamiento de la red y por lo tanto en esta línea se propone un sistema de red de mayor profundidad que demande mejores tarjetas gráficas como la tarjeta profesional Tesla K40.

| GPUS TESLA PARA ESTACIONES DE TRABAJO | | |
|---|---|---|
| **Características fundamentales** | **Tesla K40** | **Tesla K20** |
| **Pico de rendimiento de operaciones en coma flotante de doble precisión** | 1.43 Tflops | 1.17 Tflops |
| **Pico de rendimiento de operaciones en coma flotante de precisión simple** | 4.29 Tflops | 3.52 Tflops |
| **Ancho de banda de memoria (ECC desactivada)** | 288 GBytes/s | 208 GBytes/s |
| **Cantidad de memoria (GDDR5)** | 12 GB | 5 GB |
| **Núcleos CUDA** | 2880 | 2496 |

Figura 144 Caracteristicas GPU TESLA

Existen además tarjetas que pueden facilitar estos trabajos son la Gforce GTX 980M o la GTX880M de 8Gb de memoria dedicada, para uso en ordenadores portátiles.

Otra propuesta interesante es realizar un entorno grafico que permita el **Telediagnostico** de manera que la aplicación a realizar se encuentre instalada en un servidor y se analice en tiempo real la mamografía enviada a tal efecto por el cliente dando respuesta inmediata del diagnóstico, así como otros datos interesantes solicitados bajo demanda como puedan ser estadísticas basadas en la información de los datos incorporados y acumulados en la propia BBDD.

Nuestro trabajo evalúa zonas de interés ya preseleccionadas. Se propone un sistema que **segmente automáticamente** zonas de interés analizando e identificando de manera autónoma estas áreas y construyendo con ellas una nueva BBDD que cumpla los requisitos idóneos para su estudio.

Como es tan amplio el ámbito de aplicación de las redes neuronales convolucionales, casi cualquier idea de procesamiento de imágenes podría caber en un futuro próximo, como

trabajar en **proyectos de cooperación de redes convolucionales** para aunar esfuerzos de inteligencia y computacionales en lo que podríamos llamar multiredes convolucionales, como alternativa más elegante quizá al *dropout* que utilizamos en nuestro desarrollo. Quizás se podría **evitar la aleatoriedad de las capas** y someterlas a valores fijos preentrenados y ayudar con técnicas de inteligencia artificial y de la "programación convencional" asistiéndola para dotarla de mayor capacidad de generalización, **condicionando ciertas conductas no aprendidas** tal como ocurre con un ser humano que además de aprender sabemos que se le puede programar y en esta línea, aplicando cierto tono de filosofía citamos a Inmanuel Kant que propuso que "nuestro conocimiento del mundo exterior depende de nuestras formas de percepción" y extrapolándolo **se podrá enseñar a la red emulando** cualquier idea cuyo patrón sea posible en su reconocimiento por un ser inteligente **ayudado de sentidos artificiales como la visión, el olfato, gusto, tacto y audición**. Es decir si somos capaces de replicar los cinco sentidos de forma artificial ya tenemos los sensores para detectar los patrones y partimos de la base que todo lo que sigue un patrón se puede enseñar. Es cuestión de tiempo. **Un posible paso** en este sentido sería usar el sistema que hemos desarrollado combinándolo con otras entradas a la red neuronal que cojan metadatos de las pacientes, como edad, hábitos, historial clínico, o respuesta a cuestionarios sobre datos que puedan descubrir correlaciones ocultas con el cáncer de mama.

Pensemos que ahora las redes se centran en detectar generalidades: caras mediante clasificaciones genéricas, para entendernos pongamos un ejemplo: ¿es una cara de perro o es una cara de gato? pero por qué no vamos un paso más allá y planteamos detectar la de un perro concreto en distintas fases de su tiempo, de sus distintos estados anímicos, humor , cambios de imagen etc., **tal como lo haría su amo entre miles de ellos** o detectar una cara de persona aunque cambien ciertos rasgos como el pelo, la barba, la edad, o porque no distinguir las imágenes de dos gemelos cuya apariencia al ojo humano es idéntica pero que con aprendizaje profundo sabría **detectar cualquier mínima diferencia**, en fin, que pronto la **búsqueda inteligente de imágenes** concretas será posible en un buscador como google o

cualquier otro, que cualquier día aparecerá como algo innovador y entonces podremos buscar mediante imágenes (si la protección de datos lo permite), todo lo relativo a amigos, conocidos o recuerdos de la infancia de los que ya perdimos la pista y que guardamos una imagen de entonces que enseñarle a la red para que nos ayude mostrándonos todos los datos disponibles de la que es poseedora y ¿por qué no **pagar nuestros servicios o antojos a través de la red con mayor seguridad** reconociendo patrones de estafa y librándonos de un buen susto? Todo un mundo por trabajar.



# Referencias


[1] Gandur, N. (s.a.). Manual de enfermería oncológica. Recuperado el 23 de septiembre de 2014

[2] ESTADÍSTICAS SANITARIAS MUNDIALES 2014 editado por la organización mundial de la salud http://apps.who.int/iris/bitstream/10665/112817/1/WHO_HIS_HSI_14.1_spa.pdf?ua=1&ua=1

[3] Real Academia Española de la Lengua

[4] Sanitas http://www.sanitas.es/sanitas/seguros/es/particulares/biblioteca-de-salud/cancer/metastasis.html

[5] Manual CTO de Medicina y Cirugía, tomo I, ISBN 84-930264-3-3.

[6] SEOM http://www.seom.org/en/prensa/el-cancer-en-espanyacom/105460-el-cancer-en-espana-2016?start=3#content

[7] INE

[8] Coleman et al., 2008. Cáncer de mama: prevención y control. http://www.who.int/topics/cancer/breastcancer/es/index1.html

[9] OMS http://www.who.int/topics/cancer/breastcancer/es/

[10] Asociación española contra el cáncer

https://www.aecc.es/SobreElCancer/CancerPorLocalizacion/CancerMama/Paginas/cancerdemama.aspx

[11] Patrick J. Lynch, medical illustrator

[12] *M.D. ANDERSON Y EL CÁNCER / ¿QUÉ ES EL CÁNCER?*. La Fundación M.D. Anderson España - Centro Oncológico. 2007.

[13] Teleformación en salud. Autor Dr. J Santisteban. Editores: Dr. J Peinado, Sr. Victor Roque. Lima, Perú. © 2,001 EHAS

[14] Geicam investigación de cáncer de mama http://geicam.org/colabora/?gclid=CLy09_it7cwCFWMq0wodQwEOzQ

[15] incidencia: Globocam 2012 http://globocam.iarc.fr

[16] Las cifras del cáncer 2014. SEOM Sociedad española de oncología medica

[17] Asco y American Society of Clinical Oncology http://www.cancer.net/about-us

[18] Breast cáncer: http://www.breastcancer.org/es/sintomas/diagnostico/estadios

[19] Asociación española contra el cáncer

[20] Detección del Cáncer de Mama: Estado de la Mamografía en México , Brandan y Vi l laseñor, Cancerología 1 (2006) : 147-162

[21] Medicina legal en patología mamaria By Santiago Delgado Bueno, Armando Tejerina, Armando Tejerina Gómez ed Diaz de santos ISBN:84-7978-513-6 M51478 2001

[22] http://www.breastcancer.org/es/noticias-investigacion/20130419

[23] http://www.serimagmedia.com/redes-neuronales-convolucionales-la-replica-en-maquina-del-aprendizaje-humano/

[24] http://tecnologia.elpais.com/tecnologia/2015/11/09/actualidad/1447090931_239441.html

[25] http://leenissen.dk/fann/wp/

[26] http://www.scielo.cl/scielo.php?pid=S0717-75262013000500010&script=sci_arttext

[27] https://es.wikipedia.org/wiki/Modelado_de_informaci%C3%B3n_de_construcci%C3%B3n

[28] https://es.wikipedia.org/wiki/Historia_del_dise%C3%B1o_asistido_por_computadora

[29] http://www.revistasbolivianas.org.bo/scielo.php?pid=S3333-77772014000100020&script=sci_arttext

[30] https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos

[31] https://es.mathworks.com/discovery/redes-neuronales.html?s_eid=ppc_25494184762&q=redes%20neuronales

[32] Precesado de señales biomédicas isbn 978-84-8427-541-1 Pedro A. Carrion perez y otros

[33] http://www.utm.mx/edi_anteriores/temas47/2Notas_47_1-Analisis_de_imagenes_de_mamog.pdf

[34] http://www.monografias.com/trabajos93/extraccion-caracteristicas-analisis-imagenes-mamografias/extraccion-caracteristicas-analisis-imagenes-mamografias.shtml

[35] EGAN,R.L., 1960. "Experience with Mammography in A Tumor Institution - Evaluation of 1,000 Studies", Radiology, 75, S. 894-900.

[36] Raffle AE, Muir Gray JA. Preface in: Screening: evidence and practice. Oxford: Oxford University Press, 2007.

[37] Calonge N. Past, present and future issues for the US Preventive Services Task Force. Conferencia inaugural. Primera Conferencia Española de Prevención y Promoción de la Salud en la Práctica Clínica. Madrid

[38] Berrington de González A, Reeves G. Mammographic screening before age 50 years in the UK: comparison of the radiation risks with the mortality benefits. Br J Cancer 2005;93:590-6

[39] Sackett D. The arrogance of Preventive Medicine. CMAJ 2002;167:363-4

[40] Gilbert Welch H. Screening mammography - A long run for a Short slide? N Engl J Med 2010;363:1276-9.

[41] MENCATTINI,A., SALMERI,M., LOJACONO,R., FRIGERIO,M., CASELLI,F., 2008. "Mammographic images enhancement and denoising for breast cancer detection using dyadic wavelet processing", IEEE Transactions on Instrumentation and Measurement, 57, S. 1422-1430.

[42] http://www-rayos.medicina.uma.es/Rmf/Tesis/Tesis_Doctoral_Raul_Mata.pdf

[43] Shih-Chung B. Lo Huai Li, K. J. Ray Liu. Fractal modeling and segmentation for the enhancement of microcalcifications in digital mammograms.IEEE Transactions on Medial Imaging, 16(6):785-798, 1997

[44] Rangaraj M. Rangayyan Joseph Edward Leo Desautels, William Mark Morrow, Raman Bhalachandra Paranjape. Region-based constrast enhancement of mammograms. IEEE Transactions on Medial Imaging, 11(3):392-405,1992

[45] A. K. Jain. Fundamental of Digital Imaging Processing Academic, 1996

[46] Nava E. Mata, R. and Sendra F. Microcalcification detection using multiresolution techniques. Proceedings of the IEEE International Conference on Pattern Recognition ICPR 2000, 4:344-348, 2000.

[47] Nava E. Mata R. and Sendra F. Análisis multiescala de mamografías y localización de zonas sospechosas de contener clusters.II Jornadas de Radiología Digital, 2001. Torremolinos

[48] Yasemín Yardimci M.Ña Gürcan and A. Enis Cetin. Automated detenction and enhancement of microcalcification in mammograms using nonlinear subband decomposition,ICASSP, 1:157-164, 1997

[49] Rangaraj M. Rangayyan Joseph Edward Leo Desautels William Mark Morrow, Raman Bhalachandra Paranjape. Region-based constrast enhancement of mammograms.IEEE Transactions on Medial Imaging, 11(3):392-405,1992

[50] Johann Friedrich Desaga Joachim Dengler, Sabina Behrens. Segmentation of microcalcifications in mammograms.IEEE Transactions on Medial Imaging,12(4):634-642, 1993.

[51] Shih-Chung B. Lo Huai Li, K. J. Ray Liu. Fractal modeling and segmentation for the enhancement of microcalcifications in digital mammograms.IEEE Transactions on Medial Imaging, 16(6):785-798, 1997

[52] Farzin Aghdasi B. Aydin Alaylioglu. A wavelet neural network for the detection of microcalcifications in multiscale enhanced digitised mammograms.AFRICON IEEE, 1(1):389-394, 1999.

[53] Rita I. Freimanis Heng-Da Cheng, Yui Man Lui. A novel approach to microcalcification detection using fuzzy logic technique.IEEE Transactions on Medial Imaging, 17(3):442-450, 1998

[54] Ruth English Ralph Highnam, Mike Brady. Detecting lm-screen artifacts in mammography using a model-based approach.IEEE Transactions on Medial Imaging, 18(10):1016-1024, 1999.

[55] Tratamiento de señales digitales mediante wavelets y su uso con Matlab By Félix Martínez Giménez, Alfredo Peris Manguillot, Francisco Ródenas Escribá ISBN 978-84-8454-387-9. 2004

[56] http://www.editorial-club-universitario.es/pdf/1036.pdf

[57] Metodo de distorsiones de la arquitectura de la glandula mamaria a partir de imágenes radiológicas Duva Alberto gomez Betancur 2011 .http://www.bdigital.unal.edu.co/7340/1/75102698.2012.pdf

[58] Comparación de algoritmos de mejora del contraste de imágenes. Universidad de Malaga Perez Serrano Longinos 2008.

[59] Mejora en la detección de microcalcificaciones en mamografías digitalizadas mediante la aplicación de arquitecturas neuronales, Pedro José García Laencina , 2004
http://repositorio.upct.es/bitstream/handle/10317/201/pfc1328.pdf?sequence=1

[60] A.Jain Fundamentals of digital image processing. Petince Hall, Englewood Cliffs, NJ,1989.

http://www.dsic.upv.es/docs/bib-dig/tesis/etd-05192006-104846/borrador_tesis_llobet.pdf

[61] J.Gonzalez. Visión por computador. ITP Paraninfo, 1999.

[62] J Viton, M Rasigni, G Rasigni et al. "metod for characterizing masses in digital mammograms".opt ENG. , Vol 35,No 12,PP, 3453-3459, 1996

[63] G Brake and N. Karssemeijer. "Single and multiescale detectión of masses in digitized mammograms". IEEE Trans. Med. Imaging, vol. 18, No. 7, pp. 628-639, 1999.

[64] Analisis de la teoría de ondiculas orientada a las aplicaciones en la energía eléctrica: Fundamentos. Rosa María de Castro Fernandez, Madrid, 2002; http://www.uta.cl/hdiaz/Documents/Fundamentos_Wavelets.pdf

[65] Suárez Cuenca JJ. Desarrollo de un sistema de diagnóstico asistido por computador para detección de nódulos pulmonares en tomografía computarizada multicorte [tesis doctoral]. Santiago de Compostela. Universidad de Santiago de Compostela; 2009.

http://www.juntadeandalucia.es/salud/servicios/contenidos/nuevaaetsa/up/AETSA_2013_4_Diag_Asist.pdf

[66] Castilla-la Mancha contará con un Centro de Excelencia en Imagen Médica Digital. Servicio de Salud de Castilla-La Mancha [Internet].19/07/2009 [citado 16/07/2013]. URL: http://sescam.jccm.es/web1/ home.do?main=/ciudadanos/noticias/20090719CentroExcelenciaImagen Medica.html. (Archivado por WebCite® en http://www.webcitation.org/6IA5uTq52)

[67] Castells X, Sala M, Ascunce N, Salas D, Zubizarreta R, Casamitjana M. Descripción del cribado del cáncer en España. Proyecto DESCRIC. Madrid: Plan de Calidad para el Sistema Nacional de Salud, Ministerio de Sanidad y Consumo. Barcelona: Agència d'Avaluació de Tecnologia i Recerca Mèdiques de Cataluña; 2007.Informes, estudios e investigación; AATRM 2006/01.

[68] Fernandez, Nicolas Luis. 2002. "Contribución al reconocimiento de objetos 2D mediante detección de bordes en imágenes en color." Universidad Politécnica de Madrid.

[69] Método de detección de distorsiones de la arquitectura de la glándula mamaria a partir de imágenes radiológicas. Duván Alberto Gómez Betancur, Universidad de Colombia Facultad de Minas, Escuela de Sistemas Medellín, Colombia 2011.

[70] Bustamante, M. 2001. "Calculo de la Amplitud Dispersada en Mamografias, Usando Como Modelo de Degradacion el Filtro Bosso." PHAROS 8.

[71] Cheng, H. D. 2006. "Approaches for automated detection and classification of masses in mammograms." Pattern Recognition 39(4): 646–668.

[72] Tang, J. 2009. "Computer-aided detection and diagnosis of breast cancer with mammography: Recent advances." Information Technology in Biomedicine, IEEE Transactions on 13(2): 236–251.

[73] Rangayyan, Rangaraj M., y Thanh M. Nguyen. 2006. "Fractal Analysis of Contours of Breast Masses in Mammograms." Journal of Digital Imaging 20: 223-237.

[74] Ayres F.J., y Rangayvan R.M. 2003. "Characterization of architectural distortion in mammograms." In IEEE, p. 886-889.

[75] Ayres F.J., y Rangayvan R.M. 2005. "Characterization of architectural distortion in mammograms." Engineering in Medicine and Biology Magazine, IEEE 24(1): 59-67.

[76] Matsubara, T, Ichikawa T., Hara T., Fujita H., Kasai S., Endo T., Iwase T.. 2003. "Automated detection methods for architectural distortions around skinline and within mammary gland on mammograms." International Congress Series 1256(0): 950-955.

[77] Rangayyan, R.M., Banik S., y Desautels J.E.L. 2010. "Computer-aided detection of architectural distortion in prior mammograms of interval cancer." Journal of Digital Imaging 23(5): 611–631.

[78] Álvarez G Damián, Guevara G Martha L y Holguín L Germán A. 2006. "Preprocesamiento de imágenes aplicadas a mamografías digitales." Scientia Et Technica XII: 1-6.

[79] Tutorial de digitalización de imágenes Anne R. kenney ; Oya Y. Rieger; Richard Entlech 2000-2003 biblioteca de la universidad de Cornell , departamento de preservación y conservación

https://www.library.cornell.edu/preservation/tutorial-spanish/tutorial_Spanish.pdf

[80] Chevalier del Rio M, Lobato Busto R, Morán Penco P, Pombar Camean MA. Control de calidad en mamografía. Guía práctica. Xunta de Galicia. Consellería de Sanidade e Servicios Sociais. Dirección Xeral de Saúdade Pública 1995.

[81] Maidment ADA. Digital mammography. Seminars in Roentgenology 2003; 38: 216 – 230.

[82] SISTEMA INTELIGENTE PARA LA DETECCIÓN Y DIAGNÓSTICO DE PATOLOGÍA MAMARIA MEMORIA PARA OPTAR AL GRADO DE DOCTOR PRESENTADA POR Amparo Vilarrasa Andrés; Madrid, 2006 ISBN: 978-84-669-2950-9

[83] Haus AG, Yaffe MJ. Screen-film and digital mammography: Image quality and radiation dose considerations. Radiologic Clinics of North America 2000; 38: 871 – 898.

[84] Haus AG. Screen-film image receptors and film processing. In Haus AG, Yaffe MJ (eds): Categorical Course in Physics: Technical aspects of breast imaging Syllabus. Oak Brook, IL, Radiological Society of North America 1994; p85.

[85] Haus AG, Yaffe MJ. Screen-film and digital mammography: Image quality and radiation dose considerations. Radiologic Clinics of North America 2000; 38: 871 – 898.

[86] Maidment ADA. Digital mammography. Seminars in Roentgenology 2003; 38: 216 – 230.

[87] Flanagan DA, Gladding SB, Lovell FR. Can scintimammography reduce “unnecessary” biopsies?. Am Surg 1998; 64: 670.

[88] Jensen J.R., 1996, Introductory Digital Image Processing (Second Edition), Ed. Prentice Hall, pp 316.

[89] Alejandro Díaz Sotolongo. Método para la clasificación del cancer de mamas utilizando características de textura , Departamento de Computación, Facultad de Ingeniería, Universidad de Ciego de Ávila, Cuba

[90] LA MATRIZ DE CO-OCURRENCIA EN LA CLASIFICACIÓN MULTIESPECTRAL: TUTORIAL PARA LA ENSEÑANZA DE MEDIDAS TEXTURALES EN CURSOS DE GRADO UNIVERSITARIO. Universidad Nacional de La Plata, Facultad de Ciencias Agrarias y Forestales, Departamento de Ambiente y Recursos Naturales

[91] de Berg, M.; van Kreveld, M.; Overmars, Mark; Schwarzkopf, O. (2000), Computational Geometry: Algorithms and Applications, Springer, pp. 2–8.

[92] Knuth, Donald E. (1992), Axioms and hulls, Lecture Notes in Computer Science 606, Heidelberg: Springer-Verlag, p. ix+109, doi:10.1007/3-540-55611-7, ISBN 3-540-55611-7, MR 1226891.

[93] http://www.cse.yorku.ca/~aaw/Hang/quick_hull/Algorithm.html

[94] 2011 Workshops of International Conference on Advanced Information Networking and Applications Fast Two Dimensional Convex Hull on the GPU Srikanth Srungarapu Durga Prasad Reddy Kishore Kothapalli P. J. Narayanan International Institute of Information Technology, Hyderabad Gachibowli, Hyderabad, India – 500 032.

[95] http://www.jano.es/ , Revista profesionales de la salud 18.02.2011

[96] http://www.fda.gov/ U.S. Food and Drug Administration,10903 New Hampshire Avenue, Silver Spring, MD 20993,1-888-INFO-FDA (1-888-463-6332)

[97] Schulz-Wendtland R, Fuchsjäger M, Wacker T, Hermann KP. Digital mammography: an update. Eur J Radiol. 2009; 72:258-65. Medline

[98] Skaane P. Studies comparing screen-film mammography and full-field digital mammography in breast cancer screening: update review. Acta Radiol. 2009; 50:3-14. Medline

[99] Vinnicombe S, Pinto Pereira SM, McCormack VA, Shiel S, Perry N, Dos Santos Silva IM. Full-field digital versus screen-film mammography: comparison within the UK breast screening program and systematic review of published data. Radiology. 2009; 251:347-58.

[100] Schulz-Wendtland R, Fuchsjäger M, Wacker T, Hermann KP. Digital mammography: an update. Eur J Radiol. 2009; 72:258-65.

[101] Bluekens AM, Karssemeijer N, Beijerinck D, Deurenberg JJ, van Engen RE, Broeders MJ, et al. Consequences of digital mammography in population-based breast cancer screening: initial changes and long-term impact on referral rates. Eur Radiol. 2010; 20:2067-73.

[102] M. Mellado, A.M. Osa, A. Murillo, R. Bermejo, A. Burguete, M.J. Pons, N. Erdozain Influencia de la mamografía digital en la detección y manejo de microcalcificaciones. Detección de microcalcificaciones en mamografías digitales" ,Servicio de Radiología, Complejo Hospitalario de Navarra,

Pamplona, Navarra, España, Programa de Prevención de Cáncer de Mama de Navarra, Instituto de Salud Pública, Pamplona, Navarra, España. 1 Junio 2011

[103] Pablo Granitto , Lucas Uzal curso de metodos actuales en machine learning, Departamento de Computación, Facultad de Ciencias Exactas, Universidad Nacional de Río Cuarto,Ruta 36 Km. 601, Río Cuarto (5800), Córdoba, Argentina. 2016

[104] http://www.cifasis-conicet.gov.ar/granitto/ECI2014/

[105] Brandon Rohrer . Cómo elegir algoritmos para Aprendizaje automático de Microsoft Azure, 28/04/2016

https://azure.microsoft.com/es-es/documentation/articles/machine-learning-algorithm-choice/

[106]https://azure.microsoft.com/es-es/documentation/articles/machine-learning-algorithm-choice/#hoja-de-referencia-rpida-de-algoritmos-de-aprendizaje-automtico

[107] Proyecto IMED, INVESTIGACIÓN EN MÉTODOS DE AYUDA AL DIAGNÓSTICO EN CÁNCER DE MAMA, 2013 http://www.redclara.net/news/Presentaciones_dvs/Manuel_Rubio_CIEMAT.pdf

[108] CIEMAT http://www.ciemat.es/cargarAplicacionNoticias.do?identificador=195

[109] Breast cancer Digital repository 2013. http://bcdr.inegi.up.pt/

[110] MIT Technology Review, 2016 http://www.technologyreview.es/informatica/44410/los-chips-neuromorficos-tendran-una-inteligencia/

[111] William L. Hosch, Enciclopedia Britanica https://www.britannica.com/technology/machine-learning

[112] . Phil Simon (18 de marzo de 2013). Demasiado grande para ignorarlo: Caso de Estudio para grandes volúmenes de datos . Wiley. Wiley. p. pag. 89. ISBN 978-1-118-63817-0 . 89. ISBN 978-1-118-63817-0

[113] Aprendizaje automático , Wikipedia https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico

[114] Flach, Peter (2012) Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Págs. 20-21,Cambridge University Press. ISBN 978-1-107-42222-3.

[115] M. A. Gutiérrez Naranjo F. J. Martín Mateos J. L. Ruiz Reina Dpto. Ciencias de la Computación de Inteligencia Artificial, Universidad de Sevilla https://www.google.es/#q=M.+A.+Gutierrez+Naranjo+aprendizaje+automatico+modelo+probabilistico

[116] Mitchell, T. M. Machine learning, Boston, McGraw-Hill, 1997. _http://www-2.dc.uba.ar/materias/aa/aa.html#Aprendizaje probabilístico y

[117] Fernando Sancho Caparrini, Introducción al Aprendizaje Automático .Última modificación: 16 de julio de 2017. http://www.cs.us.es/~fsancho/?e=75

[118] AI and Games Pioneer, A L Samuel , 24 March 2015, i-programmer http://i-programmer.info/history/people/669-a-l-samuel-ai-and-games-pioneer.html?start=1

[119] http://dmi.uib.es/~abasolo/intart/3-aprendizaje.html

[120] Universidad de Oviedo , departamento de Inteligencia Artificial, http://www.aic.uniovi.es/ssii/SSII-T7-IntroduccionAlAprendizaje.pdf

[121] Lior Rokach and Oded Maimon (2008). Data mining with decision trees: theory and applications. World Scientific. ISBN 978-981-277-171-1.

[122] Apuntes de arboles de decisión Universidad de Nebrija https://www.nebrija.es/~cmalagon/inco/apuntes_mios/arboles_de_decision.pdf

[123] K-vecinos, wikipedia https://es.wikipedia.org/wiki/K-vecinos_m%C3%A1s_cercanos

[124] Apuntes ,Aprendizaje Automático para el Análisis de Datos, Grado en estadística y empresa. Ricardo Aler Mur. http://ocw.uc3m.es/ingenieria-informatica/aprendizaje-automatico-para-el-analisis-de-datos-2013/ocw7-knn.pdf

[125] Oldelmar rodriguez http://www.oldemarrodriguez.com/yahoo_site_admin/assets/docs/Presentaci%c3%b3n_-_KNN.20085205.pdf

[126] Fix, E.; J.L. Hodges (1989). «(1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)». International Statistical Review / Revue Internationale de Statistique 57 (3): 233-238. doi:10.2307/1403796.

[127] Núria Macía Antolinez y Ester Bernardo Mansillas ,2007, Metricas de complejidad para la transformación del problema de la detección del cáncer basado en mamografías.Revista del centro de investigación. Universidad La Salle,Julio diciembre , año/vol. 7, número 208

[128] Ho ,TK; Basu, M. (2002). “Complexity measures of supervised classification problems”. IEEE Transaction on Pattern Analsis and Machine Inteligemce, 3(24), 289-300.

[129] Ho, T.K.; Bernardó Mansilla, E (2006). “Data complexity and domains of competence of classifiers”. Pattern Recognitión.

[130] María Aldehuela Lucena, 10 de junio de 2005. ANÁLISIS COMPARATIVO ENTRE MÉTODOS ESTADÍSTICOS Y DE MINERÍA DE DATOS

[131] Fernando Sancho Caparrini, Métodos combinados de aprendizaje 13 de Diciembre de 2015 http://www.cs.us.es/~fsancho/?e=106

[131] Fernando Sancho Caparrini, Métodos combinados de aprendizaje 13 de Diciembre de 2015 http://www.cs.us.es/~fsancho/?e=106

[132] Combinación de Clasificadores http://www.infor.uva.es/~isaac/doctorado/Cap06_CombClas.pdf

[133] Yoav Freund and Robert E. Schapire. A decision-teoretic generalization of online learning and an application to boosting. In Second European Conference on Computational Learning Theory, 1995.

[134] Raquel Ubeda Navarro, Escuela Técnica Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Valencia, Proyecto Final de Carrera Aplicación de técnicas de Boosting para detección de matriculas , 2004 Valencia

[135] S. L. de la Fuente-Bermúdez a; F. J. Ornelas-Rodríguez a; J. J. González-Barbosaa; J. B. Hurtado-Ramos a; M. Romero del Valle b; y J. A. Salinas-Rincónc, DETECCIÓN DE MASAS Y ASIMETRÍAS EN MAMOGRAFÍAS CON DISTINTOS PATRONES PARENQUIMALES, UTILIZANDO ENTROPÍA DE TSALLIS-HAVRDA-CHARVÁT.
http://wwwoptica.inaoep.mx/tecnologia_salud/2015/memorias/pdf/MyT2015_44_E.pdf

[136] P.A. Viola y M. J. Jones. Rapid object detection using a boosted cascade of simple features. pp. 511-518, 2001.

[137] M. Gema Díaz Pizarro , Detección de lesiones en mamografías y Ecografías para el diagnostico del Cancer de mama, 2014
https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&cad=rja&uact=8&ved=0ahUKEwjm

y7GN8ovOAhWEVxoKHaEyAmcQFggrMAI&url=https%3A%2F%2Fruidera.uclm.es%2Fxmlui%2Fbitstream%2Fhandle%2F10578%2F3681%2FTFG_Diaz%2520Pizarro.pdf%3Fsequence%3D3&usg=AFQjCNFYoyfcELh-V-Rt4YXDBe7pTfck-Q&bvm=bv.127984354,d.d2s

[138] Clasificación de algoritmos Mathworks
http://es.mathworks.com/discovery/aprendizajeautomatico.html?requestedDomain=es.mathworks.com

[139] Bourel, M; Apprentissage statistique par agrégation de modèles, tesis doctoral, Universidad de la República, Uruguay-Université d'Aix-Marseille, France. En curso. Memoria de Trabajos de Difusión Científica y Técnica, núm. 10 (2012) ISSN 1510-7450 • ISSN (en línea) 1688-9584
http://www.um.edu.uy/docs/2_metodos_de_agregacion_de_modelos_y_aplicaciones.pdf

[140] Breiman, L; Random forests, Machine Learning, 2001. 45(1): p. 5–32.

[141] Breiman, L., Friedman, J.,Olshen, R., y Stone, C; Classification and Regression Trees.Belmont, CA 1984: Chapman & Hall.

[143] Fernando Díaz Gómez ,E. U. de Informática , Aprendizaje automático: métodos y aplicaciones,

Dpto. de Informática, E. U. de Informática, Uva
http://www.eui.uva.es/doc/anuncios/curso.2013.2014/aprendizajeAutomatico.pdf

[144] Wikipedia https://es.wikipedia.org/wiki/Karl_Popper

[145] Wikipedia https://es.wikipedia.org/wiki/Ronald_Fisher

[146] German Andres Morales España, Alvaro Gomez Ruiz. Estudio e implementación de una herramienta basada en máquinas de soporte vectorial aplicada a la localización de Fallas en sistemas de distribución, Universidad Industrial de Santander, Bucaramanga 2005.
http://tangara.uis.edu.co/biblioweb/tesis/2005/118738.pdf

[147] Ecured http://www.ecured.cu/index.php/Funci%C3%B3n_Kernel

[148] Yoamel Acosta Morales, Implementación y evaluación de kernels en la predicción de actividad biológica ,05/02/2015 http://vinculando.org/beta/implementacion-evaluacion-kernels-prediccion-actividad-biologica.html

[149] G. Moncecchi, Grupo PLN, Instituto de Computación, Facultad de Ingeniería, Universidad de la República, Uruguay, Laboratoire MoDyCo, Université Paris Ouest, Nanterre La Défense, France XI Jornadas de Informática e Investigación Operativa, Montevideo, Abril 2010. http://www.fing.edu.uy/~gmonce/kernel_methods_jiio.pdf

[150] Carlos Alonso González, SVM:Máquinas de Vectores Soporte, Grupo de Sistemas Inteligentes, Departamento de Informática, Universidad de Valladolid http://www.infor.uva.es/~calonso/MUI-TIC/MineriaDatos/SVM.pdf

[151] V. Vapnik. the nature of statistical learning Theory. Springer Verlag, New York, 1995

[152] *Philip Wolfe (1961). "A duality theorem for non-linear programming". Quarterly of Applied Mathematics 19 : 239–244. "Un teorema de dualidad para la programación no lineal" Trimestral de Matemática Aplicada. 19: 239-244.*

[153] Delta de Kronecker https://es.wikipedia.org/wiki/Delta_de_Kronecker

[154] Platt, John (1998), secuencial mínimo de optimización: un algoritmo rápido de Formación Support Vector Machines.

[155] *Chang, Chih-Chung; Lin, Chih-Jen (2011). Lin, Chih-Jen (2011). "LIBSVM: A library for support vector machines". ACM Transactions on Intelligent Systems and Technology 2 (3). "LIBSVM: Una biblioteca para máquinas de vectores soporte" ACM Transactions on Intelligent Systems y Tecnología*

[156] Autor: José Luis Alba Castro , Curso de doctorado: Decisión, estimación y clasificación.. Departamento de Teoría de la Señal y Comunicaciones http://web.archive.org/web/20140801145654/http://www.gts.tsc.uvigo.es/~jalba/doctorado/SVM.pdf

[157] Enrique J Carmona ,Tutorial de máquinas de vectores de soporte (SVM) . Dpto. de Inteligencia Artificial, ETS de Ingeniería Informática, Universidad Nacional de Educación a Distancia (UNED) (2014)

[158] Francisco José Ribadas Pena, SVM Support Vector Machines (Máquinas de Vectores Soporte) Modelos de Razonamiento y Aprendizaje,2012 http://ccia.ei.uvigo.es/docencia/MRA/1213/transparencias/Tema6.pdf

[159] Introducción a la Inteligencia Artificial , Aprendizaje http://dmi.uib.es/~abasolo/intart/3-aprendizaje.html

[160] Víctor Tomás T. Mariano, Introducción a las Redes Neuronales, Historia de las redes neuronales. Universidad Autonoma del estado de Hidalgo-Huejutla, Instituto de Ciencias Básicas e Ingeniería, Sistemas Computacionales,2011.

[161] Minsky, Marvin; Papert, Seymour (1969). Perceptrons: an introduction to computational geometry. «The perceptron has shown itself worthy of study despite (and even because of!) its severe limitations. It has many features to attract attention: its linearity; its intriguing learning theorem; its clear paradigmatic simplicity as a kind of parallel computation. There is no reason to suppose that any of these virtues carry over to the many-layered version. Never- theless, we consider it to be an important research problem to elucidate (or reject) our in- tuitive judgement that the extension to multilayer systems is sterile.»

[162] Autor: James Anderson, Redes Neuronales, Editorial: ALFAOMEGA,ISBN: 9789701512654,EAN: 9789701512654

[163] Hugo Flores Gutiérrez, Polilibro Redes Neuronales Artificiales 1. http://hugo-inc.com/RNA/indice.html

[164] Raquel Flórez López, José Miguel Fernández Fernández ,Las Redes Neuronales Artificiales. Fundamentos teóricos y aplicaciones practicas. Editorial Netbiblo, S.L, ISBN 978-84-9745-246-5

https://books.google.es/books?id=X0uLwi1Ap4QC&pg=PA20&lpg=PA20&dq=James+Anderson+asociador&source=bl&ots=gMNFlomuWg&sig=C7mVjRBb1Pt-jPxsVTmAplK-R-I&hl=es&sa=X&ved=0ahUKEwij74-K8ZbOAhVFOBQKHXCZCa8Q6AEINDAE#v=onepage&q=James%20Anderson%20asociador&f=false

[165] John Hopfield, https://en.wikipedia.org/wiki/John_Hopfield

[166] Hopfield RNA, https://es.wikipedia.org/wiki/Hopfield_(RNA)

[167] W.S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity.

Bulletin of Mathematical Biophysics, 5:115–133, 1943

[168] López-Muñoz, F; Boya, J., Alamo, C. (16 de octubre de 2006). «Neuron theory, the cornerstone of neuroscience, on the centenary of the Nobel Prize award to Santiago Ramón y Cajal». Brain Research Bulletin 70: 391-405

[169] Paniagua, R.; Nistal, M.; Sesma, P.; Álvarez-Uría, M.; Fraile, B.; Anadón, R. y José Sáez, F. (2002). Citología e histología vegetal y animal. McGraw-Hill Interamericana de España, S.A.U. ISBN 84-486-0436-9.

[170] Randall, D.; Burggren, W. y French, K. (1998). Eckert Fisiología animal (4ª edición). ISBN 84-486-0200-5.

[171] La neurona, Coast.pink enciclopedia, http://coast.pink/neurona_3201976.html

[172] La neurona wikipedia https://es.wikipedia.org/wiki/Neurona#cite_note-24

[173] Llinás, Rodolfo (1988). «The Intrinsic Electrophysiological Properties of Mammalian Neurons: Insights into Central Nervous System Function». Science 242: 1654-1664

[174] Solé y Manrubia, 1996, p. 514

[175] MQL5. Redes neuronales: de la teoría a la práctica. 2014 https://www.mql5.com/es/articles/497

[176] Aitor Gómez Maguregui, Modelos de computación II, apuntes ampliados, http://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjeiaeqnZnOAhVEOxQKHcuiARQQFggcMAA&url=ftp%3A%2F%2Fdecsai.ugr.es%2Fpub%2Fusuarios%2Fcastro%2FActividades%2FRedes-Neuronales%2FApuntes%2FApuntes%2520Ampliados.doc&usg=AFQjCNHLz6XQymUbrG3Tt3AJBMf4OC69fw&bvm=bv.128617741,d.d24

[177] Introducción a las redes neuronales artificiales, http://rna.50webs.com/tutorial/RNA_intro.html

[178] Red neuronal artificial https://es.wikipedia.org/wiki/Red_neuronal_artificial

[179] Características de las redes neuronales http://thales.cica.es/rd/Recursos/rd98/TecInfo/07/capitulo3.html

[180] Los neurocomputadores, Publicaciones SEMANA S.A, 2016 http://www.semana.com/vida-moderna/tecnologia/articulo/los-neurocomputadores/11370-3.

[181] Nettalk (red neuronal artificial) Wikipedia, https://en.wikipedia.org/wiki/NETtalk_(artificial_neural_network)

[182] Ángel Riviére, Obras escogidas Vol 1, Diálogos sobre psicología: de los cómputos mentales a los significados de la conciencia. ed panamericana, ISBN 84-7903-615-X pag 182 ,2003

[183] Perceptron Simple y Adaline, Redes de Neuronas Artificiales (Grupo 5 - RAI - UC3M 2011/2012), http://www.lab.inf.uc3m.es/~a0080630/redes-de-neuronas/perceptron-simple.html

[184] Perceptrón multicapa, Wikipedia, https://es.wikipedia.org/wiki/Perceptr%C3%B3n_multicapa

[185] Propagación hacia atras https://es.wikipedia.org/wiki/Propagaci%C3%B3n_hacia_atr%C3%A1s

[186] Marco Antonio Valencia Reyes, Cornelio Yáñez Márquez, Luis Pastor Sánchez Fernández, Algoritmo Backpropagation para Redes Neuronales conceptos y aplicaciones, INSTITUTO POLITÉCNICO NACIONAL ,CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN No. 125 Serie: VERDE Fecha: 2006

[187] Redes multicapa ftp://decsai.ugr.es/pub/usuarios/castro/Actividades/Redes-Neuronales/Apuntes/Apuntes%20Javier%20Rodriguez%20Blazquez/Redes%20Multicapa%20-%20Algoritmo%20Backpropagation.pdf

[188] RED NEURONAL Backpropagation http://es.slideshare.net/mentelibre/red-neuronal-backpropagation

[189] Aníbal Guerra, Joel Rivas, Detección de microcalcificaciones en imágenes mamográficas usando redes neuronales, Rev. Fac. Ing. UCV v.26 n.3 Caracas sep. 2011, versión impresa ISSN 0798-4065 http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S0798-40652011000300002

[190] Marta Fernandez Campa, Aprendizaje Multi-Tarea Selectivo utilizando Redes Neuronales Artificiales, Universidad rey Juan Carlos, 2009. https://eciencia.urjc.es/bitstream/handle/10115/1725/PFC_Marta_Fernandez_Camba.pdf?sequence=1&isAllowed=y

[191] Claudia Joana Santamaría Moya,. Construcción de una base de datos de imágenes de mamografía para la identificación de microcalcificaciones", Universidad Tecnológica de Pereira, Pereira-Risaralda 2014.

http://recursosbiblioteca.utp.edu.co/dspace/bitstream/handle/11059/4236/621367S232.pdf?sequence=1&isAllowed=y

[192] Paul Acquatella Tutorial de Backpropagation - Un algoritmo de entrenamiento para redes neuronales15 Mar 2012

http://es.mathworks.com/matlabcentral/fileexchange/35659-tutorial-de-backpropagation-un-algoritmo-de-entrenamiento-para-redes-neuronales/content/backpropagation.m

[193] Xabier Basogain Olabe Redes Neuronales Artificiales y sus aplicaciones. Centro: Escuela Superior de Ingeniería de Bilbao. http://www.ciberesquina.una.edu.ve:8080/2014_2/350_E.pdf

[194] Y. Bengio, A. Courville, and P. Vincent., "Representation Learning: A Review and New Perspectives," IEEE Trans. PAMI, special issue Learning Deep Architectures, 2013 (en inglés)

[195] Fast Artificial Neural Network Library http://leenissen.dk/fann/wp/

[196] Convolución https://es.wikipedia.org/wiki/Convoluci%C3%B3n

[197] Autoencoder http://ufldl.stanford.edu/wiki/index.php/Visualizing_a_Trained_Autoencoder

[198] Carlos Zahumenszky, A la primera 'Skynet' de Google le gustan los gatitos http://www.xataka.com/historia-tecnologica/a-la-primera-skynet-de-google-le-gustan-los-gatitos, Junio 2012

[199] Cony sturn, En una demostración en video de una conferencia dada en Tianjin, China, Rashid habló en inglés, y luego sonó su misma voz, pero hablando en chino mandarín. noviembre 2012 https://www.fayerwayer.com/2012/11/microsoft-muestra-sistema-para-traducir-con-tu-voz-del-ingles-al-chino-mandarin/,

[200] Robert D. Hof, 2013, Aprendizaje profundo, MIT Technology Review https://www.technologyreview.es/informatica/42965/aprendizaje-profundo/

[201] Sonia.Tangaro "MAGIC-5: an Italian mammographic database of digitised images for research" , Istituto Nazionale di Fisica Nucleare, Via Amendola 173, 70126 Bari, Italy Dipartimento di Fisica, Università di Lecce, Italia .2008 http://www.ncbi.nlm.nih.gov/pubmed/18536871

[202] J Suckling, The mini-MIAS database of mammograms http://peipa.essex.ac.uk/info/mias.html

[203] The Digital Database for Screening Mammography. Heath, M., y otros. 2000, Proc. Int. Workshop Dig. Mammography, págs. 212-218.

[204] Esther Ristori Bogajo y otros, Bases de datos de mamografía digital, estudio comparativo.

http://www-rayos.medicina.uma.es/Rmf/RadiolDigital/DDSM.htm

[205] "DDSM: Digital Database for Screening Mammography" University of South Florida Digital Mammography

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

[206] Breast Imaging Editado por Etta Pisano D. pag 144

[207] Martin Thijssen, Jan Hendriks, Leon van Erning, La mamografía digital: Nijmegen, 1998 Editado por nico karssemeijer, pag 191

[208] Bases de datos de mamografías http://www.mammoimage.org/databases/

[209] Julia Oliveira, MO Gueld, A. de A. Araújo, B. Ott, TM Deserno, "Toward a standard reference database for computer-aided mammography", Proc. SPIE 6915, Medical Imaging 2008: Computer-Aided Diagnosis, 69151Y (March 17, 2008); doi:10.1117/12.770325; http://dx.doi.org/10.1117/12.770325

[210] IRMA (Image Retrieval in Medical Applications) http://ganymed.imib.rwth-aachen.de/irma/index_en.php

[211] Bayesian Decision Support in Medical Screening 2009,http://www.diagnijmegen.nl/index.php/NWO_-_Bayesian_Decision_Support_in_Medical_Screening_%28B-SCREEN%29

[212] Ball, J. E. and L. M. Bruce (2007). Digital Mammogram Spiculated Mass Detection and Spicule Segmentation using Level Sets. Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE

[213] Brem R, Baum J, Lechner M, Kaplan S, Souders S, Gill L, et al. Improvement in sensitivity of screening mammography with computer-aided-detection: A multiinstitutional trial. AJR Am J Roentgenol. 2003;181:687–93.

[214] Warren L, Wood S, D'Orsi C, Feig S, Kopans D, O'Shaughnessy K, et al. Potential contribution of computer-aided-detection to the sensitivity of screening mammography. Radiology. 2000; 215:554–62.

[215] Birdwell R, Ikeda MD, O'Shaughnessy K, Sickles E. Mammographic characteristics of 115 missed cancers later detected ARTICLE IN PRESS 16 S.M. Sánchez Gómez with screening mammography and the potential utility of computer-aided-detection. Radiology. 2001; 219:192–202.

[216] L. MOYANO PÉREZ "Detección de microcalcificaciones en mamografía digitalizada mediante wavelets", ETSIT PFC Universidad Politécnica de Madrid. 2002

[217] MANZANO LIZCANO,J.A., SÁNCHEZ AVILA,C,"A microcalcification detection system for digital mammography using the contourlet transform and SVM", ICNAAM 2005 International Conference on numerical analysis and applied mathematics.Official conference of the European Society of Computational Methods in Sciences and Engineering ECSMSE, S. 373-377, 2005.

[218] Meyer, F. Iterative image transformations for an automatic screening of cervical smears. The Journal of Histochemistry and Cytochemistry, 128-135., 1979

[219] Juan Luis Cerrato Vargas y Joel Quintanilla Dominguez," SISTEMA DE DETECCIÓN DE MICROCALCIFICACIONES EN MAMOGRAFÍA DIGITALIZADA APLICANDO PROCESAMIENTO DE IMAGEN E INTELIGENCIA ARTIFICIAL" Jovenes en la Ciencia, Revista de divulgación Científica. 2014

http://www.jovenesenlaciencia.ugto.mx/index.php/jovenesenlaciencia/article/view/679/pdf2

[220] Ojeda-Magaña, B., Quintanilla-Domínguez, J., Ruelas, R., & Andina, D.. Images sub-segmentation with the pfcm clustering algorithm. The 7th IEEE International Conference on Industrial Informatics (págs. 499-503). Cardiff UK: IEEE, (2009)

[221] Juan Cerrato, Joel Quintanilla, José Gordillo, José Rico y José Barrón. J."Interfaz gráfica de usuario para la detección de microcalcificaciones mediante análisis de mamografía digitalizada"

Universidad Tecnológica del Suroeste de Guanajuato, 2014
http://oa.upm.es/36881/1/INVE_MEM_2014_197840.pdf

[222]Software ImageJ https://imagej.nih.gov/ij/download.html

[223] Eduardo Laorden Fiter "Descripción, comparación y ejemplos de uso de las funciones de la toolbox de procesado digital de imágenes de Matlab" 2012
http://oa.upm.es/14016/2/PFC_EDUARDO_LAORDEN_FITER_B.pdf

[224] YAKAMI DICOM Tools (Free DICOM Viewer/Converter/etc.) http://www.kuhp.kyoto-u.ac.jp/~diag_rad/intro/tech/dicom_tools.html

[225] LECUN, Yann, et al. Gradient-based learning applied to document recognition [en línea]. Proceedings of the IEEE, 1998, vol. 86, no 11, p. 7. 2014 http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

[226] Erick Zamora Redes neuronales convolucionales https://es.scribd.com/doc/295974900/Redes-Neuronales-Convolucionales

[227] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks" http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf.

[228] Página oficial de matlab http://es.mathworks.com/products/matlab/

[229] Indice de funciones Matconvnet http://www.vlfeat.org/matconvnet/functions/

[230] Dropout: A simple Way to Prevent Neural Networks from Overfitting. Srivastava et al. https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf - Mayo 2016

[231] Santiago Delgado Bueno,Armando Tejerina,Armando Tejerina Gómez "Medicina legal en patología mamaria". https://books.google.es/books?id=HM2qld76xyUC&pg=PA122&dq=Stegner+microcalcificaci%C3%B3n&hl=es&sa=X&ved=0ahUKEwi3pIbj2eTOAhWqAcAKHQdlBhIQ6AEIJzAA#v=onepage&q=Stegner%20microcalcificaci%C3%B3n&f=false

[232] Simoff S., D. C. (2002). Multimedia Data Mining between Promise and Problems (Tercera ed.). SIGKDD Explorations.

[233] Han, J. Y. (2001). Data Mining. Concepts and Techniques. 548. Editorial Morgan Kaufmann.

[234] EVANS,A.J., KUTT,E., RECORD,C., WALLER,M., BOBROW,L., MOSS,S., 2007. "Radiological and pathological findings of interval cancers in a multi-centre, randomized, controlled trial of mammographic screening in women from age 40-41 years", Clinical Radiology, 62, S. 348-352.

[235] Raffle AE, Muir Gray JA. Preface in: Screening: evidence and practice. Oxford: Oxford University Press, 2007.

[236] Calonge N. Past, present and future issues for the US Preventive Services Task Force. Conferencia inaugural. Primera Conferencia Española de Prevención y Promoción de la Salud en la Práctica Clínica. Madrid. Junio 2007.

[237] Bozek, J., Mustra, M., Delac, K., & Grgic, M. (2009). A survey of image processing algorithms in digital mammography. Recent Advances in Multimedia Signal Processing and Communications , 231, 631-657.

[238] Cheng, H. D., Shi, X., Min, R., Hu, L., Cai, X., & Du, H. N. (2006). Approaches for automated detection and classification of masses in mammograms. Pattern Recognition , 39 (4), 646-668.

[239] Vallejo Delgado N, R. J. (2012). Aplicación de técnicas de minería de datos para el diagnóstico prematuro de cáncer.

[240] Rosete Suárez A, R. D. (2009). Predicción de pacientes diabéticos. Preprocesado para Minería de Datos. Informatica Médica.

[241] Bañobre Corpas Y, B. G. Diagnóstico de Enfermedades de Transmisión Sexual mediante técnicas de Inteligencia Artificial. . Universidad de las Ciencias Informáticas, La Habana.

[242] AL-SAIYD, N., TALAFHA,S., 2015. "Mammographic image enhancement tachniques - A survey", International Journal of Computer Science and Information Technology & Security, 5, S. 200-206.

[243] KUMAR, N.H., AMUTHA,S., BABU,D.R., 2012. "Enhancement of mammographic images using morphology and wavelet transform", International Journal of Computer Technology & Applications, 3, S. 192-198.

[244] Sickles, E. A. (2007). El espectro de las asimetrías en la mama: características radiológicas, estudio diagnóstico y tratamiento. Clínicas Radiológicas de Norteamérica, 45, 765-771.

[245] Malagelada, A. O. (2007). Automatic mass segmentation in mammographic images. España: PhD thesis, Universitat de Girona.