



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, 17 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 cotutor

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 17 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



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

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 tres redes o algoritmos distintos: Back-propagation para entrenamiento supervisado, Deep Belief Network para entrenamiento no supervisado y Support Vector Machine como problema de clasificación.

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, Deep Belief Network, Support Vector Machine, 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 three different types of network or algorithms: Back-propagation for supervised training, Deep Belief Network for unsupervised training and Support Vector Machines as classification problem.

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, Deep Belief Network, Support Vector Machine, 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 Redes Neuronales Artificiales..... | 5 |
| 2.1 Historia..... | 5 |
| 2.2 Características..... | 5 |
| 2.3 Aprendizaje..... | 6 |
| 2.4 Estado actual..... | 6 |
| Capítulo 3 Herramientas..... | 7 |
| 3.1 Herramienta..... | 7 |
| 3.1.1 Algoritmos..... | 7 |
| 3.1.2 Funcionamiento de la herramienta..... | 10 |
| 3.1.3 ¿Cómo se llegó a elegir PyBrain?..... | 11 |
| 3.1.4 Conclusiones..... | 12 |
| Capítulo 4 Datos..... | 13 |
| 4.1 Obtención..... | 13 |
| 4.2 Contenido de la Base de datos..... | 13 |
| 4.3 Resultados de la investigación del HEAL..... | 14 |
| Capítulo 5 Desarrollo..... | 15 |

| | |
|---|-----------|
| 5.1 Preprocesado..... | 15 |
| 5.1.1 Eliminación de columnas poco significativas..... | 15 |
| 5.1.2 Sustitución de datos nulos..... | 16 |
| 5.1.3 Eliminación de cabeceras..... | 16 |
| 5.2 Desarrollo de Scripts..... | 16 |
| Capítulo 6 Análisis..... | 17 |
| 6.1 Explicación de los análisis..... | 17 |
| 6.1.1 Back-Propagation..... | 17 |
| 6.1.2 Cáncer de mama..... | 19 |
| 6.1.3 Melanoma..... | 21 |
| 6.1.4 Cáncer de pulmón..... | 24 |
| 6.1.5 Resultados finales usando el subconjunto de test..... | 27 |
| Capítulo 7 Conclusiones y líneas futuras..... | 31 |
| Capítulo 8 Summary and Conclusions..... | 32 |
| Capítulo 9 Presupuesto..... | 33 |
| Capítulo 10 Apéndice: Scripts..... | 34 |
| 10.1 Script para Cáncer de mama..... | 34 |
| 10.2 Script para Melanoma..... | 34 |
| 10.3 Script para Cáncer de pulmón..... | 34 |

Índice de figuras

| | |
|---|----|
| Figura 3.1: Diagrama de flujo de uso de la librería PyBrain..... | 10 |
| Figura 4.1: Tablas originales obtenidas de la presentación de resultados del proyecto del HEAL..... | 14 |
| Figura 6.1: Gráfica de resultados para análisis de neuronas ocultas en cáncer de mama..... | 19 |
| Figura 6.2: Gráfica de resultados de análisis de Learning Rate para cáncer de mama..... | 20 |
| Figura 6.3: Gráfica de resultados de análisis de Momentum para cáncer de mama..... | 21 |
| Figura 6.4: Gráfica de resultados de análisis de neuronas ocultas para melanoma..... | 22 |
| Figura 6.5: Gráfica de resultados de análisis de Learning Rate para melanoma..... | 23 |
| Figura 6.6: Gráfica de resultados de análisis de Momentum para melanoma | 24 |
| Figura 6.7: Gráfica de resultados de análisis de neuronas ocultas para cáncer de pulmón..... | 25 |
| Figura 6.8: Gráfica de resultados de análisis de Learning Rate para cáncer de pulmón..... | 26 |
| Figura 6.9: Gráfica de resultados de análisis de momentum para cáncer de pulmón..... | 27 |

Índice de tablas

| | |
|---|----|
| Tabla 6.1: Resultados de la primera prueba de Back-Propagation para Cáncer de mama..... | 19 |
| Tabla 6.2: Resultados del análisis de neuronas ocultas para Cáncer de mama..... | 19 |
| Tabla 6.3: Resultados de análisis de Learning Rate para Cáncer de mama | 20 |
| Tabla 6.4: Resultados de análisis de Momentum para Cáncer de mama.... | 21 |
| Tabla 6.5: Resultados de primera prueba de backpropagation para melanoma..... | 22 |
| Tabla 6.6: Resultados de análisis de neuronas ocultas para melanoma..... | 22 |
| Tabla 6.7: Resultados de análisis de learning rate para Melanoma..... | 23 |
| Tabla 6.8: Resultados de análisis de momentum para melanoma..... | 23 |
| Tabla 6.9: Resultados de análisis de neuronas ocultas para Cáncer de Pulmón..... | 25 |
| Tabla 6.10: Resultados de análisis de Learning Rate para cáncer de pulmón | 26 |
| Tabla 6.11: Resultados de análisis de momentum para cáncer de pulmón. . | 27 |
| Tabla 6.12: Valor de variables finales para back-propagation en cáncer de mama..... | 28 |
| Tabla 6.13: Resultados de test final con Back-Propagation para cáncer de mama..... | 28 |
| Tabla 6.14: Valor de variables finales para back-propagation en cáncer de mama..... | 28 |
| Tabla 6.15: Resultados de test final con Back-Propagation para melanoma | |

| | |
|--|----|
| | 29 |
| Tabla 6.16: Resultados de test final con Back-Propagation para cáncer de pulmón..... | 30 |

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 varios tipos de redes neuronales: RNA con Back-Propagation como algoritmo de entrenamiento, Support Vector Machine y Deep Belief Network para entrenamiento no supervisado.

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 quizás 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:

<https://github.com/etsiull/medicalpybrain>

Capítulo 2

Redes Neuronales Artificiales

En este capítulo se explicará a fondo la principal tecnología sobre la que se elabora este trabajo: las Redes Neuronales Artificiales.

2.1 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' y ha servido de inspiración para el desarrollo de otros modelos neuronales.

2.2 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.

2.3 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.

2.4 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.

Capítulo 3

Herramientas

En este capítulo se realizará un análisis sobre las características y pormenores de la librería de redes neuronales para Python conocida como Pybrain.

3.1 Herramienta

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.

3.1.1 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.

- 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, siendo el número 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.

3.1.2 Funcionamiento de la herramienta

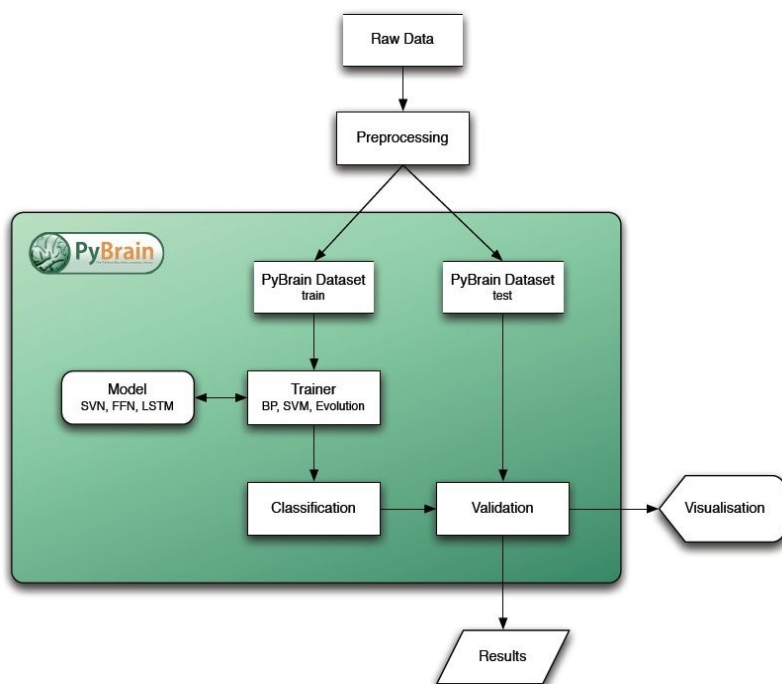


Figura 3.1: Diagrama de flujo de uso de la librería PyBrain

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.

3.1.3 ¿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.

3.1.4 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.

Capítulo 4

Datos

En esta sección se tratarán las bases de datos usadas durante el proyecto, explicando en detalle como se obtuvieron, así como el preprocesado llevado a cabo para su optimización.

4.1 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.

4.2 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.*

4.3 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.

| Criteria | Training | | | Test | | |
|------------------|----------|-------------|-------------|----------|-------------|-------------|
| | 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

| Criteria | Training | | | Test | | |
|------------------|----------|-------------|-------------|----------|-------------|-------------|
| | 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

| Criteria | Training | | | Test | | |
|------------------|----------|-------------|-------------|----------|-------------|-------------|
| | 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 4.1: Tablas originales obtenidas de la presentación de resultados del proyecto del HEAL

Capítulo 5

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.

5.1 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.

5.1.1 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).

5.1.2 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.

5.1.3 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.

5.2 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.

Capítulo 6

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.

6.1 Explicación de los análisis

6.1.1 Back-Propagation

Para el algoritmo de Back-Propagation se analizarán tres variables, a parte de una prueba inicial para comparar con los resultados finales:

- Prueba Inicial

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.

FALTA EXPLICAR LOS ANÁLISIS DE FUTUROS ALGORITMOS

EN LOS SIGUIENTES PÁRRAFOS NO SE HA TENIDO EN CUENTA LA ORDENACIÓN Y LA COLOCACIÓN DE LAS TABLAS Y FIGURAS CON RESPECTO AL TEXTO, YA QUE FALTAN DATOS POR PONER QUE ALTERARÍAN EL FORMATO DE NUEVO

6.1.2 Cáncer de mama

- Back-Propagation

El dataset de validación para el cáncer de mama con el que se realizó la primera prueba contiene un 53,41% de casos positivos y un 46,59% de casos negativos.

Los resultados fueron los siguientes:

- Primera prueba

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

Tabla 6.1: Resultados de la primera prueba de Back-Propagation para Cáncer de mama

- Análisis de neuronas ocultas

[illegible]

Tabla 6.2: Resultados del análisis de neuronas ocultas para Cáncer de mama

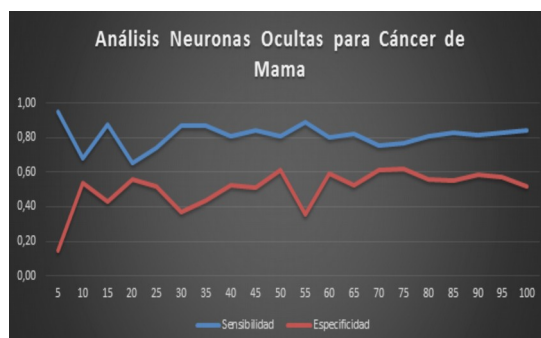


Figura 6.1: Gráfica de resultados para análisis de neuronas ocultas en cáncer de mama

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) | |
|---------------|----------------------------------|--|
| 0,0001 | 22 | |
| 0,0005 | 4 | |
| 0,001 | 3 | |
| 0,005 | 4 | |
| 0,01 | 11 | |
| 0,05 | 200 | |

Tabla 6.3: Resultados de análisis de Learning Rate para Cáncer de mama

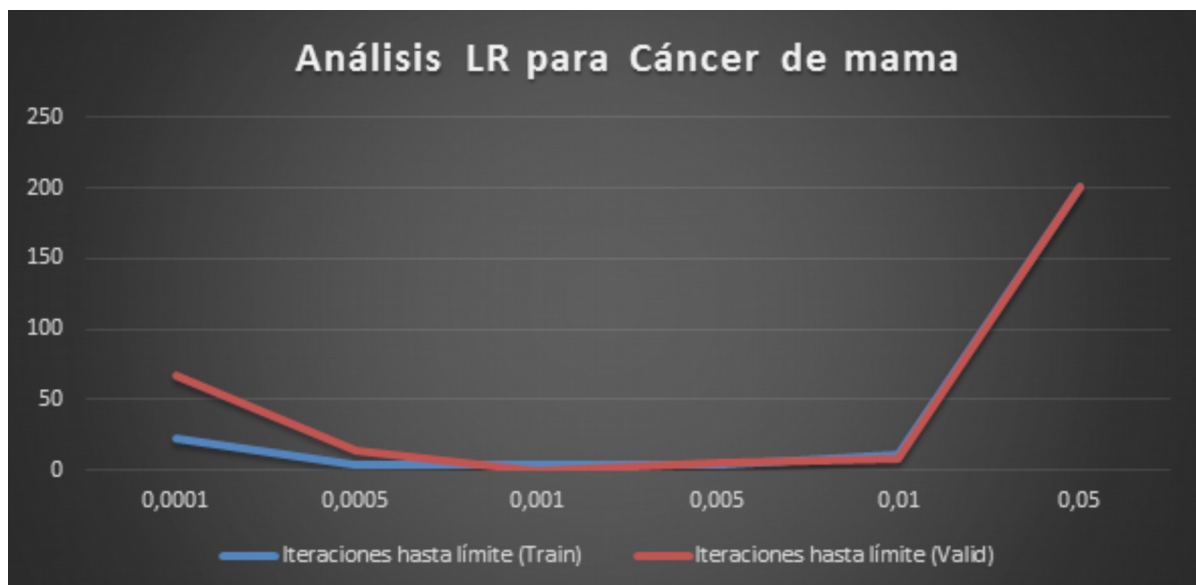


Figura 6.2: Gráfica de resultados de análisis de Learning Rate para cáncer de mama

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 Resu | | |
|--------------|---------------------------------|--|
| Momentum | Iteraciones hasta límite(Train) | |
| 0,1 | 2 | |
| 0,2 | 3 | |
| 0,3 | 2 | |
| 0,4 | 2 | |
| 0,5 | 3 | |
| 0,6 | 13 | |
| | | |
| | | |
| | | |

Tabla 6.4: Resultados de análisis de Momentum para Cáncer de mama

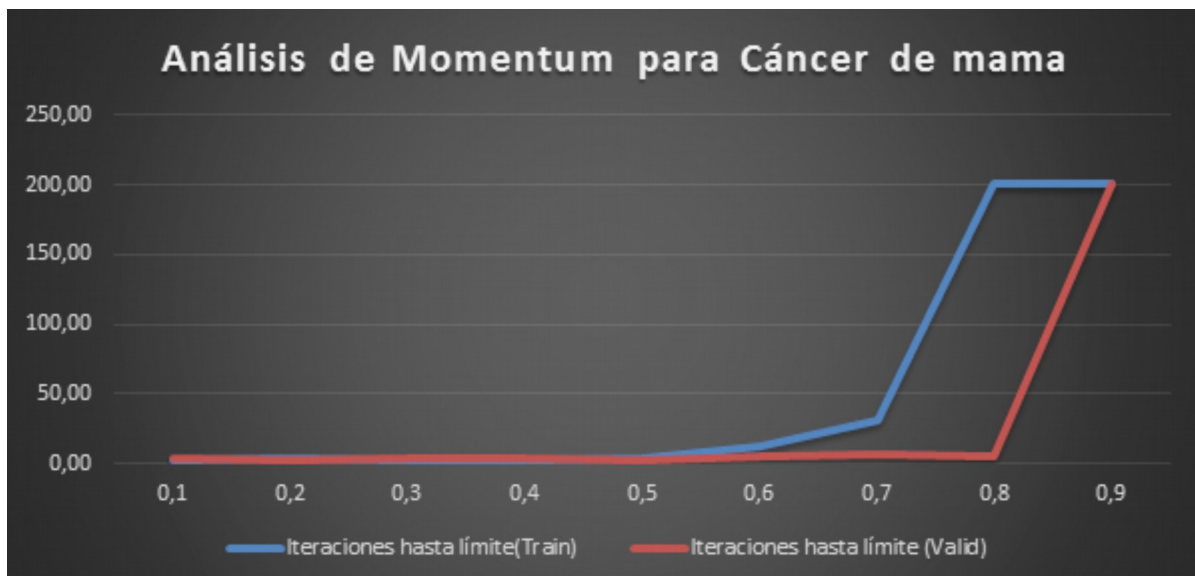


Figura 6.3: Gráfica de resultados de análisis de Momentum para cáncer de mama

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 los resultados de las cinco pruebas son más homogéneos por lo que se ha decidido usar ese valor.

6.1.3 Melanoma

- 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 | Sensibilidad | Especificidad |
|-----------|-----------|------------------|------------------|----------|--------------|---------------|
| 78 | 66 | 63 | 19 | 0,634 | 0,804 | 0,512 |

Tabla 6.5: Resultados de primera prueba de backpropagation para melanoma

- Análisis de Neuronas Ocultas

[illegible]

Tabla 6.6: 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

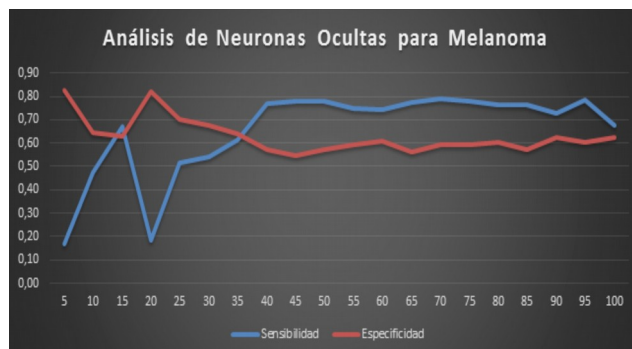


Figura 6.4: Gráfica de resultados de análisis de neuronas ocultas para melanoma

| Learning rate | Iteraciones hasta límite(Trai | |
|---------------|-------------------------------|--|
| 0,0001 | 200 | |
| 0,0005 | 24 | |
| 0,001 | 14 | |
| 0,005 | 5 | |
| 0,01 | 23 | |
| 0,05 | 200 | |

Tabla 6.7: Resultados de análisis de learning rate para Melanoma

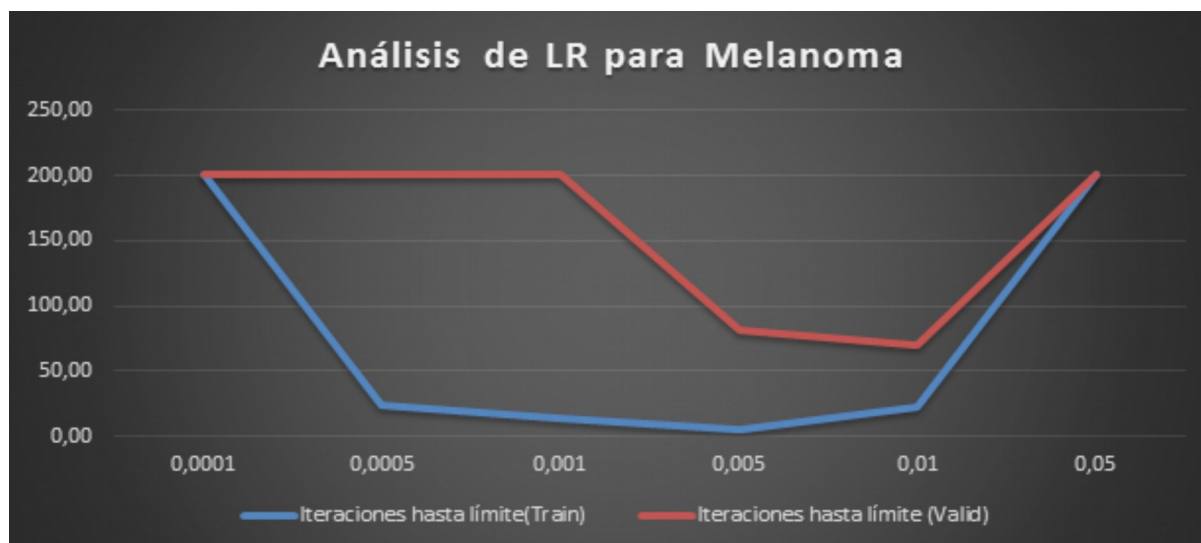


Figura 6.5: Gráfica de resultados de análisis de Learning Rate para melanoma

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

| Momentum | Iteraciones hasta límite(Train) | |
|----------|---------------------------------|--|
| 0,1 | 9 | |
| 0,2 | 7 | |
| 0,3 | 8 | |
| 0,4 | 7 | |
| 0,5 | 10 | |
| 0,6 | 9 | |
| 0,7 | 3 | |
| | | |
| | | |

Tabla 6.8: Resultados de análisis de momentum para melanoma

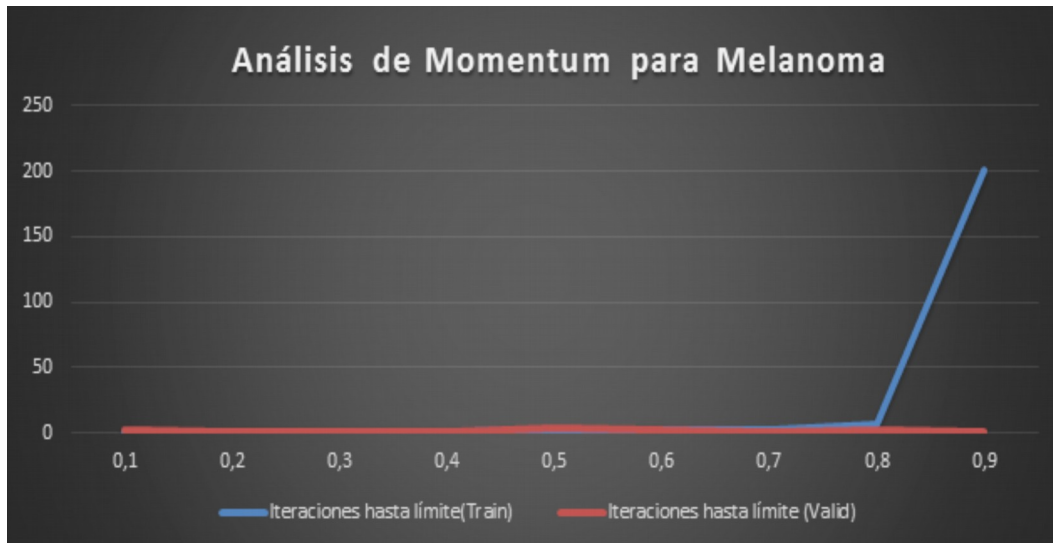


Figura 6.6: Gráfica de resultados de análisis de Momentum para melanoma

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.

6.1.4 Cáncer de pulmón

- 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 | 0,840 |

- Análisis de Neuronas Ocultas

[illegible]

Tabla 6.9: Resultados de análisis de neuronas ocultas para Cáncer de Pulmón

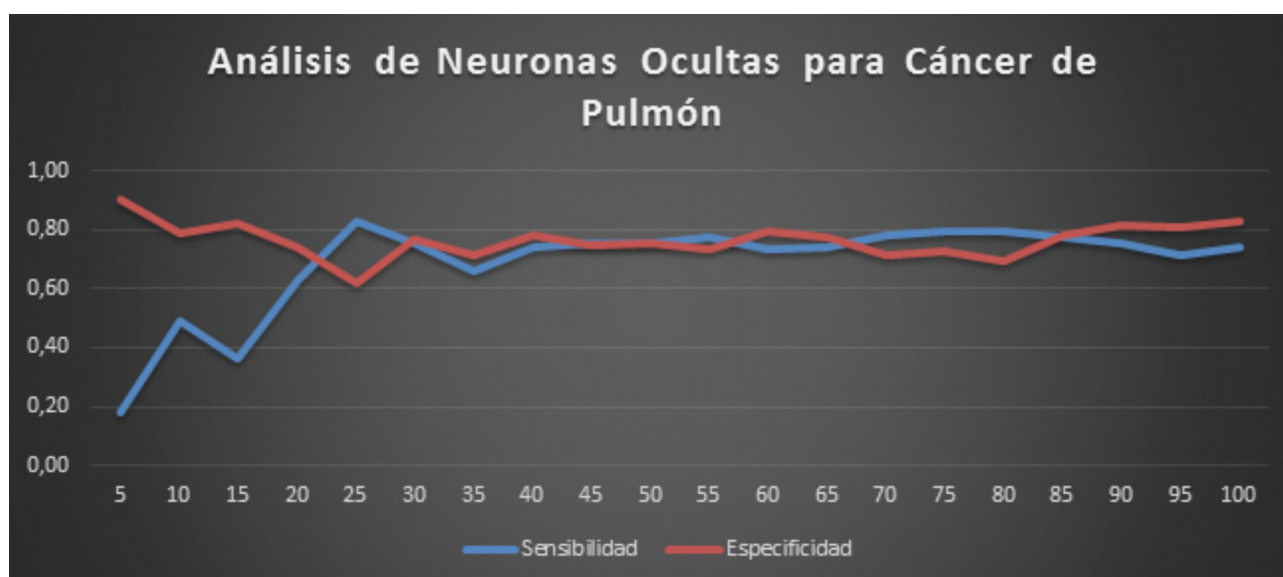


Figura 6.7: Gráfica de resultados de análisis de neuronas ocultas para cáncer de pulmón

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

| Learning Rate | Iteraciones hasta límite (Tr | |
|---------------|------------------------------|--|
| 0,0001 | 22 | |
| 0,0005 | 3 | |
| 0,001 | 2 | |
| 0,005 | 2 | |
| 0,01 | 8 | |
| 0,05 | 200 | |

Tabla 6.10: Resultados de análisis de Learning Rate para cáncer de pulmón

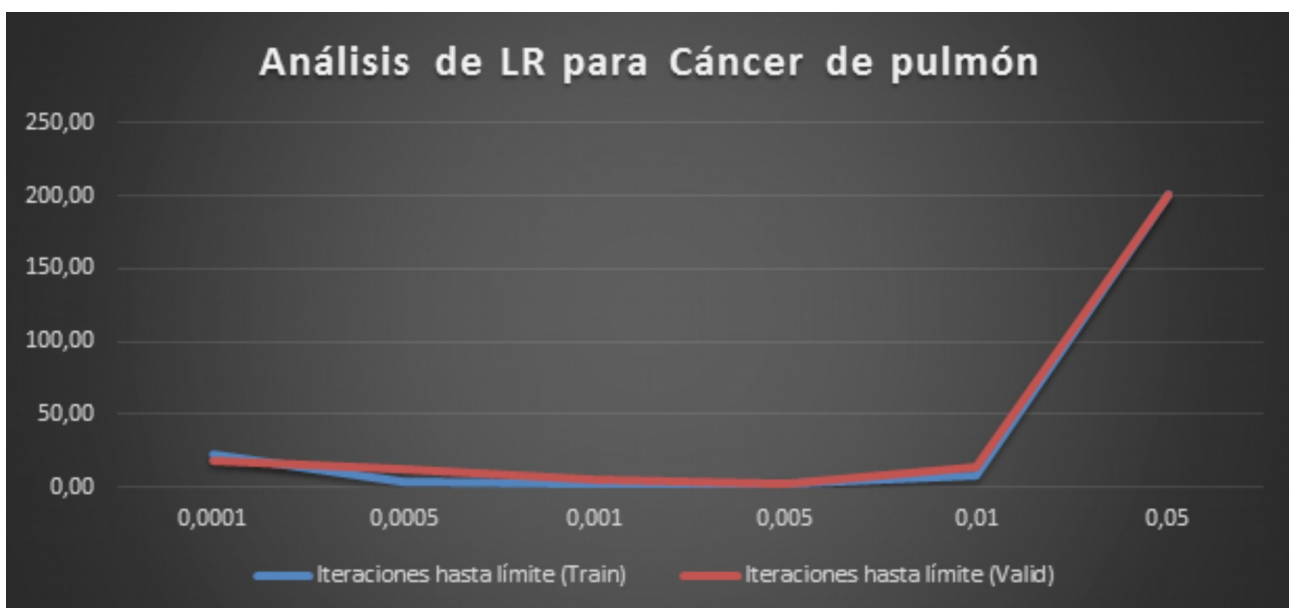


Figura 6.8: Gráfica de resultados de análisis de Learning Rate para cáncer de pulmón

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

| Momentum | Iteraciones hasta límite(Train) | |
|----------|---------------------------------|--|
| 0,1 | 1 | |
| 0,2 | 1 | |
| 0,3 | 1 | |
| 0,4 | 1 | |
| 0,5 | 2 | |
| 0,6 | 2 | |
| 0,7 | 2 | |
| | | |
| | | |

Tabla 6.11: Resultados de análisis de momentum para cáncer de pulmón

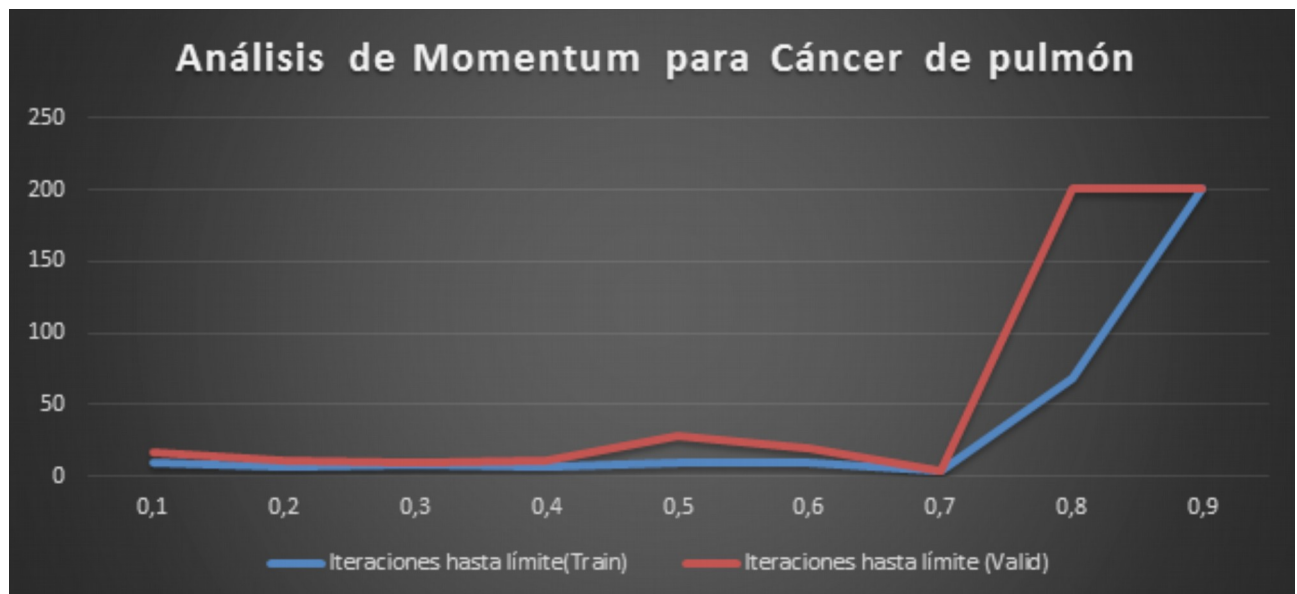


Figura 6.9: Gráfica de resultados de análisis de momentum para cáncer de pulmón

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**.

6.1.5 Resultados finales usando el subconjunto de test

- 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 6.12: 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 | Sensibilidad | Especificidad |
|-----------|-----------|------------------|-----------------|----------|--------------|---------------|
| 80 | 37 | 33 | 27 | 0,661 | 0,748 | 0,529 |

Tabla 6.13: 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.

- Algoritmo 2
- Algoritmo 3
- Melanoma
 - 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 |
|------------------|---------------|----------|
| 95 | 0,005 | 0,3 |

Tabla 6.14: 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 | Sensibilidad | Especificidad |
|-----------|-----------|------------------|------------------|----------|--------------|---------------|
| 94 | 72 | 32 | 29 | 0,731 | 0,764 | 0,692 |

Tabla 6.15: Resultados de test final con Back-Propagation para melanoma

- Algoritmo 2
- Algoritmo 3
- Cáncer de Pulmón
 - 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 |
|------------------|---------------|----------|
| 75 | 0,005 | 0,7 |

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 | Sensibilidad | Especificidad |
|-----------|-----------|------------------|------------------|----------|--------------|---------------|
| 228 | 204 | 127 | 32 | 0,731 | 0,877 | 0,616 |

Tabla 6.16: Resultados de test final con Back-Propagation para cáncer de pulmón

- Algoritmo 2
- Algoritmo 3

FALTA CAPÍTULO DE COMPARACIÓN CON LOS RESULTADOS DEL HEAL

Capítulo 7

Conclusiones y líneas futuras

FALTA TERMINAR ANÁLISIS PARA PODER COMPLETAR ESTE APARTADO

Capítulo 8

Summary and Conclusions

FALTA TERMINAR ANÁLISIS PARA PODER COMPLETAR ESTE APARTADO

Capítulo 9

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 10

Apéndice: Scripts

10.1 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

10.2 Script para Melanoma

10.3 Script para Cáncer de pulmón

Bibliografía