Universidad de La Laguna

Escuela Superior de Ingeniería y Tecnología
Sección de Ingeniería Informática

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, 27 de junio de 2016

D. **Patricio García Báez**, con N.I.F. 43.356.987-D profesor Titular de Universidad 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 profesor Titular de Universidad adscrito al Departamento de Informática y Sistemas de la Universidad de Las Palmas de Gran Canaria, como co-tutor

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*"

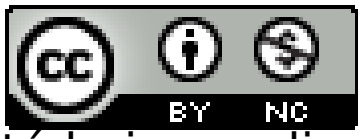
ha sido realizada bajo su dirección por D. **Alberto Fariña Barrera**, con N.I.F. 43.836.555-G.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 27 de junio 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



Resumen

El objetivo de este trabajo ha sido el desarrollo de tres Redes Neuronales Artificiales (RNA), una para cada tipo de cáncer tratado, que son: 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 llevarán a cabo estudios con dos redes o algoritmos distintos: Back-propagation, un entrenamiento supervisado y Counter-propagation, un algoritmo que mezcla el entrenamiento no supervisado con 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.*

Abstract

The goal of this project has been the development of three Artificial Neural Networks, or ANN as its acronym, that identifies the most accurate way possible the presence of different types of cancer in a person, such as breast cancer, melanoma or 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.

There will be studies for two different types of network or algorithms: Back-propagation, a supervised training, and Counter-propagation, an algorithm that combines supervised and unsupervised training.

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.*

Índice general

Capítulo 1 Introducción.....1
1.1 Antecedentes.....1
1.2 Objetivos.....2
1.3 Alcance.....3
1.4 Programa de Apoyo a Trabajos Finales .....Libres
.....4
Capítulo 2 Método, herramientas y datos.....5
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
2.2 Herramientas.....6
2.2.1 Algoritmos.....7
2.2.2 Funcionamiento de la herramienta.....10
2.2.3 ¿Cómo se llegó a elegir PyBrain?.....11
2.2.4 Conclusiones.....12
2.3 Datos.....12
2.3.1 Obtención.....12
2.3.2 Contenido de la Base de datos.....13
2.3.3 Resultados de la investigación del HEAL.....13
Capítulo 3 Desarrollo y análisis.....15
3.1 Desarrollo.....15
3.1.1 Preprocesado.....15
3.1.2 Desarrollo de Scripts.....16
3.2 Explicación de los análisis.....17

3.3 Cáncer de mama.....19
3.3.1 Back-Propagation.....19
3.3.2 Counter-Propagation.....23
3.4 Melanoma.....26
3.4.1 Back-Propagation.....26
3.4.2 Counter-Propagation.....29
3.5 Cáncer de pulmón.....33
3.5.1 Back-Propagation.....33
3.5.2 Counter-propagation.....36
3.6 Resultados finales usando el subconjunto de test.....40
3.6.1 Cáncer de mama.....40
3.6.2 Melanoma.....41
3.6.3 Cáncer de Pulmón.....42
3.7 Comparación de resultados con el HEAL.....44
Capítulo 4 Conclusiones y líneas futuras.....45
Capítulo 5 Summary and Conclusions.....46
Capítulo 6 Presupuesto.....47
Capítulo 7 Apéndice: Scripts.....48
7.1 Script para Cáncer de mama.....48
7.2 Script para Melanoma.....48
7.3 Script para Cáncer de pulmón.....48
Bibliografía y referencias.....49

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.....14
Figura 3.1: Gráfica de resultados para análisis de neuronas ocultas en cáncer de mama.....20
Figura 3.2: Gráfica de resultados de análisis de Learning Rate para cáncer de mama.....21
Figura 3.3: Gráfica de resultados de análisis de Momentum para cáncer de mama.....22
Figura 3.4: Gráfica de resultados de análisis de neuronas del mapa auto-organizado para cáncer de mama.....23
Figura 3.5: Gráfica de resultados de análisis de Learning Rate para Cáncer de Mama con Counterprop.....24
Figura 3.6: Gráfica de resultados de análisis de momentum en cáncer de mama para Counterprop.....25
Figura 3.7: Gráfica de resultados de análisis de neuronas ocultas para melanoma.....27
Figura 3.8: Gráfica de resultados de análisis de Learning Rate para melanoma con backprop.....28
Figura 3.9: Gráfica de resultados de análisis de Momentum para melanoma con backprop.....29
Figura 3.10: Gráfica de resultados de análisis de neuronas para mapa auto-organizado para melanoma.....30
Figura 3.11: Gráfica de resultados de análisis de learning rate para melanoma con Counter-prop.....31
Figura 3.12: Gráfica de resultados de análisis de momentum para melanoma con Counter-prop.....32
Figura 3.13: Gráfica de resultados de análisis de neuronas ocultas para cáncer de pulmón.....34

Figura 3.14: Gráfica de resultados de análisis de Learning Rate para cáncer de pulmón.....35
Figura 3.15: Gráfica de resultados de análisis de momentum para cáncer de pulmón.....36
Figura 3.16: Gráfica de resultados de análisis de neuronas del mapa auto-organizado para cáncer de pulmón.....37
Figura 3.17: Gráfica de análisis de learning rate para cáncer de pulmón con counter-prop.....38
Figura 3.18: Gráfica de resultados de análisis de momentum para cáncer de pulmón con counter-prop.....39

Índice de tablas

Tabla 3.1: Resultados de la primera prueba de Back-Propagation para Cáncer de mama.....19
Tabla 3.2: Resultados de análisis de neuronas ocultas para Cáncer de mama.....20
Tabla 3.3: Resultados de análisis de learning rate para cáncer de mama (Back-prop).....21
Tabla 3.4: Resultados de análisis de momentum para cáncer de mama (Back-prop).....22
Tabla 3.5: Resultados de primera prueba de cáncer de mama para counter-propagation.....23
Tabla 3.6: Resultados de análisis de neuronas del mapa auto-organizado para cáncer de mama.....23
Tabla 3.7: Resultados de análisis de Learning Rate para Cáncer de mama con Counter-prop.....24
Tabla 3.8: Resultados de análisis de momentum para cáncer de mama con Counterprop.....25
Tabla 3.9: Resultados de primera prueba de backpropagation para melanoma.....26
Tabla 3.10: Resultados de análisis de neuronas ocultas para melanoma.....26
Tabla 3.11: Resultados de análisis de Learning Rate para melanoma con Backprop.....27
Tabla 3.12: Resultados de análisis de momentum para Melanoma con backprop.....28
Tabla 3.13: Resultados de primera prueba de backpropagation para melanoma.....29
Tabla 3.14: Resultados de análisis de neuronas para mapa auto-organizado para melanoma.....30
Tabla 3.15: Resultados de análisis de Learning Rate para

melanoma con Counter-prop.....31
Tabla 3.16: Resultados de análisis de momentum para melanoma con Counter-prop.....32
Tabla 3.17: Resultados de análisis de neuronas ocultas para cáncer de pulmón.....33
Tabla 3.18: Resultados de análisis de learning rate para cáncer de pulmón con Back-prop.....34
Tabla 3.19: Resultados de análisis de momentum para cáncer de pulmón con Back-prop.....35
Tabla 3.20: Resultados de análisis para neuronas del mapa auto-organizado para cáncer de pulmón.....37
Tabla 3.21: Resultados de análisis de learning rate para cáncer de pulmón con counter-prop.....38
Tabla 3.22: Resultados de análisis de momentum para cáncer de pulmón con counter-prop.....39
Tabla 3.23: Valor de variables finales para back-propagation en cáncer de mama.....40
Tabla 3.24: Resultados de test final con Back-Propagation para cáncer de mama.....40
Tabla 3.25: Valor de variables finales para Counter-propagation en cáncer de mama.....41
Tabla 3.26: Valor de variables finales para back-propagation en cáncer de mama.....41
Tabla 3.27: Resultados de test final con Back-Propagation para melanoma.....42
Tabla 3.28: Resultados de test final con Back-Propagation para cáncer de pulmón.....43

Capítulo 1
Introducción

En este trabajo se propone la elaboración de tres Redes Neuronales Artificiales, 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 como algoritmo de entrenamiento y Counter-Propagation, una red que combina un mapa auto-organizado con una red con aprendizaje Back-propagation.

Se utiliza como herramienta PyBrain, 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 las variables características de cada uno de los algoritmos, para finalmente realizar una comparativa entre los tres algoritmos, 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 las redes neuronales 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, 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), 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, pero elaborado con árboles de decisión genéticos, cuyos resultados se compararán con los resultados de este trabajo en uno de los capítulos.

Otro de los casos más conocidos quizas sea el “Global Neural Network Cloud Service for Breast Cancer” de Brittany Wenger, 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, 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:

- Desarrollar un script de Python que sea capaz 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 las variables de la red neuronal para intentar optimizar los resultados obtenidos.
- Usar la misma base de datos para realizar pruebas con otros tipos de redes neuronales y algoritmos de clasificación para comprobar los resultados entre ellos.

1.3 Alcance

a) **1. Preparación**
   I. Búsqueda de información acerca de RNAs
   II. Búsqueda y elección de herramientas para la realización de las RNA.
   III. Estudio del trabajo realizado por el laboratorio austriaco (HEAL)

b) **2. Preprocesado**
   I. Análisis de las bases de datos que se utilizarán en el proyecto para detectar posibles dudas y resolverlas.
   II. Elección de herramienta de preprocesado para las bases de datos
   III. Preprocesamiento de la base de datos para eliminar datos nulos, campos innecesarios y ajustar valores
   IV. Selección de las particiones de la base de datos que se usarán para entrenamiento, validación y testeo

c) **3. Diseño y Desarrollo**
   I. Elección del modelo de RNA que se utilizará en el proyecto. (Redes de base radial, Perceptrón Multicapa,…)
   II. Diseño de la red
   III. Entrenamiento y validación de la RNA.
   IV. Optimización de los parámetros de la RNA

d) **4. Evaluación de la red neuronal**
   I. Realización de pruebas con los datos de la partición de testeo
   II. Creación de gráficas de errores y precisión para la presentación de la RNA
   III. Comparación de resultados con el laboratorio HEAL

e) **5. Finalización**
   I. Preparación de la memoria del TFG
   II. 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 del Trabajo de Fin de Grado [1]:

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 serían las redes neuronales como las herramientas usadas para su tratamiento y desarrollo, así como las bases de datos que fueron proporcionadas por .

2.1 Redes Neuronales Artificiales

2.1.1 Historia

Los dos pioneros en al 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' 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 capas, que normalmente suelen ser tres:

- La capa de entrada: esta suele contener una neurona por cada variable que se le pasará a la red.
- La capa oculta: está compuesta 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: esta contiene una neurona por cada salida deseada de la red.

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 algoritmos, pero como ejemplo, se explicará el funcionamiento del algoritmo Back-Propagation que será usado en el proyecto.

Este algoritmo aplica un patrón a la entrada de la red como estímulo, este se propaga desde la primera capa a través de las capas superiores hasta que genera una salida. Dicha salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas. Esta señal se propaga hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Las neuronas de la capa oculta solo reciben una fracción de la señal total, basándose aproximadamente en la contribución que haya hecho cada neurona a la salida original.

Después del entrenamiento, cuando se presente un patrón arbitrario de entrada, que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

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 hasta meteorología para predecir el tiempo o en economía para predecir las fluctuaciones en el mercado financiero.

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, a pesar de que Pybrain contiene una gran variedad de algoritmos para 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

La propagación hacia atrás de errores o retro propagación (del inglés backpropagation) 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, que esté incompleto, las neuronas de la capa oculta de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante

su entrenamiento.

Este será uno de los algoritmos estudiados durante el proyecto.

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. 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 $\eta^+ > 1$. η^+ suele ser considerado empíricamente como 1,2 y η^- como 0,5.

Support Vector Machines

Una 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, siento el numero de casillas mucho menor al universo de posibles entradas.

Deep Belief Network

Una 'Deep Belief Network' (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.

- **Otros algoritmos**

Counter-Propagation

Se ha decidido explicar este algoritmo a parte de los anteriormente nombrados, ya que no se podría clasificar como supervisado o no supervisado y es uno de los que se usarán durante el proyecto.

El algoritmo Counter-propagation combina lo mejor de dos mundos, usando tanto métodos de aprendizaje tanto supervisado como no supervisado.

La primera parte del algoritmo es un mapa auto-organizado o mapa de Kohonen, 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 MxM, 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.

La salida del mapa en forma de array se pasa a la segunda parte del algoritmo, formada por una red neuronal común con back-propagation pero sin neuronas ocultas, cuyo número de entradas será el número de neuronas del mapa.

2.2.2 Funcionamiento de la herramienta

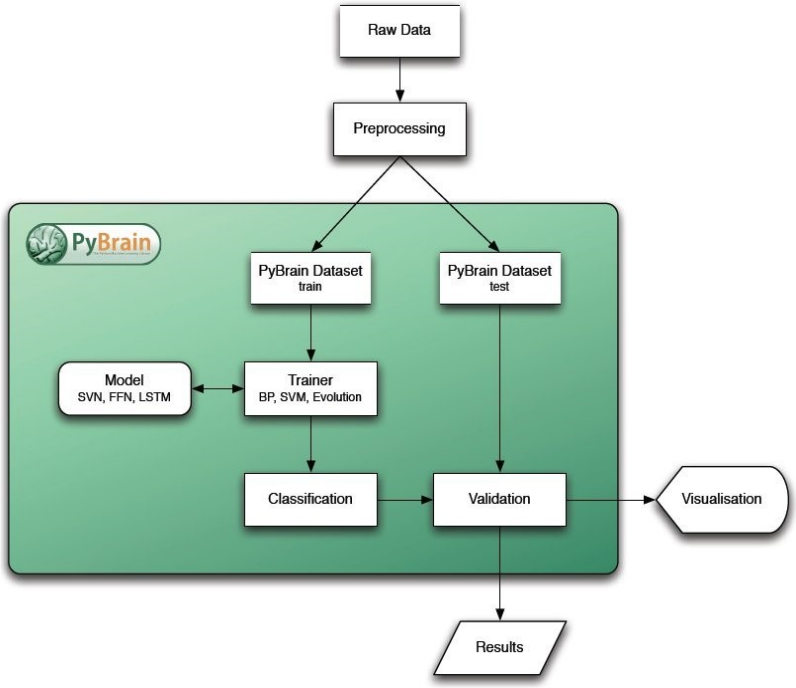


Figura 2.1: Diagrama de flujo de uso de la librería PyBrain

Como se puede ver en este diagrama de flujo 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(numcols-1, 10)

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(numcols-1, numhidden, 10, 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 Learning Rate 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 elegido 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 compresión 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, uno para cada tipo de cáncer, conteniendo cada uno de ellos distinto número de entradas, véase 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.*

2.3.3 Resultados de la investigación del HEAL.

Al contrario que este proyecto, 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 este apartado se indicarán 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

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.

- **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 '0' para que el filtro de Weka los reconociera. Esto hubiera provocado que las columnas como 'sexo' o 'cáncer' perdieran datos, ya que el valor '0' representa que un paciente es varón y que no tiene cáncer respectivamente en estas columnas, pero esto se solucionó de manera sencilla, ya que Weka proporciona la utilidad de seleccionar a qué columnas desea aplicarse el filtro seleccionado.

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).Sustitución de datos nulos

- **Sustitución de datos nulos**

Para las columnas restantes, aunque tuvieran menos del 50% de datos nulos, 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 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.

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 Explicación de los análisis

- **Back-Propagation**

Para el algoritmo de Back-Propagation se analizarán tres

variables: neuronas ocultas, learning rate y momentum, a parte de una prueba inicial para comparar con los resultados finales.

Primera Prueba

Las pruebas iniciales se usarán 25 neuronas ocultas, 0,0001 de Learning Rate y 0,1 de Momentum, llegando a un máximo de 100 iteraciones de entrenamiento. Estas pruebas servirán como trampolín a análisis posteriores, teniendo una referencia de partida que intentar mejorar con dichos análisis. 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 Learning Rate 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.

Learning Rate

En siguiente lugar, se realizará el análisis de la variable Learning Rate. 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 Learning Rate 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 Learning Rate 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 Learning Rate: $0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05$ (Se pretendía realizar pruebas con valores mayores, pero la red neuronal provocaba un overflow).

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 Learning Rate, 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.

- **Counter-Propagation**

Para el algoritmo de Counter-Propagation se analizarán tres variables: neuronas del mapa y el learning rate y momentum de la red neurona con back-propagation. 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 learning rate 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 comenzado 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.

Learning rate

El estudio para la variable Learning rate 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 0.07.

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 Learning rate.

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

3.3.1 Back-Propagation

- **Primera prueba**

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibili-dad	Especifici-dad
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

- **Análisis de neuronas ocultas**

Medias		
Neuronas Ocultas	Sensibilidad	Especificidad
5	0,95	0,14
10	0,68	0,54
15	0,87	0,43
20	0,65	0,56
25	0,74	0,51
30	0,87	0,37
35	0,87	0,43
40	0,81	0,52
45	0,84	0,51
50	0,81	0,61
55	0,89	0,36
60	0,80	0,59
65	0,82	0,53
70	0,76	0,61
75	0,77	0,62
80	0,81	0,56
85	0,83	0,55
90	0,81	0,58
95	0,83	0,57
100	0,84	0,51

Tabla 3.2: Resultados de análisis de neuronas ocultas para Cáncer de mama

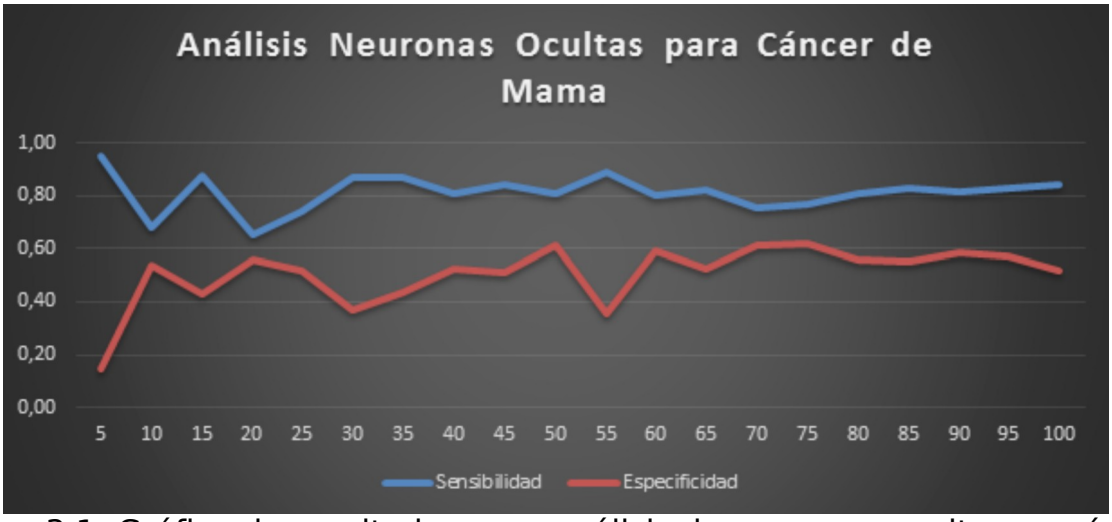


Figura 3.1: Gráfica de resultados para análisis de neuronas ocultas en cáncer de mama

Se ha decidido **usar 50 neuronas ocultas**. 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 Learning Rate**

Learning Rate	Iteraciones hasta límite (Train)	Iteraciones hasta límite (Valid)
0,0001	22	67
0,0005	4	14
0,001	3	8
0,005	4	5
0,01	11	8
0,05	200	200

Tabla 3.3: Resultados de análisis de learning rate para cáncer de mama con back-prop

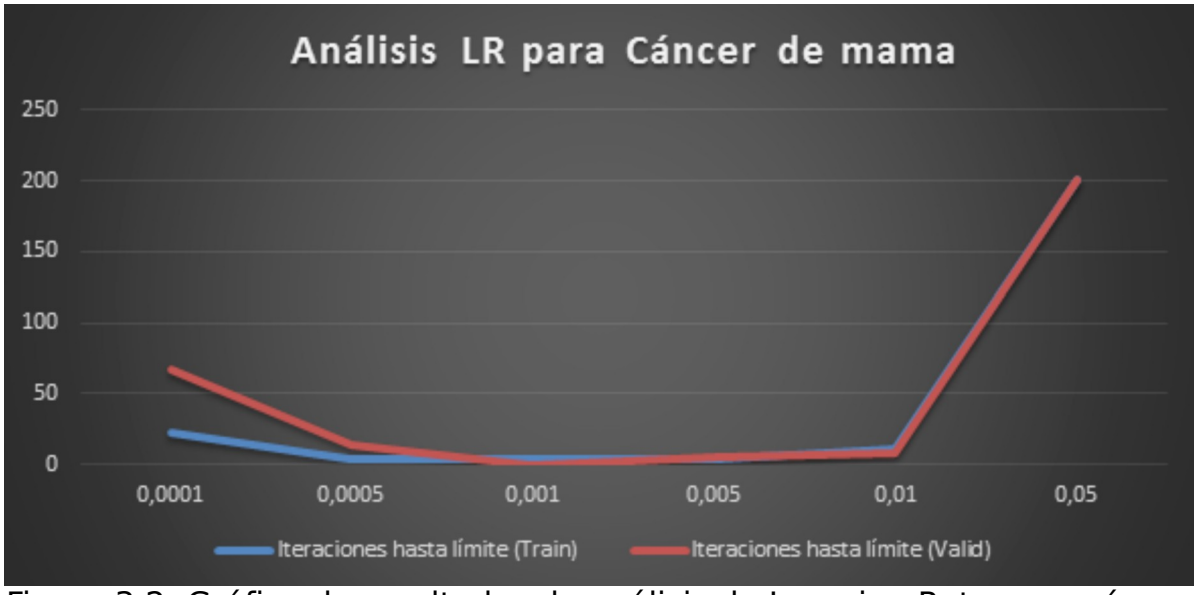


Figura 3.2: Gráfica de resultados de análisis de Learning Rate para cáncer de mama con back-prop

Se utilizará **0.005 de Learning Rate**. 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**

Mejores Resultados		
Momentum	Iteraciones hasta límite(Train)	Iteraciones hasta límite (Valid)
0,1	2	3
0,2	3	2
0,3	2	3
0,4	2	3
0,5	3	2
0,6	13	5
0,7	31	7
0,8	200	5
0,9	200	200

Tabla 3.4: Resultados de análisis de momentum para cáncer de mama con back-prop

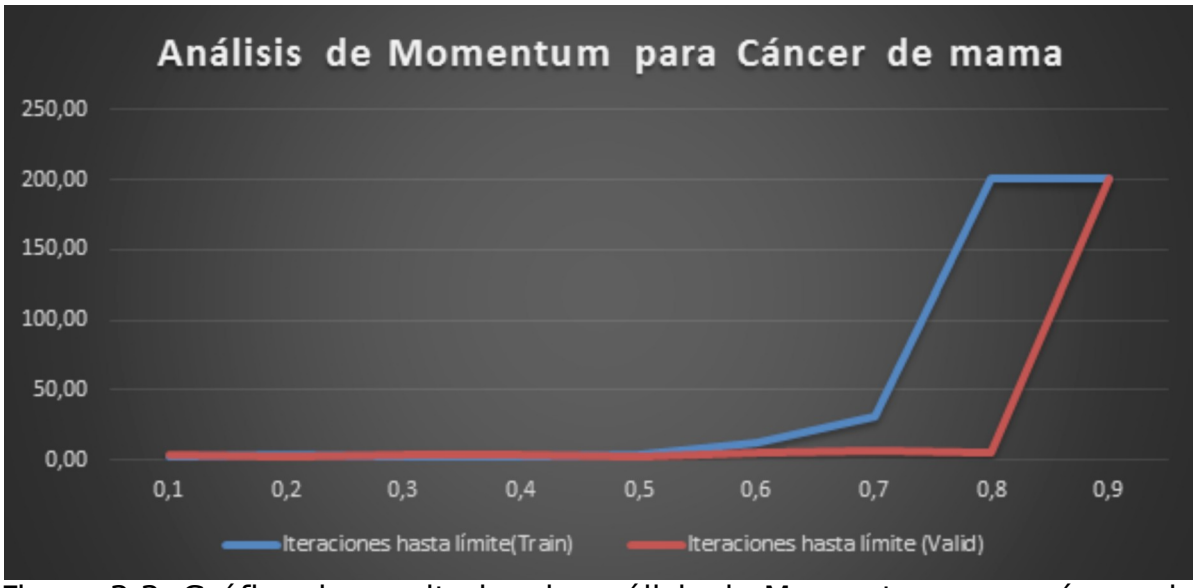


Figura 3.3: Gráfica de resultados de análisis de Momentum para cáncer de mama con back-prop

Durante los cinco primeros casos de estudio, 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	Sensibili-dad	Especifici-dad
100	62	10	3	0,920	0,971	0,861

Tabla 3.5: Resultados de primera prueba de cáncer de mama para counter-propagation

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

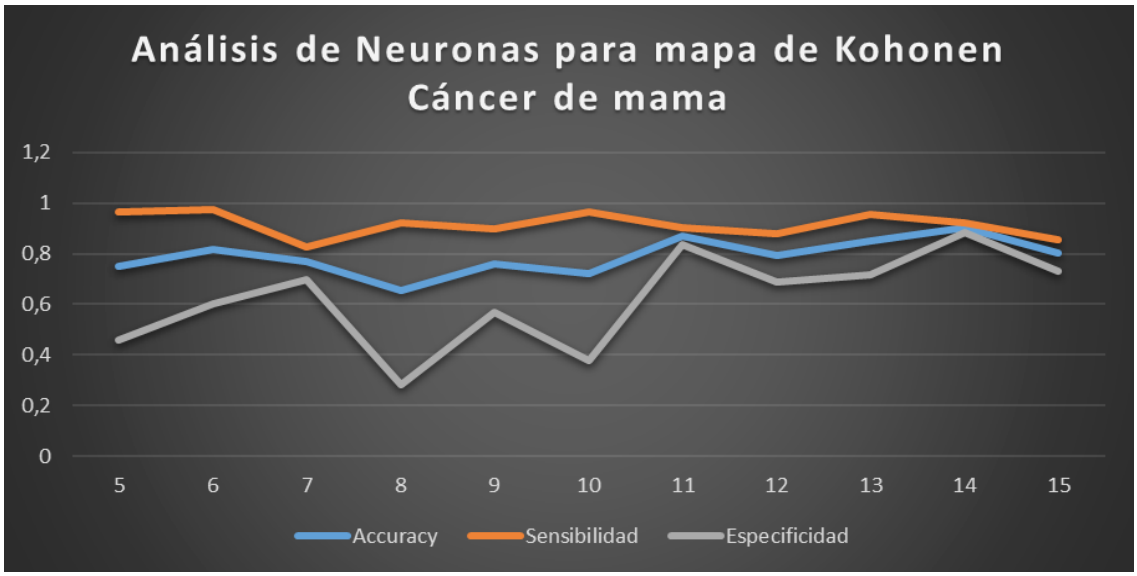


Figura 3.4: Gráfica de resultados de análisis de neuronas del mapa auto-organizado para cáncer de mama

Tal y como se esperaba los resultados con este segundo algoritmo son gratamente superiores a los obtenidos con el back-propagation por sí solo.

Se ha decidido **usar 14 neuronas** para el mapa auto-organizado, por lo que resultará un mapa de 14x14 (196) neuronas. El mejor resultado de esta instancia fue de 93,18% de accuracy, 97,08% de sensibilidad y 88,88% de especificidad.

- **Análisis de Learning Rate**

Medias		
Learning Rate	Iteraciones(Train)	Iteraciones(Valid)
0,0001	200	200
0,0005	139,4	139,4
0.001	132.2	132

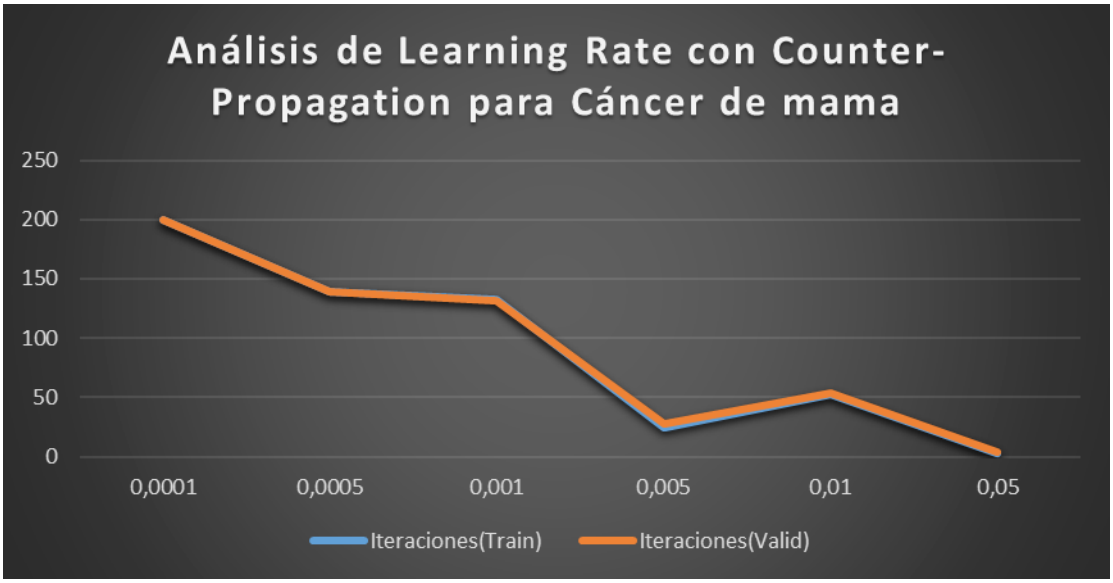


Figura 3.5: Gráfica de resultados de análisis de Learning Rate para Cáncer de Mama con Counterprop

Se puede ver como la iteración en la que baja del 0.07 de error va disminuyendo a medida que el learning rate se acerca a 1. por ello se elegirá **usar 0.05 de Learning Rate** ya que es el menor de todos.

- **Análisis de Momentum**

Medias		
Momentum	Iteraciones(Train)	Iteraciones(Valid
0,1	7,8	12
0,2	49,6	55,2
0,3	5,8	5,8
0,4	4,8	5,6
0,5	3	4
0,6	1	1,4
0,7	40,6	41,4
0,8	1	1,2
0,9	41	1,8

Tabla 3.8: Resultados de análisis de momentum para cáncer de mama con Counterprop

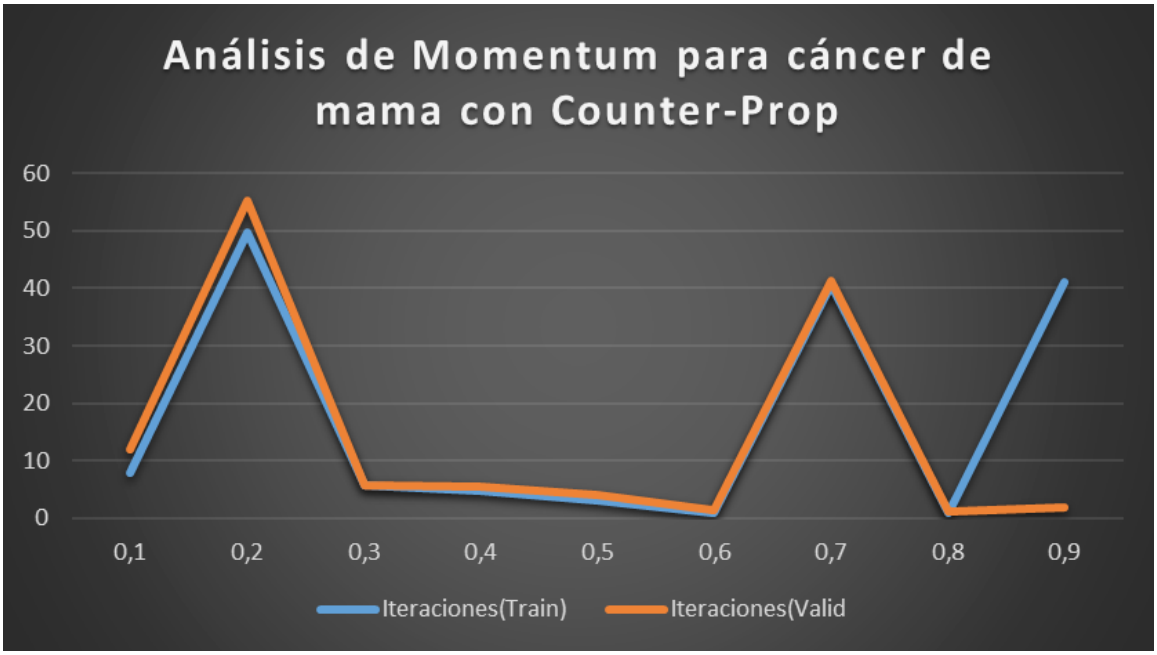


Figura 3.6: Gráfica de resultados de análisis de momentum en cáncer de mama para Counterprop

En este caso las instancias de 0.6 y 0.8 son muy parecidas, por lo que se decidió usar como clasificador el accuracy medio de ambas, dando como ganador a la instancia de 0.8. Una de las cinco pruebas

para ese valor devolvió un accuracy de 0.96, lo que es el mejor resultado de cualquier prueba durante el análisis para este cáncer de mama.

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.

Se han obtenido los siguientes resultados

- **Primera prueba**

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibili-dad	Especifici-dad
78	66	63	19	0,634	0,804	0,512

Tabla 3.9: Resultados de primera prueba de backpropagation para melanoma

- **Análisis de Neuronas Ocultas**

Medias		
Neuronas Ocultas	Sensibilidad	Especificidad
5	0,17	0,82
10	0,47	0,64
15	0,67	0,63
20	0,19	0,82
25	0,52	0,70
30	0,54	0,68
35	0,61	0,64
40	0,77	0,57
45	0,78	0,55
50	0,78	0,57
55	0,75	0,59
60	0,74	0,61
65	0,78	0,56
70	0,79	0,59
75	0,78	0,59
80	0,76	0,60
85	0,76	0,57
90	0,73	0,62
95	0,79	0,60
100	0,68	0,62

Tabla 3.10: Resultados de análisis de neuronas ocultas para melanoma

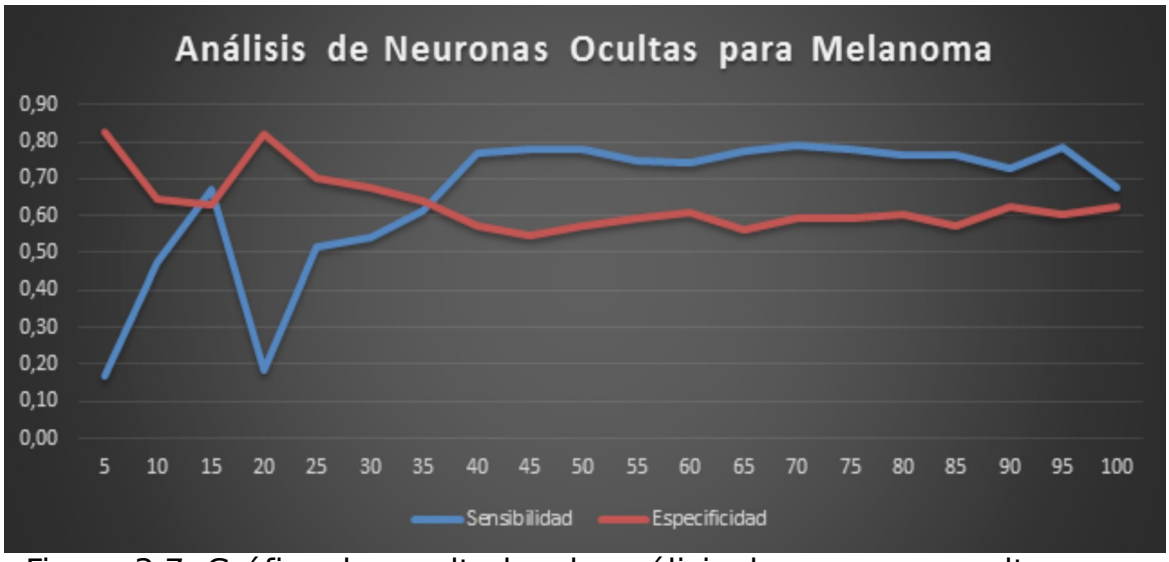


Figura 3.7: Gráfica de resultados de análisis de neuronas ocultas para melanoma

Los resultados en el caso del 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 y una muy buena especificidad en comparación.

- **Análisis de Learning Rate**

Mejores resultados		
Learning rate	Iteraciones hasta límite(Train)	Iteraciones hasta límite (Valid)
0,0001	200	200
0,0005	24	200
0,001	14	200
0,005	5	82
0,01	23	70
0,05	200	200

Tabla 3.11: Resultados de análisis de Learning Rate para melanoma con Backprop

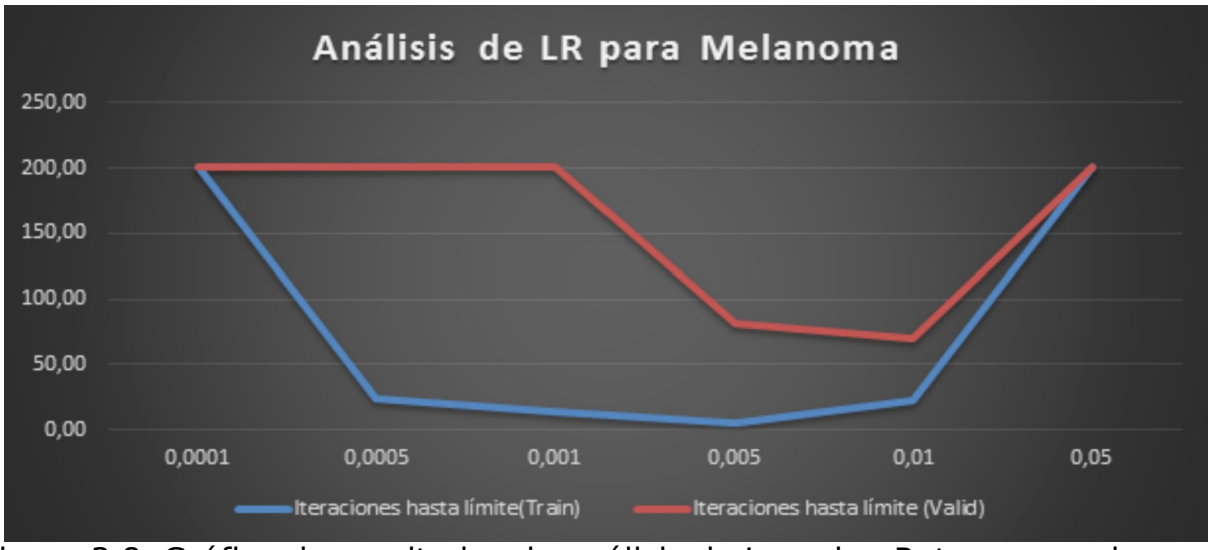


Figura 3.8: Gráfica de resultados de análisis de Learning Rate para melanoma con backprop

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

- **Análisis de Momentum**

Mejores resultados		
Momentum	Iteraciones hasta límite(Train)	Iteraciones hasta límite (Valid)
0,1	9	16
0,2	7	11
0,3	8	9
0,4	7	11
0,5	10	28
0,6	9	19
0,7	3	4
0,8	68	200
0,9	200	200

Tabla 3.12: Resultados de análisis de momentum para Melanoma con backprop

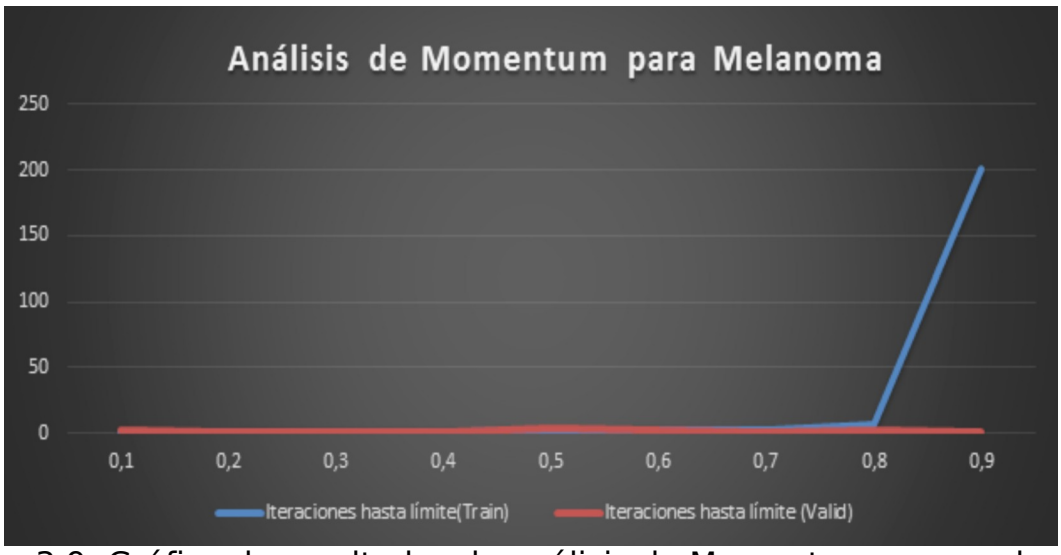


Figura 3.9: Gráfica de resultados de análisis de Momentum para melanoma con backprop

Como en el caso anterior las primeras pruebas 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.

3.4.2 Counter-Propagation

- **Primera prueba**

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibili-dad	Especifici-dad
84	112	17	13	0,863	0,866	0,868

Tabla 3.13: Resultados de primera prueba de backpropagation para melanoma

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

Medias			
Neuronas	Accuracy	Sensibilidad	Especificidad
5	0,80	0,80	0,81
6	0,85	0,89	0,82
7	0,83	0,87	0,81
8	0,83	0,84	0,84
9	0,79	0,74	0,83
10	0,83	0,78	0,88
11	0,82	0,78	0,86
12	0,79	0,81	0,79
13	0,78	0,76	0,80
14	0,77	0,78	0,76
15	0,78	0,78	0,79

Tabla 3.14: Resultados de análisis de neuronas para mapa auto-organizado para melanoma

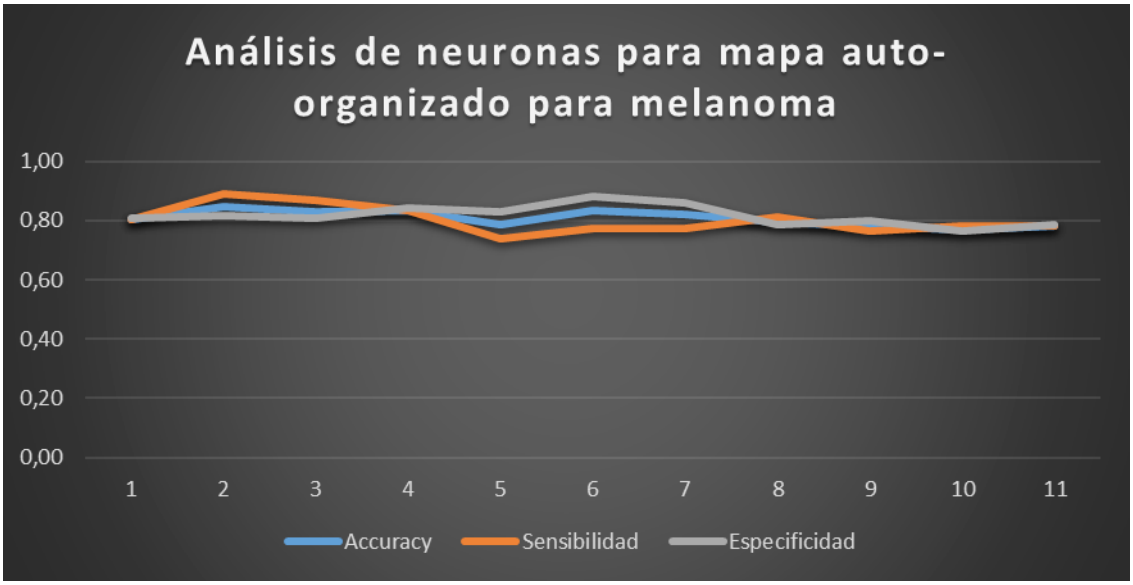


Figura 3.10: Gráfica de resultados de análisis de neuronas para mapa auto-organizado para melanoma

En este caso parece ser que los mejores resultados surgen de un número de neuronas más bajas. Se escogerá usar 6 neuronas ocultas, ya que tuvo el mayor accuracy y la mayor sensibilidad. El mejor resultado dentro de esta instancia fue de un 88% de accuracy, 94% de sensibilidad y 85% de especificidad

- **Análisis de Learning Rate**

Medias		
Learning Rate	Iteraciones(train)	Iteraciones(valid)
0,0001	200	200
0,0005	171,2	176,6
0,001	200	200
0,005	64,6	72,2
0,01	29,6	64,4
0,05	3,2	8

Tabla 3.15: Resultados de análisis de Learning Rate para melanoma con Counter-prop

Al igual que para el cáncer de mama con Counter-propagation se

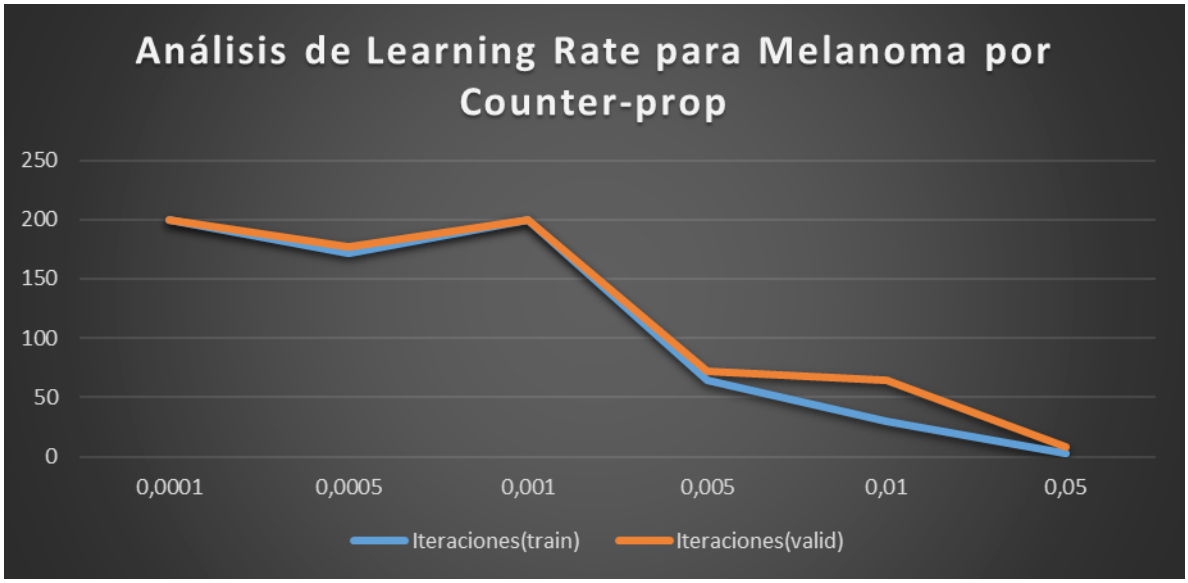


Figura 3.11: Gráfica de resultados de análisis de learning rate para melanoma con Counter-prop

ve claramente como a medida que el Learning rate 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 Learning Rate**, cuyo mejor resultado fue de 92% de accuracy, 91% de sensibilidad y 93% de especificidad.

- **Análisis de momentum**

Medias		
Momentum	Iteraciones(train)	Iteraciones(valid)
0,1	6	9,2
0,2	9,4	15,8
0,3	4,6	6,8
0,4	4	9,6
0,5	5,6	15,4
0,6	3,8	7,2
0,7	51,2	133,2
0,8	41,2	41,6
0,9	0,8	1,8

Tabla 3.16: Resultados de análisis de momentum para melanoma con Counter-prop

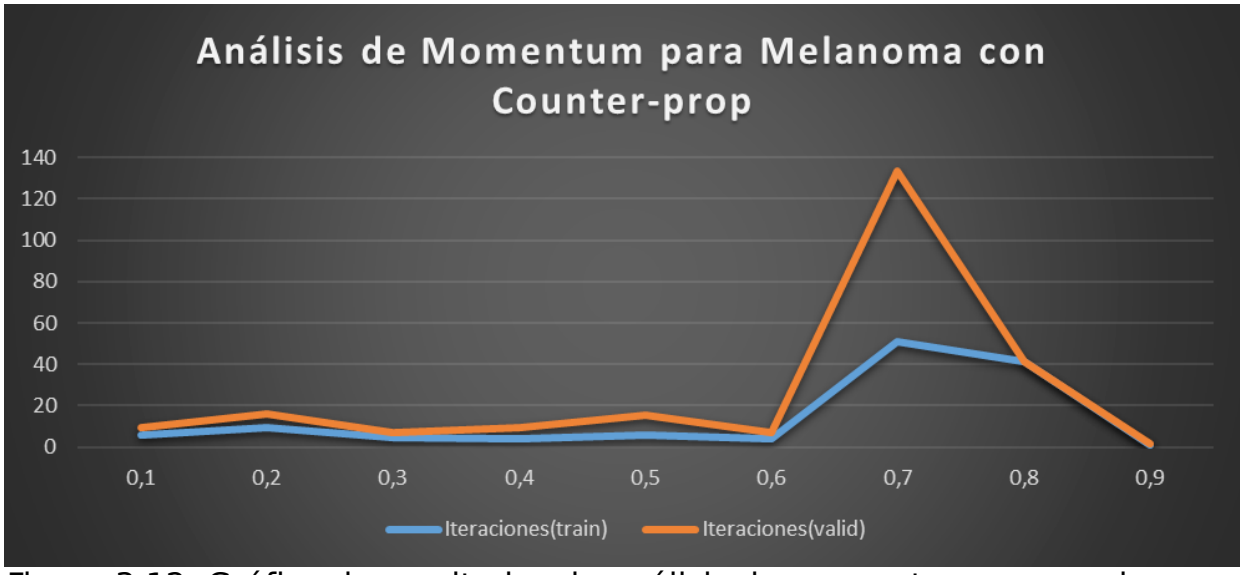


Figura 3.12: Gráfica de resultados de análisis de momentum para melanoma con Counter-prop

Los resultados no parecen variar mucho con el paso del tiempo, excepto para las instancias de 0.7 y 0.8, pero le mejor resultado sin duda es para **0.9 de momentum** cuyo mejor resultado individual fue de 92% de accuracy, 96% de sensibilidad y 90% de especificidad.

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	Sensibili-dad	Especifici-dad
166	294	56	75	0,778	0,689	0,846

- **Análisis de Neuronas Ocultas**

Medias		
Neuronas Ocultas	Sensibilidad	Especificidad
5	0,18	0,90
10	0,49	0,79
15	0,36	0,82
20	0,63	0,74
25	0,83	0,62
30	0,76	0,77
35	0,66	0,71
40	0,74	0,78
45	0,75	0,75
50	0,76	0,75
55	0,78	0,73
60	0,73	0,80
65	0,74	0,77
70	0,78	0,71
75	0,80	0,73
80	0,80	0,69
85	0,77	0,78
90	0,75	0,81
95	0,71	0,81
100	0,74	0,83

Tabla 3.17: Resultados de análisis de neuronas ocultas para cáncer de pulmón

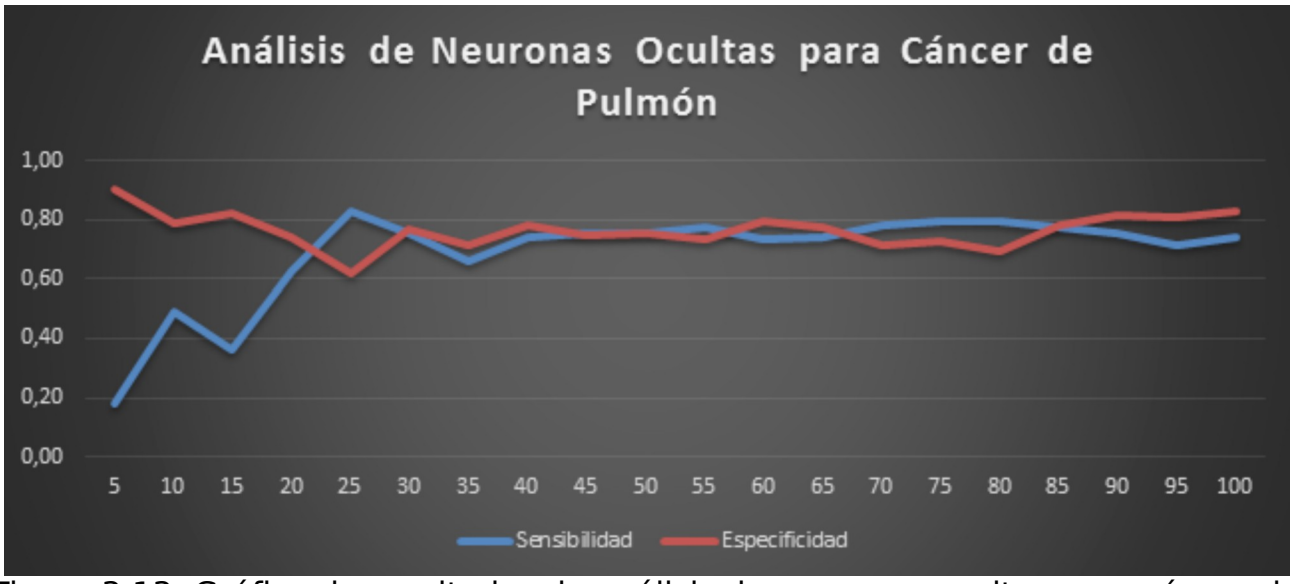


Figura 3.13: Gráfica de resultados de análisis de neuronas ocultas para cáncer de pulmón

Obviamente por tener más casos con los que entrenar la red neuronal, 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**¸ 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 Learning Rate**

Mejores resultados		
Learning Rate	Iteraciones hasta límite (Train)	Iteraciones hasta límite (Valid)
0,0001	22	18
0,0005	3	12
0,001	2	5
0,005	2	2
0,01	8	14
0,05	200	200

Tabla 3.18: Resultados de análisis de learning rate para cáncer de pulmón con Back-prop

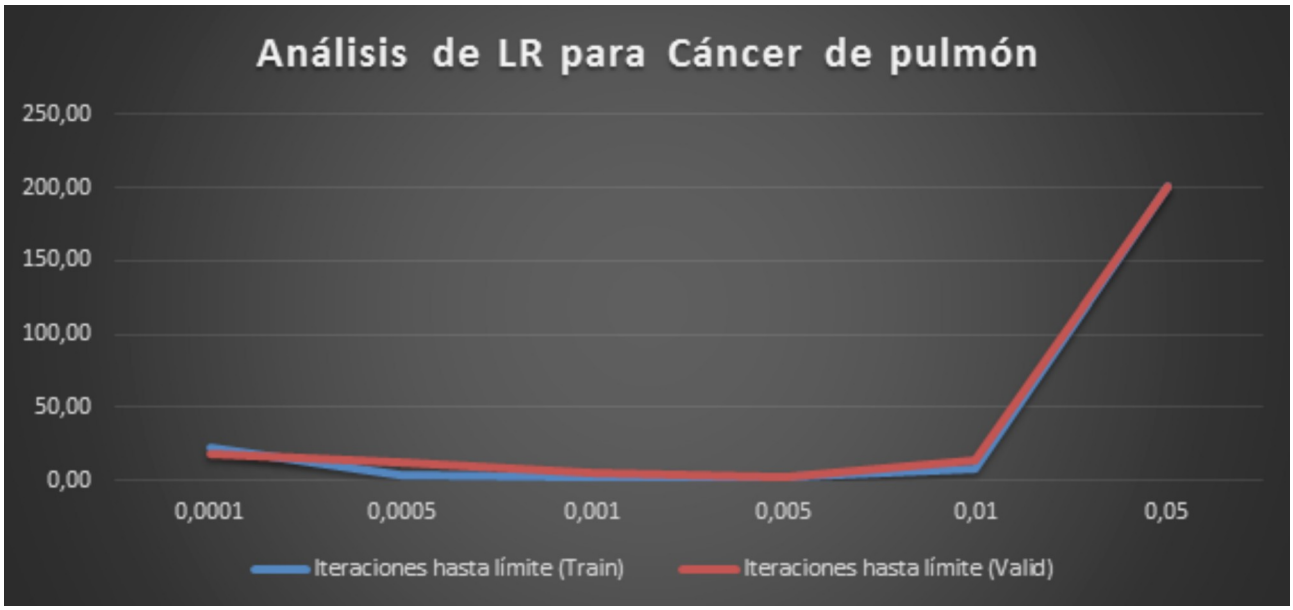


Figura 3.14: Gráfica de resultados de análisis de Learning Rate para cáncer de pulmón

De nuevo para el este cáncer se elegirá usar **0.005 de Learning Rate**, obteniendo los mejores resultados 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**

Mejores resultados		
Momentum	Iteraciones hasta límite(Train)	Iteraciones hasta límite (Valid)
0,1	1	2
0,2	1	1
0,3	1	1
0,4	1	1
0,5	2	3
0,6	2	2
0,7	2	1
0,8	7	2
0,9	200	1

Tabla 3.19: Resultados de análisis de momentum para cáncer de pulmón con Back-prop

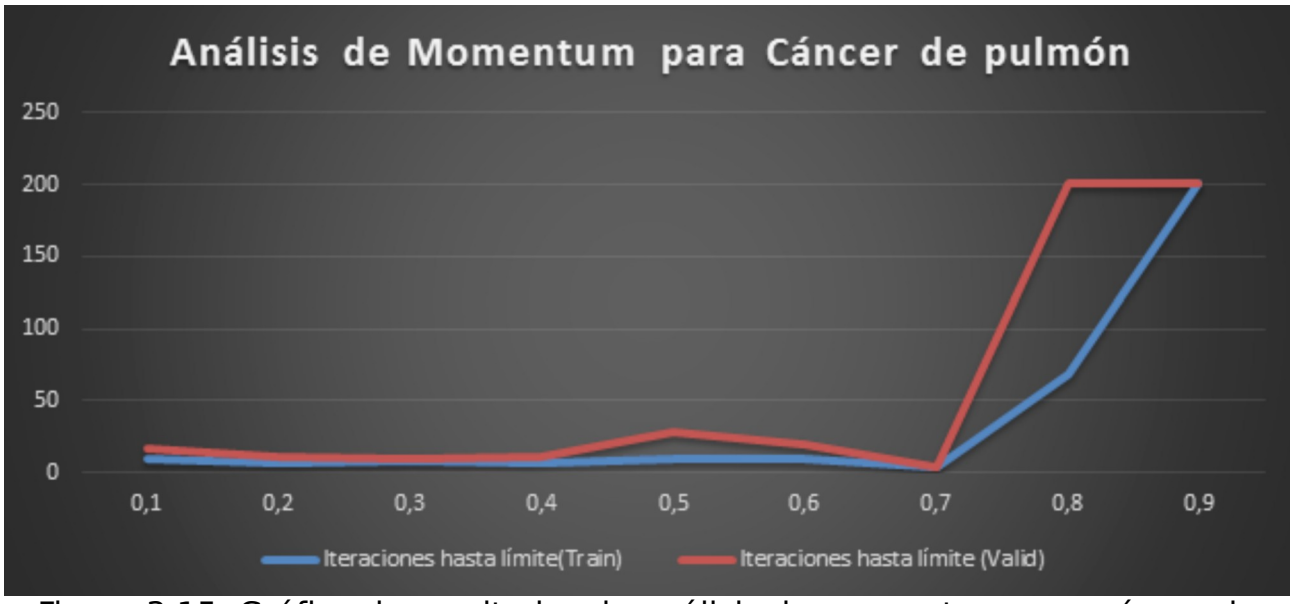


Figura 3.15: Gráfica de resultados de análisis de momentum para cáncer de pulmón

En este último caso del cáncer de pulmón, 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**.

3.5.2 Counter-propagation

- **Primera prueba**

Positivos	Negativos	Falsos positivos	Falsos negativos	Accuracy	Sensibili-dad	Especifici-dad
240	341	9	1	0,983	0,996	0,974

Ya desde la primera prueba se puede vislumbrar que este algoritmo tendrá un buen comportamiento, superando holgadamente a primeras pruebas para cánceres anteriores.

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

Medias			
Neuronas	Accuracy	Sensibilidad	Especificidad
5	0,94	0,96	0,93
6	0,92	0,98	0,88
7	0,94	0,94	0,94
8	0,91	0,93	0,89
9	0,91	0,96	0,88
10	0,92	0,93	0,92
11	0,95	0,95	0,94
12	0,95	0,94	0,95
13	0,91	0,97	0,87
14	0,92	0,96	0,89
15	0,95	0,97	0,93

Tabla 3.20: Resultados de análisis para neuronas del mapa auto-organizado para cáncer de pulmón

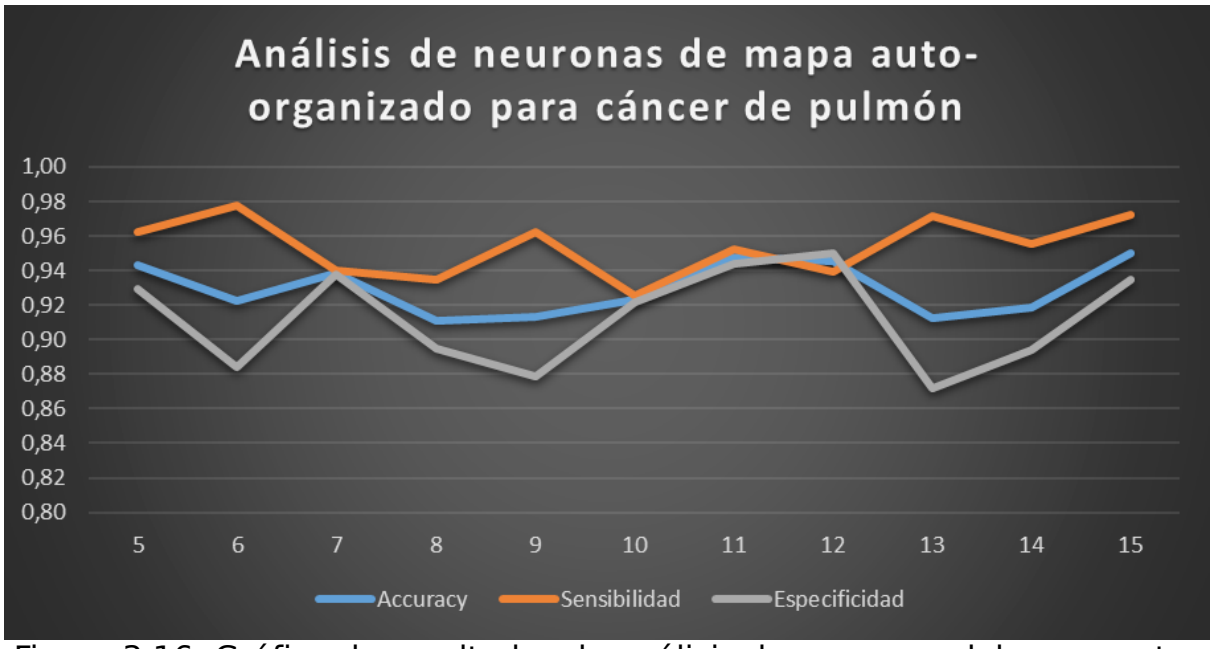


Figura 3.16: Gráfica de resultados de análisis de neuronas del mapa auto-organizado para cáncer de pulmón

Se escogerá usar **15 neuronas**, ya que a pesar de tener el mismo accuracy que las instancias 11 o 12, su sensibilidad es mayor. El mejor resultado para 15 neuronas fue de 97% de accuracy, 98% de sensibilidad y 97% de especificidad.

- **Análisis de Learning Rate**

Medias		
Learning Rate	Iteraciones(train)	Iteraciones(valid)
0,0001	200	200
0,0005	200	200
0,001	115	113,2
0,005	21,2	20,8
0,01	9,2	9,8
0,05	2	2,6

Tabla 3.21: Resultados de análisis de learning rate para cáncer de pulmón con counter-prop

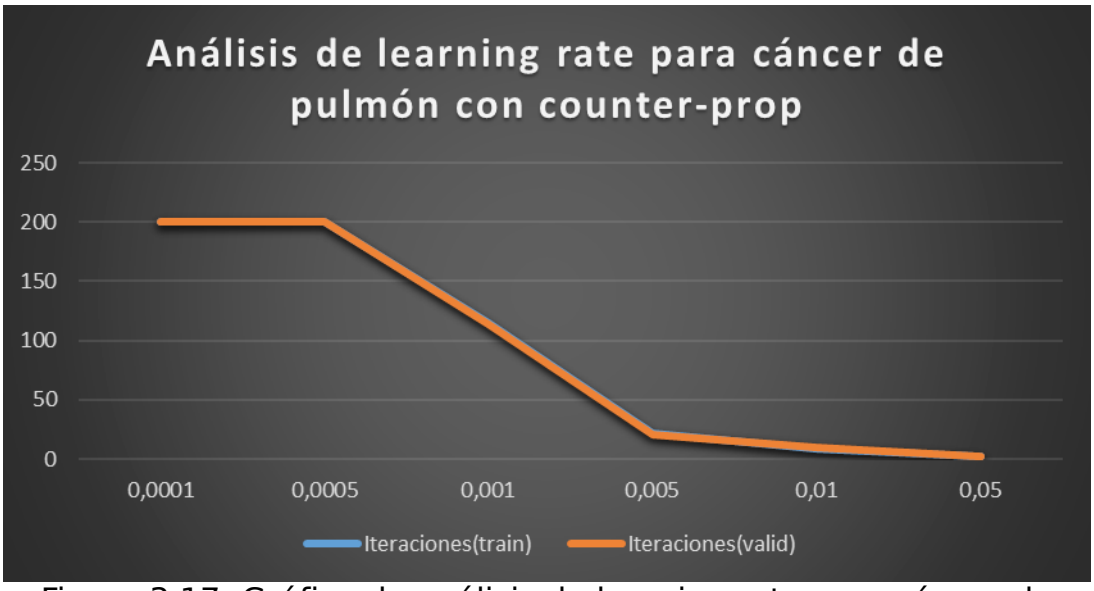


Figura 3.17: Gráfica de análisis de learning rate para cáncer de pulmón con counter-prop

Al igual que en los otros dos tipos de cáncer con este algoritmo, el learning rate será de 0.05. Esta variable también fue la misma para los 3 casos del algoritmo back-propagation. El mejor resultado de las 5 pruebas para la instancia que resultó elegida fue de 98% de accuracy, 99% de sensibilidad y 98% de especificidad.

- **Análisis de momentum**

Medias		
Momentum	Iteraciones(train)	Iteraciones(valid)
0,1	4,4	4,8
0,2	5,2	6,2
0,3	3,4	4,2
0,4	2,6	2,8
0,5	1,4	2,2
0,6	1,8	2,4
0,7	1,6	2,4
0,8	1,2	1,6
0,9	0,6	1

Tabla 3.22: Resultados de análisis de momentum para cáncer de pulmón con counter-prop

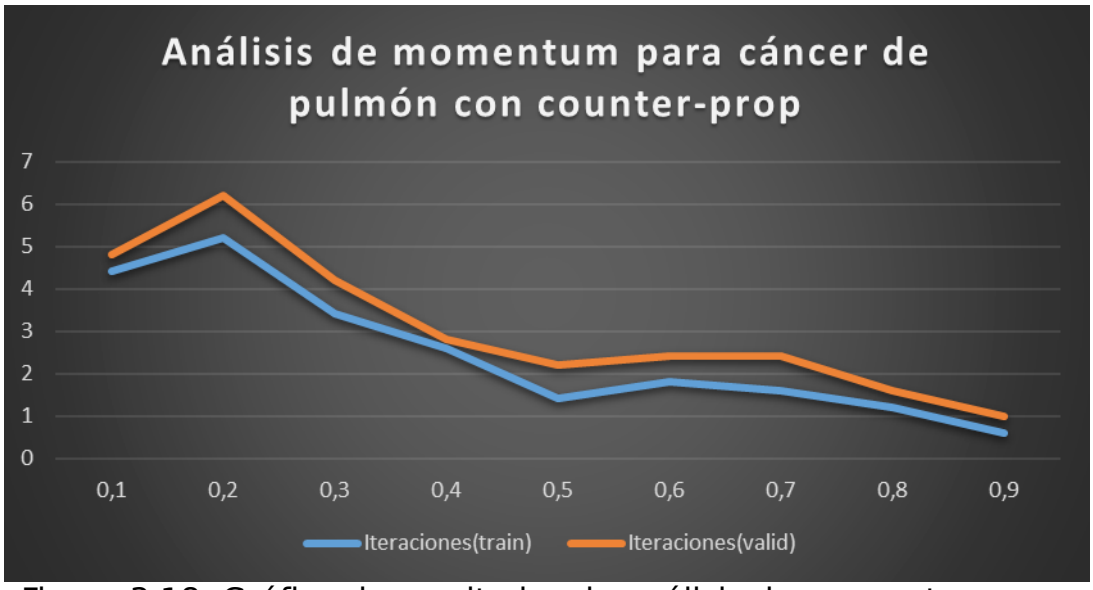


Figura 3.18: Gráfica de resultados de análisis de momentum para cáncer de pulmón con counter-prop

Como se puede observar en la tabla, las iteraciones medias para alcanzar el límite de error de 0.07 son menores para la instancia de 0.9 de momentum, por lo que se elegirá usar esta. El mejor resultado de las cinco pruebas para esa instancia fue de 95% de accuracy, 99% de sensibilidad y 93% de especificidad.

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	Learning Rate	Momentum
50	0,005	0,1

Tabla 3.23: Valor de variables finales para back-propagation en cáncer de mama

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 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 siguientes:

Positivos	Negativos	Falsos Positivos	Falsos Negativo	Accuracy	Sensibili-dad	Especifici-dad
80	37	33	27	0,661	0,748	0,529

Tabla 3.24: 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	Learning Rate	Momentum
14	0,05	0,8

Tabla 3.25: Valor de variables finales para Counter-propagation en cáncer de mama

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 60,45% de pacientes con cáncer y un 39,55 de pacientes sin cáncer.

Positivos	Negativos	Falsos positivos	Falsos Negativos	Accuracy	Sensibili-dad	Especifici-dad
98	76	2	0	0,989	1	0,974

Tabla 3.26: Resultados de test final con Counter-Propagation para Cáncer de mama

Se obtienen unos resultados para el dataset de test casi perfectos, donde el accuracy es casi un 99% y la sensibilidad de un 100%, lo que implica que cada paciente con cáncer fue diagnosticado correctamente, objetivo que era nuestra prioridad.

3.6.2 Melanoma

- Back-Propagation

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

Neuronas Ocultas	Learning Rate	Momentum
95	0,005	0,3

Tabla 3.27: Valor de variables finales para back-propagation en cáncer de mama

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 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	Sensibili-dad	Especifici-dad
94	72	32	29	0,731	0,764	0,692

Tabla 3.28: 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	Learning Rate	Momentum
6	0,05	0,9

Tabla 3.29: Valor de variables finales para Counter-Propagation en Melanoma

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 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	Sensibili-dad	Especifici-dad
107	98	6	16	0,903	0,870	0,942

Tabla 3.30: Resultados de test final con Counter-propagation para melanoma

De nuevo unos resutados que mejoran bastante los del algoritmo back-propagation, aunque en este caso destaca que la especificidad es más alta que la sensibilidad, lo cual, no es lo más indicado, pero parece ser un patrón después de haber visto los resultados durante los análisis.

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	Learning Rate	Momentum
75	0,005	0,7

Tabla 3.31: 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 siguientes:

Positivos	Negativos	Falsos Positivos	Falsos Negativos	Accuracy	Sensibili-dad	Especifici-dad
228	204	127	32	0,731	0,877	0,616

Tabla 3.32: Resultados de test final con Back-Propagation para cáncer de pulmón

- Counter-propagation

Neuronas del mapa	Learning Rate	Momentum
15	0,05	0,9

Tabla 3.33: Valor de variables para Counter-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 siguientes:

Positivos	Negativos	Falsos positivos	Falsos Negativos	Accuracy	Sensibili-dad	Especifici-dad
254	326	5	6	0,981	0,977	0,985

Tabla 3.34: Resultados de test final con Counter-propagation para cáncer de pulmón

3.6.4 Conclusiones del análisis

Los resultados son claros, el Counter-Propagation funciona mucho mejor que el Back-Propagation. Esto se debe a que el Back-Propagation puede caer en mínimos locales durante el entrenamiento, mientras que el Counter-Propagation, añadiendo la fase previa de entrenamiento no supervisado con el mapa de Kohonen, evita que el Back-Propagation posterior caiga en minimos locales, encontrando siempre el mínimo global.

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.

3.7.1 Cáncer de mama

	Accuracy	Sensibilidad	Especificidad
HEAL	74,35%	91,88%	78,76%
Proyecto	98,9%	100%	97,4%

Tabla 3.35: Comparación de resultados entre HEAL y proyecto para cáncer de mama

3.7.2 Melanoma

	Accuracy	Sensibilidad	Especificidad
HEAL	73,15%	66,67%	78,76%
Proyecto	90,3%	87%	94,2%

Tabla 3.36: Comparación de resultados entre HEAL y proyecto para Melanoma

3.7.3 Cáncer de Pulmón

	Accuracy	Sensibilidad	Especificidad
HEAL	87,3%	77,61%	94,37%
Proyecto	98,1%	97,7%	98,5%

3.7.4 Conclusiones

Como se puede observar, los resultados obtenidos en este proyecto con el algoritmo Counter-Propagation son significativamente mejores que los obtenidos por el HEAL con su

algoritmo de árboles genéticos, superando en los tres tipos de cáncer el 90% de accuracy y en el Cáncer de mama y Cáncer de pulmón superando el 98%.

Se obtienen también altos porcentajes de sensibilidad, que superan en 20 puntos a las sensibilidades del HEAL para el caso del cáncer de mama y cáncer de pulmón.

Capítulo 4
Conclusiones y líneas futuras

Al término del proyecto, se han conseguido todos los objetivos propuestos en un principio. Se elaboraron varias redes distintas y se estudió su aplicación para cada uno de los tres tipos de cáncer, obteniendo resultados muy satisfactorios para lo esperado con el algoritmo Counter-propagation.

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.

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. Various nets were developed and analyzed for every one of the three types of cancer treated, obtaining amazing results with Counter-propagation algorithm.

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.

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 Depp 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 €

Capítulo 7
Apéndice: Scripts

7.1 Script para Cáncer de mama

NO HE AÑADIDO LOS ENLACES A GITHUB PARA ESPERAR A QUE HAGA EL MERGE DE LA RAMA DEVEL Y LA MASTER, PORQUE EL ENLACE CAMBIARÍA

7.2 Script para Melanoma

7.3 Script para Cáncer de pulmón

Bibliografía y referencias

[1]Enlace a Github del proyecto:
https://github.com/etsiiull/medicalpybrain