



# Trabajo de Fin de Grado

---

## Detección de distintos tipos de cáncer mediante Redes Neuronales Artificiales

*Cancers detection using Artificial Neural Networks*

Alberto Fariña Barrera

---

La Laguna, 4 de julio de 2016

D. **Patricio García Báez**, con N.I.F. 43.356.987-D profesor contratado doctor adscrito al Departamento de Ingeniería informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Carmen Paz Suárez Araujo**, con N.I.F. 43.640.373-N profesora Titular de Universidad adscrita al Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria, como co-tutora

## **C E R T I F I C A ( N )**

Que la presente memoria titulada:

*“Detección de distintos tipos de cáncer mediante Redes Neuronales Artificiales”*

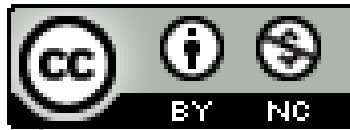
ha sido realizada bajo su dirección por D. **Alberto Fariña Barrera**, con N.I.F. 43.836.555-G.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 4 de julio de 2016.

# Agradecimientos

Me gustaría agradecer a mis tutores Patricio García Báez y Carmen Paz Suárez Araujo por el apoyo y la ayuda prestada durante el desarrollo del trabajo y de esta memoria. También me gustaría agradecer a mi compañero Óscar Trujillo Acosta por haberme ayudado en algunos aspectos del proyecto. Agradecer a mi amiga Rita Hernández Pérez, estudiante del Grado de Medicina por el apoyo y la ayuda a entender la parte médica del proyecto. Por último, a mi familia por haberme ayudado a llegar hasta aquí.

# Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial 4.0 Internacional.

## **Resumen**

El objetivo de este trabajo ha sido la detección de tres tipos de cáncer mediante redes neuronales artificiales y comparar la eficacia del sistema diseñado con otras soluciones propuestas para el mismo problema, basadas en computación evolutiva. Los tres tipos de cáncer han sido: cáncer de mama, melanoma y cáncer de pulmón.

Se dispone para el proyecto de una base de datos proporcionada por el Heuristic and Evolutionary Algorithms Laboratory (HEAL). Dicha base de datos tiene tres tablas, una para cada uno de los tres tipos de cáncer citados anteriormente que contendrán datos de pacientes con cáncer y pacientes sanos para que la red pueda ser entrenada y validada.

Se llevaron a cabo estudios con dos clases de arquitecturas neuronales diferentes, una supervisada, Back-Propagation, y una híbrida, Counter-Propagation. Esta presenta un algoritmo de entrenamiento no supervisado con otro de entrenamiento supervisado.

El uso de este sistema no pretende ser el de un diagnosticador final, sino más bien una herramienta de soporte para los oncólogos, que les permita hacer un diagnóstico inicial del paciente y derivar a pruebas más concienzudas.

**Palabras clave:** *Red Neuronal Artificial, Back-Propagation, Counter-Propagation, Detección de cáncer, PyBrain.*

## **Abstract**

The goal of this project has been the detection of three types of cancer using artificial neural networks and comparing the efficiency of the designed system with other proposed solutions for the same problem based on evolutionary computing. The three types of cancer are: breast cancer, melanoma and respiratory system cancer.

A database is provided for the project by Heuristic and Evolutionary Algorithms Laboratory (HEAL). This database contains three tables, one for each type of cancer named before with data about patients with and without cancer so the network can be trained and validated.

Studies were conducted with two different types of neural architecture, a supervised one, Back-Propagation, and a hybrid one, Counter-Propagation. This last architecture shows an unsupervised algorithm along with a supervised algorithm.

This system doesn't pretend to be a final test, but more like a support tool for oncologists, that allow them to do an initial test and refer to more important tests if necessary.

**Keywords:** *Artificial Neural Network, Back-Propagation, Counter-Propagation, Cancer detection, PyBrain.*

# Índice general

<b>Capítulo 1 Introducción.....</b>	<b>1</b>
1.1 Antecedentes.....	1
1.2 Objetivos.....	2
1.3 Actividades a realizar.....	3
1.4 Programa de Apoyo a Trabajos Finales Libres.....	4
<b>Capítulo 2 Método, herramientas y datos.....</b>	<b>5</b>
2.1 Redes Neuronales Artificiales.....	5
2.1.1 Historia.....	5
2.1.2 Características.....	5
2.1.3 Aprendizaje.....	6
2.1.4 Estado actual.....	6
2.1.5 Redes o arquitecturas estudiadas en el proyecto.....	6
2.2 Herramientas.....	7
2.2.1 Algoritmos.....	8
2.2.2 Funcionamiento de la herramienta.....	10
2.2.3 ¿Cómo se llegó a elegir PyBrain?.....	12
2.2.4 Conclusiones.....	12
2.3 Datos.....	12
2.3.1 Obtención.....	13
2.3.2 Contenido de la Base de datos.....	13
2.3.3 Resultados de la investigación del HEAL.....	14
<b>Capítulo 3 Desarrollo y análisis.....</b>	<b>16</b>
3.1 Desarrollo.....	16
3.1.1 Preprocesado.....	16
3.1.2 Desarrollo de Scripts.....	17
3.2 Pruebas a realizar y parámetros estudiados.....	18

3.2.1 Back-Propagation.....	18
3.2.2 Counter-Propagation.....	19
3.3 Cáncer de mama.....	20
3.3.1 Back-Propagation.....	21
3.3.2 Counter-Propagation.....	23
3.4 Melanoma.....	25
3.4.1 Back-Propagation.....	25
3.4.2 Counter-Propagation.....	28
3.5 Cáncer de pulmón.....	30
3.5.1 Back-Propagation.....	30
3.5.2 Counter-propagation.....	33
3.6 Resultados finales usando el subconjunto de test.....	35
3.6.1 Cáncer de mama.....	35
3.6.2 Melanoma.....	37
3.6.3 Cáncer de Pulmón.....	38
3.6.4 Conclusiones del estudio.....	39
3.7 Comparación de resultados con el HEAL.....	39
3.7.1 Cáncer de mama.....	40
3.7.2 Melanoma.....	40
3.7.3 Cáncer de Pulmón.....	41
3.7.4 Conclusiones.....	41
<b>Capítulo 4 Conclusiones y líneas futuras.....</b>	<b>42</b>
<b>Capítulo 5 Summary and Conclusions.....</b>	<b>43</b>
<b>Capítulo 6 Presupuesto.....</b>	<b>44</b>
<b>Bibliografía y referencias.....</b>	<b>45</b>
Bibliografía.....	45
Referencias.....	45



# Índice de figuras

Figura 2.1: Diagrama de flujo de uso de la librería PyBrain.....	10
Figura 2.2: Tablas originales obtenidas de la presentación de resultados del proyecto del HEAL en el EUROCAST 2015 – Capítulo “Computer Aided Systems Theory pp 350-358” .....	15
Figura 3.1: Resultados del análisis de neuronas ocultas en cáncer de mama.....	21
Figura 3.2: Resultados del análisis de ratio de aprendizaje para cáncer de mama con Back-Propagation.....	22
Figura 3.3: Resultados del análisis de momentum para cáncer de mama con Back-Propagation.....	22
Figura 3.4: Resultados del análisis de neuronas del mapa auto-organizado para cáncer de mama.....	23
Figura 3.5: Resultados del análisis del ratio de aprendizaje para cáncer de mama con Counter-Propagation.....	24
Figura 3.6: Resultados del análisis de momentum para cáncer de mama con Counter-Propagation.....	25
Figura 3.7: Resultados de análisis de neuronas ocultas para melanoma.....	26
Figura 3.8: Resultados de análisis de ratio de aprendizaje para melanoma con Back-Propagation.....	27
Figura 3.9: Resultados de análisis de momentum para melanoma con Back-Propagation.....	27
Figura 3.10: Resultados del análisis de neuronas del mapa auto-organizado para melanoma.....	28
Figura 3.11: Resultados del análisis de ratio de aprendizaje para melanoma con Counter-Propagation.....	29
Figura 3.12: Resultados del análisis de momentum para melanoma con Counter-Propagation.....	30
Figura 3.13: Resultados de análisis de neuronas ocultas para	

cáncer de pulmón.....	31
Figura 3.14: Gráfica de resultados de análisis de ratio de aprendizaje para cáncer de pulmón.....	32
Figura 3.15: Resultados de análisis de momentum para cáncer de pulmón.....	32
Figura 3.16: Resultados del análisis de neuronas del mapa auto-organizado para cáncer de pulmón.....	33
Figura 3.17: Resultados de análisis de ratio de aprendizaje para cáncer de pulmón con Counter-Propagation.....	34
Figura 3.18: Resultados del análisis de momentum para cáncer de pulmón con Counter-Propagation.....	35

# Índice de tablas

Tabla 2.1: Ratio de pacientes con y sin cáncer en las tablas de la base de datos.....	14
Tabla 3.1: Resultados de la primera prueba de Back-Propagation para cáncer de mama.....	21
Tabla 3.2: Resultados de primera prueba de cáncer de mama para Counter-Propagation.....	23
Tabla 3.3: Resultados de primera prueba de Back-Propagation para melanoma.....	25
Tabla 3.4: Resultados de primera prueba de Back-Propagation para melanoma.....	28
Tabla 3.5: Resultados de primera prueba para cáncer de pulmón con Back-Propagation.....	30
Tabla 3.6: Resultados de primera prueba para cáncer de pulmón con Counter-Propagation.....	33
Tabla 3.7: Valor de variables finales para back-propagation en cáncer de mama.....	35
Tabla 3.8: Resultados de test final con Back-Propagation para cáncer de mama.....	36
Tabla 3.9: Valor de variables finales para Counter-Propagation en cáncer de mama.....	36
Tabla 3.10: Resultados de test final con Counter-Propagation para Cáncer de mama.....	36
Tabla 3.11: Valor de variables finales para Back-Propagation en cáncer de mama.....	37
Tabla 3.12: Resultados de test final con Back-Propagation para melanoma.....	37
Tabla 3.13: Valor de variables finales para Counter-Propagation en Melanoma.....	37
Tabla 3.14: Resultados de test final con Counter-Propagation para	

melanoma.....	38
Tabla 3.15: Valor de variables finales para Back-Propagation en cáncer de pulmón.....	38
Tabla 3.16: Resultados de test final con Back-Propagation para cáncer de pulmón.....	38
Tabla 3.17: Valor de variables para Counter-Propagation en cáncer de pulmón.....	39
Tabla 3.18: Resultados de test final con Counter-propagation para cáncer de pulmón.....	39
Tabla 3.19: Comparación de resultados entre HEAL y proyecto para cáncer de mama.....	40
Tabla 3.20: Comparación de resultados entre HEAL y proyecto para Melanoma.....	40
Tabla 3.21: Comparación de resultados entre HEAL y proyecto para cáncer de pulmón.....	41

# Capítulo 1

## Introducción

En este trabajo se propone la elaboración de tres Redes Neuronales Artificiales (RNAs) [\[1\]](#), una para cada tipo de cáncer propuesto (cáncer de mama, melanoma, cáncer de pulmón), que detecten la presencia de la enfermedad en un paciente con una alta probabilidad de acierto. Para ello se llevarán a cabo estudios sobre dos tipos de redes neuronales: RNA con Back-Propagation(BPN) [\[2\]](#) como algoritmo de entrenamiento y Counter-Propagation(CPN) [\[3\]](#), una arquitectura que combina un mapa auto-organizado con un Perceptrón Simple.

Se utiliza como herramienta PyBrain [\[4\]](#), una librería de RNA sobre el lenguaje Python, que permite que la creación y entrenamiento de redes se haga de manera sencilla e intuitiva.

Para elaborar la red neuronal más eficiente posible se realizan análisis en profundidad para determinar el valor de los parámetros más característicos de cada una de las RNAs, para finalmente realizar una comparativa entre ellas, para seleccionar el que mejor determine si un paciente tiene cáncer o no.

El resultado de este proyecto tiene como objetivo servir de ayuda a médicos a la hora de obtener un primer diagnóstico rápido de un paciente, para, si es necesario, derivar a pruebas más concienzudas con la mayor brevedad posible.

### 1.1 Antecedentes

Muchos son los antecedentes de la inteligencia artificial durante el último siglo. El campo en sí comenzó su expansión a mediados de la década de los 40, llegando a un parón cuando los ordenadores de la época no podían abarcar tanta capacidad de cómputo como la que se necesitaba. Al comenzar a miniaturizarse cada vez más los transistores de un procesador, y por ello, a evolucionar de manera

exponencial la capacidad de cómputo, el campo de las redes neuronales volvió al alza [5], siendo actualmente uno de los sectores más importantes para empresas como Google, Amazon o IBM con su famosa IA, Watson, que llegó a ganar el concurso estadounidense 'Jeopardy'.

Enfocando ahora en las redes neuronales y la medicina oncológica, el caso que más concierne este proyecto es el del Heuristic and Evolutionary Algorithm Laboratory (HEAL) [6], el mismo laboratorio de investigación austriaco que cedió las bases de datos para este proyecto, elaboró un diagnosticador basado en las mismas bases de datos, cuyos resultados se compararán con los resultados de este trabajo en uno de los capítulos.

Otro de los casos más conocidos quizás sea el “Global Neural Network Cloud Service for Breast Cancer” de Brittany Wenger [7], que es capaz de detectar si un tumor es maligno con un 99% de acierto. En el caso de este proyecto, las diferencias son que se hará para tres tipos de cáncer y que las bases de datos usadas no serán tan grandes, por lo que los resultados pueden no ser tan altos.

Otra de las variables posibles del uso de redes neuronales la vemos en el caso de dos profesores y un alumno del Instituto Politécnico Nacional de México [8], que crearon una red neuronal que detecta tumores malignos a partir de las imágenes de mamografías, por lo que no solo con datos empíricos puede entrenarse una red neuronal aplicada a la medicina, también con datos extraídos de una imagen digital.

En conclusión, las muchas y variadas incursiones de las redes neuronales en el mundo de la medicina, y más concretamente en el tema de la oncología, proporciona una gran base con la que comparar los resultados de este proyecto para evaluar su éxito.

## **1.2 Objetivos**

Los objetivos principales de este proyecto son los siguientes:

- Desarrollo de una red neuronal artificial para la detección de tres tipos de cáncer.
- Desarrollar scripts de Python que sean capaces de crear, entrenar y mostrar los resultados de una red neuronal, usando la librería PyBrain y las bases de datos proporcionadas y preprocesadas.
- Realizar estudios sobre los parámetros de las RNAs para

intentar optimizar los resultados obtenidos.

- Compara los resultados obtenidos con los de otros autores que usen la misma base de datos.

## **1.3 Actividades a realizar**

Se han dividido las actividades a realizar durante este proyecto de la siguiente manera:

### **1. Preparación**

1. Búsqueda de información acerca de RNAs
2. Búsqueda y elección de herramientas para la realización de las RNAs
3. Estudio del trabajo realizado por el laboratorio HEAL

### **2. Preprocesado**

1. Análisis de las bases de datos que se utilizarán en el proyecto para detectar posibles dudas y resolverlas
2. Elección de herramienta de preprocesado para las bases de datos
3. Preprocesamiento de la base de datos para eliminar datos nulos, campos innecesarios y ajustar valores
4. Selección de las particiones de la base de datos que se usarán para entrenamiento, validación y testeo

### **3. Diseño y Desarrollo**

1. Elección del modelos de RNAs que se utilizará en el proyecto
2. Diseño de la red
3. Entrenamiento y validación de las RNAs
4. Optimización de los parámetros de las RNAs

### **4. Evaluación de las RNAs**

1. Realización de pruebas con los datos de la partición de testeo
2. Creación de gráficas de errores y precisión para la presentación de la RNA
3. Comparación de resultados con el laboratorio HEAL

### **5. Finalización**

1. Preparación de la memoria del TFG

2. Preparación de la defensa de la memoria del TFG

## **1.4 Programa de Apoyo a Trabajos Finales Libres**

Este trabajo de fin de grado está adherido al programa de apoyo a trabajos finales libres de la Oficina de Software Libre de La Universidad de La Laguna.

Puede encontrar todos los ficheros del proyecto, así como documentación, en el Github institucional de la Escuela Superior de Ingeniería y Tecnología [\[9\]](#).



# Capítulo 2

## Método, herramientas y datos

En este capítulo se explicará a fondo la principal tecnología sobre la que se elabora este trabajo, tanto el método que serán las redes neuronales como las herramientas usadas para su tratamiento y desarrollo, así como las bases de datos que fueron proporcionadas por HEAL.

### 2.1 Redes Neuronales Artificiales

#### 2.1.1 Historia

Los dos pioneros en la investigación de redes neuronales fueron Warren McCulloch y Walter Pitts, que propusieron un modelo matemático de neurona donde cada una de ellas está dotada de un conjunto de entradas y salidas. Cada entrada está afectada por un peso y la activación de la neurona se calcula mediante la suma de los productos de cada entrada y la salida es una función de esta activación. Este modelo se conoce como “Neurona de McCulloch-Pitts” [\[10\]](#) y ha servido de inspiración para el desarrollo de otros modelos neuronales.

#### 2.1.2 Características

Las redes neuronales constan de un conjunto de neuronas como la de McCulloch-Pitts organizadas en tres tipos de capas:

- La capa de entrada: esta suele contener una neurona por cada variable que se le pasará a la red.
- Capas ocultas: están compuestas por un número variable de neuronas dependiendo de la longitud que abarque el problema que agilizarán el trabajo de aprendizaje. No hay un número definido de capas ocultas, puede haber varias, no haber ninguna o solo una.

- La capa de salida: contiene una neurona por cada salida deseada de la red, conformando por tanto el resultado de la misma.

El método más común de interconexión de estas capas es de manera 'uno a todos', donde el resultado de la neurona de una capa se propaga a todas las neuronas de la capa siguiente.

### **2.1.3 Aprendizaje**

El aprendizaje en una red neuronal puede realizarse partiendo de varios tipos algoritmos. En general podemos hablar de paradigmas supervisados y no supervisados, en función de que usen una señal de supervisión externa. Uno de estos algoritmos es el Back-Propagation, de aprendizaje supervisado y que se estudiará durante el proyecto.

### **2.1.4 Estado actual**

Actualmente las redes neuronales viven su segunda edad de oro, siendo usadas en gran cantidad de proyectos y en una gran variedad de ámbitos, desde la medicina, como es el caso de este trabajo o con los ejemplos vistos en el capítulo de Antecedentes, hasta en meteorología para predecir el tiempo o en economía para predecir las fluctuaciones en el mercado financiero.

### **2.1.5 Redes o arquitecturas estudiadas en el proyecto**

Son dos los tipos de redes neuronales con las que se llevaron a cabo estudios en el proyecto:

- **RNAs con Back-Propagation**

La propagación hacia atrás de errores o retro propagación (del inglés Back-Propagation) es un algoritmo de aprendizaje supervisado que se usa para entrenar redes neuronales artificiales. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

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

aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

Después del entrenamiento, cuando se presente un patrón arbitrario de entrada, las neuronas de la capa oculta de la red responderán con una cierta activación, que propagándose hasta la capa de salida conseguirán activar aquellas neuronas en las que el patrón de entrada se asemeje a aquellas características que hayan aprendido a reconocer durante su entrenamiento.

Dos de las variables más significativas del Back-Propagation son el “ratio de aprendizaje”(ratio de aprendizaje) y el “momentum”, ambas modifican la manera en la que el algoritmo realiza el aprendizaje a partir de los datos. El ratio de aprendizaje es el ratio en el que los parámetros son modificados en la dirección del gradiente, mientras que el momentum es el ratio con el que el gradiente del paso anterior es usado en el paso actual.

- **Arquitectura híbrida Counter-Propagation**

Se trata de una arquitectura que combina tanto aprendizaje no supervisado como aprendizaje supervisado.

La primera parte del algoritmo es un mapa auto-organizado o mapa de Kohonen [\[11\]](#), este mapa es un tipo de red neuronal artificial que es entrenada usando aprendizaje no supervisado para producir una representación discreta del espacio de las muestras de entrada. Pasada una entrada a un mapa de tamaño  $M \times M$ , la salida puede tener dos formas, se puede devolver la posición  $X,Y$  de la neurona del mapa que se ha activado con esa entrada o una lista o array con todas las neuronas del mapa representadas como 0, excepto la que se activó que se marcará con un 1. Esta capa permite simplificar el problema inicial de forma que la siguiente capa pueda resolverlo mas fácilmente.

La salida del mapa en forma de matriz se pasa a la segunda parte del algoritmo, formada por una red neuronal Perceptrón Simple [\[12\]](#), similar a back-propagation pero sin neuronas ocultas, cuyo número de entradas será el número de neuronas del mapa.

## 2.2 Herramientas

Para este proyecto, se ha decidido usar la librería PyBrain. En las secciones posteriores se conocerá más a fondo sus características y funcionamiento, así como el por qué de su elección.

El objetivo de la librería PyBrain es ofrecer una variedad de algoritmos fáciles de usar y potentes para la realización de tareas de machine learning esto permite su uso tanto para estudiantes dando sus primeros pasos en la inteligencia artificial, así como para investigadores de primer nivel.

### **2.2.1 Algoritmos**

En este caso se analizarán los algoritmos de aprendizaje, tanto supervisado como no supervisado que contiene PyBrain, destacando especialmente aquellos basados en gradientes o métodos de exploración.

- **Aprendizaje supervisado**

El aprendizaje supervisado es una técnica para deducir una función a partir de datos de entrenamiento. Los datos de entrenamiento consisten de pares de objetos (normalmente vectores): una componente del par son los datos de entrada y el otro, los resultados deseados. El objetivo del aprendizaje supervisado es el de crear una función capaz de predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos, los datos de entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente.

#### **Back-Propagation (BPN)**

Uno de los algoritmos usados durante el proyecto, ya explicado a fondo en la sección 2.1.5.

#### **R-Prop**

R-Prop es el acrónimo de “Resilient Back-Propagation”. En este caso se toma en cuenta el signo de la derivada parcial para cada peso para tratar de acelerar la búsqueda del óptimo. Si ocurre un cambio de signo de la derivada parcial de la función de error total comparado con la anterior iteración el valor de actualización para ese peso se multiplicará por un factor  $\eta^- < 1$ . Si no existe un cambio de signo se multiplica el valor de actualización por un factor  $\eta^+ > 1$ .  $\eta^+$  suele ser considerado empíricamente como 1,2 y  $\eta^-$  como 0,5.

## **Support Vector Machines**

Una Support Vector Machine (SVM) es un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio posible. Cuando las nuevas muestras se ponen en correspondencia con dicho modelo, en función de su proximidad pueden ser clasificadas a una u otra clase.

Más formalmente, una SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que puede ser utilizado en problemas de clasificación o regresión. Una buena separación entre las clases permitirá una clasificación correcta.

- **Aprendizaje no supervisado**

Un algoritmo de aprendizaje no supervisado se distingue del aprendizaje supervisado por el hecho de que no hay un conocimiento a priori. En este caso, un conjunto de datos de objetos de entrada es tratado como un conjunto de variables aleatorias, siendo construido un modelo de densidad para el conjunto de datos.

## **K-Means Clustering**

El clustering o agrupamiento por K-means tiene como objetivo la partición de un conjunto de  $n$  observaciones en  $k$  grupos en el que cada observación pertenece al grupo más cercano a la media.

## **PCA/pPCA**

El “Principal Component Analysis” (Análisis de Componentes Principales en español) es una técnica estadística utilizada para reducir la dimensionalidad de un conjunto de datos. Técnicamente, busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados.

El pPCA es el enfoque probabilístico del PCA, usando como estimador de máxima verosimilitud un algoritmo EM (Expectation-Maximization).

## **LSH**

El “Locally-Sensitive Hashing” es un algoritmo para la resolución del problema Nearest Neighbor Search en espacios de alta dimensionalidad separando las entradas de manera que cada una de ellas sea mapeada en una casilla con alta probabilidad, siendo el número de casillas mucho menor al universo de posibles entradas.

## Deep Belief Network

Una “Deep Belief Network” (DBN) (Red de Creencia Profunda) es un tipo de red neuronal compuesto por múltiples capas de variables latentes, con conexiones entre las capas, pero no entre unidades dentro de cada capa. Una variable latente es una variable que no se observa directamente, sino que es inferida a partir de otras variables.

Si una DBN es entrenada de manera no supervisada con un conjunto de ejemplos de entrada puede aprender a reconstruir probabilísticamente dichas entradas. Puede ser entrenada de manera supervisada después de este paso para mejorar la fase de clasificación.

- **Redes Neuronales Híbridas**

### Counter-Propagation

Uno de los algoritmos usados durante el proyecto, ya explicado a fondo en la sección 2.1.5.

En este caso es necesario indicar que la herramienta PyBrain no contiene este algoritmo explícitamente, sino que se ha desarrollado un script que combina la implementación del mapa de Kohonen que sí contiene PyBrain con un BPN sin capa oculta, de manera que se puede simular el comportamiento de un CPN.

### 2.2.2 Funcionamiento de la herramienta

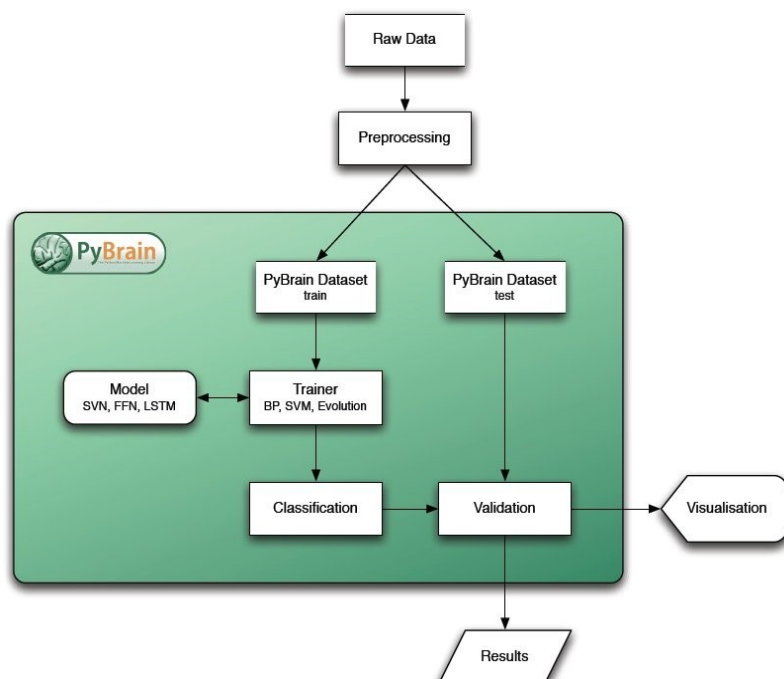


Figura 2.1: Diagrama de flujo de uso de la librería PyBrain

Como se puede ver en este diagrama de flujo de la figura 2.1 la herramienta comienza convirtiendo unos datos ya preprocesados en 'Datasets', unos tipos de datos propios de la herramienta. A continuación, se pasa ese Dataset a un Trainer creado con anterioridad que, como su nombre indica, se encargará de entrenar el modelo de Red Neuronal con los datos proporcionados. Por último, se procede a la clasificación y a la validación de la red con el Dataset de testeo.

Ahora se mostrarán unos ejemplos de código en la consola de Python para la creación de redes, trainers, datasets, etc.

- `patternDS = SupervisedDataSet(numInputs, numOutputs)`

Este ejemplo muestra la creación de un Dataset, donde los dos parámetros pasados al constructor son el número de parámetros distintos que se observarán y las distintas salidas que podría tener la red, por ese orden.

- `net = buildNetwork(numInputs, numHidden, numOutputs, bias = True)`

Las redes se crean con la función "buildNetwork" al que se deben pasar el número de parámetros, el número de neuronas ocultas que se desean, las posibles salidas e indicar si se quiere usar un Bias.

- `trainer = BackpropTrainer(net, trainDS, learningrate = myLearningRate, momentum = myMomentum)`

El trainer es bastante sencillo, solo es necesario que se le pasen la red, el dataset de entrenamiento y los parámetros de ratio de aprendizaje y momentum, que suelen ser menos que 1 y mayores que cero.

- `trainerror = trainer.trainUntilConvergence(verbose = True, trainingData = trainDS, validationData = validDS, maxEpochs = 10)`

Por último, se entrena la red a través de una de las funciones del trainer, en este caso se indica si se desea usar el modo Verbose, los dataset de entrenamiento y validación, así como el número máximo de iteraciones. En 'trainerror' se contendrán dos vectores con los errores de entrenamiento y validación respectivamente que podrán ser representados en una gráfica posteriormente para tener una representación visual del resultado del entrenamiento.

### **2.2.3 ¿Cómo se llegó a elegir PyBrain?**

Obviamente se barajaron varias alternativas antes de que la balanza se inclinara por utilizar PyBrain. La primera duda fue si escoger una herramienta con entorno gráfico o una de las muchas librerías para varios lenguajes que facilitan la programación de redes neuronales. Se decidió usar una librería por su flexibilidad, las herramientas con entorno gráfico, a pesar de su facilidad de uso, suelen no incluir todas las características posibles. Por poner un ejemplo, un programa podría no incluir un visor para las gráficas de aprendizaje, pero con una librería se pueden añadir funcionalidades con otras librerías y realizar dichas tareas.

Una vez seleccionado esto, era necesario elegir un lenguaje de programación en el que buscar una librería y se eligió Python debido a que es un lenguaje interpretado y no compilado y para realizar los cientos de pruebas que son necesarias implican cientos de ejecuciones y es mejor que sea en un script.

Las búsquedas de librerías de redes neuronales para Python arrojaron dos resultados principales: PyBrain y Neurolab. Esta última fue descartada por dos motivos:

- PyBrain funciona de manera que creas la red neuronal y posteriormente se le aplica el algoritmo de aprendizaje, pero Neurolab relaciona ambas cosas con una misma función del constructor, por lo que se hace imposible aplicar varias funciones de aprendizaje a la misma red sin crearla dos veces.
- PyBrain tiene su propio tipo de datos para gestionar las relaciones entre las entradas y los targets en el aprendizaje supervisado, los 'Datasets', algo que Neurolab no tiene, lo que dificulta la gestión de los datos.

### **2.2.4 Conclusiones**

La librería es bastante completa para lo que se desea realizar en este proyecto, contando con varios métodos de entrenamiento de redes neuronales supervisadas, métodos de visualización externos usando la librería 'matplotlib' y realiza todo esto bastante ágilmente, pero se echa en falta una interfaz gráfica de usuario como con la que cuenta el Microsoft Azure Machine Learning Studio que ayude a una mejor comprensión de los datos, y el flujo del proceso.

## **2.3 Datos**

En este apartado se presentan los datos proporcionados para el



proyecto, explicando su obtención y contenido.

### **2.3.1 Obtención**

Las bases de datos fueron proporcionadas por el Heuristic and Evolutionary Algorithm Laboratory o HEAL, un laboratorio de investigación austriaco experto en algoritmos heurísticos. Los datos son los mismo que usó dicho laboratorio para un proyecto propio en el que elaboraron un diagnosticador para los mismo tipos de cáncer que se tratan en este proyecto, pero en su caso usando árboles genéticos en lugar de redes neuronales.

Se recibieron tres ficheros formato CSV (comma-separated values), uno para cada tipo de cáncer, conteniendo cada uno de ellos distinto número de entradas: 706 entradas para el fichero de cáncer de mama, 905 entradas para el fichero de melanoma y 2363 entradas para el fichero de cáncer de pulmón.

Los ficheros están organizados de manera que cada fila es una entrada representado a un paciente y las columnas representan variables a analizar.

### **2.3.2 Contenido de la Base de datos**

Los tres ficheros contienen datos para casi 4000 pacientes y 32 variables en total. Estas variables son:

- Variables no relacionadas con análisis sanguíneos: *Cancer (1/0), Edad, sexo*
- Variables relacionadas con los análisis sanguíneos: *AST, ALT, GT37, BUN, Creatinina, CH37, LD37, Ácido Úrico, Bilirrubina, Colesterol, Colesterol HDL, Hierro, Ferritina, Transferrina, CRP, BSG1, Leucocitos, Neutrófilos, Linfocitos, Monocitos, Eosinófilos, Basófilos, Eritrocitos, Hemoglobina, Hematocritos, MCV, Plaquetas.*

Para cada tipo de cáncer el número de pacientes sanos y con cáncer en la base de datos difiere, por eso, en la tabla 2.1 se indica dicho ratio.

	Cáncer de mama	Melanoma	Cáncer de pulmón
Pacientes enfermos	377 (54,11%)	314 (39,30%)	628 (31,48%)
Pacientes sanos	320 (45,89%)	485 (60,70%)	1367 (68,52%)
Total	697	799	1995

Tabla 2.1: Ratio de pacientes con y sin cáncer en las tablas de la base de datos

### **2.3.3 Resultados de la investigación del HEAL.**

La investigación llevada a cabo por el HEAL no se basa en redes neuronales, sino en árboles de decisión genéticos, pero fueron usados los mismos datos que se usarán en este proyecto, así que es una gran oportunidad de comparar los resultados de ambos.

En la figura 2.2 se muestran los resultados de HEAL, para en un capítulo final, llevar a cabo la comparación entre estos y los de este proyecto.

	Training			Test		
Criteria	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
Best Accuracy	81.87%	89.79%	72.53%	74.36%	84.29%	62.65%
Best Sensitivity	76.63%	95.29%	54.63%	74.08%	92.67%	52.16%
Best of Both	77.05%	95.55%	55.25%	74.36%	91.88%	53.70%

Table 6: Breast cancer ensemble results

	Training			Test		
Criteria	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
Best Accuracy	83.20%	79.52%	86.39%	73.92%	66.19%	80.62%
Best Sensitivity	83.65%	81.43%	85.57%	73.04%	67.14%	78.14%
Best of Both	83.20%	80.24%	85.77%	73.15%	66.67%	78.76%

Table 7: Melanoma ensemble results

	Training			Test		
Criteria	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity
Best Accuracy	92.47%	87.25%	96.27%	87.26%	76.71%	94.95%
Best Sensitivity	92.85%	89.06%	95.61%	87.64%	78.31%	94.44%
Best of Both	92.93%	88.96%	95.83%	87.30%	77.61%	94.37%

Table 8: Respiratory system cancer ensemble results

Figura 2.2: Tablas originales obtenidas de la presentación de resultados del proyecto del HEAL en el EUROCAST 2015 – Capítulo “Computer Aided Systems Theory pp 350-358”

# Capítulo 3

## Desarrollo y análisis

En este apartado se mostrarán los análisis en profundidad llevados a cabo para optimizar cada una de las redes neuronales para cada uno de los algoritmos usados, así como la fase de desarrollo en la cual se realizó el preprocesado de la base de datos, así como el desarrollo de los scripts.

### 3.1 Desarrollo

Durante la fase de desarrollo se llevó a cabo el preprocesado de la base de datos, así como la elaboración de los scripts que se usarían para entrenar las redes neuronales.

#### 3.1.1 Preprocesado

Este apartado se dividirá en cada uno de los métodos seguidos para el preprocesamiento hasta obtener las base de datos listas para ser leídas por la red neuronal. Dicho preprocesamiento de las bases de datos fue llevado a cabo en el software Weka [\[13\]](#).

- **Eliminación de columnas poco significativas**

En primer lugar se decidió eliminar las columnas que tuvieran más de un 50% de datos nulos, ya que se consideró que serían poco significativas e influirían poco en el entrenamiento de la red neuronal, aunque en primer lugar hubo que sustituir los valores nulos “-1” por “”(vacío) para que el filtro de Weka los reconociera.

A su vez se aprovechó para eliminar columnas que no eran necesarias para el proyecto, como la que indicaba el número de identificación de paciente. Las únicas variables no relacionadas con datos sanguíneos que se mantuvieron por ser útiles para el proyecto fueron: Edad, Sexo y Cancer (0/1).

También fueron eliminadas filas donde, al igual que las columnas,

muchos de los datos estuvieran perdidos, habiendo sido eliminado un 1,27% de los datos para el cáncer de mama, un 11,71% de los datos para el Melanoma y un 15,57% de los datos para el cáncer de pulmón.

- **Sustitución de datos nulos**

Para las columnas restantes, aunque tuvieran menos del 50% de datos nulos o perdidos, aun había muchos, por lo que fue necesario realizar un filtro de sustitución de datos nulos. Este filtro sustituye los datos a '0' por la media de la columna de manera que afecten de la menor manera posible a lo significativos que son dichos datos.

- **Normalización de los datos**

Para un mayor entendimiento de los datos por parte de las redes neuronales se normalizarán los resultados por columnas para que se sitúen entre los valores 0 y 1, también usando un filtro de Weka.

- **Eliminación de cabeceras**

Por último fue necesario eliminar las cabeceras que indicaban a qué pertenecía cada columna, ya que impedían la correcta lectura por parte del script de Python.

### **3.1.2 Desarrollo de Scripts**

Como se ha dicho anteriormente, los scripts se elaboraron en Python y usando la librería PyBrain. Todos los scripts están divididos en tres partes bien diferenciadas:

- Lectura de datos y creación de datasets
- Creación y entrenamiento de la red neuronal
- Cálculo e impresión de resultados

Todos los códigos se pueden observar en el repositorio Github del proyecto [\[9\]](#).

Fueron elaborados distintos códigos para los análisis y para los resultados finales, pero son simples modificaciones del código inicial para formatear la salida de manera deseada o realizar cálculos en bucle.

## **3.2 Pruebas a realizar y parámetros estudiados**

Se detallan a continuación las pruebas a realizar y los parámetros estudiados en los dos modelos de RNAs analizados.

### **3.2.1 Back-Propagation**

Para el algoritmo de Back-Propagation se analizarán tres variables: neuronas ocultas, ratio de aprendizaje y momentum, a parte de una prueba inicial para comparar con los resultados finales.

#### **Primera Prueba**

Las pruebas iniciales serán una primera implementación de la red usando valores predeterminados, que serán 25 neuronas ocultas, 0,0001 de ratio de aprendizaje y 0,1 de momentum, llegando a un máximo de 100 iteraciones de entrenamiento. Estas pruebas servirán de referencia de partida que intentar mejorar con los análisis posteriores. Los porcentajes obtenidos serán calculados con el dataset de validación.

#### **Neuronas Ocultas**

Posteriormente, se procederá a hacer pruebas con varios números distintos de neuronas ocultas, para observar en cuál de ellos la red neuronal tiene un mejor resultado. Las pruebas se realizarán modificando el número de neuronas ocultas de 5 a 100 con pasos de 5 neuronas (5, 10, 15,...,100), pero dejando el resto de variables con el mismo valor que en la prueba inicial, es decir, un ratio de aprendizaje de 0,0001 y un momentum de 0,1, ya que estas variables influyen en el aprendizaje de la red y no en la red en sí misma.

Para cada número de neuronas se harán 5 pruebas obteniendo la media de ellos como representante de la instancia en cuestión.

Las pruebas se realizarán usando los valores de sensibilidad y especificidad. La sensibilidad indica la capacidad del estimador para dar como casos positivos los casos realmente enfermos, es decir, la proporción de enfermos correctamente identificados. Mientras que la especificidad indica todo lo contrario, la proporción de sanos correctamente identificados.

En este caso se buscará un valor de sensibilidad lo más alto

posible sin despreciar tampoco el valor de especificidad, ya que es bastante más importante detectar correctamente a un paciente que tiene cáncer que detectar como sano a un paciente que no lo tiene.

### **Ratio de aprendizaje**

En siguiente lugar, se realizará el análisis de la variable ratio de aprendizaje. Hasta ahora ha tenido un valor de 0,0001, pero se procederá a realizar pruebas para obtener el mejor resultado posible. Las pruebas consistirán en hallar con qué valor de ratio de aprendizaje se alcanza un error determinado en menos tiempo.

Observando valores de error en las pruebas previas se ha decidido que dicha cota sea 0,13, por lo cual el script calculará la iteración en la que el error baja de dicho valor. Existen casos en los que el error no llega a bajar de ese valor durante las iteraciones que tiene determinadas el entrenamiento, por lo cual se harán 5 pruebas por cada valor de ratio de aprendizaje usando el mejor de ellos como representante de esa instancia intentando ignorar los casos en los que no se llegue a ese límite en el caso del error de entrenamiento como en el error de validación.

Se ha decidido limitar el entrenamiento a 200 iteraciones, ya que aproximadamente a esa altura el valor de error será continuo. Se analizará el comportamiento de la red neuronal con los siguientes valores de ratio de aprendizaje: 0,0001, 0,0005, 0,001, 0,005, 0,01 y 0,05. Se pretendía realizar pruebas con valores mayores, pero la red neuronal provocaba un desbordamiento.

### **Momentum**

Por último, se realizará el análisis de la única variable que falta para el algoritmo de Back-Propagation. Dicho análisis será muy similar al realizado con el ratio de aprendizaje, ya que se fijará como objetivo en reducir el número de iteraciones posibles.

En esta ocasión se realizarán estudios desde 0,1 hasta 0,9 de momentum, como siempre con 5 iteraciones para cada instancia, seleccionando el mejor como representante de la misma.

### **3.2.2 Counter-Propagation**

Para el algoritmo de Counter-Propagation se analizarán tres variables: neuronas del mapa y el ratio de aprendizaje y momentum del perceptrón simple. Previamente como en el caso anterior se realizará una prueba inicial como referencia.

### **Primera prueba**

Para la primera prueba del análisis se usarán datos intermedios de los que se planea analizar, por ello la variable de ratio de aprendizaje de la capa de el perceptrón simple tomará 0,005, la de momentum 0,5 y el número de neuronas del mapa será de 10.

### **Neuronas del mapa auto-organizado**

Las pruebas con las neuronas del mapa se realizarán comenzando en 5, lo que generará un mapa de 25 neuronas (5x5) avanzando de uno en uno hasta llegar a 15 generando un mapa de 15x15 neuronas. Se analizarán los resultados finales de accuracy, sensibilidad y especificidad, realizándose cinco pruebas con cada numero de neuronas y obteniendo la media como representante de esa instancia.

### **Ratio de aprendizaje**

El estudio para la variable ratio de aprendizaje será prácticamente idéntica al algoritmo anterior, excepto que variará el umbral a partir del cual se obtendrá la iteración límite. En este caso, observando los valores de error de las pruebas previas se usará un umbral de 0,13.

### **Momentum**

Los análisis para el momentum también serán idénticos a los realizados para el caso del algoritmo Back-Propagation, utilizando el mismo umbral que el usado con el análisis del ratio de aprendizaje.

## **3.3 Cáncer de mama**

El dataset de validación para el cáncer de mama con el que se realizaron las pruebas contiene un 53,41% de casos positivos y un 46,59% de casos negativos. Las tablas de datos para cualquier análisis se podrán consultar en el Github del proyecto [\[9\]](#).



### 3.3.1 Back-Propagation

- **Primera prueba**

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibilidad	Especificidad
83	45	37	11	0,723	0,883	0,549

Tabla 3.1: Resultados de la primera prueba de Back-Propagation para cáncer de mama

La primera prueba de la tabla 3.1 arroja resultados decentes, pero la especificidad es demasiado baja, a pesar de que la sensibilidad es bastante buena.

- **Análisis de neuronas ocultas**

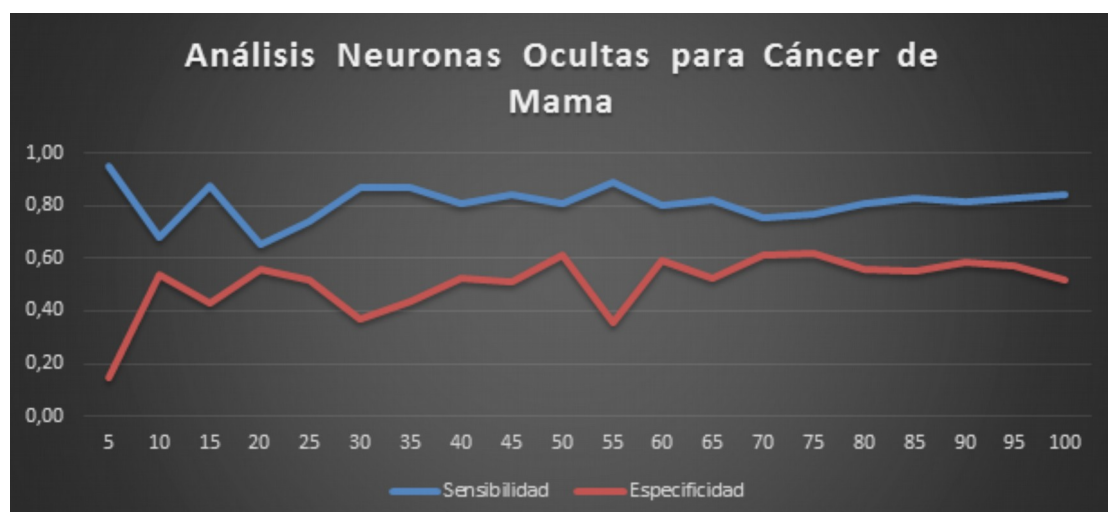


Figura 3.1: Resultados del análisis de neuronas ocultas en cáncer de mama

Con los resultados de la figura 3.1 se ha decidido **usar 50 neuronas ocultas**. Con este número de neuronas la sensibilidad es de un 81% y la especificidad de un 61%. A pesar de no ser la sensibilidad más alta de la tabla, tiene la mejor relación, entre las dos variables, ya que, sin perder mucha sensibilidad con respecto a los mayores valores, se gana bastante especificidad.

- **Análisis de ratio de aprendizaje**

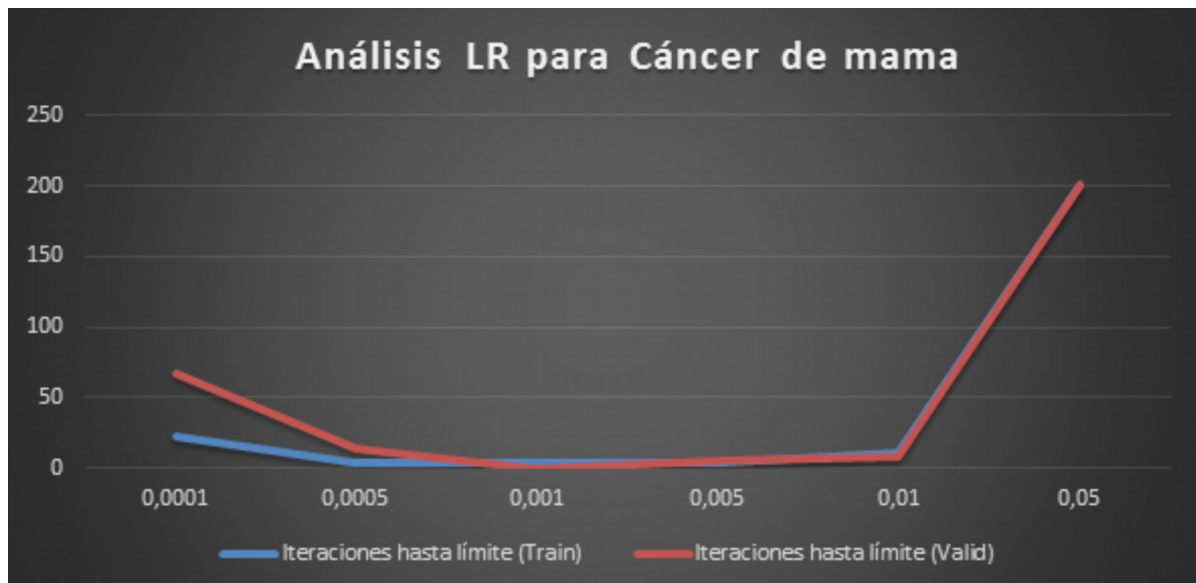


Figura 3.2: Resultados del análisis de ratio de aprendizaje para cáncer de mama con Back-Propagation

Por los resultados de la figura 3.2 se utilizará **0,005 de ratio de aprendizaje**. Con este valor se obtuvo el menor número de iteraciones para el error de validación (5) y el segundo mejor para el error de entrenamiento (4).

- **Análisis de Momentum**

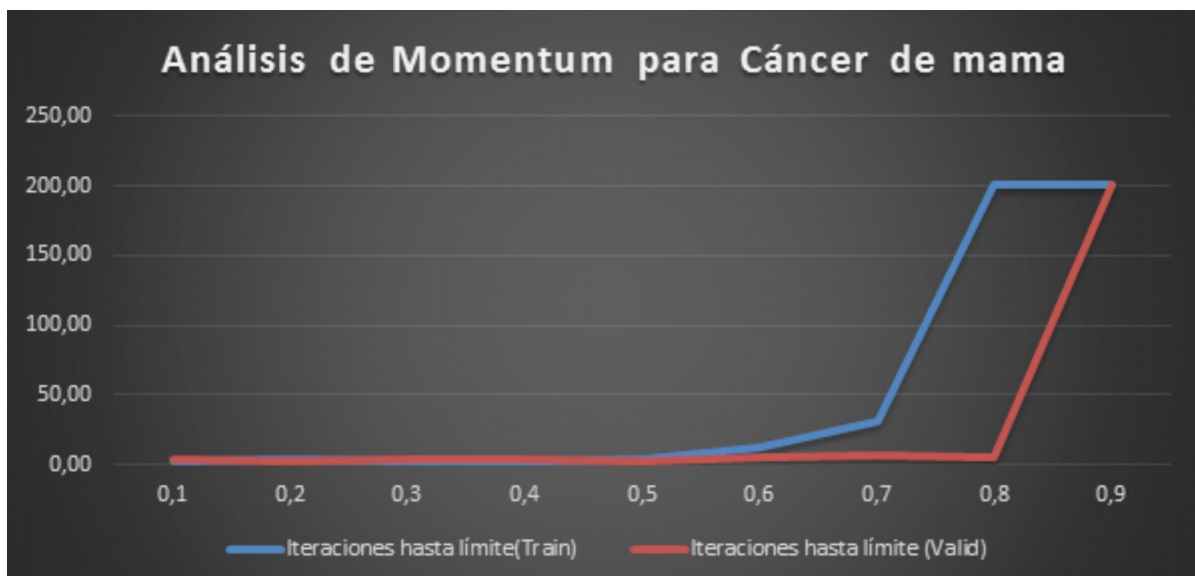


Figura 3.3: Resultados del análisis de momentum para cáncer de mama con Back-Propagation

Durante los cinco primeros casos de estudio de la figura 3.3, los resultados son prácticamente idénticos, variando entre 3 y 2 iteraciones, tanto para entrenamiento como para validación, pero en el caso de **0,1 de momentum** los resultados de las cinco pruebas son más homogéneos por lo que se ha decidido usar ese valor.

### 3.3.2 Counter-Propagation

- **Primera prueba**

Positivos	Negativos	Falsos positivos	Falsos negativos	Accuracy	Sensibilidad	Especificidad
83	28	54	11	0,627	0,883	0,341

Tabla 3.2: Resultados de primera prueba de cáncer de mama para Counter-Propagation

Los resultados de la tabla 3.2 en este caso son un poco peores a los presentados en la primera prueba de Back-Propagation con un 10% menos de accuracy.

- **Análisis de neuronas del mapa auto-organizado**

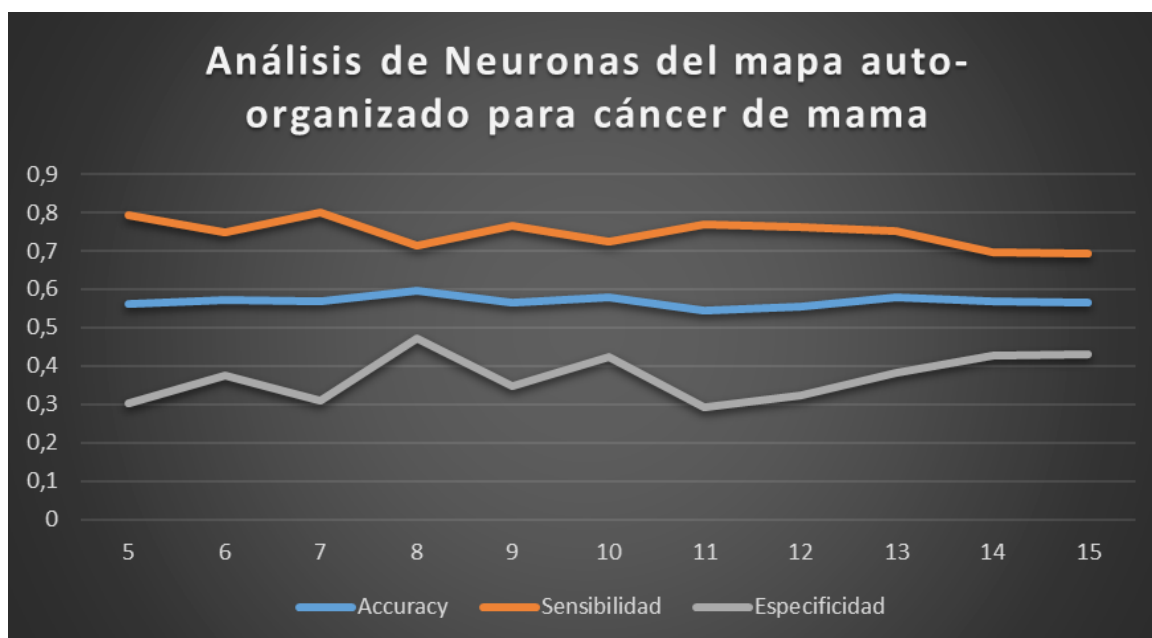


Figura 3.4: Resultados del análisis de neuronas del mapa auto-organizado para cáncer de mama

Estos resultados de la figura 3.4 no son tan buenos como se esperaba, alcanzando por poco el 60% de accuracy de media. Se ha decidido usar **10 neuronas**, cuyos resultados son de media: 58% de accuracy, 72% de sensibilidad y 42% de especificidad.

- **Análisis de ratio de aprendizaje**

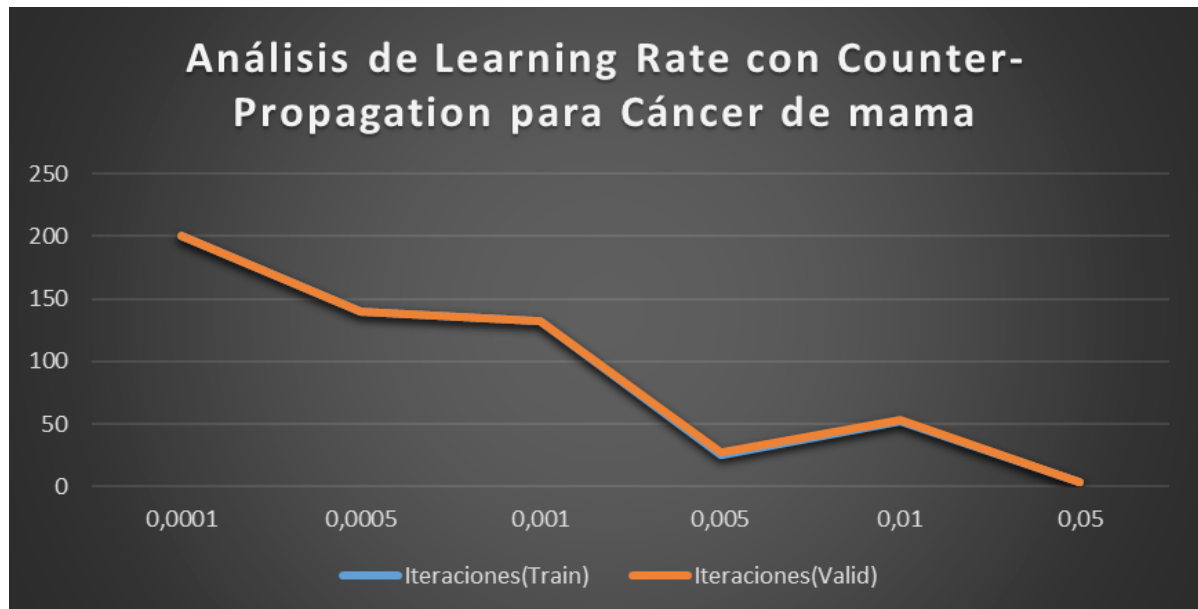


Figura 3.5: Resultados del análisis del ratio de aprendizaje para cáncer de mama con Counter-Propagation

Se puede ver en la figura 3.5 como la iteración en la que baja del 0,13 de error va disminuyendo a medida que el ratio de aprendizaje se acerca a 1. por ello se elegirá **usar 0,05 de ratio de aprendizaje** ya que es el menor de todos.

- **Análisis de momentum**

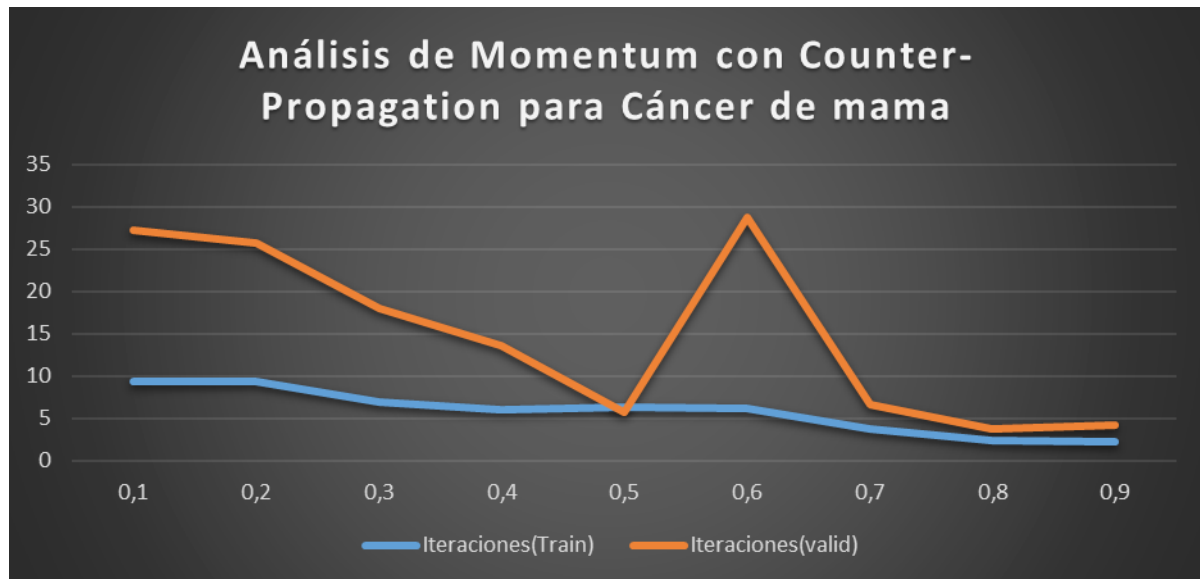


Figura 3.6: Resultados del análisis de momentum para cáncer de mama con Counter-Propagation

En este caso, también se ve claramente que, en la figura 3.5 a medida que aumenta el momentum, el número de iteraciones antes del umbral disminuye, se podrían usar los valores 0,8 o 0,9 ya que sus resultados son bastante idénticos, pero se ha decidido usar **0,9 de momentum** por tener un mejor accuracy sus resultados.

## 3.4 Melanoma

### 3.4.1 Back-Propagation

La base de datos de validación para este tipo de cáncer contiene un 42,92% de pacientes con cáncer y un 57,08% de pacientes sin cáncer.

- **Primera prueba**

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibilidad	Especificidad
78	66	63	19	0,634	0,804	0,512

Tabla 3.3: Resultados de primera prueba de Back-Propagation para melanoma  
Los resultados de la tabla 3.3 en este caso son un poco peores a

los de las primeras pruebas para cáncer de mama.

- **Análisis de Neuronas Ocultas**

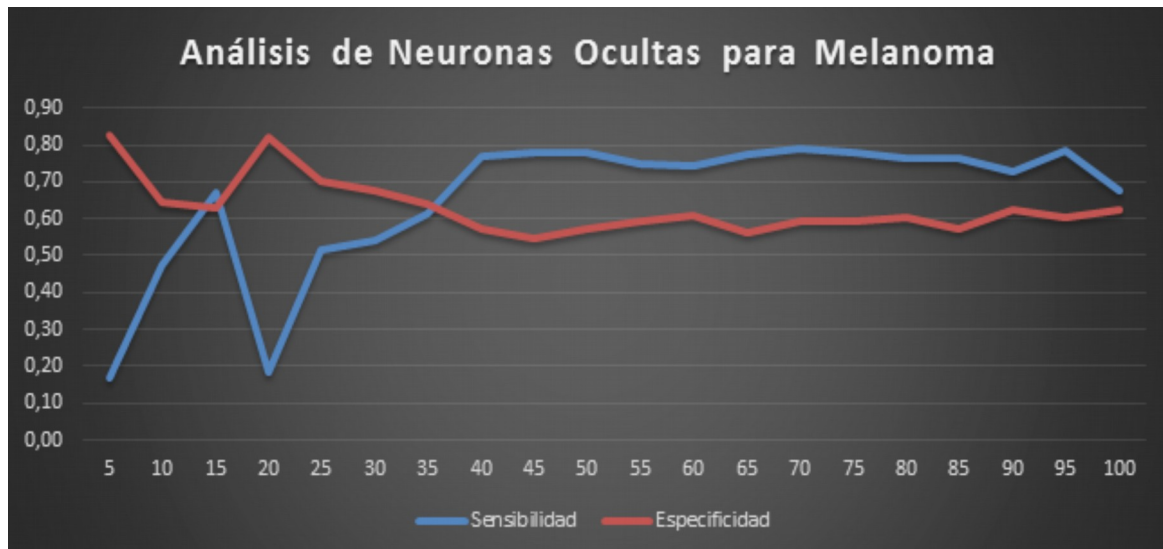


Figura 3.7: Resultados de análisis de neuronas ocultas para melanoma

Los resultados de la figura 3.7 sobre el análisis de neuronas ocultas para melanoma son peores que el cáncer de mama en cuanto a sensibilidad, ya que aquí no superan el 80%. Esto puede deberse a la mayor existencia de casos de pacientes sanos que pacientes con cáncer en el conjunto usado para entrenar la red. Se elegirá **usar 95 neuronas ocultas**, esta instancia obtuvo la mejor sensibilidad (79%) y una muy buena especificidad (60%) en comparación.

- **Análisis de ratio de aprendizaje**

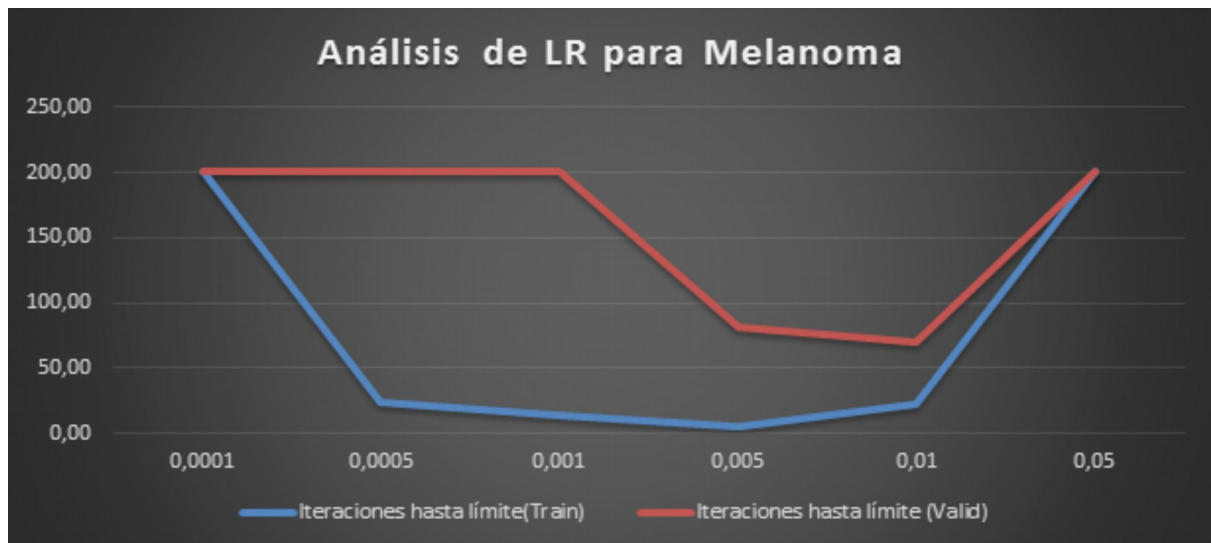


Figura 3.8: Resultados de análisis de ratio de aprendizaje para melanoma con Back-Propagation

En este caso se ven más dificultades en la figura 3.8 para alcanzar la cota de error de 0,13 que con el cáncer de mama. Se ha decidido seleccionar **0,005 de ratio de aprendizaje**, al igual que en caso anterior, ya que, a pesar de no tener el mínimo de iteraciones para el error de validación (82), sí que tiene el mínimo de iteraciones (5) para el error de entrenamiento y la relación es mejor.

- **Análisis de Momentum**

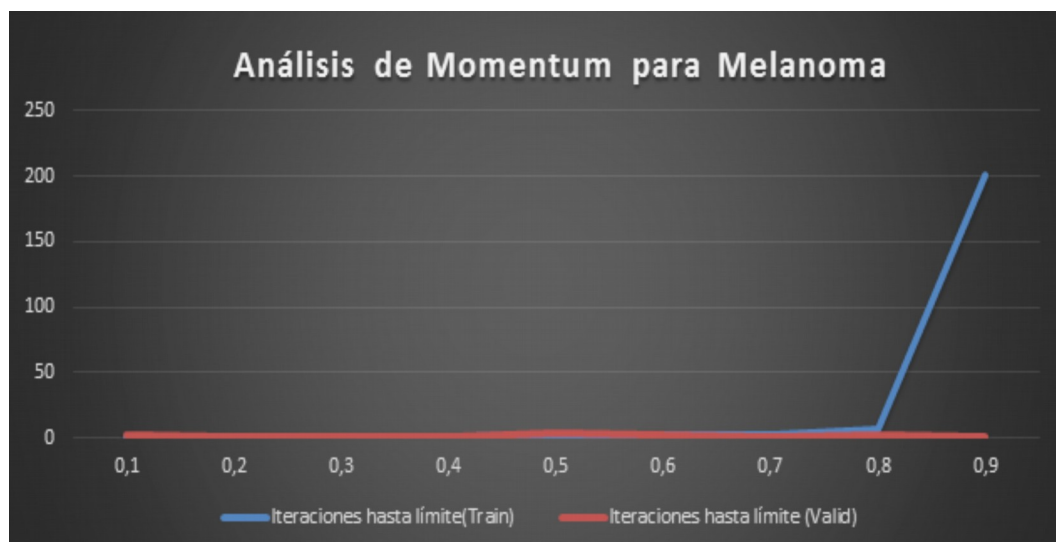


Figura 3.9: Resultados de análisis de momentum para melanoma con Back-Propagation

Como en el caso anterior las primeras pruebas de la figura 3.9 son bastantes parecidas, pero elegiremos alguna de las tres que tardan solo una iteración para ambos errores (0,2, 0,3 y 0,4). Teniendo en cuenta el resto de pruebas realizadas para cada uno de esas tres instancias, la más homogénea es **0,3 de momentum**.

### 3.4.2 Counter-Propagation

- **Primera prueba**

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibilidad	Especificidad
31	101	14	54	0,660	0,365	0,878

Tabla 3.4: Resultados de primera prueba de Back-Propagation para melanoma

La primera prueba para el Melanoma con la arquitectura neuronal Counter-Propagation de la tabla 3.4 mejora un poco los obtenidos con el Back-Propagation en cuanto a accuracy, pero tiene una sensibilidad demasiado baja para lo que se espera.

- **Análisis de neuronas del mapa auto-organizado**

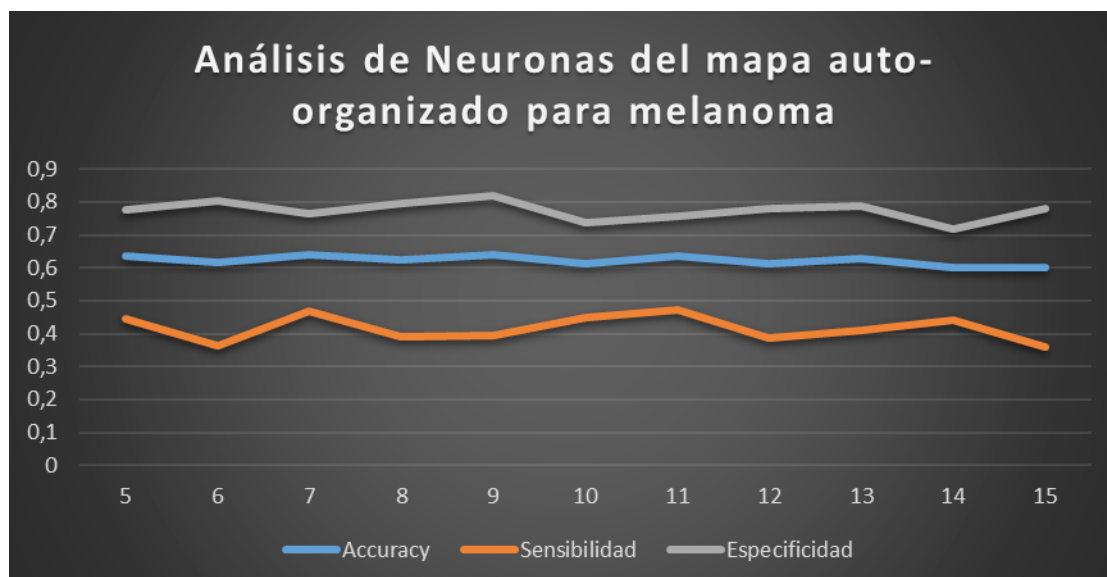


Figura 3.10: Resultados del análisis de neuronas del mapa auto-organizado para melanoma



Al igual que el análisis de las neuronas para el mapa auto-organizado para el cáncer de mama, en este caso, en la figura 3.10 son bastante estables desde el principio hasta el final. Se ha elegido como mejor resultado **11 neuronas** por tener la mejor sensibilidad de todas las pruebas (47%), para el segundo mejor accuracy (64%), ya que parece ser que la sensibilidad será bastante baja durante todas las pruebas para el melanoma.

- **Análisis de ratio de aprendizaje**

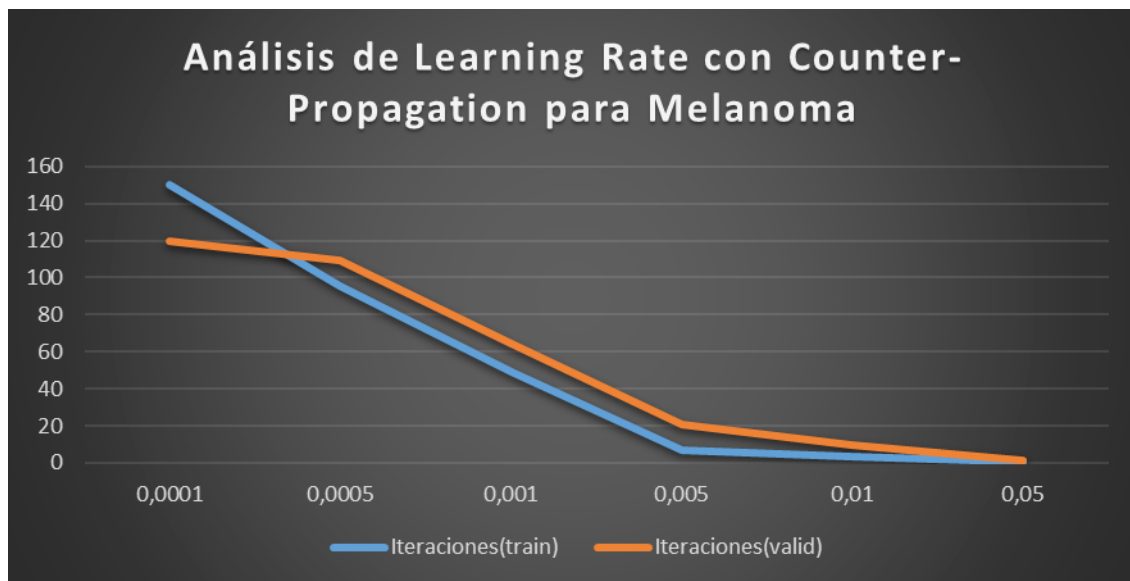


Figura 3.11: Resultados del análisis de ratio de aprendizaje para melanoma con Counter-Propagation

En la figura 3.11 se ve claramente como a medida que el ratio de aprendizaje se acerca a 1, mejora el número de iteraciones necesarias para alcanzar el límite. Sin lugar a duda, la mejor instancia es la de **0,05 de ratio de aprendizaje**, cuyo mejor resultado fue de 0,8 iteraciones de media para el error de entrenamiento y 1,2 para el error de validación.

- **Análisis de momentum**

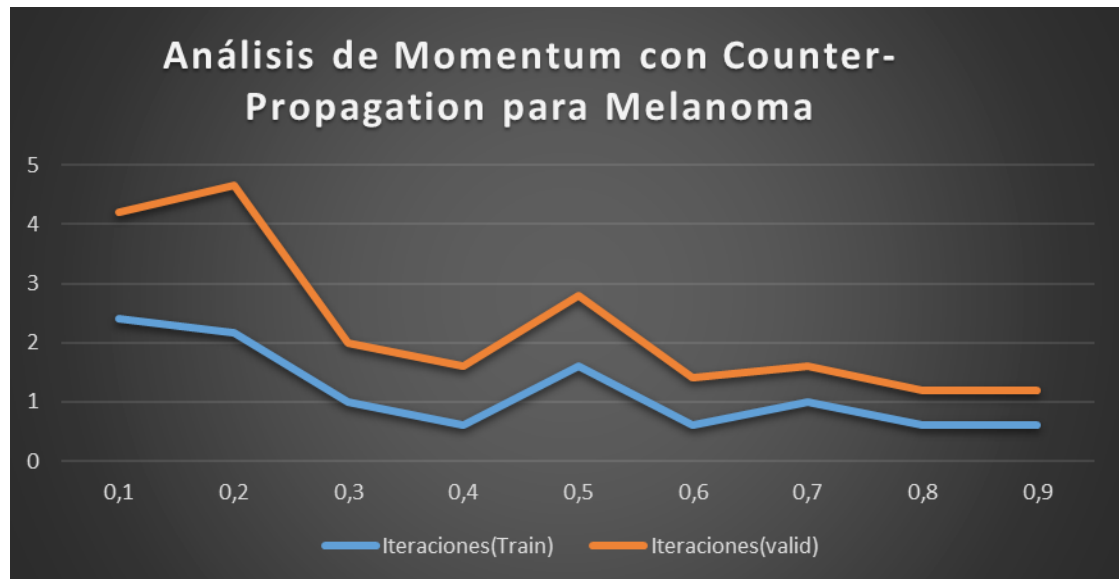


Figura 3.12: Resultados del análisis de momentum para melanoma con Counter-Propagation

Los resultados no parecen variar mucho con el paso del tiempo en la figura 3.12, excepto para las instancias de 0,8 y 0,9 cuyos resultados son idénticos y a su vez, los mejores, pero se ha decidido usar **0,9 de momentum** por tener un mejor accuracy sus pruebas.

## 3.5 Cáncer de pulmón

### 3.5.1 Back-Propagation

La base de datos para el cáncer de pulmón es la más extensa de las tres por lo que se esperan los mejores resultados de todos los casos. El conjunto de validación con el que se llevarán a cabo los análisis tiene un 40.78% de pacientes con cáncer y un 59.22% de pacientes sin cáncer.

- **Primera prueba**

Positivos	Negativos	Falso Positivos	Falso Negativos	Accuracy	Sensibilidad	Especificidad
166	294	56	75	0,778	0,689	0,846

Tabla 3.5: Resultados de primera prueba para cáncer de pulmón con Back-Propagation

La primera prueba de la tabla 3.5 es la mejor primera prueba de los tres cánceres para Back-Propagation con 77,8% de accuracy.

- **Análisis de Neuronas Ocultas**

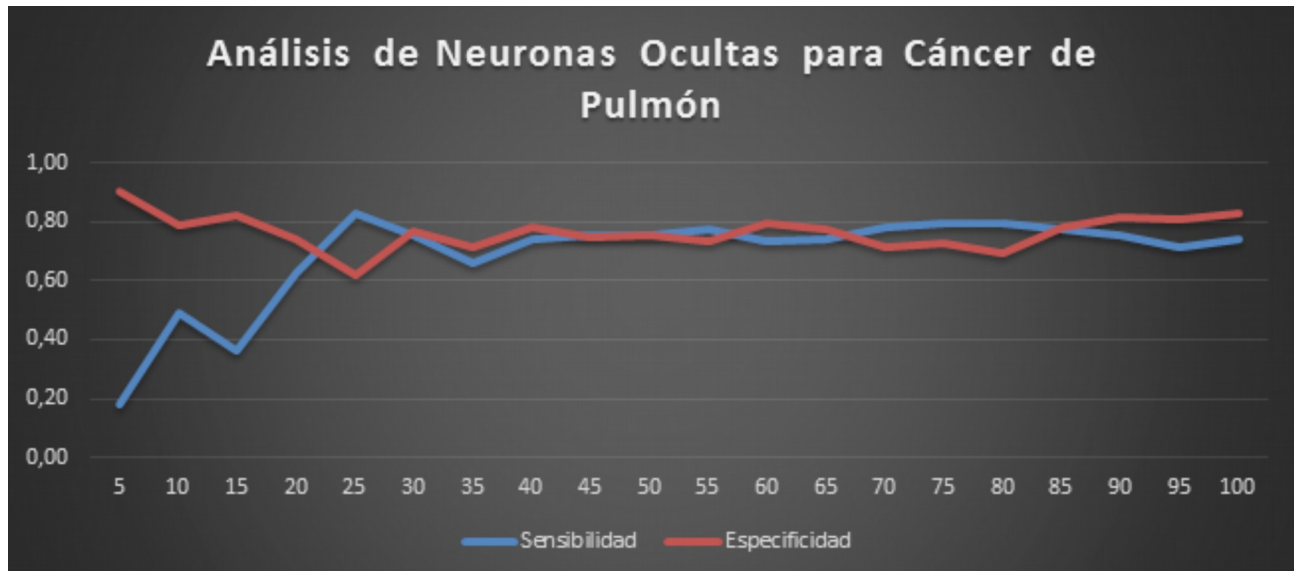


Figura 3.13: Resultados de análisis de neuronas ocultas para cáncer de pulmón

Obviamente por tener más casos con los que entrenar la red neuronal, en la figura 3.13 se demuestra que los resultados son mejores, no tanto en cuanto a sensibilidad, pero sí que se nota una gran diferencia en Especificidad. Se ha decidido usar **75 neuronas ocultas** con 80% de sensibilidad y 73% de especificidad, debido a que tiene la mayor sensibilidad, descartando el resultado con mejor relación entre ambas variables, que es el caso de las 100 neuronas ocultas, por considerar que dicha sensibilidad es muy baja, por lo que se prefirió perder 10 puntos de especificidad por ganar 6 en sensibilidad.

- **Análisis de ratio de aprendizaje**

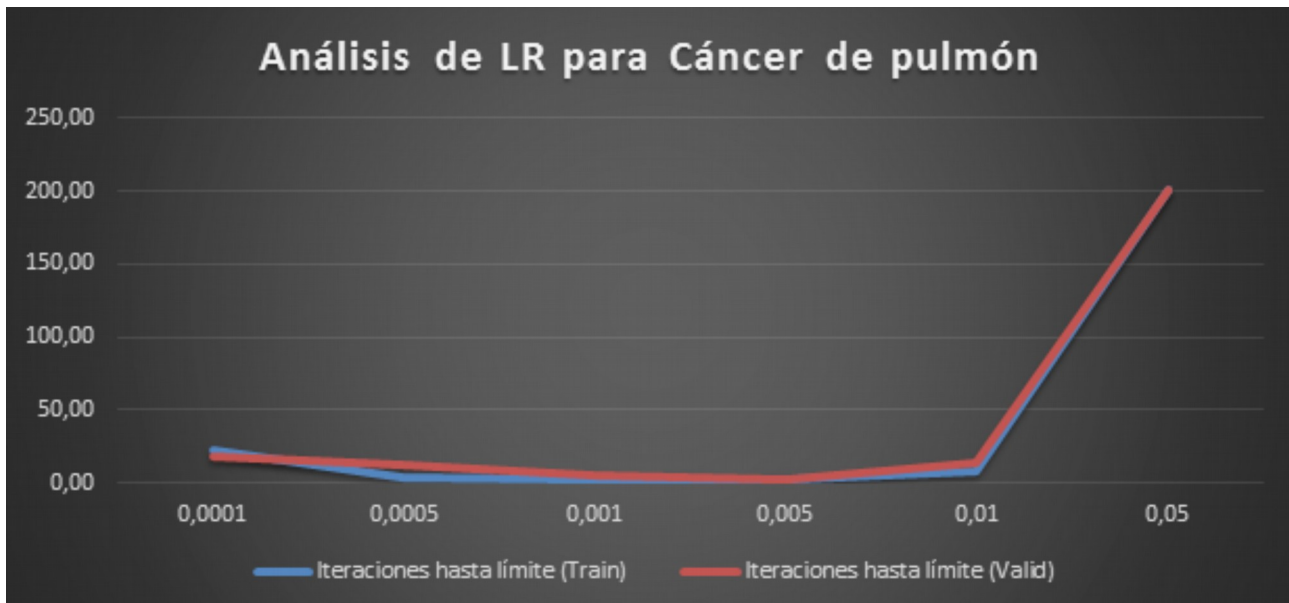


Figura 3.14: Gráfica de resultados de análisis de ratio de aprendizaje para cáncer de pulmón

De nuevo para el este cáncer se elegirá usar **0,005 de ratio de aprendizaje**, obteniendo los mejores resultados en la figura 3.14 en general, tal y como se esperaba con 2 iteraciones para alcanzar la cota, tanto en error de entrenamiento como en error de validación.

- **Análisis de Momentum**

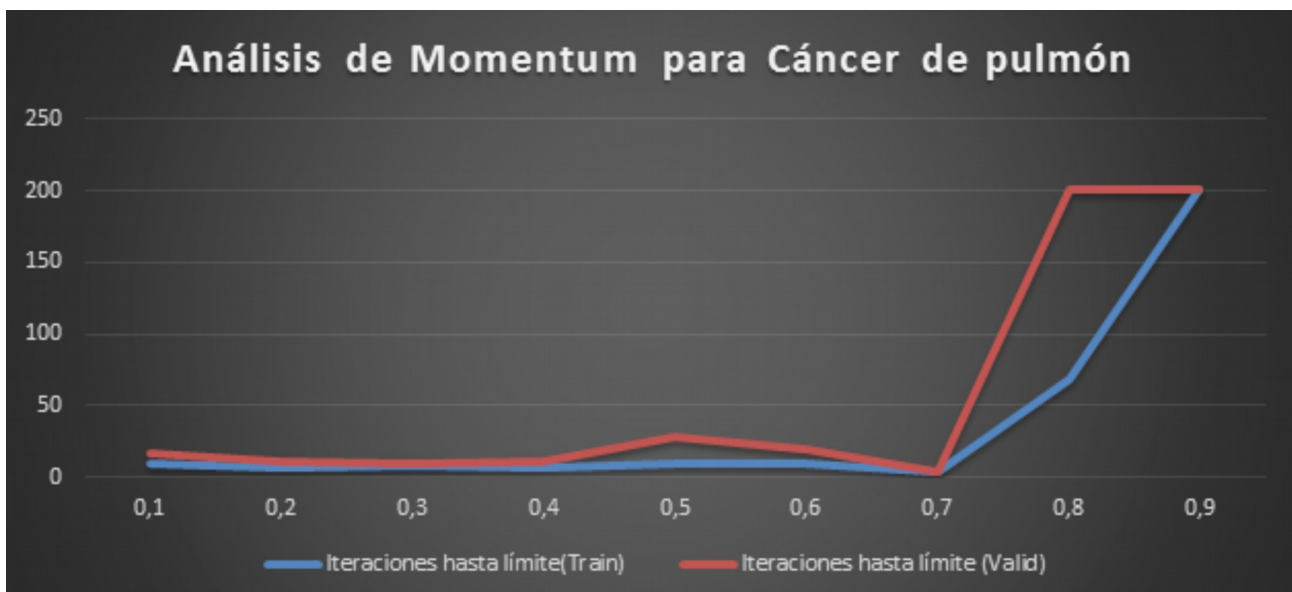


Figura 3.15: Resultados de análisis de momentum para cáncer de pulmón

En este último caso del cáncer de pulmón en la figura 3.15, los resultados son más dispares que los anteriores y solo hay un resultado mejor que todos los demás, para la instancia de **0,7 de momentum** con 1 iteración para entrenamiento y validación.

### 3.5.2 Counter-propagation

- **Primera prueba**

Positivos	Negativos	Falsos positivos	Falsos negativos	Accuracy	Sensibilidad	Especificidad
41	311	37	110	0,705	0,272	0,894

Tabla 3.6: Resultados de primera prueba para cáncer de pulmón con Counter-Propagation

Ya desde la primera prueba de la tabla 3.6 se puede vislumbrar que este algoritmo se comportará de manera parecida al melanoma por tener bajos niveles de sensibilidad, aunque su accuracy es un poco mejor, superando el 70%.

- **Análisis de neuronas para mapa auto-organizado**

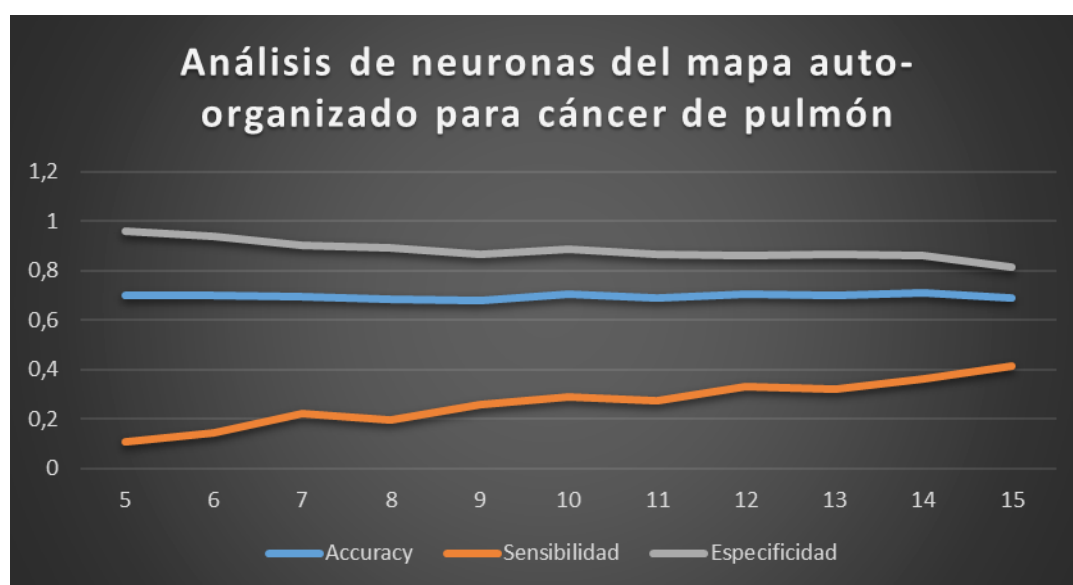


Figura 3.16: Resultados del análisis de neuronas del mapa auto-organizado para cáncer de pulmón

Se escogerá usar **15 neuronas**, ya que tiene la mejor sensibilidad de la figura 3.16. La baja sensibilidad general para estos datos con

este algoritmo obligan a escoger casi siempre la instancia que mejor sensibilidad consiga, ya que es necesario un alto valor de esta variable en este estudio.

- **Análisis de ratio de aprendizaje**

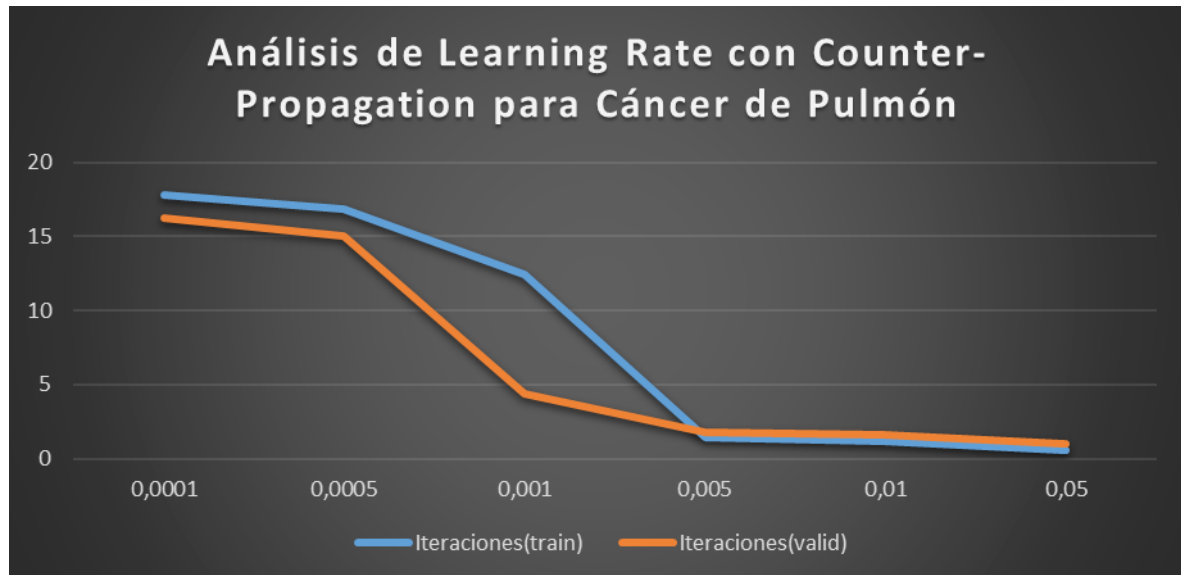


Figura 3.17: Resultados de análisis de ratio de aprendizaje para cáncer de pulmón con Counter-Propagation

Al igual que en los otros dos tipos de cáncer con este algoritmo, en la figura 3.17, el **ratio de aprendizaje será de 0,05**, donde se alcanza el umbral de error en tan solo dos iteraciones. Esta variable también fue la misma para los 3 casos del algoritmo Back-Propagation.

- **Análisis de momentum**

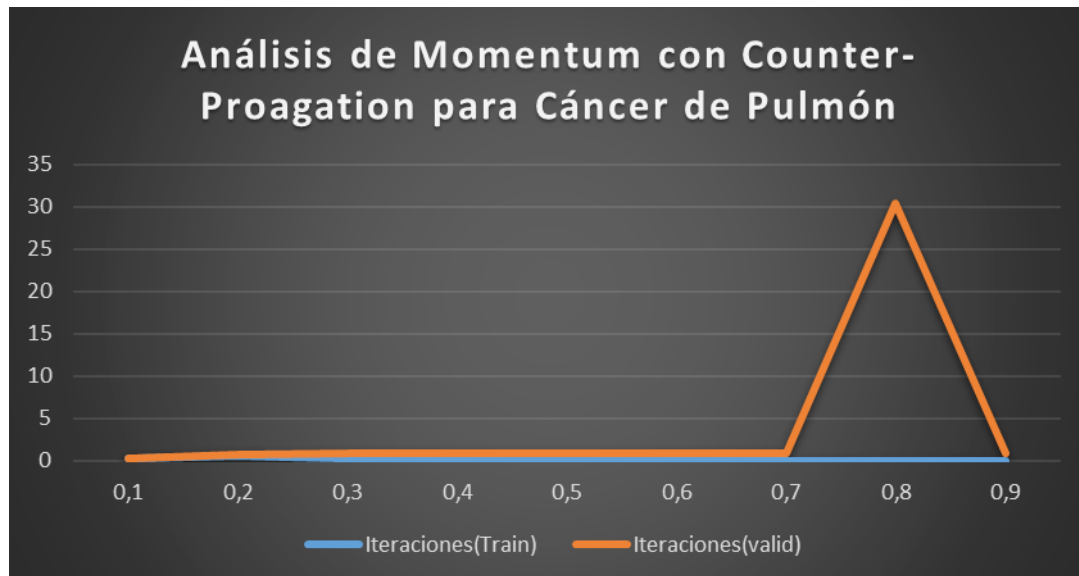


Figura 3.18: Resultados del análisis de momentum para cáncer de pulmón con Counter-Propagation

Como se puede observar en la figura 3.18 , las iteraciones medias para alcanzar el límite de error de 0,13 son prácticamente estables, solo alcanzando un pico en la instancia de 0,8. De hecho para las instancias de 0,5, 0,6, 0,7 y 0,9 los resultados medios son idénticos, por lo que se podría escoger cualquiera de ellas, pero se ha decidido escoger **0,9 de momentum** por tener la mejor sensibilidad y accuracy de todas ellas.

## 3.6 Resultados finales usando el subconjunto de test

### 3.6.1 Cáncer de mama

- **Back-Propagation**

Para el cáncer de mama, los análisis de las variables del algoritmo Back-Propagation dejaron los siguientes resultados:

Neuronas Ocultas	ratio de aprendizaje	Momentum
50	0,005	0,1

Tabla 3.7: Valor de variables finales para back-propagation en cáncer de mama

Con los valores de la tabla 3.7 se elaborará una red final, pero en este caso obteniendo los resultados con el dataset de testeo, en lugar de con el de validación. El dataset de testeo tiene un 60,45% de pacientes con cáncer y un 39,55 de pacientes sin cáncer.

Para este dataset y con los valores de variables anteriormente nombrados, los resultados fueron los de la tabla 3.8

Positivos	Negativos	Falsos Positivos	Falsos Negativo	Accuracy	Sensibilidad	Especificidad
80	37	33	27	0,661	0,748	0,529

Tabla 3.8: Resultados de test final con Back-Propagation para cáncer de mama

Se puede ver que en general, son un poco más bajos que los obtenidos durante el entrenamiento, algo normal, considerando un margen al trabajar con datos con los que no se había trabajado antes.

#### • Counter-Propagation

Para el cáncer de mama, los análisis de las variables del algoritmo Counter-propagation dejaron los siguientes resultados:

Neuronas del mapa	ratio de aprendizaje	Momentum
10	0,05	0,9

Tabla 3.9: Valor de variables finales para Counter-Propagation en cáncer de mama

Con estos valores de la tabla 3.9 se elaborará una red final, pero en este caso obteniendo los resultados con el dataset de testeo, en lugar de con el de validación. El dataset de testeo tiene un 60,45% de pacientes con cáncer y un 39,55 de pacientes sin cáncer.

Positivos	Negativos	Falsos positivos	Falsos Negativos	Accuracy	Sensibilidad	Especificidad
88	26	44	19	0,644	0,822	0,371

Tabla 3.10: Resultados de test final con Counter-Propagation para Cáncer de mama

Se obtienen en la tabla 3.10 unos resultados para el dataset de test un poco peores que el Back-Propagation en cuanto a accuracy, pero 8 puntos mejores en sensibilidad, por lo que se decidirá usar



este algoritmo como mejor de los dos.

### 3.6.2 Melanoma

- **Back-Propagation**

Para el melanoma, los análisis de las variables del algoritmo back-propagation dejaron los siguientes resultados:

Neuronas Ocultas	ratio de aprendizaje	Momentum
95	0,005	0,3

Tabla 3.11: Valor de variables finales para Back-Propagation en cáncer de mama

Con estos valores de la tabla 3.11 se elaborará una red final, pero en este caso obteniendo los resultados con el dataset de testeo, en lugar de con el de validación. El dataset de testeo tiene un 54,18% de pacientes con cáncer y un 45,82% de pacientes sin cáncer.

Para este dataset y con los valores de variables anteriormente nombrados, los resultados fueron los de la tabla 3.12:

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibilidad	Especificidad
94	72	32	29	0,731	0,764	0,692

Tabla 3.12: Resultados de test final con Back-Propagation para melanoma

- **Counter-Propagation**

Las variables obtenidas en los análisis para el algoritmo Counter-Propagation con el melanoma fueron:

Neuronas del mapa	ratio de aprendizaje	Momentum
11	0,05	0,9

Tabla 3.13: Valor de variables finales para Counter-Propagation en Melanoma

Con estos valores de la tabla 3.13 se elaborará una red final, pero en este caso obteniendo los resultados con el dataset de testeo, en lugar de con el de validación. El dataset de testeo tiene un 54,18% de pacientes con cáncer y un 45,82% de pacientes sin cáncer.

Para este dataset y con los valores de variables anteriormente nombrados, los resultados fueron los siguientes:

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibilidad	Especificidad
46	84	40	30	0,650	0,605	0,677

Tabla 3.14: Resultados de test final con Counter-Propagation para melanoma

En este caso, los resultados son claramente inferiores a los obtenidos con el algoritmo Back-Propagation, por lo que con este tipo de cáncer se usará el primer algoritmo para las comparaciones con los resultados del HEAL.

### 3.6.3 Cáncer de Pulmón

- **Back-Propagation**

Para el cáncer de pulmón, los análisis de las variables del algoritmo Back-Propagation dejaron los siguientes resultados:

Neuronas Ocultas	ratio de aprendizaje	Momentum
75	0,005	0,7

Tabla 3.15: Valor de variables finales para Back-Propagation en cáncer de pulmón

Con estos valores se elaborará una red final, pero en este caso obteniendo los resultados con el dataset de testeo, en lugar de con el de validación. El dataset de testeo tiene un 43,99% de pacientes con cáncer y un 56,01% de pacientes sin cáncer.

Para este dataset y con los valores de variables anteriormente nombrados, los resultados fueron los de la tabla 3.15.

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibilidad	Especificidad
228	204	127	32	0,731	0,877	0,616

Tabla 3.16: Resultados de test final con Back-Propagation para cáncer de pulmón

- **Counter-propagation**

Neuronas del mapa	ratio de aprendizaje	Momentum
15	0,05	0,9

Tabla 3.17: Valor de variables para Counter-Propagation en cáncer de pulmón

Con estos valores de la tabla 3.17 se elaborará una red final, pero en este caso obteniendo los resultados con el dataset de testeo, en lugar de con el de validación. El dataset de testeo tiene un 43,99% de pacientes con cáncer y un 56,01% de pacientes sin cáncer.

Para este dataset y con los valores de variables anteriormente nombrados, los resultados fueron los de la tabla 3.18.

Positivos	Negativos	Falsos positivos	Falsos Negativos	Accurac y	Sensibilidad	Especificidad
99	254	80	66	0,707	0,600	0,760

Tabla 3.18: Resultados de test final con Counter-propagation para cáncer de pulmón

Los resultados para Counter-Propagation se acercan a lo aceptable, pero tienen una sensibilidad demasiado baja para considerarlos, por lo que se usará el algoritmo Back-Propagation.

### 3.6.4 Conclusiones del estudio

Entre las dos redes neuronales usadas para el estudio durante el proyecto, el Back-Propagation ha sido mejor para melanoma y cáncer de pulmón, mientras que el Counter-Propagation solo ha destacado en el cáncer de mama debido a su alta sensibilidad. Por el contrario, dicho algoritmo para los otros dos tipos de cáncer ha tenido resultados de sensibilidad bastante bajos, claramente provocados por una baja concentración de pacientes con cáncer en los datasets, si esto cambiara, posiblemente el Counter-Propagation podría llegar a superar los resultados del Back-Propagation para esos dos tipos de cáncer.

## 3.7 Comparación de resultados con el HEAL

En esta última sección del análisis se compararán los resultados

obtenidos con el algoritmo Counter-Propagation, con los obtenidos por el HEAL. Estos resultados pueden observarse en la figura 2.2.

Se compararán solo los resultados del subconjunto de test, sin tener en cuenta, los mejores resultados de la fase de entrenamiento. También se tendrán en cuenta las filas eliminadas por el preprocesado. A su vez, se comparará con los resultados del compañero Óscar Trujillo que obtuvo sus mejores resultados con un “decision jungle”.

### 3.7.1 Cáncer de mama

	Accuracy	Sensibilidad	Especificidad	Filas Eliminadas
HEAL	74,36%	<b>91,88%</b>	53,70%	0%
Counter-Propagation	64,4%	82,2%	37,1%	1,27%
Decision Jungle	<b>79,70%</b>	86,90%	<b>68,57%</b>	1,27%

Tabla 3.19: Comparación de resultados entre HEAL y proyecto para cáncer de mama

### 3.7.2 Melanoma

	Accuracy	Sensibilidad	Especificidad	Filas Eliminadas
HEAL	73,15%	66,67%	78,76%	0%
Back-Propagation	73,1%	<b>76,4%</b>	69,2%	11,71%
Decision Jungle	<b>74,90%</b>	69,90%	<b>80,76%</b>	11,71%

Tabla 3.20: Comparación de resultados entre HEAL y proyecto para Melanoma

### 3.7.3 Cáncer de Pulmón

	Accuracy	Sensibilidad	Especificidad	Filas Eliminadas
HEAL	87,3%	77,61%	94,37%	0%
Back-Propagation	73,1%	87,7%	61,6%	15,57%
Decision Jungle	<b>94,10%</b>	<b>91,20%</b>	<b>96,37</b>	15,57%

Tabla 3.21: Comparación de resultados entre HEAL y proyecto para cáncer de pulmón

### 3.7.4 Conclusiones

Como se puede observar, los resultados varían de un lugar a otro. En el caso del cáncer de mama, no se ha conseguido superar en ningún aspecto los resultados del HEAL, mientras que en melanoma y cáncer de pulmón se obtienen unas sensibilidades 10 puntos mejores a las obtenidas por el HEAL, a cambio de un accuracy menor. Pero como la sensibilidad es un factor bastante importante en este caso, se pueden considerar como resultados aceptables.

# **Capítulo 4**

## **Conclusiones y líneas futuras**

Al término del proyecto, se han conseguido todos los objetivos propuestos en un principio. La arquitectura híbrida Counter-Propagation podría haber tenido unos mejores resultados de contar con otras bases de datos con unos ratios de pacientes sanos y enfermos más nivelados. Mientras, las comparaciones con el HEAL han arrojado datos dispares, consiguiendo mejores datos de sensibilidad en melanoma y cáncer de pulmón, pero un peor resultado global en cáncer de mama.

Las posibles líneas futuras del proyecto podrían empezar por aumentar el rango de pruebas para las variables, es decir, en lugar de analizar los resultados para valores entre 5 y 100 de neuronas para la capa oculta de Back-propagation, usar valores entre 5 y 200 o hasta encontrar un punto en el que los resultados comiencen a decaer significativamente. También se pueden considerar pruebas sin sustituir los datos nulos o perdidos, para observar si modifican notablemente el resultado de las redes.

Otra opción de futuro puede ser seguir probando con otros tipos de redes neuronales o algoritmos, como Support Vector Machine o Deep Believe Network.

En cuanto a su aplicación práctica en el campo de la medicina se puede elaborar una interfaz gráfica para la introducción de datos de manera más accesible, permitiendo a doctores realizar un primer diagnóstico rápidamente.

# **Capítulo 5**

## **Summary and Conclusions**

At the end of the project, all objectives proposed have been achieved. Counter-Propagation hybrid architecture could have achieved better results if other databases with a better sick/healthy ratio. HEAL results comparison throw some up and downs, getting better sensitivity results in melanoma and respiratory system cancer, but a worse global results in breast cancer.

A possible future for the project could be increasing the range of the tested variables, instead of testing hidden neurons of Back-propagation between 5 and 100, they could be tested between 5 and 200 or until finding a point where the results starts decreasing significantly. Also, could be made tests not replacing missing data, observing if it affects the result in a big way.

Another option for the future of the project could be continuing with testing other types of neural networks and algorithms like Support Vector Machine or Deep Belief Network.

Talking about a possible real life application in medicine, a GUI could be developed, allowing an easy input of data by a doctor, obtaining a first diagnosis as soon as possible.

# Capítulo 6

## Presupuesto

Para calcular el presupuesto se ha tenido en cuenta el número de horas dedicadas a cada una de las fases del proyecto.

Fases	Horas	Euros/hora	Subtotal
Estudio Previo	40	20	800 €
Desarrollo	80	30	2400 €
Análisis	200	30	6000 €
Comparativa	20	20	400 €
Total:			9600 €



# Bibliografía y referencias

## Bibliografía

Redes Neuronales y Sistemas Borrosos – Bonifacio Martín del Brío y Alfredo Sanz Molina – Ed. Ra-Ma, 2ª edición, 2001

Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones – Celestino Ordóñez Galán – Ed. Ra-Ma, 1ª edición, 1995

## Referencias

- [1] Redes Neuronales Artificiales:  
<http://www.uta.cl/charlas/volumen16/Indice/Ch-csaavedra.pdf>
- [2] Back-Propagation: <https://en.wikipedia.org/wiki/Backpropagation>
- [3] Counter-Propagation network:  
[http://www.inf.ufrgs.br/~engel/data/media/file/cmp121/CPN\\_Freeman%20.pdf](http://www.inf.ufrgs.br/~engel/data/media/file/cmp121/CPN_Freeman%20.pdf)
- [4] PyBrain: <http://pybrain.org/>
- [5] Historia de la Inteligencia Artificial:  
[http://biblioteca.itam.mx/estudios/estudio/estudio10/sec\\_16.html](http://biblioteca.itam.mx/estudios/estudio/estudio10/sec_16.html)
- [6] Heuristic and Evolutionary Algorithm Laboratory (HEAL):  
<http://heal.heuristiclab.com/>
- [7] Global Neural Network Cloud Service for Breast Cancer:  
<https://cloud4cancer.appspot.com/>
- [8] Redes neuronales para detectar cáncer en mamografías:  
<http://www.eluniversal.com.mx/articulo/ciencia-y-salud/salud/2015/06/9/detectan-cancer-de-mama-con-redes-neuronales>
- [9] Repositorio del TFG en Github:  
<https://github.com/etsiitull/medicalpybrain>

[10] Neurona de McCulloch-Pitts:

[https://es.wikipedia.org/wiki/Neurona\\_de\\_McCulloch-Pitts](https://es.wikipedia.org/wiki/Neurona_de_McCulloch-Pitts)

[11] Mapas auto-organizados o Mapas de Kohonen:

<http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema5dm.pdf>

[12] Perceptrón simple:

<http://www.lcc.uma.es/~jmortiz/archivos/Tema4.pdf>

[13] Weka: <http://www.cs.waikato.ac.nz/ml/weka/>