UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE SISTEMAS INFORMÁTICOS

Diagnóstico de la enfermedad de Parkinson usando deep learning y grabaciones de voz mediante teléfono móvil

Proyecto Fin de Grado

Grado en Ingeniería del Software

Curso académico 2018-2019

Autor:

Belén García-Botija Aldana

Tutores:

Alfonsa García López

Luis Miguel Pozo Coronado

A Alfonsa y Luis, por la dedicación y amabilidad que han tenido conmigo. Gracias por vuestra implicación y confianza.
A todos esos profesores con tanta vocación, por transmitir vuestra energía, conocimientos y ganas de aprender.
A mis amigos, por su apoyo incondicional y todos esos momentos divertidos.
A More, porque sin su ayuda y cariño todo sería más difícil.
A mis padres, por apoyarme siempre tanto y creer en mí.

Resumen

La enfermedad del Parkinson (EP) es un desorden neurodegenerativo crónico del que se desconoce la causa. Es la segunda enfermedad neurodegenerativa más común y se espera que para 2050 el número de diagnosticados se triplique. Sus síntomas, además de temblores, rigidez muscular o falta de equilibrio, comprenden trastornos en la voz y el habla en fases tempranas de la enfermedad.

Su proceso de diagnosis y seguimiento, hoy en día, es lento y complicado al no existir ninguna prueba específica. En su defecto, el personal sanitario tiene que evaluar concienzudamente el historial clínico, síntomas y exámenes físicos y neurológicos de cada paciente, y ofrecer un diagnóstico a menudo poco preciso. Es por ello, que se estima que hasta un 25 % de los diagnosticados como enfermos de Parkinson realmente padecen otra enfermedad, es decir, el proceso actual provoca numerosos falsos positivos.

En los últimos años se ha desarrollado mucha investigación dirigida al uso del análisis de la voz como método de diagnosis y seguimiento de la enfermedad de Parkinson. Técnicas de minería de datos aplicadas a características vocales, extraídas de grabaciones realizadas en condiciones controladas, han permitido la discriminación entre enfermos y sanos con altas tasas de éxito y proponer el análisis de voz para el seguimiento de la enfermedad. Los importantes avances en TIC han abierto la posibilidad del seguimiento online de la enfermedad mediante el análisis de grabaciones de voz recogidas con teléfonos móviles, que permiten disponer de datos de un gran número de pacientes.

Con todo esto, mi Proyecto de Fin de Grado pretende dar una alternativa al diagnóstico actual de la enfermedad a través de grabaciones de voz de los pacientes. En este proyecto se han utilizado los datos de la actividad de voz de mPower, que entre otras cosas incluye más de 64000 grabaciones de voz, para discriminar enfermos de Parkinson y controles mediante un análisis básico de los datos, una extracción de características y la aplicación de algoritmos de minería de datos y de aprendizaje profundo.

Este avance supone una mejora considerable en la vida de las personas al proponer un diagnóstico y seguimiento más preciso de la enfermedad sin necesidad de acudir asiduamente a los centros médicos.

Abstract

Parkinson's disease (PD) is a chronic neurodegenerative disorder whose cause is unknown. It is the second most common neurodegenerative disease and it is expected that by 2050 the number of diagnosed will triple. Its symptons, in addition to tremors, muscle rigidity or lack of balance, include disorders in the voice and speech in early phases of the disease.

Its process of diagnosis and monitoring, nowadays, is slow and complex because there is no specific test. Failing this, health personnel must thoroughly evaluate the clinical history, symptoms and physical and neurological examinations of each patient, and propose a diagnosis that is often not very precise. That is why it is estimated that up to 25 % of those diagnosed as Parkinson's patients actually suffer from another illness, in other words, the current process causes numerous false positives.

In recent years, much research has been conducted aimed at the use of voice analysis as a method of diagnosis and monitoring of Parkinson's disease. Data mining techniques applied to vocal characteristics, extracted from recordings made under controlled conditions, have allowed discrimination between sick and healthy people with high success rates and propose voice analysis for the monitoring of the disorder. Important advances in ICT have opened the possibility of online monitoring of the disease through the analysis of voice recordings collected with mobile phones, which allow having data from a large number of patients.

With all this, my end-of-degree project aims to provide an alternative to the current diagnosis of the illness through voice recordings of patients. In this project we have used mPower voice activity data, which among other things includes more than 64,000 voice recordings, to discriminate Parkinson's patients and controls through a basic analysis of the data, a feature extraction and the application of data mining and deep learning algorithms.

This advance represents a considerable improvement in the lives of people by proposing a diagnosis and more accurate monitoring of the disorder without having to go frequently to medical centers.

Índice

Índice de tablas

Índice de figuras

1. Introducción

1.1. Contexto

Los primeros indicios de la enfermedad del Parkinson (EP) los encontramos en antiguos textos indios que datan del año 2500 a.C. En ellos, se describen episodios de temblores e incluso parálisis en algunos individuos.

A pesar de que numerosas personalidades hagan referencia a estos síntomas a lo largo de la historia, no es hasta 1817 cuando James Parkinson, un cirujano británico, publica "An essay on the shaking palsy". En él, logra agrupar diversos síntomas bajo una única dolencia, la EP. Sin embargo, no se conocería como enfermedad de Parkinson hasta 1880, cuando el neurólogo francés Jean-Marie Charcot añadió nuevos síntomas y la bautizó. [*Historia de la Enfermedad del Parkinson,* 2019]

La EP es una enfermedad neurodegenerativa crónica de causa desconocida que afecta al sistema nervioso central, provocando síntomas motores y no motores. Es la segunda enfermedad neurodegenerativa más común por detrás del Alzheimer [Catalán y Álamo, 2019].

Los síntomas principales son temblores, rigidez muscular, falta de equilibrio y trastornos del habla y voz, entre otros. Estos trastornos del habla afectan al 90 % de los pacientes y suceden en etapas iniciales de la enfermedad [Roberts-South, 2017].

El habla de los enfermos de la EP se caracteriza por una intensidad monótona, con ataques lentos, excesivas pausas para respirar y repeticiones de sílabas. A día de hoy, esta dolencia no tiene una prueba específica para su diagnóstico, por lo que el personal sanitario debe basarse en la historia clínica del paciente, los síntomas y un examen físico y neurológico. Es por todo ello, que el estudio de la voz puede ser un gran método de diagnóstico y de seguimiento de la enfermedad.

Otro aspecto a tener en cuenta, es que este nuevo método no invasivo ofrecería una mayor agilidad en el proceso de diagnosis y monitorización. Los sistemas de salud podrían ahorrar bastantes costes y esfuerzos y los pacientes no tendrían que desplazarse tan asiduamente a los centros de salud para las revisiones.

Desde hace una década, bastantes investigadores han mostrado gran interés por ofrecer una solución para el diagnóstico del Parkinson mediante la voz. En un principio, se utilizaban grabaciones de voz realizadas en laboratorio de las que se extraían una serie de características, con la intención de usarlas como predictores para clasificar enfermos de Parkinson y controles sanos. Estas grabaciones, por lo general, solían ser vocales sostenidas, ya que como se demostró en [Sakar et al., 2013], ofrecen una mayor información que las palabras o frases cortas.

Habitualmente, se realizaba una selección de las características extraídas para mejorar la efectividad de los métodos de data mining (KNN, SVM o bosques aleatorios). (ver [Tan, Steinbach y Kumar, 2006]). Este es el caso de trabajos como [Chen et al., 2013], [Little et al., 2009] y [Trister et al., 2016]. De esta manera, se analizaban las características y, para evitar redundancias y simplificar el problema, se eliminaban aquellas que estuvieran fuertemente correlacionadas. Para la selección hay diversos algoritmos a seguir y encontramos una excelente comparación de cuatro de ellos en [Tsanas, Little, McSharry, Spielman et al., 2012].

Esta selección de características sigue presente y ocupa la mayor parte de los artículos e incluso los hay específicos como [Soliman et al., 2016]. Mediante estos métodos tradicionales, es posible discriminar si un paciente sufre la EP o no, e incluso se puede predecir cuál es su nivel de UPDRS. La UPDRS (en inglés, Unified Parkinson's Disease Rating Scale) es una escala para la valoración de la enfermedad del Parkinson que mide síntomas motores y no motores, y es muy útil para realizar un seguimiento de los pacientes.

Cabe destacar que hasta ahora la gran mayoría de los estudios han evaluado sus modelos empleando validación cruzada o similares, sin reparar en que se encuentran diferentes grabaciones de un mismo individuo tanto en el entrenamiento como en el test. Este puede ser un motivo significativo por el cual obtienen resultados tan optimistas. Tan solo en [Sakar et al., 2013] se plantean esto mismo y proponen un sistema de validación en el que todas las grabaciones de un individuo las resumen en una a través de medidas de centralización y dispersión, evitando así este problema.

Otro factor común en las investigaciones ha sido el número de participantes implicados en el proyecto, que en general es menor de 45, pudiendo realizar cada uno numerosas grabaciones. No obstante, hay excepciones como Arora y Tsanas, que en 2016 emplearon grabaciones telefónicas de miles de pacientes con las que realizaron la selección de características tradicional [Arora y Tsanas, 2016]. Consiguieron resultados algo pobres, aunque abrieron la posibilidad de emplear la voz como biomarcador de la EP, incluso con grabaciones telefónicas de baja calidad.

Las grabaciones utilizadas en este proyecto se han obtenido del proyecto mPower, que consiste en una aplicación móvil destinada a recoger información relevante de enfermos de Parkinson. En ella, se realizan diversos cuestionarios y encuestas, además de tests de voz, de memoria, de pulsaciones sobre la pantalla del dispositivo y de la destreza al caminar del individuo. Estos datos son exportados a la plataforma Synapse, que se encarga de distribuirlos a los investigadores para que de manera colaborativa se analicen y se consigan resultados positivos para la enfermedad. [Bot et al., 2016]

Este Proyecto Fin de Grado (PFG) pretende diagnosticar la EP mediante grabaciones telefónicas de los pacientes. Igualmente, se pretende ofrecer a los enfermos y a sus familiares una mayor calidad de vida diagnosticando esta dolencia en fases tempranas y realizando un seguimiento online de la enfermedad.

Para ello, se utilizan redes de neuronas artificiales que aprenden a reconocer si un paciente tiene Parkinson o no. Este aprendizaje lo hacen por sí mismas de manera supervisada generalizando comportamientos y reconociendo patrones sobre dos conjuntos de datos. El primer conjunto son características de sonido extraídas de archivos de voz grabados por pacientes y el segundo, esos archivos de voz en crudo directamente.

1.2. Objetivos

El objetivo principal de este proyecto consiste en la búsqueda de modelos eficaces de detección de la enfermedad del Parkinson. Para ello, se utilizarán datos de voz recogidos a través de grabaciones telefónicas y se implementarán redes de neuronas basadas en Deep Learning.

Objetivos específicos

Para alcanzar el objetivo principal, es necesario cubrir los siguientes objetivos específicos:

1. Aprender lenguaje R en el entorno de desarrollo RStudio.
2. Conocer las posibilidades de la librería Keras.
3. Estudiar el estado del arte, tanto en lo relativo a métodos de deep learning como de investigaciones previas sobre el mismo tema realizadas con data mining.
4. Plantear diferentes experimentos para generar y evaluar diversos modelos posibles.
5. Realizar una comparativa de resultados entre archivos de audio en crudo y características extraídas de esos audios, además de con otros trabajos de investigadores destacados.

1.3. Estructura del documento

Tras esta introducción, el resto del Proyecto de Fin de Grado se estructura en las siguientes secciones:

Fundamentos teóricos: Se expone el concepto de deep learning y machine learning y su funcionamiento. Además, se clasifican los diferentes tipos de redes neuronales, se presentan las métricas que evalúan un modelo de machine learning y las dos herramientas predominantes en este contexto (TensorFlow y Keras).

Datos: Se comenta el origen y el proceso de obtención de los datos empleados en este proyecto. También se analizan sus características y los diferentes conjuntos de datos creados para el desarrollo de los experimentos.

Metodología: Se presenta la metodología seguida, tanto en este PFG como en los distintos experimentos.

Experimentos y análisis de resultados: Se exponen los resultados de los experimentos organizados y se hace una comparativa con otros trabajos relacionados.

Conclusiones: Se desarrollan las conclusiones obtenidas de la investigación y la experiencia personal de haber realizado este proyecto.

Impacto social: Se analiza el impacto social que supone este PFG dentro de esta línea de investigación.

Trabajos futuros: Se enumeran aquellas propuestas que aportarán mayor valor al resultado obtenido.

2. Fundamentos teóricos

2.1. ¿Qué es y cómo funciona el Deep Learning?

Para poder definir deep learning, primero debemos situar este campo en la Inteligencia artificial.

La Inteligencia artificial nació en la década de 1950 y trata de automatizar tareas intelectuales que normalmente realizan los humanos. Por lo tanto, la Inteligencia Artificial es un campo muy amplio que incluye Machine learning y Deep learning.

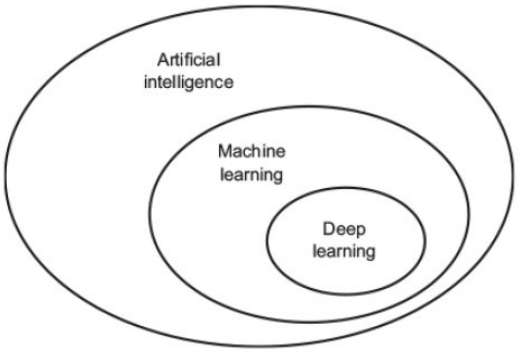

Figura 1: Inteligencia artificial, Machine learning y Deep learning.
Imagen obtenida de [Chollet y Allaire, 2018].

Machine learning se basa en la pregunta de si los ordenadores serían capaces de aprender por sí mismos cómo realizar alguna tarea. Esto supuso un nuevo paradigma de programación, en el que las personas ya no indican las reglas (programas), los datos a tratar y esperan las respuestas, sino que el sistema de machine learning recibe los datos, las respuestas esperadas y es el propio sistema entrenado el que obtiene las reglas.

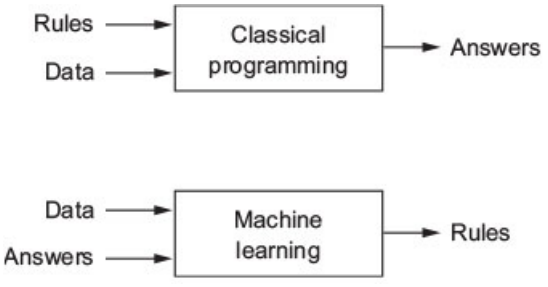

Figura 2: Programación clásica VS Machine Learning.
Imagen obtenida de [Chollet y Allaire, 2018].

Por lo tanto, cualquier algoritmo de machine learning necesita tres elementos: datos de entrada, ejemplos de las salidas esperadas y una manera de medir si el algoritmo está funcionando correctamente y determinar la distancia entre la salida real y la esperada. Esta medida se utiliza para ajustar el modelo y es lo que se denomina aprendizaje. La esencia principal de estos algoritmos es realizar una transformación

útil de los datos de entrada de manera que estas nuevas representaciones acerquen el modelo a las salidas esperadas. Las transformaciones son aprendidas por el modelo a través de una exposición numerosa de ejemplos.

Con todo ello, el deep learning es un campo específico del machine learning que se centra en el aprendizaje mediante capas sucesivas de representaciones cada vez más significativas que forman una red de neuronas artificiales. La cantidad de capas se conoce como profundidad del modelo. Actualmente, se suelen construir modelos de decenas y cientos de capas.

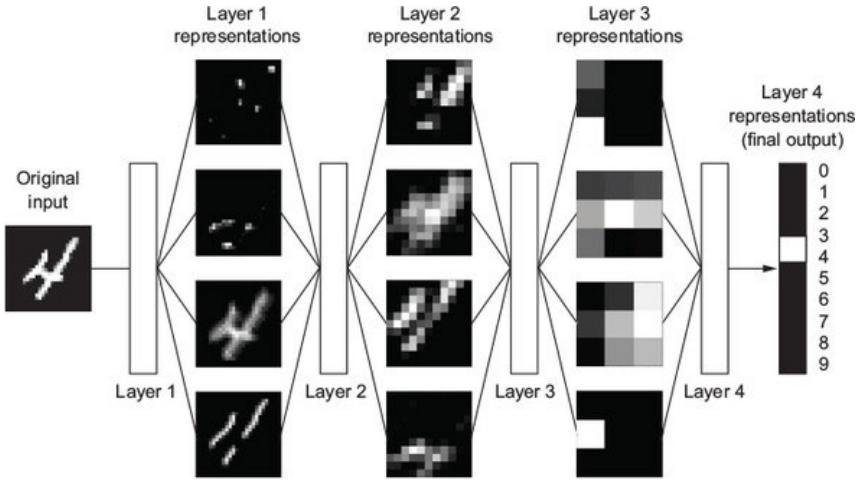

Figura 3: Representaciones de las capas de un modelo clasificador de dígitos. Imagen obtenida de [Chollet y Allaire, 2018].

Este aprendizaje de representaciones se lleva a cabo mediante los pesos de las capas (conjunto de números) que almacenan la transformación que realiza cada capa a los datos que le llegan. Así, se puede considerar que el aprendizaje consiste en encontrar el conjunto de pesos para todas las capas de manera que el modelo siempre clasifique las entradas con sus salidas correspondientes.

Como encontrar esos valores específicos es muy complejo, teniendo en cuenta que puede haber millones de parámetros, la forma de acercar la salida lo más posible a la esperada es mediante la función pérdida, que mide esta distancia e informa cómo de bien lo ha hecho la red para cada ejemplo. Con esta información, un optimizador ajusta los pesos con el objetivo de minimizar la función pérdida. Este ajuste se conoce como algoritmo de propagación hacia atrás (en inglés, *backpropagation algorithm*) y es el algoritmo principal en deep learning.

Al inicio, los pesos de la red se inicializan aleatoriamente por lo que la red implementa varias transformaciones aleatorias obteniendo una función pérdida muy elevada. Conforme se van sucediendo los ejemplos, los pesos se van ajustando y se minimiza la función pérdida, es lo que se conoce como entrenamiento. En este momento las salidas obtenidas son lo más similares posibles a las deseadas y se considera que la red está entrenada.

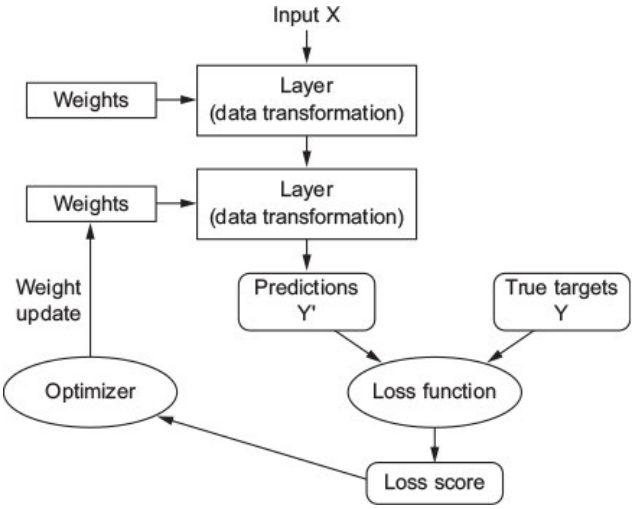

Figura 4: Funcionamiento Deep Learning.
Imagen obtenida de [Chollet y Allaire, 2018].

El ajuste de los pesos durante el entrenamiento es lo que puede resultar más complejo. Dado un coeficiente de peso individual en la red, ¿cómo saber si el coeficiente debe aumentarse o disminuirse y en qué medida? Como todas las operaciones utilizadas en la red son diferenciables, se puede calcular el gradiente de la función pérdida con respecto a los coeficientes de la red y moverlos en la dirección opuesta al gradiente, lo que disminuiría la función pérdida.

Dada una función derivable, los mínimos relativos se encuentran donde la derivada sea 0 y el menor valor de ellos será el mínimo absoluto de la función. En nuestro caso, habría que encontrar la combinación de pesos que minimice la función pérdida buscando los puntos en los que el gradiente sea 0. Al contar con incluso millones de parámetros, esto sería imposible. En su defecto, los coeficientes de los pesos varían en la dirección contraria del gradiente pero de manera gradual. Así los nuevos valores de los pesos se calcularían: $W = W - (step * gradient)$.

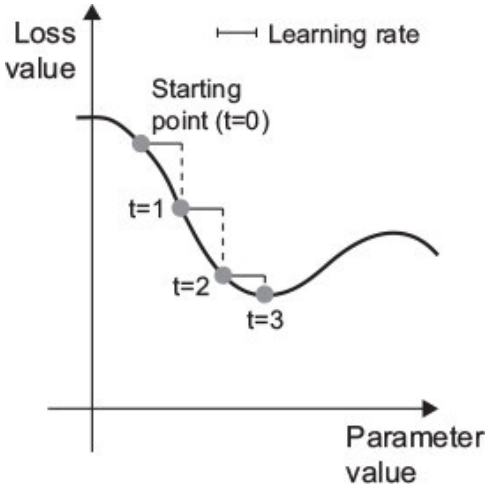

Figura 5: Descenso del gradiente de una red con un parámetro y un solo ejemplo.
Imagen obtenida de [Chollet y Allaire, 2018].

El problema de este proceso es el tamaño del "paso" a dar, también conocido como tasa de aprendizaje. Si es muy pequeño, podríamos atascarnos en un mínimo local. Si, por el contrario, es muy grande, nos llevaría a puntos de la curva totalmente aleatorios. Por eso, es un factor muy importante a elegir.

Hay varias formas de realizar la actualización de pesos: tras cada ejemplo, tras todos los ejemplos del conjunto de datos y la más común, tras un número determinado de ejemplos que se conoce como mini-batch gradient descent.

El algoritmo explicado hasta ahora se denomina SGD (en inglés, *stochastic gradient descent*), pero existen múltiples variantes que se conocen como optimizadores que no solo tienen en cuenta el valor de los gradientes en el momento, sino también actualizaciones anteriores de los pesos. Algunos de estos optimizadores son: Adagrad, Adam, RMSProp...

Para poder optimizar adecuadamente los pesos de cada capa, entra en escena el algoritmo de propagación hacia atrás que ya se mencionó anteriormente. Este, a partir del valor final de la función pérdida, recorre desde las últimas capas hasta las primeras valorando la contribución de cada parámetro en esa pérdida y así, los actualiza y ajusta con mayor precisión.

Debido al extenso uso de frameworks como TensorFlow, este algoritmo no hay que implementarlo a mano, pues ya existe una función que calcula el gradiente de un conjunto de operaciones y bastaría con una llamada a esta. De este modo, la implementación de redes de neuronas es cada vez más sencilla.

2.2. Tipos de redes según su conectividad

En deep learning las diferentes arquitecturas neuronales se pueden clasificar según la conectividad de las capas y neuronas.

Las más comunes son las redes de capas densas, cuyas neuronas se encuentran totalmente conectadas con las neuronas de la capa siguiente y realizan operaciones lineales. Son capaces de aprender patrones globales que incluyen todo el espacio de entrada.

Por otro lado, también se encuentran las redes convolucionales en las que las "conexiones" entre sus capas son operaciones convolucionales. Estas últimas son capaces de aprender pequeños patrones locales, que con la sucesión de las distintas capas, aprenden patrones mayores como composición de los más pequeños.

Las redes convolucionales suelen utilizarse para el análisis y la clasificación de imágenes. Por ello, se explicará su funcionamiento como si se trataran imágenes.

Las convoluciones operan sobre tensores 3D llamados mapas de características formados por los ejes de altura, anchura y profundidad (también conocido como eje de canales). Si se trata de una imagen RGB, la profundidad es 3, al tener 3 canales: rojo, verde y azul; y si es una imagen en blanco y negro, la profundidad es 1. La operación de convolución extrae fragmentos del mapa de entrada (el primer mapa serían los píxeles de la propia imagen) y aplica la misma transformación a todos los fragmentos, produciendo un mapa de características de salida. Este tiene las mismas dimensiones que el de entrada, pero la profundidad pasa a ser un parámetro de la

capa porque deja de representar los canales y pasa a representar filtros, los cuales codifican independientemente un aspecto específico de los datos de entrada.

El mecanismo de la convolución consiste en "deslizar una ventana" sobre el mapa de características de entrada y realizar una serie de operaciones para extraer fragmentos. Estos fragmentos se obtienen a través de un producto tensorial con una misma matriz denominada núcleo de convolución (kernel). Como se aplican varios kernel, su conjunto es lo que hemos denominado filtros. El kernel toma inicialmente valores aleatorios y mediante el algoritmo de propagación hacia atrás se van ajustando sus valores. Cada ubicación espacial en el mapa de características de entrada corresponde a la misma ubicación en el de salida.

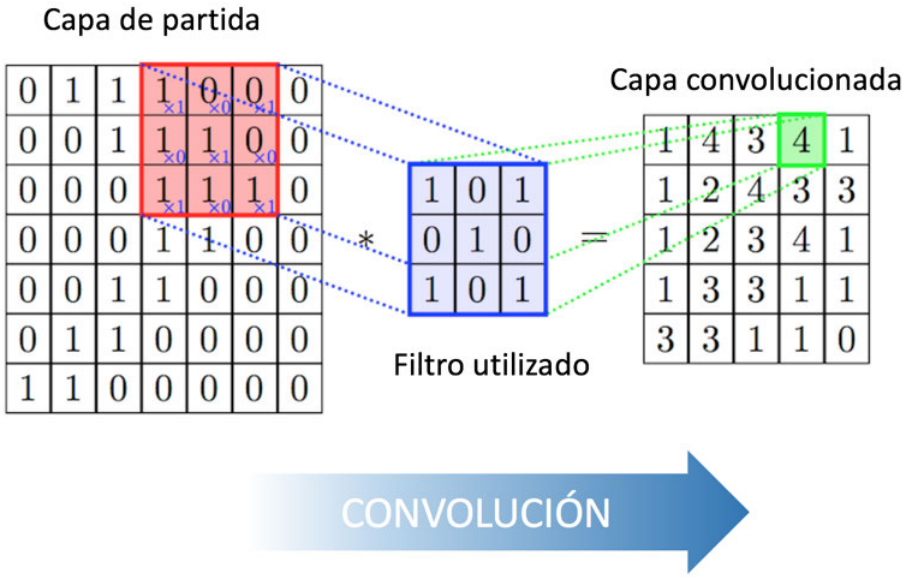

Figura 6: Operación de convolución en un espacio 2D.
Imagen obtenida de `http://www.diegocalvo.es/red-neuronal-convolucional/`

Otra operación muy importante en las redes convolucionales es el *max pooling*. Esta, consiste en extraer ventanas de los mapas de características de entrada y seleccionar el valor máximo, de tal manera que prevalezcan las características más importantes que detecta cada filtro. Es habitual que el max pooling simplifique el mapa de características a la mitad, lo cual reduce el número de parámetros a entrenar, y permite que las capas de convolución sucesivas analicen un espacio mayor, pudiendo identificar patrones más grandes.

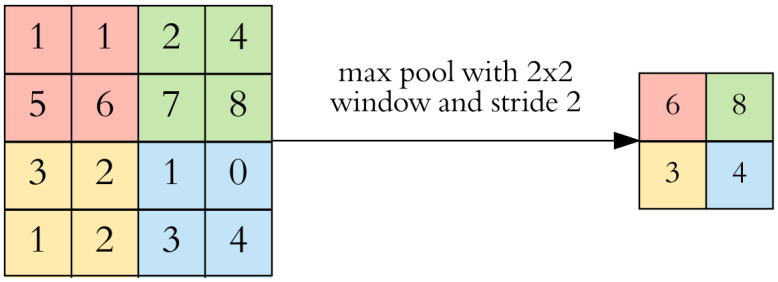

Figura 7: Funcionamiento del max pooling.
Imagen obtenida de `https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2`

De la misma manera que funcionan las convoluciones 2D para el tratamiento de imágenes, se pueden utilizar las convoluciones 1D para extraer fragmentos de secuencias. Por lo tanto, las capas convolucionales unidimensionales son capaces de reconocer patrones locales en una secuencia. La operación max pooling funciona igual también que en las convolucionales 2D, es decir, se extrae el valor máximo de una subsecuencia, y así se reduce la longitud de las entradas 1D.

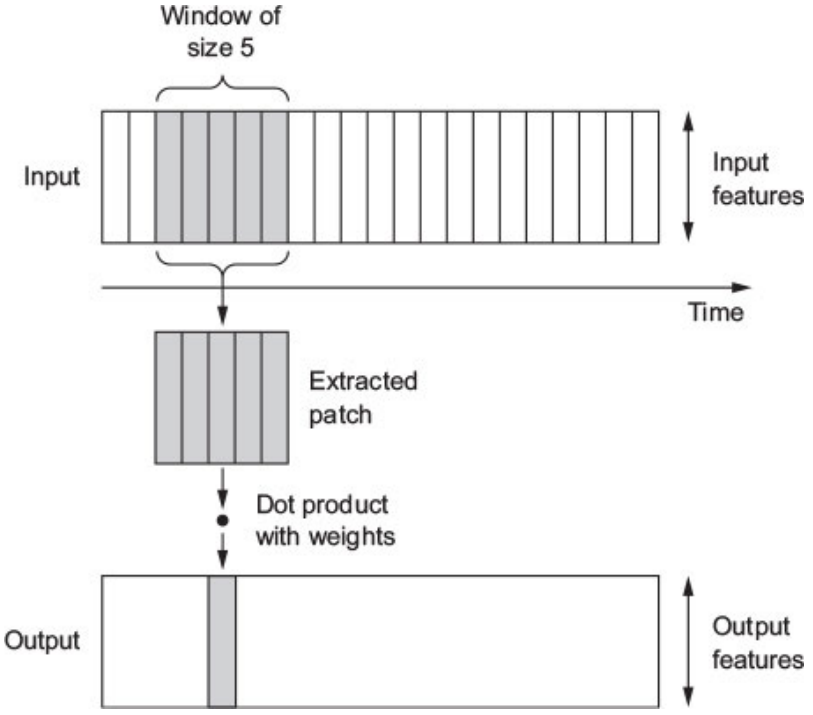

Figura 8: Funcionamiento de las convoluciones 1D.
Imagen obtenida de [Chollet y Allaire, 2018]

2.3. Fundamentos del machine learning

Todo problema de machine learning cuenta con un conjunto de variables, características o predictores $X = (X_1, X_2, ..., X_n)$ y unas respuestas Y de manera que se busca f / $f(X) = Y$.

Los algoritmos de machine learning se dividen en cuatro tipos de aprendizaje.

- Aprendizaje supervisado: Es el más común. Consiste en aprender a asignar datos de entrada a objetivos conocidos dado un conjunto de ejemplos. La gran mayoría de las aplicaciones de deep learning pertenecen a esta categoría.
- Aprendizaje no supervisado: Consiste en encontrar transformaciones de los datos de entrada sin objetivos conocidos, como en el supervisado, para compresión o eliminación de datos, o para comprender las correlaciones en los datos. Se suele utilizar en el análisis de datos y, a menudo, antes de abordar un problema supervisado para entender mejor los datos.
- Aprendizaje auto-supervisado: Es aprendizaje supervisado pero los objetivos, en vez de ser anotaciones dadas, los genera un algoritmo heurístico a partir de los datos de entrada.
- Aprendizaje por refuerzo: Un agente recibe información del entorno y aprende a elegir acciones que maximizarán alguna recompensa.

Para el aprendizaje de tipo supervisado existen varios conceptos a tener en cuenta:

- Muestra o entrada: Ejemplo determinado que se introduce en el modelo formado por un vector de los valores de las variables junto con la respuesta correspondiente que se denomina etiqueta.
- Predicción o salida: El resultado generado por el modelo.
- Objetivo: Aquello que debería predecir el modelo, es decir, la respuesta real.
- Error de predicción o valor pérdida: Medida de la distancia entre la predicción del modelo y el objetivo.
- Clases: Categorías definidas en un problema de clasificación.
- Anotaciones: Todos los objetivos de un conjunto de datos, normalmente establecidos por humanos (en el caso del aprendizaje auto-supervisado, vendrían dados por un algoritmo metaheurístico).
- Epoch: Cada iteración sobre el conjunto de entrenamiento.
- Clasificación binaria: Problema de clasificación en el que una entrada puede ser de dos clases. Habitualmente denotadas como 0 y 1.
- Clasificación multiclase: Problema de clasificación en el que una entrada puede ser de más de dos clases.
- Clasificación multietiqueta: Problema de clasificación en el que cada entrada puede pertenecer a varias clases.
- Codificación one-hot: Codificación de las clases en forma de array de manera que el valor "1" se asigne a la clase que corresponda en cada momento. Se

debe llevar a cabo si las clases no son de tipo numérico y, aunque lo sean, es aconsejable ya que el algoritmo de la red de neuronas será más eficiente. En la siguiente figura tenemos tres clases de colores: si la entrada pertenece a la clase "red", esta es la que tendría un 1 y las demás clases un 0. Y así sucedería con las demás clases.

```
red,    green,  blue
1,      0,      0
0,      1,      0
0,      0,      1
```

Figura 9: Ejemplo de codificación one-hot con tres clases.

- Regresión escalar: Tarea donde el objetivo es un valor escalar continuo.
- Regresión vectorial: Tarea en la que el objetivo es un conjunto de valores continuos.

2.3.1. Preprocesado de datos

Para poder introducir los datos en una red de neuronas hay que tratarlos previamente y darles un formato determinado. Así será mucho más sencillo para ella y obtendremos resultados acordes a lo esperado. Esto incluye la vectorización, la normalización, el tratamiento de valores vacíos y la extracción de características.

- Vectorización: Independientemente de los datos que se estén tratando, todas las entradas y objetivos en una red neuronal deben ser tensores de coma flotante y en algún caso tensores de enteros. Los tensores es la manera que tiene Tensorflow de manejar vectores, matrices y estructuras de datos de mayor dimensión.
- Normalización: Antes de introducir los datos en la red de neuronas, hay que normalizar cada característica independientemente, ya que no se recomienda introducir valores muy grandes o datos demasiado heterogéneos en los que cada característica se encuentre en un rango muy diferente. Si se hace, puede activar grandes actualizaciones de gradiente que evitarán que la red converja. Por este motivo, hay varias formas de normalizar un conjunto de datos. Las dos más comunes son: transformar los datos para que tengan media 0 y desviación típica 1 (ver ecuación 1) y forzar a que todos se encuentren en el rango [0,1] (ver ecuación 2).

$$X_{norm} = \frac{X - X_{mean}}{X_{stddev}} \tag{1}$$

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2}$$

- Tratamiento de valores vacíos: En general, se pueden sustituir los valores vacíos por 0 si este no tiene significado alguno. La red aprenderá durante el entrenamiento que 0 significa que falta el valor y lo ignorará. En el caso de que se espere que en los datos de test falten valores, pero en los de entrenamiento no, la red no aprenderá a ignorar los valores vacíos. Para ello, se recomienda generar ejemplos de entrenamiento con valores vacíos.
- Extracción de características: Se aconseja presentar los datos al modelo de tal manera que se facilite el trabajo a la red de neuronas. De esta forma, si se conoce el problema lo suficiente, podremos realizar transformaciones de los datos para que las características que se introducen en la red sean más representativas y sencillas. Aunque el deep learning realmente no necesita estas transformaciones, porque es capaz de extraerlas de datos en crudo por sí mismo, la selección de características tiene sus ventajas: resolver problemas de forma más elegante usando menos recursos innecesarios y además con menos datos, ya que la red solo puede aprender estas características si hay disponibles muchos datos de entrenamiento.

2.3.2. Evaluación de modelos

Una vez que se han tratado los datos iniciales para introducirlos en la red y se ha entrenado el modelo de la mejor manera posible, lo último es evaluar el modelo.

Para la evaluación siempre se procede de la misma forma: se divide el conjunto de datos en dos subconjuntos: entrenamiento y test. Del conjunto de entrenamiento se extrae a su vez otro subconjunto, el de validación. Por lo tanto, con los datos del conjunto de entrenamiento se entrena el modelo, con el de validación se evalúa y se va ajustando, y una vez que se cree que está listo, se reentrena el modelo con los datos de entrenamiento al completo y se hace la prueba definitiva con el conjunto de test.

La razón por la que no se dividen los datos únicamente en entrenamiento y test es que, al desarrollar un modelo, siempre hay que ajustar el número de capas y el tamaño de las mismas (lo que se conoce como hiperparámetros para diferenciarlos de los parámetros que serían los pesos de las capas). Para realizar este ajuste se necesita cierto feedback, que se consigue del rendimiento del modelo sobre el conjunto de validación. Este proceso siempre provoca cierto sobreajuste sobre el conjunto aunque no haya sido entrenado directamente con él. Si se ajustan los hiperparámetros y se repite el experimento demasiadas veces, llega un momento en el que se filtra demasiada información del conjunto de validación y éste deja de ser válido como método para evaluar el modelo. Es por todo ello, que se necesita un conjunto de test que nunca antes haya visto el modelo, incluso indirectamente, para poder medir la generalización y la calidad real del modelo.

Hay varias técnicas que organizan la validación de los modelos.

- Validación simple: Se separa un subconjunto para test, otro para entrenar y como no se debe ajustar el modelo con el conjunto de test, hay que reservar también datos para el conjunto de validación. Esta técnica presenta un inconveniente, pues si se dispone de un conjunto reducido de datos, es posible que

los conjuntos de validación y test no sean lo suficientemente representativos. Si esto ocurriese, es preferible implementar alguna de las dos siguientes técnicas.

- Validación K-fold (o validación cruzada): Con este enfoque, se dividen los datos en K particiones iguales. Para cada partición i, se entrena el modelo en las K-1 restantes y se evalúa con la partición i. El resultado final del rendimiento del modelo es el promedio de los K resultados obtenidos. Este método también admite utilizar un conjunto de validación como en la validación simple para ajustar el modelo previamente. En algunos casos, este tipo de validación se emplea como entrenamiento del modelo y se dispone de otro conjunto de test con datos nunca vistos para evaluar el modelo.

Figura 10: Ejemplo de validación cruzada con 10 particiones. Imagen obtenida de `http://karlrosaen.com/ml/learning-log/2016-06-20/`

- Validación K-fold iterada con mezcla: Esta técnica es para situaciones en las que dispongamos de muy pocos datos. Consiste en aplicar la validación K-fold repetidamente, pero mezclando los datos antes de cada separación en particiones. El resultado final del rendimiento del modelo es el promedio de los resultados obtenidos en cada ejecución de la validación K-fold. El mayor inconveniente es el coste de ejecución al entrenar y evaluar numerosos modelos.

Para elegir adecuadamente el tipo de validación de nuestro modelo hay que tener bien presentes las características de los datos. Si ya se ha elegido qué validación se va a usar, hay que tener cuidado con cómo formamos los diferentes conjuntos de datos. Esto es, asegurarse que los conjuntos de entrenamiento y de test son representativos por sí mismos y que no hay un mismo dato en ambos; deben ser conjuntos disjuntos.

2.4. Métricas y matriz de confusión

Una vez el modelo ha sido entrenado, solo queda evaluarlo con datos nuevos de test. Con ellos, el modelo elabora predicciones que, al compararlas con los objetivos generan la matriz de confusión. Esta matriz es útil para medir la efectividad del modelo mediante una serie de medidas: accuracy, recall, precisión y fscore.

Matriz de confusión

La matriz de confusión es una herramienta utilizada en el aprendizaje supervisado que permite una visualización rápida del desempeño de un modelo. Las filas de la matriz representan el número de predicciones de cada clase realizadas por el modelo, mientras que las columnas, los valores reales de cada clase. De esta manera, si nos fijamos en el valor de una celda, obtendremos el número de casos en los que se ha predicho una clase y la clase a la que realmente corresponden.

		Objetivo	
		Positivos	**Negativos**
Predicción	**Positivos**	True Positive	False Positive
	Negativos	False Negative	True Negative

Tabla 1: Apariencia de una matriz de confusión

- Verdaderos positivos (VP): resultados positivos que el modelo ha predicho correctamente como positivos.
- Verdaderos negativos (VN): resultados negativos que el modelo ha predicho correctamente como negativos.
- Falsos positivos (FP): resultados negativos que el modelo ha predicho erróneamente como positivos.
- Falsos negativos (FN): resultados positivos que el modelo ha predicho erróneamente como negativos.

Métricas de evaluación

A través de la matriz de confusión se puede evaluar la calidad del modelo con una serie de medidas.

- Accuracy (exactitud): Proporción de datos clasificados correctamente.

$$Accuracy = \frac{N\ de\ predicciones\ correctas}{N\ total\ de\ predicciones}$$

O de otra manera:

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

- Recall (sensibilidad): Representa la proporción de positivos predichos correctamente entre el total real de positivos.

$$Recall = \frac{VP}{VP + FN}$$

- Precisión: Representa la proporción de positivos predichos correctamente entre el total de positivos predichos.

$$Precision = \frac{VP}{VP + FP}$$

- Fscore: Es una medida de la exactitud del test que consiste en la media armónica del recall y la precisión.

$$Fscore = \ 2 \ * \ \frac{Recall * Precision}{Recall + Precision}$$

2.5. Overfitting y underfitting

La finalidad de todo algoritmo de machine learning reside en predecir correctamente respuestas ante datos totalmente nuevos. Si un modelo de machine learning empieza a funcionar peor con datos nuevos que con los datos de entrenamiento, se puede afirmar que el modelo sufre sobreajuste (en inglés, overfitting). Para tratar de evitarlo, hay que encontrar la justa medida entre optimización y generalización. La optimización es el proceso de ajuste del modelo para conseguir el mayor rendimiento posible de los datos de entrenamiento, mientras que la generalización hace referencia a cómo de bien se comporta el modelo con datos nunca vistos anteriormente. El objetivo es conseguir una buena generalización, pero el problema es que solo se puede ajustar el modelo a través de los datos de entrenamiento.

Al comenzar el entrenamiento, el valor pérdida en el conjunto de entrenamiento es igual de bajo que en el de test, lo que se conoce como underfitting, es decir, el modelo se encuentra por debajo de sus posibilidades y es capaz todavía de aprender más. Pero, en un determinado número de iteraciones, el modelo deja de generalizar y las medidas de validación se estancan e incluso empeoran. Esto significa que el modelo está empezando a tener sobreajuste y a aprender patrones específicos de los datos de entrenamiento, pero que resultan engañosos o irrelevantes con datos nuevos.

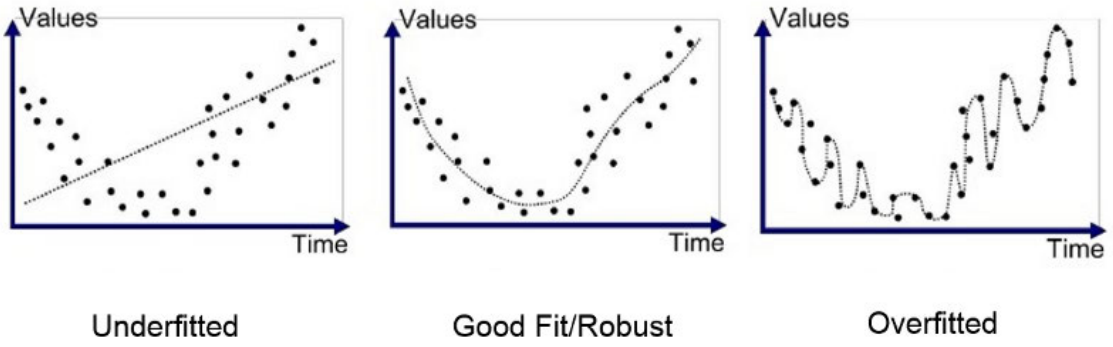

Figura 11: Representación de overfitting, underfitting y caso óptimo.
Imagen obtenida de `https://medium.com/datos-y-ciencia/machine-learning-c~omo-desarrollar-un-modelo-desde-cero-cc17654f0d48`

Para prevenir el sobreajuste, la mejor solución es adquirir más datos de entrenamiento porque generalizará más. No obstante, si esto no es posible, hay otras técnicas:

- Reducir la capacidad de la red: La forma más fácil de reducir el sobreajuste es reducir el tamaño del modelo, esto es, el número de capas y número de neuronas por capa, que determinan el número de parámetros que pueden ajustarse en el modelo. Un modelo con muchos parámetros tiene mayor capacidad de memorización con la que poder "aprenderse de memoria" cada entrada de datos y su objetivo. De esta manera, no clasificaría adecuadamente datos nuevos y no generalizaría nada. Con todo ello, si el modelo tiene una capacidad de memorización reducida, el proceso de optimización se centrará en los patrones más significativos, lo cual seguramente generalizará mejor.

- Añadir regularización a los pesos: Como se ha visto en la técnica anterior, un modelo más simple tiene menos probabilidades de sufrir sobreajuste. Por ello, otra manera de reducirlo es establecer restricciones a la complejidad de la red forzando que los pesos sean valores pequeños. Es lo que se llama regularización de los pesos y se realiza agregando a la función de pérdida un coste asociado a tener pesos grandes. Hay dos tipos, la regularización L1 en la que el coste añadido es proporcional al valor absoluto de los coeficientes de los pesos; y la regularización L2 en la que el coste es proporcional al cuadrado del valor de los coeficientes de los pesos.

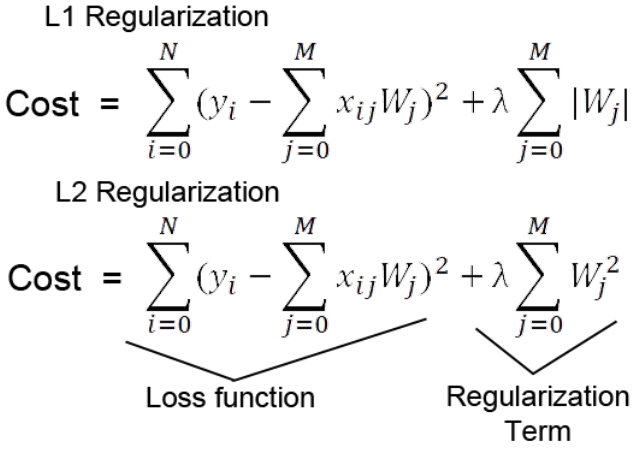

Figura 12: Expresiones matemáticas de la función pérdida con regularización L1 y L2. Imagen obtenida de `https://medium.com/datos-y-ciencia/machine-learning-c~omo-desarrollar-un-modelo-desde-cero-cc17654f0d48`

- Añadir dropout: Es una de las técnicas de regularización más efectiva. Consiste en eliminar aleatoriamente, dejando a 0, un cierto número de salidas en una capa durante el entrenamiento. Provoca el mismo efecto que reducir la capacidad de la red. Al ejecutar el test, no se descartan unidades, sino que los valores de salida de la capa se reducen en un factor igual al valor de dropout. Así se compensa que haya más unidades activas que en el entrenamiento. La mayoría de las veces estas dos operaciones se implementan durante el entrenamiento, dejando el test sin cambios.

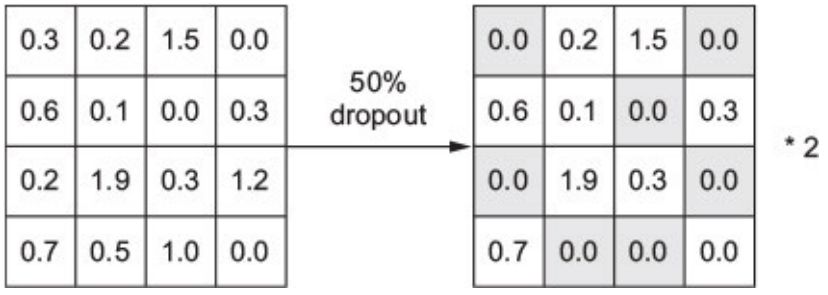

Figura 13: Ejemplo de dropout aplicado únicamente en el entrenamiento. Imagen obtenida de [Chollet y Allaire, 2018]

2.6. Tensorflow y Keras

Dentro del mundo del machine learning existen numerosas herramientas de trabajo. De entre todas, destacan dos principalmente: Tensorflow y Keras.

Tensorflow es una librería de código abierto para aprendizaje automático desarrollada por Google que utiliza como forma de programación grafos de flujo de datos. Los grafos están formados por operaciones matemáticas representadas en los nodos, y las salidas y entradas de estos que son vectores multidimensionales de datos: los tensores. Mediante una serie de tareas es capaz de construir y entrenar redes neuronales que detecten patrones y correlaciones análogos al aprendizaje y razonamiento de los humanos.

La arquitectura flexible de TensorFlow le permite realizar sus cálculos en múltiples CPUs y GPUs.

Además, proporciona APIs de Python, C++, Java, Go, Rust y Haskell. También hay bibliotecas de terceros como es el caso de C#, R, Scala, Julia y OCaml.

Por otra parte, Keras es una librería de redes neuronales de código abierto escrita en Python y creada por el ingeniero de Google François Chollet. Puede ejecutarse sobre Tensorflow, Microsoft Cognitive Toolkit o Theano.

El objetivo de Keras es actuar como interfaz, en lugar de framework de machine learning. Ofrece una mayor abstracción, haciendo más sencillo el desarrollo de modelos de deep learning independientemente del backend computacional utilizado.

Da soporte a las redes neuronales estándar (con capas densas), a las redes convolucionales y a las recurrentes. La principal ventaja es que ya tiene implementado bloques para gestionar a un alto nivel las capas, funciones objetivo, funciones de activación, optimizadores matemáticos, además de otras funcionalidades como el dropout, la normalización y el pooling.

Con todo ello, Tensorflow y Keras son una combinación perfecta para desarrollar modelos de deep learning por su sencillez de uso, potencia y eficacia. De hecho, en 2017 el equipo de Tensorflow comenzó a dar soporte a Keras.

3. Datos

Los datos empleados en este PFG se han obtenido del proyecto mPower desarrollado por Sage Bionetworks [*mPower: Mobile Parkinson Disease Study* 2019]. Este consiste en una aplicación móvil que recoge información relevante a través de los sensores del dispositivo y cuestionarios, para intentar comprender mejor la enfermedad del Parkinson mediante la búsqueda de patrones.

Cualquier persona enferma o no, mayor de edad, residente en Estados Unidos y con acceso a un móvil, podrá participar en el estudio. Una vez registrados, realizarán una encuesta demográfica y de manera mensual deberán cumplimentar un cuestionario de evaluación de la EP (PDQ-8) y una serie de preguntas para evaluar la UPDRS, pero centrándose en los síntomas motores (MDS-UPDRS). En cuanto a actividades, se proponen cuatro tareas diferentes que se deben realizar tres veces al día denominadas: Memory, Tapping, Voice y Walking (ver Figura 14). A los participantes que están diagnosticados como enfermos de Parkinson se les pide que las realicen justo antes y después de tomar la medicación y en cualquier otro momento. En el caso de los participantes de control, pueden realizar las actividades en cualquier momento del día.

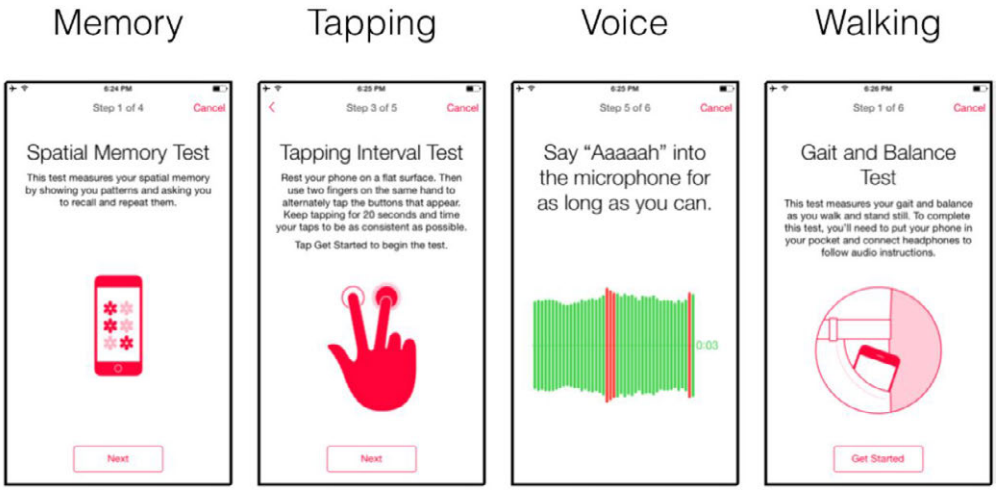

Figura 14: Las actividades propuestas por la aplicación de mPower.
Imagen obtenida de `https://www.synapse.org/#!Synapse:syn4993293/wiki/247861`

Los creadores de mPower sostienen que sus datos deben ser compartidos inmediatamente, incluso antes de completar su propio análisis. De esta manera, se podrían conseguir soluciones más rápidas y óptimas que mejoren la vida de las personas que padecen esta dolencia. Desde su punto de vista, aquellos que reutilizan datos de otras investigaciones son *data scientists* y no parásitos de la investigación como defienden otros investigadores. [J.Wilbanks y Friend, 2016]

Para poder compartir esta información tan sensible, han desarrollado un proceso electrónico de consentimiento informado en el que se incluyen preferencias sobre el intercambio de datos. Así, los participantes pueden decidir si comparten o no sus datos codificados con investigadores cualificados de todo el mundo. Además, en cualquier momento del proceso pueden cambiar sus preferencias.

Para llevar a cabo la difusión de los datos se apoyan en la plataforma Synapse (`https://www.synapse.org/`), que se encarga de almacenarlos, organizarlos y distribuirlos a los investigadores para que de manera colaborativa se analicen. Así mismo, Synapse cuenta con distintos niveles de usuarios que ofrecen diferentes permisos y responsabilidades: *Anonymous*, *Registered*, *Certified* y *Validated*.

	Anonymous	**Registered**	**Certified**	**Validated**
Browse Public Project Catalog	X	X	X	X
Browse Public file Catalog	X	X	X	X
Create a Project		X	X	X
Add Wiki Content		X	X	X
Download Files/Tables (depends on fulfilling the Conditions for Use, if any, of the File or Table)		X	X	X
Upload Files/Tables			X	X
Add Provenance			X	X
Access to Bridge Data				X

Tabla 2: Tipos de usuario en Synapse. Basada en `https://docs.synapse.org/articles/accounts_certified_users_and_profile_validation.html`

Los datos del proyecto mPower proceden de estudios de investigación realizados a través de una aplicación móvil autoguiada, por lo que se les considera *Bridge Data*. En Synapse, los únicos usuarios con acceso a este tipo de datos son los validados. Para llegar a serlo hay que:

- Demostrar que se conocen y se entienden los términos y condiciones de uso de Synapse, en cuanto a intercambio de datos y ética a aplicar, mediante un breve examen de 15 preguntas.
- Validar la identidad a través de una carta académica de un superior, una carta notarial o una copia de una licencia profesional.
- Hacer una declaración pública del uso previsto de los datos.
- Aceptar explícitamente un juramento por el que uno se adhiere a un código de comportamiento y se compromete a cumplir con las condiciones de uso específicas de los datos.

Tras realizar todo este proceso, ya consideran que se es un investigador cualificado y por lo tanto mPower cumpliría su compromiso de intercambio de datos.

Los datos que interesan en nuestro caso son los relativos a la actividad de la voz. Estos consisten en ficheros de audio (.wav) que recogen grabaciones de los participantes manteniendo la vocal 'a' con un volumen estable durante 10 segundos [Bot et al., 2016].

En un principio, contamos con 64724 grabaciones de 5733 participantes diferentes: 1091 participantes diagnosticados con Parkinson (37382 grabaciones) y 4758 de control (27342 grabaciones).

Cabe destacar que estos datos contienen incongruencias, pues encontramos que un mismo paciente tiene grabaciones identificado como sano y a su vez otras como diagnosticado. Es por ello que la suma de enfermos y sanos no corresponde con el

total real. Al advertir este fallo con la investigación bastante avanzada, únicamente se tomaron medidas en los datos de audio en crudo (AudioCrudo.csv). En los experimentos donde se utilizan otros datos, se asume el error. El procedimiento para corregirlo fue primero determinar cuáles eran los pacientes afectados y revisar cuántas grabaciones tenían como sanos y cuántas como enfermos. Si había una diferencia clara, las grabaciones consideradas de la clase minoritaria se tomaron como de la clase con mayor número de grabaciones. En caso de duda, se descartaron todas las grabaciones del paciente en cuestión.

Antes de comenzar la investigación se realizó un filtrado inicial por el que únicamente se seleccionaron las grabaciones que se realizaron justo antes de tomar la medicación y las de los que no toman medicación (participantes de control). El motivo es que, para el caso de los enfermos, justo antes de tomar la medicación puede ser el momento en el que los síntomas se muestren con mayor claridad, por lo tanto es el caso más extremo. Esta idea viene apoyada por [Pinho et al., 2018], quienes defienden que la levodopa, medicación para la EP, mejora características sonoras de la voz como la frecuencia fundamental (F0) y el jitter.

Este filtrado también vino determinado por los participantes que, a su vez, cuentan con datos demográficos necesarios para la investigación (edad y género). De esta manera, el conjunto de datos se nos reduce considerablemente, quedándonos con 24160 grabaciones de 1618 participantes diferentes: 425 participantes diagnosticados con Parkinson (8267 grabaciones) y 1245 de control (15893 grabaciones).

	Nº Participantes	**Nº Grabaciones**
Enfermos	425	8267
Control	1245	15893
Totales	**1670***	**24160**

Tabla 3: Características de los datos de CaractAudioDemog.csv

A partir de este conjunto de grabaciones se crearon dos bases de datos. La primera incluye 62 características de sonido para cada grabación, las cuales se utilizaron posteriormente para entrenar las redes de neuronas con las que se realiza el diagnóstico que determina si el paciente tiene o no la EP. Estas características fueron extraídas a través del software de código abierto OpenSmile en el marco del proyecto: *Modelos de minería de datos para el diagnóstico precoz de enfermedades neurodegenerativas*, financiado por el programa PROINCE de CyTMa2 de la Universidad Nacional de La Matanza (Buenos Aires), en el que han participado, por parte de la UPM, Francisco Díaz y Alfonsa García. Estas características, junto a las variables edad y sexo, se encuentran en el fichero CaractAudioDemog.csv y se pueden clasificar en 5 grupos:

- 10 que informan sobre la frecuencia fundamental F0.
- 10 relacionadas con el volumen.
- 6 que representan las medias y desviaciones típicas del Jitter local, el Shimmer local y la tasa HNR (harmonic to noise rate).
- 18 relacionadas con las frecuencias F1, F2, F3.
- 18 consideradas como “otras”.

Todas las características se pueden encontrar en los anexos.

La segunda base de datos consiste simplemente en las grabaciones de voz en crudo. Para poder introducir los audios en la red de neuronas, hubo que manipularlos para darles el formato adecuado. Esto se llevó a cabo por medio de un programa en Python (ver anexos), el cual a partir del segundo 2.5 toma un cuarto de segundo del que se extraen 10000 entradas consecutivas, que representan un muestreo de la onda sonora en ese intervalo. Estos nuevos datos se recogen en el fichero AudioCrudo.csv.

	Nº Participantes	**Nº Grabaciones**
Enfermos	400	8120
Control	1192	15253
Totales	**1592**	**23373**

Tabla 4: Características de los datos de AudioCrudo.csv

4. Metodología

4.1. Metodología del proyecto

Este PFG es el resultado de una beca de colaboración en un proyecto de investigación con mis tutores. En su desarrollo ha sido posible aplicar ligeramente algunos principios ágiles, como es el caso de las reuniones semanales en las que se realizaba una retrospectiva de la semana: avances, problemas y futuras tareas. También se ha contado con diferentes fases que podrían considerarse *sprints*. Estas han estado presentes durante todo el proceso hasta la finalización del proyecto, ya que se ha intentado aplicar una continua mejora de las mismas. Todos estos ápices ágiles realmente no conforman ninguna metodología ágil específica. En la siguiente figura se puede comprobar la duración de las diferentes fases.

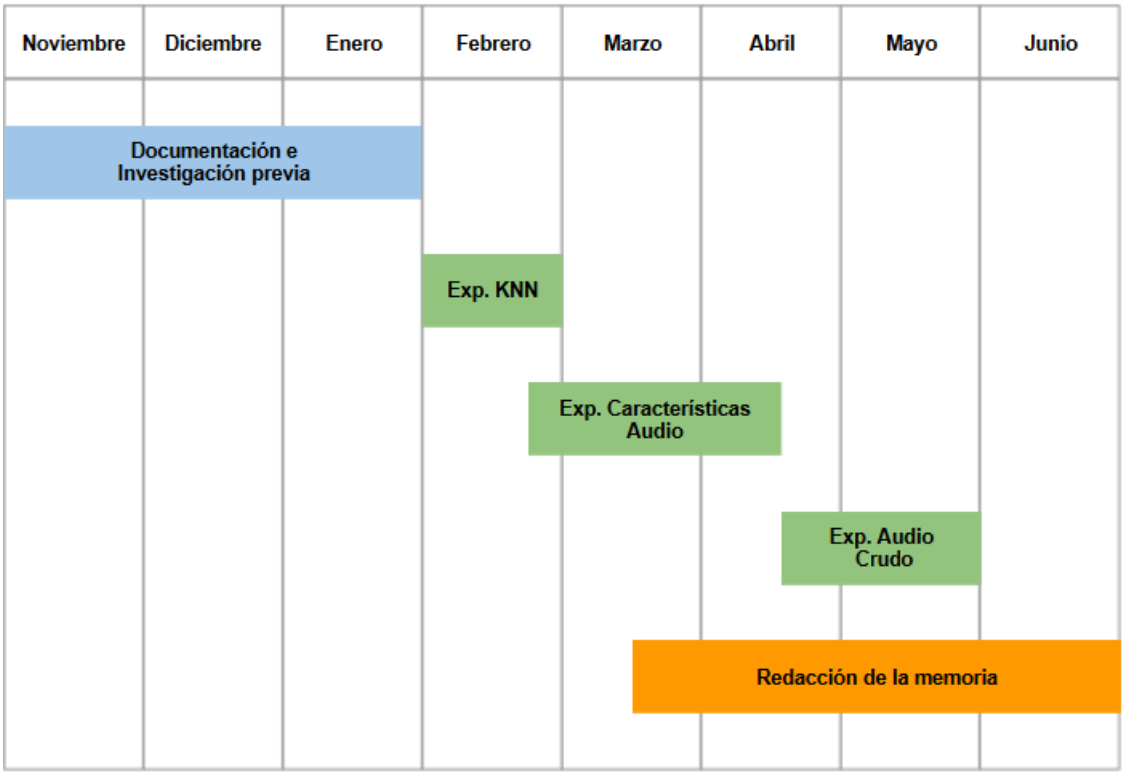

Figura 15: Cronograma del proyecto.

En nuestro caso no existe un cliente definido, sino que es la propia línea de investigación la que va determinando los pasos a seguir. Es por ello, que los cambios son constantes y son consecuencia de aplicar el método científico.

El método científico consiste en un conjunto de pasos ordenados que se emplean para adquirir nuevos conocimientos. Es la metodología predominante en las investigaciones y busca minimizar la posible subjetividad del científico. Sus pasos principales son los siguientes:

- **Observación:** Consiste en la recopilación de hechos acerca del problema.
- **Hipótesis y predicción:** Se formula una hipótesis a partir de los datos obtenidos en la observación.
- **Experimento:** Se realiza el experimento oportuno para probar la hipótesis.
- **Análisis y conclusión:** Se analizan los resultados obtenidos y se decide si se acepta o rechaza la hipótesis planteada. En caso de rechazarse, se comenzaría el ciclo de nuevo hasta llegar a aceptar una de las hipótesis.

Figura 16: Pasos del método científico. Imagen obtenida de `https://medium.com/@reyesrichie/como-aplicar-el-metodo-cientifico-a-crear-una-empresa-1f405e20be06`

En nuestro proyecto, el método científico se aplica con el objetivo de originar nuevas ideas y por lo tanto nuevos experimentos con los que se obtengan mejores resultados.

4.2. Metodología de los experimentos

En este proyecto se han realizado diversos experimentos que podemos agrupar en las 3 fases que se muestran en el cronograma anterior de la Figura 15: KNN con características de audio, redes de neuronas con características de audio y redes de neuronas con audios en crudo.

En todas ellas, ha habido que seleccionar un conjunto de entrenamiento y otro de test con una proporción aproximada de 80 % y 20 %, respectivamente, de los registros disponibles. Una aportación de este proyecto respecto a trabajos anteriores es que hemos comparado distintas estrategias a la hora de seleccionar el conjunto de test. Dado que, como suele ser habitual en este tipo de bases de datos, cada individuo tiene varias grabaciones, una selección aleatoria del test implica con una alta probabilidad que haya grabaciones de una misma persona en el conjunto de entrenamiento y en el de test. Esto provoca que los resultados del test estén distorsionados, sean demasiado optimistas y no sean completamente fiables y verosímiles. Por este motivo, se han definido experimentos seleccionando los conjuntos de entrenamiento y de test de dos maneras distintas: de forma aleatoria y tomando registros consecutivos para asegurarnos de que no haya grabaciones de un mismo individuo en ambos conjuntos.

Para evitar todo esto, también se ha hecho algún experimento tomando una sola grabación por paciente. Aunque se disminuyen considerablemente los datos disponibles, se evita el problema de la selección de conjuntos.

Igualmente, se ha introducido en algún experimento la variable *edad* para realizar una comparación con aquellos que no cuentan con ella y ver así su influencia.

Como se ha mostrado en la sección *Datos*, tras el filtrado inicial contamos con unos datos muy desequilibrados, pues hay bastantes más grabaciones de personas sanas que de enfermas. Es por ello, que en algunos experimentos se pretende equilibrar el problema tomando todas las grabaciones de los enfermos y el mismo número de grabaciones de sanos escogidas de manera aleatoria.

KNN

En el caso de los experimentos con KNN (ver [Tan, Steinbach y Kumar, 2006]), se han hecho varias selecciones de características teniendo en cuenta la correlación entre ellas, para poder eliminar aquellas características redundantes que podrían provocar ruido, y los resultados de [Giuliano et al., 2019]. En cuanto al número de vecinos a evaluar, será reducido, pues si tratamos con el problema desequilibrado, al haber más registros de participantes sanos, siempre predecirá como sano. Los resultados se han evaluado mediante la tasa de éxito y las matrices de confusión.

Redes de neuronas

En todos los experimentos con redes de neuronas se ha contado con un conjunto de validación para ajustar los parámetros del modelo sin que tenga que intervenir el conjunto test. Con él, ha sido posible modificar el número de epochs de entrenamiento de 5 en 5 hasta alcanzar el óptimo, el número de capas y sus neuronas siempre en potencias de 2, la regularización: se ha probado tanto regularización L2 como dropout y en el caso de las redes convolucionales, el tamaño del kernel.

También se han ejecutado experimentos en los que se diferencia a hombres y mujeres, pues el grosor y longitud de las cuerdas vocales es distinto y hace que la frecuencia fundamental de la voz masculina sea menor que la de la femenina [Puts, Doll e Hill, 2014]. Por ello, al tratar todas las grabaciones a la vez puede que estemos confundiendo al modelo y sea interesante esta división por género.

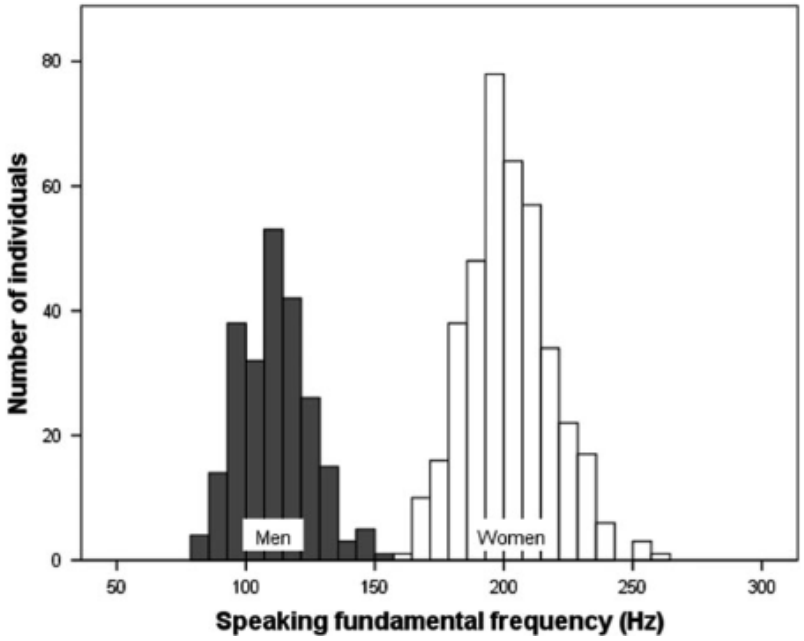

Figura 17: Frecuencias fundamentales en hombres y mujeres. Imagen obtenida de `https://pdfs.semanticscholar.org/d3f6/82f904882591488fa2bd82b7de35ca4e4b1d.pdf`

Los experimentos se han implementado en lenguaje R en el entorno de desarrollo RStudio y se han ejecutado sobre la GPU que es mucho más eficiente y rápido que cualquier CPU. Para la programación de las redes de neuronas se ha utilizado la librería Keras y para KNN, la librería *class*.

Se han agrupado en un fichero batch para automatizar su ejecución y cada fichero R, con su experimento correspondiente, exporta sus resultados a un fichero txt para poder consultarlos en cualquier momento.

Los resultados que se ofrecen corresponden al mejor modelo obtenido tras 50 ejecuciones, en el caso de los experimentos con características de sonido, y 20 ejecuciones en los de audio en crudo. Estos resultados se han evaluado a través de la tasa de éxito (accuracy), recall, precisión, fscore y la matriz de confusión. A su vez, a modo de curiosidad, se muestra el tiempo que ha tardado en entrenar el mejor modelo. El fscore es la medida más relevante en nuestro caso, pues al tratarse de un diagnóstico, interesan tanto el recall como la precisión. Además, es con la que se comparan los modelos y se decide cuál es el mejor.

Como se ha mencionado anteriormente, se han utilizado redes de neuronas con características de audio y con audios en crudo. Para las primeras se han empleado capas densas, las cuales se encuentran totalmente conectadas entre ellas y son las más comunes. Para los audios en crudo, se han empleado capas convolucionales conectadas al final con capas densas. Aunque las redes convolucionales suelen estar destinadas a la visión artificial, como estas aprenden pequeños patrones locales, pareció buena idea aplicarlas a ficheros de voz en los que se podrían dar estos patrones en la dimensión temporal. La única diferencia con respecto a las imágenes es que en vez de analizar un espacio 2D, en nuestro caso es 1D.

Todos los experimentos se han ejecutado en un equipo con una memoria RAM de 8GB la cual es suficiente para llevar a cabo los experimentos con características de sonido, pero no para los de audios en crudo pues hay que tratar varios gigas de datos. El problema no es la ejecución de los modelos sino el preprocesado de datos. Para ello, mis tutores lo realizaron en un equipo con 16 GB de RAM, y a través de un fichero RData exportaron los datos sobre los que ya pude ejecutar los experimentos al completo.

5. Experimentos y análisis de resultados

En esta sección se pretende mostrar la línea de investigación seguida en el proyecto, los experimentos realizados y sus resultados. Por último, se hará una comparativa de los resultados obtenidos con trabajos de investigadores destacados en este ámbito.

5.1. KNN

Aunque el objetivo principal de este trabajo es tratar con redes de neuronas, también se ha querido estudiar cómo se comportan estos datos de voz con algoritmos tradicionales de data mining como KNN.

Se ha probado con $k = 5$ y $k = 10$ y tres posibles selecciones de características:

Sel0: 14 características de sonido y la edad. Estas son: F1, F6, F9, F15, F21, F24, F26, F29, F33, F35, F39, F45, F51 y F57.
Sel1: 14 características de sonido. Las mismas que en Sel0.
Sel2: 5 características de sonido que son: F1, F15, F21, F24, F29.

1. Datos no equilibrados. CaractAudioDemog.csv

Contamos con 24160 grabaciones de 1618 pacientes del fichero CaractAudioDemog.csv. Cada registro consiste en 62 características de voz y la edad y el sexo del paciente como datos demográficos.

Para validar el modelo se ha usado un conjunto de test de 5000 registros, en un caso seleccionado de manera aleatoria, y en otro seleccionando registros consecutivos. También se ha probado validación cruzada 10-fold. Los resultados correspondientes a la tasa de éxito han sido los siguientes:

		$k = 5$	$k = 10$
TestAlea	Sel0	88.5 %	87.1 %
	Sel1	77.7 %	75.9 %
	Sel2	69.3 %	70 %
TestCons	Sel0	72.7 %	73.3 %
	Sel1	62.7 %	62.2 %
	Sel2	62.5 %	63.2 %
ValCruzada	Sel0	87.5 %	86.8 %
	Sel1	76.3 %	78.3 %
	Sel2	70 %	70 %

Tabla 5: Resultados KNN con datos no equilibrados de CaractAudioDemog.csv.

Como puede verse en la tabla anterior, los resultados obtenidos en los experimentos que cuentan con el conjunto de test elegido con registros consecutivos, son bastante inferiores a los de los otros dos tipos de experimentos (test escogido de forma aleatoria y validación cruzada como método de evaluación). Con ello, se demuestra que una selección cuidadosa de los conjuntos, proporciona resultados más acordes al rendimiento real del modelo en el caso de introducir datos nuevos.

Además, en las matrices de confusión de gran parte de los experimentos, se puede apreciar que la mayoría de los registros se clasifican como sanos, y en algunos de ellos, entre los clasificados como enfermos hay más errores que aciertos.

Por otra parte, aunque los resultados son mejores si se usa la variable edad, no parece adecuado hacerlo, porque la distribución de edades en el conjunto de enfermos es bastante diferente a la del conjunto de sanos.

2. Datos equilibrados. CaractAudioDemog.csv

Partiendo del mismo fichero de datos, se separan los 8267 registros correspondientes a enfermos y del resto se seleccionan aleatoriamente otros 8267 correspondientes a sanos. En ambos conjuntos se apartan 1000 registros consecutivos para el conjunto de test. Esto garantiza el mismo número de registros de sanos y enfermos, pero no de personas. Por eso se comprueba la distribución de personas en los conjuntos de entrenamiento y test.

Personas	Train	Test
Sanos	971	162
Enfermos	343	82

Tabla 6: Distribución de personas en KNN con datos equilibrados de CaractAudioDemog.csv.

Los resultados de la tasa de éxito son los siguientes:

	$k = 5$	$k = 10$
Sel0	66.3 %	65.5 %
Sel1	53.5 %	54 %
Sel2	55 %	55 %

Tabla 7: Resultados KNN con datos equilibrados de CaractAudioDemog.csv.

En este caso se supera discretamente el azar. Estos malos resultados se reflejan en las matrices de confusión, en las que se observa que o se clasifican como sanos la mayoría de los registros, o en algún caso, bastantes sanos se clasifican como enfermos. Esto nos indica que el algoritmo no es capaz de discernir correctamente entre sanos y enfermos, y por lo tanto, no por equilibrar el problema se obtienen mejores resultados.

También se han realizado experimentos similares utilizando validación cruzada y seleccionando el conjunto de test de forma aleatoria, y aunque mejoran los resultados, no ofrecen ninguna novedad respecto a los experimentos ejecutados en el apartado anterior.

3. Datos no equilibrados. Sólo un registro por paciente. Datos iniciales

Para estos experimentos, se han empleado los datos iniciales que consisten en 64724 registros con características de sonido y sin datos demográficos. Estos registros corresponden a 5733 personas, 4688 sanos y 1045 enfermos. Seleccionamos un sólo registro por persona, la primera grabación, y separamos 500 elegidos aleatoriamente

para el conjunto de test. Hacemos dos posibles selecciones de características, atendiendo a la matriz de correlaciones, una con 5 (Sel1) y otra con 10 (Sel2), siendo estas únicamente características de sonido.

Sel1: F1, F11, F21, F24, F35.
Sel2: F1, F11, F21, F24, F26, F29, F33, F35, F51, F57.

Los resultados obtenidos de la tasa de éxito son:

	$k = 5$	$k = 10$
Sel1	71.8 %	76.8 %
Sel2	76.4 %	81.2 %

Tabla 8: Resultados KNN de un registro por paciente con datos iniciales.

A pesar de seleccionar solo un registro por paciente para evitar: grabaciones de una misma persona en los conjuntos de entrenamiento y de test, y que el desajuste de datos (pacientes con registros sanos y enfermos) distorsione el modelo, se sigue teniendo el mismo problema, esto es, en las matrices de confusión se refleja que la gran mayoría de los registros se clasifican como sanos y los que hay como enfermos son más errores que aciertos.

5.2. Redes de neuronas. Características de sonido

Una vez vistos los resultados obtenidos con el algoritmo KNN, nos centraremos en las redes de neuronas y en el impacto que provocan en los resultados, tanto las 62 características de sonido de las grabaciones como los datos demográficos de los pacientes del fichero CaractAudioDemog.csv. Durante el proyecto se han realizado bastantes más experimentos con diferentes configuraciones de la red, pero solo se muestran los más representativos.

Cabe destacar que los resultados que se muestran son fruto de un modelo de red neuronal formada por 7 capas densas con un dropout de 0.3 y 10 epochs de entrenamiento.

1. Datos no equilibrados. Selección aleatoria del conjunto test

En los primeros experimentos realizados, se probó únicamente con las características de sonido y añadiendo como predictor la edad de los pacientes. Para validar el modelo se han seleccionado 5000 registros para el conjunto de test de manera aleatoria.

	Accuracy (Train/Test)	Recall	Precision	Fscore	T_{ejec}
Caract.	79.83 % / 78.78 %	64.5 %	71.07 %	67.62	26.25s
Caract. + Edad	85.74 % / 85.78 %	81.5 %	76.9 %	79.12	31.29s

Tabla 9: Resultados de características con edad. Selección aleatoria del conjunto de test.

Más adelante, dividimos hombres y mujeres para evaluarlos por separado como ya hicieron [Tsanas, Little, McSharry y Ramig, 2010]. La selección de los conjuntos sigue siendo aleatoria y se toman 4000 registros para el conjunto de test en el caso

de los hombres y 2000 para las mujeres. La distribución del número de grabaciones y personas distinguiendo hombres y mujeres es la siguiente:

	HOMBRES		**MUJERES**	
	Nº Participantes	**Nº Grabaciones**	**Nº Participantes**	**Nº Grabaciones**
Enfermos	244	4571	181	3696
Control	926	11911	319	3982
Totales	**1139***	**16482**	**479***	**7678**

Tabla 10: Distribución de hombres, mujeres y grabaciones en CaractAudioDemog.csv.

En el caso de las mujeres, se puede comprobar que el número de grabaciones de diagnosticados y sanos es bastante similar y por lo tanto es prácticamente un problema equilibrado. A su vez, como ya se mencionó en la sección *Datos*, hay un desajuste en el número de participantes, porque hay personas que tienen registros como sanos y como diagnosticados. Por eso, la suma de enfermos y de control es superior a la real.

En la siguiente tabla se reflejan los resultados de los experimentos divididos por género y en algunos incluyendo la edad.

	Accuracy (Train/Test)	Recall	Precision	Fscore	T_{ejec}
Caract. Hombres	82.91 % / 80.35 %	57.14 %	71 %	63.54	17.86s
Caract. Mujeres	80.17 % / 77.75 %	81.2 %	73.48 %	77.14	12.61s
Caract. Hombres + Edad	85.8 % / 84.6 %	66.52 %	77.17 %	71.45	27.08s
Caract. Mujeres + Edad	89.7 % / 87.7 %	87.88 %	87 %	87.43	18.82s

Tabla 11: Resultados de características con edad separados por género. Selección aleatoria del conjunto de test.

Como se puede observar, en los experimentos en los que interviene la edad se obtienen resultados bastantes mejores, pero como se comentó en los resultados de KNN, no deberíamos tenerlos en cuenta ya que la distribución de edades en el conjunto de enfermos es muy diferente a la del conjunto de sanos.

En el caso de los experimentos en los que evaluamos exclusivamente a los hombres, en las matrices de confusión se observa que bastantes enfermos se clasifican como sanos, de ahí que el valor de recall sea bastante menor. En cuanto a las mujeres, los resultados son muy buenos al ser un problema prácticamente equilibrado y esto se refleja en las matrices de confusión. Aunque para [Tsanas, Little, McSharry y Ramig, 2010] esta separación por géneros resulte beneficiosa, en nuestro caso resulta descompensado, pues las futuras grabaciones de las mujeres se diagnosticarían con mayor fiabilidad que las de los hombres. Es por ello, que es preferible escoger el modelo del experimento que los evalúa conjuntamente.

2. Datos no equilibrados. Selección consecutiva del conjunto test

En estos experimentos, se trató de seleccionar el conjunto de test con registros consecutivos de manera que no hubiese grabaciones de una misma persona en los conjuntos de entrenamiento y de test. La selección se realizó asegurándose mediante histogramas, de que el número de enfermos y sanos era similar en ambos conjuntos. Se tomaron 4000 registros para el conjunto de test de los hombres y 2000 para el test de las mujeres. Los resultados obtenidos son los siguientes:

	Accuracy (Train/Test)	Recall	Precision	Fscore	T_{ejec}
Caract. Hombres	83.96 % / 68.45 %	33.91 %	40.46 %	36.9	16.93s
Caract. Mujeres	81.75 % / 67.2 %	80.74 %	63.45 %	71.05	7.33s

Tabla 12: Resultados de características separados por género.
Selección consecutiva del conjunto de test.

Los resultados son peores que seleccionando el test de forma aleatoria. Esto ocurre especialmente en el experimento de los hombres, en el que sus resultados son pésimos, puesto que la mayoría de los enfermos son clasificados como sanos y viceversa. En el caso de las mujeres, los resultados son mejores, aunque sigue habiendo muchos sanos que se clasifican como enfermos.

5.3. Redes de neuronas. Audio en crudo

En los dos apartados anteriores se han utilizado las características de sonido como predictores de los modelos, al igual que todos los trabajos de las referencias. La novedad que aporta este proyecto es que también se utiliza la voz directamente para introducirla en las redes de neuronas. Hay dos tipos de experimentos: el primer experimento trata simplemente de introducir en la red de neuronas las 10000 entradas que se extrajeron directamente de las grabaciones; y el segundo consiste en aplicar a los datos empleados en el experimento anterior la transformada rápida de Fourier (FFT). Realmente el último experimento no sería necesario pues la FFT se basa en operaciones lineales, las cuales son capaces de realizar las redes de neuronas. Aun así, se creyó conveniente llevar a cabo el experimento. Los datos utilizados en todos los experimentos son los pertenecientes al fichero AudioCrudo.csv. Todos los experimentos han sido ejecutados con un mismo modelo: 8 capas convolucionales intercaladas con una capa *max_pooling* junto con 4 capas densas al final que tienen dropout de 0.4 y regularización L2 de 0.01.

1. Datos equilibrados. Selección aleatoria del conjunto test

Se toman los 8120 registros de enfermos y otros 8120 correspondientes a sanos elegidos aleatoriamente. En ambos conjuntos se seleccionan también de forma aleatoria 1000 registros para el conjunto de test. Así nos podemos asegurar que a lo sumo haya un mismo paciente en ambos conjuntos y que el problema sea equilibrado en cuanto a número de registros, pero no en personas. Por eso, se comprueba la distribución de personas en los conjuntos de entrenamiento y de test.

Personas	Train	Test
Sanos	1073	436
Enfermos	394	238

Tabla 13: Distribución de personas audio en crudo y FFT.
Selección aleatoria del conjunto de test.

Los resultados obtenidos son:

	Accuracy (Train/Test)	Recall	Precision	Fscore	T_{ejec}
Audio	64 % / 65.2 %	75.2 %	62.67 %	68.36	146.25s
Audio con FFT	74.96 % / 72.25 %	87 %	67.18 %	75.81	119.77s

Tabla 14: Resultados de audio en crudo. Selección aleatoria del conjunto de test.

2. Datos equilibrados. Selección consecutiva del conjunto test

Se separan los 8120 registros correspondientes a enfermos y del resto se seleccionan aleatoriamente otros 8120 de sanos. En ambos conjuntos se toman 1000 registros consecutivos para el conjunto de test. Igual que en el caso anterior, podemos asegurar que hay el mismo número de grabaciones de enfermos que de sanos, pero no de personas. Por ello, se comprueba la distribución de estas en ambos conjuntos.

Personas	Train	Test
Sanos	953	143
Enfermos	324	77

Tabla 15: Distribución de personas audio en crudo y FFT.
Selección de registros consecutivos para el conjunto de test.

Los resultados obtenidos son:

	Accuracy (Train/Test)	Recall	Precision	Fscore	T_{ejec}
Audio	70.64 % / 59.05 %	71.1 %	57.29 %	63.45	155.39s
Audio con FFT	76.19 % / 60.45 %	75.2 %	58.06 %	65.53	115.42s

Tabla 16: Resultados de audio en crudo.
Selección de registros consecutivos para el conjunto de test.

Como ha ocurrido hasta el momento, los resultados de los experimentos con test elegido de forma aleatoria son bastante mejores que los que su conjunto de test se ha formado seleccionando registros consecutivos. Además, han funcionado mejor los experimentos a los que se aplicó al audio la FFT.

Por otra parte, en estos experimentos se ha dado la curiosa situación de que el modelo clasificaba todos los registros como sanos o como enfermos, lo que ha provocado que tengamos que programar el rechazo de estos modelos para evitar errores como una división por cero o escoger como mejor modelo, por su fscore, modelos inútiles. Esto sucede especialmente (un 80 % de las ejecuciones) en los experimentos en los que no se aplica la FFT.

En cuanto a las matrices de confusión, encontramos un patrón común: los enfermos los clasifica muy bien, pero los registros sanos ronda el 50 %, de ahí que el recall sea tan alto y la precisión mucho más baja.

Como curiosidad, destacar que los modelos con audio en crudo tardan unas 5 veces más en entrenar que los modelos con características de sonido. Esto sucede por el número de entradas y el número de parámetros a entrenar en cada modelo.

	Nº entradas	Nº parámetros
Características de sonido	62/63	∼16000
Audio en crudo	10000	∼5000000
Audio en crudo + FFT	5000	∼2500000

Tabla 17: Número de entradas y parámetros en el modelo de cada tipo de experimento.

5.4. Comparativa con otros trabajos anteriores

Una interesante representación de trabajos realizados en los últimos años en este mismo ámbito, pueden ser: [Little et al., 2009], [Tsanas, Little, McSharry, Spielman et al., 2012], [Chen et al., 2013] y [Arora y Tsanas, 2016]. Todos ellos se enfrentaron al problema del diagnóstico de la enfermedad del Parkinson mediante características de sonido, extraídas de grabaciones de vocales sostenidas, y utilizando métodos de data mining.

[Little et al., 2009]

- Cuentan con 195 grabaciones de 31 personas, de las cuales 23 padecen la EP.
- De 17 características de sonido, seleccionan 10 no correlacionadas y prueban combinaciones.
- Para cada combinación, utilizan una clasificación con SVM (máquinas de soporte vectorial). La combinación con mejores resultados es la elegida.
- Validan los resultados mediante bootstrapping con 50 repeticiones.
- Finalmente, con 4 características de sonido logran una tasa de éxito de 91,4 %.

[Tsanas, Little, McSharry, Spielman et al., 2012]

- Cuentan con 263 grabaciones de 43 personas, de las cuales 33 están diagnosticados de la EP.
- Disponen de 132 características de sonido. Para su selección emplean 4 algoritmos diferentes: LASSO, mRMR, RELIEF, LLBFS.
- Prueban el rendimiento de los modelos con diferentes selecciones de características tanto con clasificadores SVM como con bosques aleatorios.
- Validan los resultados mediante validación cruzada con 10 particiones y lo realizan 100 veces.

- Finalmente, tomando 10 características seleccionadas con el algoritmo RELIEF y mediante SVM, consiguen aproximadamente un 99 % de tasa de éxito.

[Chen et al., 2013]

- Cuentan con 195 grabaciones de 31 personas, de las cuales 23 padecen la EP.
- Disponen de 22 características de sonido y hacen una selección de las mismas a través de la técnica PCA.
- Prueban diferentes modelos mediante el algoritmo Fuzzy KNN cambiando el número de vecinos y la intensidad fuzzy.
- Evalúan los resultados mediante validación cruzada con 10 particiones.
- Finalmente, alcanzan un 96 % de tasa de éxito.

[Arora y Tsanas, 2016]

- Cuentan con 2799 grabaciones telefónicas de 1507 enfermos y 15486 de 8394 sanos.
- Disponen de 309 características de sonido.
- Utilizan bosques aleatorios como clasificador.
- Para evaluar los modelos, realizan validación cruzada con 10 particiones 100 repeticiones.
- Finalmente, logran un modelo con un 63,8 % de recall y un 67,2 % de precisión.

Como se ha comentado al inicio, todos utilizan técnicas de data mining, por lo que suelen realizar una selección de características. Sin embargo, en nuestro caso al desarrollar modelos de deep learning no es necesaria esta selección, pues es el propio algoritmo el que aprende qué características son más determinantes.

En cuanto a los resultados que obtienen, excepto Arora y Tsanas, son realmente buenos. De hecho, para encontrar resultados parcialmente similares en los experimentos llevados a cabo, habría que remontarse: a los experimentos KNN con selección del conjunto de test de forma aleatoria y los que evalúan los modelos con validación cruzada; y a los experimentos deep learning con características de sonido que incluyen la variable edad y selección aleatoria del test. Los experimentos con audio en crudo quedan muy lejos de estos resultados.

Igual que en nuestro caso se han obtenido los mejores resultados con selección aleatoria del conjunto de test y evaluando con el método de validación cruzada, todos ellos evalúan sus modelos con validación cruzada y bootstrapping. De ahí que alcancen resultados tan optimistas que no son reales, pues están evaluando sus modelos con datos que ya han visto estos. Por ello, cuando se tiene mayor cuidado en la selección del conjunto de test eligiendo registros consecutivos, obtenemos peores resultados, pero sí que se pueden considerar representativos del conocimiento real del modelo.

También hay que destacar que las grabaciones con las que tratan los tres primeros trabajos están realizadas en condiciones de laboratorio, por lo que la calidad de las mismas es mucho mayor que las grabaciones telefónicas utilizadas por Arora y Tsanas y en este proyecto. Es quizás por este motivo, por el que sus resultados son más similares a los nuestros.

6. Conclusiones

Tal y como se ha ido comentando en los resultados de los experimentos, hay una serie de conclusiones comunes a todos.

La más importante es que en la selección de conjuntos de entrenamiento y de test hay que tener especial cuidado, pues ha quedado demostrado que seleccionándolos aleatoriamente o evaluando los modelos a través de la validación cruzada, se consiguen resultados demasiado buenos y optimistas. Sin embargo, eligiendo los conjuntos con la precaución de que no haya grabaciones de un mismo paciente en ambos, se obtienen resultados más limitados, pero indican el conocimiento real del modelo y la posible respuesta de este a datos completamente nuevos.

También se ha observado que cuando se usa la variable edad como predictor, los resultados mejoran. No obstante, no deberíamos tenerla en cuenta, pues la distribución de edades en el conjunto de enfermos es muy diferente a la del conjunto de sanos, y no queremos que a partir de cierta edad del individuo, el modelo lo clasifique siempre como enfermo.

Los resultados obtenidos del audio en crudo no son demasiado positivos, por lo que habría que probar otros métodos. De esta manera, el diagnóstico del Parkinson mediante la voz en crudo queda todavía abierto a nuevas investigaciones.

Con relación a los datos utilizados en la investigación, procedentes del proyecto mPower, hay que destacar la incongruencia de datos: ciertos individuos tienen grabaciones como sano y otras como enfermo. Esto puede confundir a los modelos, por ello proponemos que mPower establezca algún tipo de control para que esto no suceda.

En cuanto a mi experiencia personal, ha sido muy enriquecedor poder formar parte de un equipo de investigación de mi escuela con tanta experiencia. Gracias a ello, he tomado conciencia sobre qué implica seguir una línea de investigación en el ámbito informático.

Cuando me ofrecieron realizar este proyecto, una de las cosas que me entusiasmó fue la posibilidad de aumentar mis conocimientos sobre inteligencia artificial a través de un escenario real. He podido experimentar ampliamente con esta tecnología a lo largo del proyecto, lo que me ha permitido conocer el gran potencial que tiene. Sin embargo, también he podido comprobar que se necesita un potente equipo informático donde llevar a cabo los experimentos, ya que la implementación de esta tecnología supone una computación muy pesada.

El hecho de haber realizado un PFG con tanta repercusión social es algo que me ha motivado desde el principio. Considero que los resultados obtenidos ayudarán, junto con otros trabajos, a miles de personas que sufren esta terrible enfermedad.

7. Impacto social

Dado que este proyecto se enmarca dentro de una línea de investigación que aún no ha dado un resultado tangible, no podemos decir que tenga un impacto directo en la sociedad. Sin embargo, es innegable que participa en el avance de la investigación del diagnóstico de la EP mediante la voz.

Cuando en un futuro próximo se alcance un modelo de red neuronal con unos resultados relevantes, será totalmente viable la implementación de una aplicación móvil que, mediante grabaciones de voz, permita determinar con bastante certeza si una persona padece o no la enfermedad de Parkinson. Esto resultaría de gran interés, puesto que a día de hoy no existe ninguna prueba específica para su diagnóstico (y mucho menos tan ágil).

Por otra parte, si se consiguiese un modelo de clasificación multiclase que determine la gravedad o el estadio en el que se encuentra un paciente, se podría hacer un seguimiento de su evolución de manera telemática. De esta forma, no haría falta que los pacientes se desplazasen tan frecuentemente a los centros de salud para dicho seguimiento, lo cual es una gran ventaja si pensamos en pacientes de avanzada edad. Además, esto significaría un claro ahorro en tiempo y esfuerzos, que supondría un impacto positivo en la economía de la sanidad y la descongestión de los hospitales de las grandes ciudades.

De cualquier modo, el fin de esta investigación es mejorar la calidad de vida de los enfermos de Parkinson y de sus familiares, y esto es algo que veremos en los próximos años con unos resultados muy esperanzadores.

8. Trabajos futuros

Aunque en este PFG se ha desarrollado una línea de investigación implementando diversos experimentos y se han llegado a una serie de conclusiones, no significa que la investigación haya terminado, sino todo lo contrario. Aún quedan bastantes posibilidades que indagar.

- Realizar una nueva extracción de características de sonido, quizás más similares a las que utilizan otros trabajos de las referencias.
- Probar *data augmentation*. Como se dispone de un menor número de grabaciones de enfermos, una manera de aumentarlas sería tomar otro intervalo de las grabaciones. Así, contaríamos con el doble de grabaciones de enfermos y tendríamos más información disponible.
- Probar los coeficientes cepstrales de frecuencias-mel en los modelos de audio en crudo, que dan información de la variación temporal de la distribución de frecuencias, muy utilizados en el reconocimiento de voz.
- Una vez encontrado un modelo adecuado, se podría implementar una aplicación móvil multidispositivo que fuese capaz de diagnosticar la EP en tiempo real.

Referencias

[1] *Historia de la Enfermedad del Parkinson,* URL: http://www.neurowikia.es/content/historia-de-la-enfermedad-de-parkinson?quicktabs_block_views_popular_block=1 (visitado 22-03-2019).

[2] M. J. Catalán y A. Rodríguez del Álamo. *Definición de la enfermedad de Parkinson.* URL: https://www.parkinsonmadrid.org/el-parkinson/el-parkinson-definicion/ (visitado 22-03-2019).

[3] A. Roberts-South. *Improving Communication in Parkinson's Disease.* URL: http://www.pdf.org/winter%5C_14%5C_comunication (visitado 09-04-2017).

[4] B. E. Sakar, M. E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin y O. Kursun. «Collection and analysis of a Parkinson speech dataset with multiple tipes of sound recordings». En: *IEEE Journal of Biomedical and Health Informatics* 17.4 (2013), págs. 828-834.

[5] P. Tan, M. Steinbach y V. Kumar. *Introduction to Data Mining.* New York: Pearson International Edition, 2006.

[6] H. L. Chen, C. C. Huang, X. G. Yu, X. Xu, X. Sun, G. Wang y S. J. Wang. «An Efficient Diagnosis System for Detection of Parkinson's Disease Using Fuzzy K-nearest Neighbor Approach». En: *Expert Syst. Appl.* 40.1 (ene. de 2013), págs. 263-271. ISSN: 0957-4174.

[7] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman y L. O. Ramig. «Suitability of dysphonia measurements for telemonitoring of Parkinson's Disease». En: *IEEE Transactions of Biomedical Engineering* 56.4 (2009), págs. 1015-1022.

[8] A. D. Trister, E. C. Neto, B. M. Bot, T. Perumal, A. Pratap, A. Klein, E. R. Dorsey, C. M. Tanner y S. H. Friend. «mPower: A smartphone-based study of Parkinson's disease provides personalized measures of disease impact». En: *Mov Disord. 31, 20th International Congress.* 2016.

[9] A. Tsanas, M. A. Little, P. E. McSharry, J. Spielman y L. O. Ramig. «Novel speech signal processing algorithms for high accuracy classification of Parkinson's disease». En: *IEEE Transactions of Biomedical Engineering* 59.5 (2012), págs. 1264-1271.

[10] A. B. Soliman, M. Fares, M. M. Elhefnawi y M. Al-Hefnawy. «Features Selection for Building an Early Diagnosis Machine Learning Model for Parkinson's Disease». En: *International Conference on Artificial Intelligence and Pattern Recognition (AIPR) IEEE.* 2016.

[11] S. Arora y A. Tsanas. «Discrimination of Parkinson's Disease participants from healthy controls using telephone-quality voice recordings». En: *Mov Disord. 31, 20th International Congress.* 2016.

[12] B. M. Bot, C. Suver, E. C. Neto, M. Kellen, A. Klein, C. Bare, M. Doerr, A. Pratap, J. Wilbanks, E. R. Dorsey, S. H. Friend y A. D. Trister. «The mPower study, Parkinson disease mobile data collected using ResearchKit». En: *Scientific Data* 3.160011 (2016), págs. 1-9.

[13] F. Chollet y J. J. Allaire. *Deep Learning with R.* New York: Manning Publications, 2018.

[14] *mPower: Mobile Parkinson Disease Study.* URL: https://www.synapse.org/mPower (visitado 26-06-2019).

[15] J.Wilbanks y H. Friend. «First, design for data sharing». En: *Nature Biotechnology* 34.4 (2016), págs. 377-379.

[16] P. Pinho, L. Monteiro, M. F. Soares, L. Tourinho, A. Melo y A. C. Nóbrega. «Impact of levodopa treatment in the voice pattern of Parkinson's disease patients: a systematic review and meta-analysis». En: *CoDAS* 30.5 (2018).

[17] M. Giuliano, A. García-López, S. Pérez, F. D. Pérez, O. Spositto y J. Bossero. «Selection of voice parameters for Parkinson's disease prediction from collected mobile data». En: *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*. Abr. de 2019, págs. 1-3.

[18] D. A. Puts, L. M. Doll y A. K. Hill. «Sexual Selection on Human Voices». En: *Evolutionary Perspectives on Human Sexual Psychology and Behavior*. Springer, 2014. Cap. 3, págs. 69-86.

[19] A. Tsanas, M. A. Little, P. E. McSharry y L. O. Ramig. «Nonlinear speech analysis algorithms mapped to a standard metric achieve clinically useful quantification of average Parkinson disease symptom severity». En: *J. Royal Society Interface* 8.59 (2010), págs. 842-855.