



Facultad de Ciencias Económicas y Empresariales

# **EL DEEP LEARNING: PERSPECTIVA GENERAL Y SU APLICACIÓN EN EL CAMPO DEL HEALTHCARE.**

Clave: 201702688

Autor: Laura García Tudela

Tutor: Carlos Martínez de Ibarreta Zorita

## ÍNDICE

Resumen: .....	3
Abstract: .....	3
Abreviaturas: .....	4
INTRODUCCIÓN.....	6
1.- ¿Qué es el DeepLearning?.....	10
2.- Redes Neuronales Artificiales .....	15
2.1- ANN y su similitud con el cerebro humano.....	15
2.2-La neurona artificial: el perceptrón .....	17
2.3- El perceptrón multicapa .....	20
2.4- El aprendizaje de las redes neuronales.....	21
3.-Redes Neuronales Convolucionales:.....	24
3.1-Tipos de capas .....	25
3.2-Arquitectura de Redes Convoluciones .....	30
4. Redes neuronales recurrentes .....	33
4.1 <i>Backpropagation through time</i> .....	34
4.2 <i>Gated cells</i> .....	35
4.3 Codificación del lenguaje.....	37
4.4 Tipos de redes neuronales recurrentes .....	38
5.-Aplicaciones actuales y futuro cercano en el campo de la salud .....	39
5.1- Procesamiento de imágenes: diagnóstico y predicción .....	40
5.2-Preprocesamiento de lenguaje: <i>electronic health records</i> .....	45
6.-Desarrollo de una Red Neuronal Convolutiva en Python .....	49
7.-Dilemas éticos: AI y DeepLearning aplicados al healthcare.....	57
8.-Conclusión.....	61
REFERENCIAS: .....	62

## **Resumen:**

Este trabajo de fin de grado realiza una revisión de las distintas aplicaciones del Deep Learning -más concretamente de las Redes Neuronales- en el sector sanitario. Para ello, en primer lugar, se ofrece una visión general del funcionamiento de las redes neuronales, así como de los tipos que actualmente encabezan el estado del arte: las Redes Neuronales Convolucionales y las Recurrentes. Respecto de los usos concretos que se están dando actualmente, se exponen aquí algunos modelos que por diversos motivos destacan sobre el resto. Los modelos expuestos buscan tanto mejorar la predicción de enfermedades como para procurar una mejora asistencial del paciente. El proyecto concluye con una reflexión acerca de la necesaria vertiente ética de los usos de la Inteligencia Artificial en un sector que usa información tan sensible como lo son los datos clínicos.

**Palabras clave:** Inteligencia Artificial, Deep Learning, Sector Sanitario, Redes Neuronales, Convoluciones, Recurrentes.

## **Abstract:**

*This article analyzes how Deep Learning models, specially Neural Networks, may be applied in the Healthcare Sector. To this aim, firstly a theoretical overview of the functioning of these models is given, as well as the types that are currently leading the state of the art: Convolutional Neural Networks and Recurrent Neural Networks. Moreover, with respect to the specific applications, an explanation of models created with sanitary purposes is presented by explaining some models that for various reasons stand out from the rest. Some of them are predictive models designed for diseases diagnosis. Others aim to ameliorate sanitary assistance as a whole. This paper concludes with a brief reflection on the necessary ethical aspect of AI.*

**Keywords:** Artificial Intelligence, Deep Learning, Healthcare Sector, Neural Networks, Convolutional, Recurrent.

## Abreviaturas:

AI	<i>Artificial Intelligence</i>
ANN	<i>Artifital Neural Networks</i>
AUC	<i>Area Under the Curve</i>
BPTT	<i>Back Propagation Through Time</i>
CNN	<i>Convolutional Neural Network</i>
CPU	<i>Central Processing Unit</i>
ECVL	<i>European Computer Vision Library</i>
EDDLL	<i>European Distributed Deep Learning Library</i>
EHR	<i>Electronic Health Record</i>
EMBC	<i>Engineering in Medicine and Biology Society)</i>
GPU	<i>Graphics Processing Unit</i>
GRU	<i>Gated Recurrent Unit</i>
IA	Inteligencia Artificial
ICD	<i>International Classification of Diseases</i>
ILSVR	<i>ImageNet Large Scale Visual Recognition</i>
ISBI	<i>International Symposium on Biomedical Imaging</i>
LOINC	<i>Logical Observation Identifiers Names and Codes</i>
LSTM	<i>Long Short Term Memory</i>
MAE	<i>Mean Absolute Error</i>
MAPE	<i>Mean Absolute Porcentual Error</i>
MINST	<i>Modified National Institute of Standards and Technology</i>
MIT	<i>Massachusetts Institute of Technology</i>
ML	<i>Machine Learning</i>
MLP	<i>Multi Layer Perceptron</i>
MRI	<i>Magnetic Resonance Imaging</i>
NDC	<i>National Drug Code</i>
OMS	<i>Organización Mundial de la Salud</i>
PCA	<i>Principal Component Analysis</i>
ReLU	<i>Rectified Linear Unit</i>
RMSE	<i>Root Mean Square Error</i>
RNN	<i>Recurrent Neural Networks</i>

RNR	Redes Neuronales Recurrentes
RNA	Redes Neuronales Artificiales
RNC	Redes Neuronales Convolucionales
ROC	<i>Receiver Operating Characteristic</i>
SLN	<i>Sentinel Lymph Node</i>
SVM	<i>Support Vector Machine</i>

# INTRODUCCIÓN

El Healthcare es uno de los sectores más importantes para la economía global. Así, siendo conscientes de la importancia de mantener un sistema de salud lo suficientemente fuerte, y de cómo el uso de la tecnología era el camino para conseguirlo, en 2019 la Unión Europea desarrolla el programa *DeepHealth*. Este programa se crea con el fin de aprovechar la gran cantidad de datos que generan los sistemas sanitarios nacionales para poder unirlos en un dataset común y que de esta forma estos dejaran de ser lo que habían sido hasta ahora; fuentes de conocimiento sin explotar.

La fuente de datos más importante para la aplicación de las tareas que la Inteligencia Artificial puede llevar al campo de la salud son las imágenes. Además, en el contexto de unificación de bases de datos (objetivo del programa *DeepHealth*), son el tipo de *input* más fácil de tratar, pues no necesitan de una fase previa de codificación que puede sin duda dificultarse aún más por las barreras lingüística. Aun así, a día de hoy, las imágenes siguen, por norma general, siendo únicamente revisadas por especialistas.

El proyecto incluye el desarrollo de dos nuevas librerías de software libre y código abierto, la *European Distributed Deep Learning Library* (EDDLL) y la *European Computer Vision Library* (ECVL), lo que facilitará la investigación y el desarrollo.

Diagnóstico, monitorización y tratamiento son las 3 claves que el DeepLearning ya es capaz de mejorar, y que resultan en una mejora significativa tanto del tratamiento del paciente, como de la eficiencia y eficacia de los sistemas sanitarios nacionales. Fue buscando la mejora en cada una de estos que el proyecto *DeepHealth* ha financiado los llamados 14 “*success stories*”, sobre los que se probó el proyecto. Se trata de aplicaciones distintas, pero todas ellas se refieren a una de las tres áreas de referencia en este campo: enfermedades neurológicas, detección del cáncer, y patología mediante diagnóstico de imágenes. Entre ellas cabe destacar las que utilizan redes neuronales profundas para el reconocimiento de imágenes, por tratarse de la gran mayoría:

- *Predictive and Populational Model for Alzheimer's Disease using Structural Neuroimaging*<sup>1</sup>
- *Skin lesion classification and segmentation*<sup>2</sup>,

---

<sup>1</sup> Universidad de Valencia

<sup>2</sup> Universidad de Modena

- *Segmentation of the lumbar region*<sup>3</sup>
- *Renal and adrenal tumour diagnosis*<sup>4</sup>
- *Lung cancer diagnosis*<sup>5</sup>
- *Deep Learning for accurate diagnosis of prostate cancer*<sup>6</sup>
- *Multiple Sclerosis Lesion Segmentation*<sup>7</sup>

Otro ejemplo claro de cómo este tipo de algoritmos destaca en el estado del arte lo encontramos en un momento clave para demostrar la eficiencia de nuestros sistemas sanitarios: la llegada de la Covid-19. Durante la pandemia, y, de hecho, aún se sigue haciendo, se desarrollaron cientos de modelos predictivos que intentaban llevar a cabo distintas tareas, siendo la más común la detección de la enfermedad.



**Figura 1:** Tipo de arquitecturas de Deep Learning usadas para la detección de pacientes con COVID-19 basado en los papers publicados. Fuente: Shoeibi, A., Khodatars, M., Alizadehsani, R., Ghassemi, N., Jafari, M., Moridian, P., ... & Shi, P. (2020). Automated detection and forecasting of covid-19 using deep learning techniques: A review. *arXiv preprint arXiv:2007.10785*.

Como puede apreciarse en la figura 1, las Redes Neuronales Convolucionales fueron el tipo de arquitectura más desarrollado. Es por ello que, aunque este proyecto busca mostrar una perspectiva general del Deep Learning, y de su aplicación en el campo del

<sup>3</sup> Fundacio Fisabio.

<sup>4</sup> Bucharest

<sup>5</sup> Universidad de Turin

<sup>6</sup> Karolinka Institutet (Sweden)

<sup>7</sup> Universidad de Lausanne

*Healthcare*, es inevitable dedicar una mayor extensión al análisis de las Redes Neuronales en general, y de entre ellas, a las Convolucionales en especial.

Según la OMS, durante 2015, los errores médicos relacionados con el uso incorrecto de fármacos y fallas de diagnóstico representaron la tercera causa de muerte en Estados Unidos y 10% de todas las muertes. (Cabrol et al., 2020) El desarrollo de algoritmos que posibilitan la producción de diagnósticos mediante el uso de la tecnología podría reducir esta componente de mortalidad. Pero, el motivo por el que escogí este tema no es solo este, sino la ya mencionada necesidad de desarrollo de un sistema sanitario lo más eficiente posible a través de herramientas que, como se mostrará en este Trabajo de Fin de Grado, están ganando cada vez más atención tanto del sector tecnológico, como del sanitario.

Respecto de la metodología utilizada para la realización de este Trabajo de Fin de Grado, que busca revisar la aplicación de técnicas específicas de Deep Learning a un campo tan concreto como el sanitario, cabe realizar una distinción por secciones, pues no en todos los apartados se han llevado a cabo las mismas pautas (en lo que a exhaustividad y tecnicismo se refiere).

En primer lugar, y dado que, aunque gracias a las asignaturas realizadas durante el grado en *Business Analytics*, las técnicas principales de Machine Learning (ML) han sido para mí ya objeto de estudio, el Deep Learning (DL) va un paso más allá. Por tanto, se realiza aquí una revisión general para comprender no solo qué es el DL, sino también qué papel juega dentro de la Inteligencia Artificial y cómo se diferencia del ML. A continuación, se pasa a una parte más técnica del proyecto, dedicada en concreto al estudio de las Redes Neuronales: desde sus elementos más básicos, pasando por explicar los primeros algoritmos con los que se empezaron a desarrollar este tipo de redes, y acabando con una explicación de las arquitecturas que están siendo ampliamente utilizadas en la actualidad.

La segunda gran parte del trabajo se dedica a mostrar, en función del tipo de red utilizada, algunas de las aplicaciones más importantes de este tipo de algoritmo en el sistema sanitario, recogiendo tanto los estudios considerados hoy paradigmáticos como algunos muy recientes. Se incluye además un ejemplo de código de implementación de una Red Neuronal Artificial sencilla. Para finalizar, y antes de concluir, dada mi formación en Derecho y mi interés por la ética, se incluye un apartado que intenta mostrar cómo, aunque estas herramientas puedan desembocar en una mejora para la sociedad, debemos ser

conscientes de los peligros que conllevan el uso de datos de tal sensibilidad como lo son los datos médicos sanitarios.

# 1.- ¿Qué es el DeepLearning?

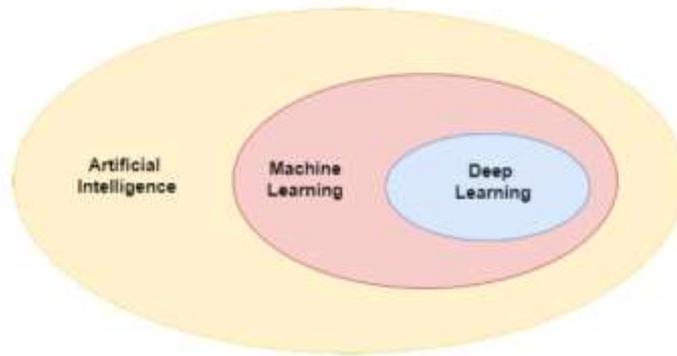
No podemos comenzar un Trabajo de Fin de Grado sobre el papel de las redes neuronales en el campo de la salud sin comenzar con unas previas nociones básicas que nos sirvan de contexto, y principalmente, nos sitúen el DL dentro del universo que actualmente conforman las distintas formas que engloban la inteligencia artificial.

Las técnicas de DL han supuesto una verdadera revolución en el campo de la inteligencia artificial. Aunque el concepto nace en la década de los 70, es gracias a la aparición de datos masivos que surge la necesidad de crear algoritmos capaces de tratar con volúmenes de datos que desde 2010 han sufrido un crecimiento exponencial.

En la última década, estas técnicas han conseguido resolver problemas que antes resultaban inalcanzables para las máquinas. Incluso en nuestro día a día, casi sin ser conscientes de ello, utilizamos el DL para una gran cantidad de tareas: al traducir un texto en internet, al desbloquear la pantalla del móvil a través del sistema de reconocimiento facial, al hacer una foto y que nuestro smartphone conozca la ubicación o qué personas aparecen a través del reconocimiento de imágenes, o el uso de asistentes digitales como Siri. El desarrollo de las técnicas de DL ha afectado a la práctica totalidad de las industrias en una gran cantidad de aspectos, y a día de hoy, gracias a la gran proliferación de datos generada por las nuevas realidades de entre las que podemos destacar las redes sociales, así como por el aumento de inversión en tecnología y computación, parecen ser más relevantes que nunca.

Sin embargo, antes de adentrarnos en el mundo del DL, es necesario explorar una serie de conceptos previos e identificar qué papel ocupa el DL en el universo que conforma la Inteligencia Artificial (en adelante IA). Los términos a explicar son: aprendizaje supervisado, no supervisado y mixto, y Machine Learning.

Existe una estrecha relación entre los conceptos de Inteligencia Artificial, Machine Learning y Deep Learning. En concreto, se trata de subgrupos:



**Figura 2:** Diferencias entre IA, Machine Learning y DeepLearning. Fuente: ichi.pro

Como se puede observar en la Figura 2, la IA contiene los conceptos de ML y DL, y aunque abarca mucho más que eso, dentro de las técnicas de la IA destacan el Machine Learning (aprendizaje automatizado) y el Deep Learning (aprendizaje profundo).

El concepto de Inteligencia Artificial data de la década de los cincuenta, con la idea de automatizar tareas que normalmente son realizadas por humanos. (Chollet, 2018) Podríamos decir que la IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano, en una gran variedad de campos. (Blanco et al., 2020)

El concepto de ML fue el más exitoso a partir de la década de los 90, siendo bastante más específico que el concepto de Inteligencia Artificial. Se habla de aprendizaje automático pues se refiere a aquellos modelos que no son programados con una serie de reglas diseñadas por humanos, sino que se entrenan o aprenden de los datos.

Dichos modelos llevan a cabo, a través de diversos algoritmos, tareas de clasificación o predicción, utilizando para ello un gran abanico de algoritmos, siendo algunos de los más comunes SVM (*Support Vector Machine*), árboles de decisión, PCA (*Principal Component Analysis*), K-means Clustering y las redes neuronales. Respecto del aprendizaje de este tipo de algoritmos, se distinguen distintos métodos que se agrupan normalmente en dos categorías: supervisado y no supervisado.

Respecto a los modelos de aprendizaje supervisado, estos requieren una mayor intervención humana pues se usan como input datos que han sido previamente etiquetados. Es decir, el modelo conoce el *input* y el *output* y aprende, a partir de estos, patrones que le permiten la aplicación a otros datos cuya solución se desconozca. Se

puede hablar así de un aprendizaje a través de ejemplos, y de ahí la importancia de la medición del error o evaluación de la *performance* del modelo.

De esta forma, tres son los elementos básicos de los modelos de Machine Learning que aplican aprendizaje supervisado (Tamir, 2020):

1. Proceso de decisión: trata de hallar los patrones en los datos que son clave para realizar la tarea.

2. Función de error: Para conocer la *performance*<sup>8</sup> del modelo. Algunos ejemplos de funciones de pérdida son RMSE (*Root Mean Square Error*), MAPE (*Mean Absolute Porcentual Error*), MAE (*Mean Absolute Error*) y entropía cruzada binaria o multiclase y *hinge loss* (para modelos de clasificación).

3. Proceso de optimización del modelo: se ajusta y actualiza el modelo para mejorarlo en mayor o menor medida dependiendo del error detectado.

En segundo lugar, en los modelos de aprendizaje no supervisado, los algoritmos no usan ningún dato etiquetado u organizado previamente para indicar cómo tendría que ser categorizada la nueva información, sino que tienen que encontrar la manera de clasificarla ellos mismos. Por tanto, este método no requiere de intervención humana. (Gershgorin, 2017) Es decir, se sirven únicamente de los datos en sí, extrayendo estos mismos patrones o correlaciones, sin hacer uso de las respuestas o *outputs*.

En muchas ocasiones se usa como paso previo antes de afrontar un problema de aprendizaje supervisado. Dos de las formas más comunes de aprendizaje no supervisado son la reducción de dimensiones y el *clustering*. (Chollet, 2018)

Sin embargo, existen otros tipos de aprendizaje que van más allá de la dicotomía del aprendizaje supervisado y no supervisado, siendo el aprendizaje auto supervisado y el aprendizaje reforzado dos grandes ejemplos. En el caso del aprendizaje auto supervisado, se realiza un etiquetado de los datos de entrada, pero a diferencia del aprendizaje supervisado, estos datos de entrenamiento se etiquetan de forma autónoma. Así, los *outputs* con los que se entrena el modelo no han sido determinados por humanos, sino que se generan directamente desde los *inputs*. Respecto al aprendizaje reforzado, cabe recalcar que se trata de un tipo de aprendizaje que se encuentra en fase de investigación

---

<sup>8</sup> Calidad de predicción o clasificación del modelo

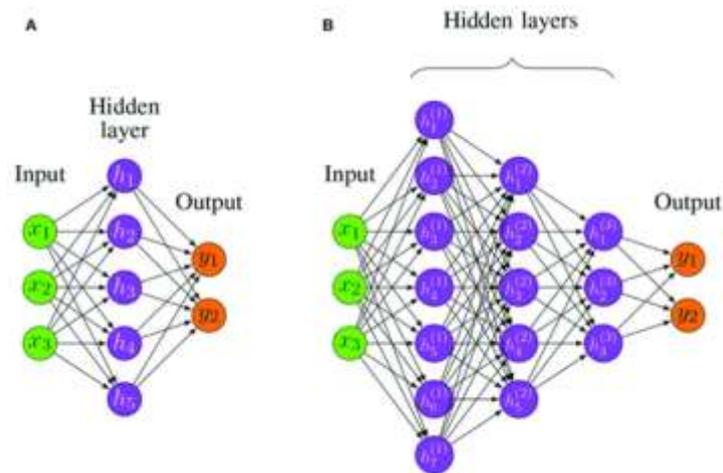
y a día de hoy no cuenta con grandes aplicaciones prácticas. (Chollet, 2018) El aprendizaje reforzado, a diferencia del aprendizaje supervisado, no cuenta con objetivos explícitos, sino que busca obtener los mejores resultados posibles mediante pruebas de acierto y error en búsqueda de experiencia, recibiendo una especie de “recompensa” o “refuerzo positivo” cada vez que el modelo acierta.

Siguiendo con nuestro análisis del universo que conforma la IA, el DL es un subgrupo del ML. Así las cosas, cabe plantearse qué aporta de especial el Deep Learning en relación con las técnicas de Machine Learning.

Las técnicas de ML se mostraban insuficientes de cara a procesar los datos en su forma original –o forma bruta. Es decir, era necesario transformar esos datos para convertirlos en formas que el modelo pueda comprender, por medio de extracción o creación de nuevas *features*. Por el contrario, el Deep Learning, al estar fundado en modelos que se sustentan en el aprendizaje basado en la representación –*Representation Learning*–, se pueden usar los datos en su forma original. (Lecun et al., 2015) Aunque desarrollaremos este concepto más adelante, las técnicas de DL usan modelos no lineales que permiten transformar los datos originales y abstraerlos en distintas capas de cara a la toma de decisiones. Esto es clave cuando nos enfrentemos a problemas cuya solución no sea lineal. Respecto a los datos originales, cabe destacar que las técnicas de Deep Learning requieren una cantidad de datos y potencia de cómputo muy superior a aquella requerida para las técnicas convencionales de Machine Learning. (Chauhan y Singh, 2018) En definitiva, el DL es una forma sofisticada de ML basada en las redes neuronales que nos permite realizar tareas mucho más complejas, para las que se necesita un volumen muy superior de datos.

Una vez analizadas las diferencias entre ML y DP, es interesante pararse a destacar las disparidades entre el concepto de *Shallow Learning* y *Deep Learning* (ambos son subconjuntos del ML). Estas técnicas están basadas en redes neuronales artificiales. A diferencia de “*shallow*” o superficial, el término “*deep*” o profundo, viene a hacer referencia al número de *hidden layers* (capas ocultas), que contiene una red neuronal. Así, cabe decir que la diferencia entre las técnicas de *Shallow* y *Deep Learning* estriba en el número de capas intermedias entre los inputs y los outputs. En el caso del *Deep Learning*, se usan una gran cantidad de capas ocultas que permiten al modelo aprender representaciones de datos con múltiples niveles de abstracción. (Vázquez, 2017) En este sentido, no existe un estándar de cara a determinar a partir de qué número de capas ocultas

se puede considerar que una red es de DL. Sin embargo, la mayoría de los expertos opinan que se consideraría DL desde el momento en el que se implemente una segunda capa oculta.



**Figura 3:** Dos ejemplos de Redes Neuronales Artificiales. (A) A shallow neural network . (B) A Deep neural network, aunque podría ser discutible pues solo tiene 3 capas ocultas. Fuente: Emmert-Streib, Frank & Yang, Zhen & Feng, Han & Tripathi, Shailesh & Dehmer, Matthias. (2020). An Introductory Review of Deep Learning for Prediction Models With Big Data. *Frontiers in Artificial Intelligence*.

Otra diferencia clave es que los algoritmos de aprendizaje profundo destacan por la escalabilidad de los datos. Es decir, existe una alta capacidad de adaptación y a medida que aumentan el volumen de datos, el modelo se adapta y mejora. Sin embargo, el *Shallow Learning*, se estabiliza en un determinado nivel de rendimiento que no mejora, al contrario, empeora cuando se añaden más datos de entrenamiento a la red.

A continuación, nos adentraremos en el concepto de redes neuronales, explicando su funcionamiento y relevancia.

## 2.- Redes Neuronales Artificiales

Aun cuando no podemos decir que las Redes Neuronales Artificiales son un subconjunto del Deep Learning-pues pueden ser redes *Deep* o *Shallow* dependiendo del número de capas que las formen- se trata del elemento vertebrador del DL, al ser a través de este tipo de algoritmo que se están consiguiendo los mayores avances. Estos avances se están consiguiendo, gracias a características de entre las que cabe destacar la posibilidad de incluir una gran cantidad de capas, la escalabilidad de los datos y la posibilidad de trabajar con problemas no lineales. A continuación, se estudiarán estas y otras muchas más en mayor profundidad.

### 2.1- ANN y su similitud con el cerebro humano

Antes de meternos de lleno en el funcionamiento de las Redes Neuronales Artificiales (más conocidas como *Artifitial Neural Networks*, de donde proviene el acrónimo ANN), es necesario comprender el porqué de su denominación.

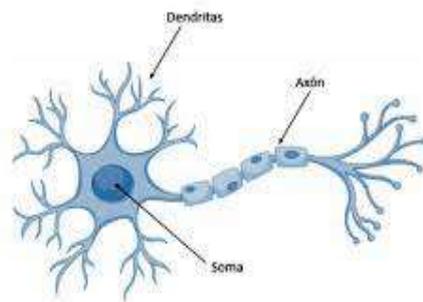
El aprendizaje está, tal y como lo conocemos los seres humanos, íntimamente ligado a una capacidad que se escapa del alcance de los algoritmos: la experiencia humana. Así, parece claro que una forma coherente de aproximarse al problema consiste en la construcción de sistemas capaces de imitar características humanas. Para intentar que las computadoras adquirieran una capacidad humana como el aprendizaje, es por ello que, las ANN se inspiran en el funcionamiento de las redes neuronales biológicas del cerebro humano.

Se han desarrollado en la literatura diversas definiciones de ANN, todas ellas haciendo referencia a su relación con los modelos biológicos. Las redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico. (Kohonen, 1988)

A partir de esta y otras definiciones que siguen la misma línea, se extrae la idea de que una Red Neuronal Artificial es un sistema que, inspirado en el sistema nervioso, permite

establecer relaciones entre *inputs* y *outputs*. Estas imitan el comportamiento del cerebro humano, donde se procesa la información en paralelo y no de manera secuencial – que es como se procesa en las formas de computación tradicionales. Se crea así la posibilidad de que las máquinas sean capaces de generalizar situaciones a través de la extracción de patrones de una serie de datos sin tratamiento previo, que serán aplicados en escenarios distintos, sin necesidad de modificación o intervención humana. Se requiere pues, a partir de esta definición, realizar una primera revisión del funcionamiento del sistema nervioso humano y por ende de su unidad básica, la neurona.

El español Ramón y Cajal, Premio Nobel de Medicina, fue el primero en identificar la neurona como unidad básica funcional del cerebro. Se cambia así la concepción que se tenía hasta el momento al demostrar que el sistema nervioso está formado por una red de células individuales o neuronas, y no por fibras nerviosas que se conectaban entre sí formando un sistema<sup>9</sup>. (Ramón y Cajal, 1911) Las tres partes principales de una neurona son: las dendritas, el cuerpo de la célula y el axón. El proceso esencial en el sistema nervioso humano y mediante el cual se lleva a cabo la conexión de esta red es la llamada sinapsis.



**Figura 4:** Representación básica neurona biológica. Fuente: García-Olalla, O (16 de septiembre de 2019). Redes Neuronales Convolucionales: qué son y cómo se entrenan.

<https://www.xeridia.com/blog/redes-neuronales-artificiales>

Las dendritas actúan como receptoras, cargando de señales el cuerpo de la célula (o soma, tal y como se denomina en la figura 4). Este realiza la suma de las señales de entrada y es el axón el encargado de llevar la señal hasta otras neuronas. Lo que ocurre en una neurona

---

<sup>9</sup> Concepción anterior o decimonónica

biológica es básicamente lo siguiente: la neurona se estimula a través de sus entradas (*inputs*) y cuando se alcanza un cierto umbral, la neurona se activa, pasando una señal hacia el axón. Pero, la cuestión que debemos preguntarnos aquí para trasladar la idea a la AI es la siguiente: ¿cómo se produce la transformación de este *input* en *output*?

Neurona biológica	Neurona artificial
- Señales que llegan a las sinapsis.	- Entradas a la neurona.
- Carácter excitador o inhibitor de las sinapsis de entrada.	- Pesos de entrada.
- Estímulo total de la neurona.	- $Net_j = \sum_{i=0}^N w_i(t) \cdot x_i(t)$ .
- Activación o no de la neurona.	- Función de activación.
- Respuesta de la neurona	- Función de salida.

**Figura 5:** Tabla comparativa neurona biológica y neurona artificial. Fuente: Redes Neuronales Artificiales. Fundamentos teóricas y aplicaciones prácticas. Raquel Flórez, José Miguel Fernández

## 2.2-La neurona artificial: el perceptrón

El modelo matemático más simple y, por ende, más estudiado de una neurona, es el perceptrón. El concepto de perceptrón fue introducido en 1957 por Frank Rosenblatt, en su obra *The perceptron—A perceiving and recognizing automaton*. El perceptrón funciona como un clasificador binario para problemas linealmente separables y es el funcionamiento del mismo el que permite que se produzca la conocida como “*forward propagation*”.

La estructura básica del perceptrón puede definirse como un elemento que admite una serie de inputs,  $x_i$ ,  $i=1, \dots, M$ , todos ellos con sus correspondientes pesos,  $w_i$ ,  $i=1, \dots, M$ . La suma ponderada de estos inputs pasa a través de la llamada función de activación ( $g$ ). A esta función de activación debe añadirse un sesgo,  $w_0$ , que nos permita corregir la función para generar el output “ $y$ ”.

$$y = g \left( w_0 + \sum_{i=1}^m x_i w_i \right)$$

**Figura 6:** Modelo matemático de perceptrón. Fuente: elaboración propia

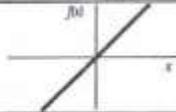
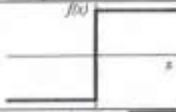
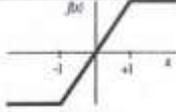
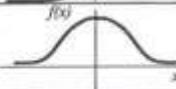
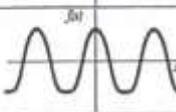
Podemos concluir que el comportamiento de la neurona artificial depende casi completamente de las funciones de activación, pues transforman la combinación de datos de entrada, pesos y sesgos en un producto que pasará- o no- a la siguiente capa; dependiendo del llamado “estado de activación”. Según este estado de activación, una neurona biológica puede activarse -excitarse- o no ante un estímulo del exterior. En el perceptrón se fija un valor umbral o *threshold*, de manera que, al igual que no todos los estímulos externos activan todas las células humanas, la señal solo se transmite si se supera este determinado valor.

La importancia de la función de activación es innegable, dado que representa simultáneamente la salida de la neurona y su estado de activación. Bajo la definición originaria de perceptrón, esta se basó en la función de Heaviside, también conocida como función de escalón. La función de escalón produce excitación o inhibición de la neurona dependiendo de si se supera o no el valor umbral, tomando 1 o 0 respectivamente. Quedaría definida como:

$$f(x) = \begin{cases} 0 & \text{si } \sum w_j x_j \leq \text{threshold} \\ 1 & \text{si } \sum w_j x_j > \text{threshold} \end{cases}$$

**Figura 7:** Función de escalón. Fuente: Spiegel, M. & Abellanas, L.: "Fórmulas y tablas de matemática aplicada", Ed. McGraw-Hill, 1988.

No todas las neuronas artificiales pueden tener únicamente dos estados de activación, sino que algunas pueden tomar cualquier valor dentro de un conjunto determinado, dependiendo de la forma que se haya dado a la misma. Así, en la figura 8 se muestran algunas de las funciones de activación más conocidas.

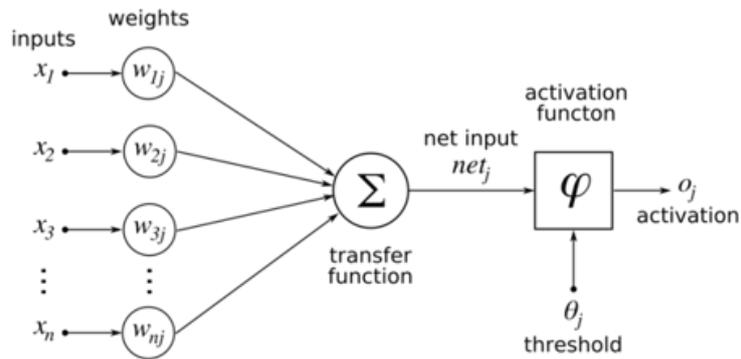
	Función	Rango	Gráfica
<b>Identidad</b>	$y = x$	$[-\infty, +\infty]$	
<b>Escalón</b>	$y = \text{sign}(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
<b>Lineal a tramos</b>	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq +l \\ +1, & \text{si } x > +l \end{cases}$	$[-1, +1]$	
<b>Sigmoidea</b>	$y = \frac{1}{1 + e^{-x}}$ $y = \text{tgh}(x)$	$[0, +1]$ $[-1, +1]$	
<b>Gaussiana</b>	$y = Ae^{-mx^2}$	$[0, +1]$	
<b>Sinusoidal</b>	$y = A \text{sen}(\omega x + \varphi)$	$[-1, +1]$	

**Figura 8:** principales funciones de activación; fuente: Palacio, F. (2003). Herramientas en GNU/Linux para estudiantes universitarios - Redes Neuronales con GNU/Linux. The Public's Library and Digital Archive. Online [Jun 2017]

Algunas de estas funciones merecen aquí un tratamiento especial. En particular, la función sigmoide es la más conocida por haber sido la más utilizada hasta la última década. La llegada del Big Data y la consecuente necesidad de aumentar el tamaño de las redes han hecho que este tipo de funciones hayan perdido popularidad dados los problemas que presenta la función sigmoide cuando se añade un gran número de capas. Es por ello que, en las Redes Neuronales de mayor profundidad, como es el caso de las Redes Convolucionales que serán tratadas en el siguiente apartado de este estudio, suele usarse la función de activación lineal rectificadora, más conocida como ReLU (*Rectified Linear Unit*).

En cuanto al aprendizaje, el algoritmo de aprendizaje del perceptrón simple pertenece al grupo de los algoritmos que se fundamentan en la corrección de errores. Los algoritmos de este tipo ajustan los pesos de manera proporcional a la diferencia existente entre la salida actual de la red neuronal y la salida deseada, con el objetivo de minimizar el error actual de la red. (Larranaga et al., 1997)

El Perceptrón Simple puede utilizarse tanto para tareas de clasificación como para la representación de funciones booleanas<sup>10</sup>, permite discriminar entre dos clases linealmente separables (el denominado problema OR), pero no entre clases linealmente no separables (el llamado XOR), lo que limita su aplicación práctica. (Minsky y Papert, 1969;1988)



**Figura 9:** Funcionamiento básico del perceptrón. Fuente: (March 14, 2013) Wikibooks: Artificial Neural Networks. [https://en.wikibooks.org/wiki/Artificial\\_Neural\\_Networks](https://en.wikibooks.org/wiki/Artificial_Neural_Networks)

### 2.3- El perceptrón multicapa

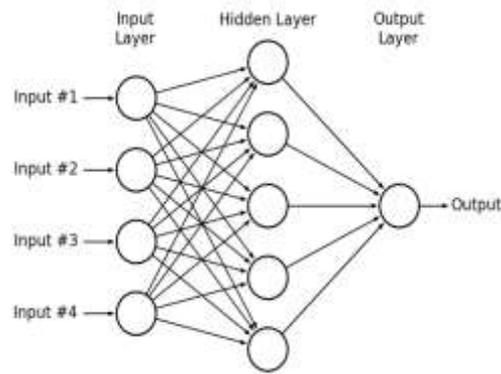
Tras las críticas de Minsky y Papert en su obra “*Perceptrons*”<sup>11</sup>, ante la imposibilidad del uso del perceptrón simple para afrontar problemas no lineales, pues se define como un elemento de cálculo universal booleano<sup>12</sup>, aparece el Perceptrón Multicapa o MLP (*Multilayer Perceptron*).

Este modelo se compone de 3 tipos de capas: la de entrada; donde se recibe la información del exterior o *inputs*, la de salida que envía el resultado hacia otras neuronas, y las llamadas capas ocultas (más conocidas como *hidden layers*), donde tiene lugar el procesamiento no lineal de los datos. Es gracias a la adición de varias capas que pueden solucionarse problemas de tipo no lineal.

<sup>10</sup> Función cuyo dominio está formado por valores binarios (0 y 1).

<sup>11</sup> 1967

<sup>12</sup> Algebra de 2 valores, que normalmente suelen ser “0” y “1”,



**Figura 10:** Ejemplo de un Perceptrón Multicapa. Pouliakis et al.(2016) Artificial Neural Networks as Decision Support Tools in Cytopathology: Past, Present, and Future. Biomedical Engineering and Computational Biology 2016:7 1–18 doi:10.4137/BECB.S31601.

Algunas investigaciones sugieren que la corteza cerebral puede ser vista como una arquitectura profunda que involucra de 5 a 10 capas durante el procesamiento de la información visual. (Fukushima, 1980) Aun cuando cada neurona sólo produce un único *output* (que dependerá de los distintos pesos), cada uno de ellos se transformará a medida que pase por las capas siguientes. Se habla incluso de que las capas ocultas (*hidden layers*) actúan como un proceso de filtrado, pues si no se supera el umbral de activación, el *output* producido en una determinada neurona no pasará a la capa siguiente y por tanto tampoco a la capa de salida.

El perceptrón multicapa añade una o más capas ocultas, en estas la función de activación es de tipo no lineal; inicialmente de tipo sigmoide, aunque en la capa de salida la función de activación puede ser lineal. Funahashi (1989) demostró que, si bien un perceptrón simple puede únicamente aproximar funciones booleanas, un MLP con una única capa oculta puede aproximar cualquier función continua. Es decir, se puede demostrar que, una red neuronal de solo 2 capas, con funciones de activación de tipo no lineal; como por ejemplo la función sigmoide, puede aproximar relaciones no lineales entre los datos de entrada y salida. El MLP es el modelo utilizado en las redes neuronales actuales, y es gracias a este que las redes neuronales se consideran aproximadores de función universal, más conocidas como “*universal function approximators*”.

## 2.4- El aprendizaje de las redes neuronales

Buscamos que la red neuronal sea capaz de resolver problemas similares, pero con distintos datos sin necesidad de intervención externa. Para ello deben encontrarse los

pesos que hagan que las predicciones difieran lo menos posible de la realidad y es en el proceso de aprendizaje o entrenamiento donde la red modifica estos pesos para que las soluciones ofrecidas por la red sean lo más ajustadas a la realidad posible.

Durante la fase de entrenamiento se mide la pérdida (diferencia entre el valor predicho y el valor real) y se buscan los pesos del modelo que hagan que esta tome el valor más bajo posible. Como resultado, una vez finalizado el entrenamiento, los pesos contendrán la información aprendida por la red. Por lo que podemos concluir que los criterios que determinan cómo se modifican los valores de los pesos son esenciales para el desarrollo de las ANN.

Uno de los factores más importantes del MLP y que puede apreciarse en la Figura 10 es que, todas las neuronas de una determinada capa ( $k$ ), se conectan con todas y cada una de las neuronas de la capa siguiente ( $k + 1$ ). Así, el método de aprendizaje debe tener en cuenta la contribución de cada una de estas al error. Para ello, se usa la llamada “*back-propagation*” o “retropropagación del error” que utiliza el algoritmo de descenso de gradiente para encontrar los pesos que hacen que la función de pérdida sea lo más próxima a 0 posible. (Voulodimos et al., 2017)

El algoritmo de descenso del gradiente busca de forma analítica el mínimo de toda función diferenciable. Dado que el mínimo de una función es un punto en el que la derivada es 0, se basa en encontrar aquellos puntos en los que la derivada llega a 0 y comprobar para cuál de ellos la función tiene el valor más bajo. Aplicado a una red neuronal, eso significa encontrar la combinación de valores de los pesos que produce la menor función de pérdida posible. Esto puede hacerse resolviendo la ecuación del gradiente ( $\text{gradient}(f)(W) = 0$  para  $W$ ). Se trata de una ecuación polinómica de  $N$  variables, donde  $N$  es el número de coeficientes de la red. (Chollet, 2018)

El algoritmo de retropropagación del error modifica sistemáticamente los pesos de las conexiones entre las neuronas de modo que la respuesta se vaya acercando cada vez más a la respuesta deseada: la que hace que el error sea lo más cercano a 0 posible.

El entrenamiento se lleva a cabo en dos etapas diferentes. En primer lugar, se calculan todas las activaciones de las neuronas de la red, tomando para los pesos sinápticos valores por defecto, es decir, aleatorios. A continuación, la respuesta de la red se compara con la realidad, obteniéndose el error global de la misma. El error que se ha calculado se propaga

en dirección opuesta a la de las conexiones sinápticas para así distinguir la aportación de cada una de las partes de la red al error denominado aquí “global”.

### **3.-Redes Neuronales Convolucionales:**

Una vez explicado el funcionamiento básico de las ANN, pasamos a adentrarnos en el desarrollo de redes neuronales profundas. Mayores volúmenes de datos requieren también la adición de capas que ayuden a una mejora de lo que hemos denominado filtrado en apartados anteriores. Esta mayor profundidad de las redes conlleva dificultades relacionadas con el entrenamiento de la red, que han inspirado la creación en los últimos años -con una tendencia especialmente alta desde 2012- de nuevos modelos, principalmente dedicados al reconocimiento de imágenes.

Entre los factores más destacados que han contribuido al enorme impulso del aprendizaje profundo se encuentra la aparición de grandes conjuntos de datos etiquetados de alta calidad y disponibles públicamente, junto con la potenciación de la computación paralela en la GPU (*Graphics Processing Unit*), que ha permitido la transición de un entrenamiento basado en la CPU (*Central Processing Unit*) a otro basado en la GPU, permitiendo así una aceleración significativa en el entrenamiento de los modelos profundos. (Voulodimos et al., 2017) Una CPU realiza el procesamiento a través de núcleos de manera secuencial, mientras que una GPU cuenta con una arquitectura en paralelo diseñada que permite resolver varias tareas al mismo tiempo.

Muchas observaciones sugieren que el entrenamiento basado en el método del gradiente descendente para redes neuronales profundas supervisadas se estanca en lo que ha sido llamado mínimo local aparente. En general, a pesar de la mejor representación obtenida utilizando una arquitectura profunda, se obtienen peores resultados que con soluciones obtenidas por medio de arquitecturas con una sola capa de neuronas ocultas. (Yu Liu, 2014) Hasta antes de 2006, las arquitecturas profundas no habían sido objeto de discusión dentro del ámbito científico debido a los pobres resultados que se habían tenido en su proceso de aprendizaje. Este comportamiento era generalmente obtenido utilizando una inicialización aleatoria de los pesos sinápticos de la red. Los únicos casos de éxito que habían sido reportados hasta ese momento habían sido las llamadas redes convolucionales. (Lee et al., 2009)

Los primeros modelos computacionales basados en estas conectividades locales entre neuronas y en transformaciones de la imagen organizadas jerárquicamente se encuentran en Neocognitron<sup>13</sup>. Cuando las neuronas con los mismos parámetros se aplican sobre parches de la capa anterior en lugares diferentes, adquiriéndose una forma de invariancia traslacional. (Fukushima, 1980) Posteriormente, autores como Lecun (1989) comienzan a diseñar redes neuronales convolucionales que emplean el gradiente de error y obtienen muy buenos resultados en una variedad de tareas de reconocimiento de patrones. Por lo que se llega a la conclusión de que las redes profundas pueden obtener un gran éxito, pero con una arquitectura de la red distinta que la que se venía desarrollando con los modelos más simples (con menos capas).

Los primeros modelos de redes neuronales convolucionales se basan en el sistema visual animal. Fue a raíz de la obra de Hubel y Wiesel (1962) que se empezaron a desarrollar redes inspiradas en la arquitectura del córtex visual animal. Estas redes utilizan el ya explicado algoritmo de descenso del gradiente, pero permiten el desarrollo de modelos profundos con un error aceptable gracias a la inclusión de diferentes tipos de capas; cada una de ellas con diferentes funciones dentro de la red. Actualmente, los sistemas de reconocimiento de patrones basados en redes neuronales convolucionales son los que tienen un mejor rendimiento y los que se desarrollan con mayor frecuencia.

### **3.1-Tipos de capas**

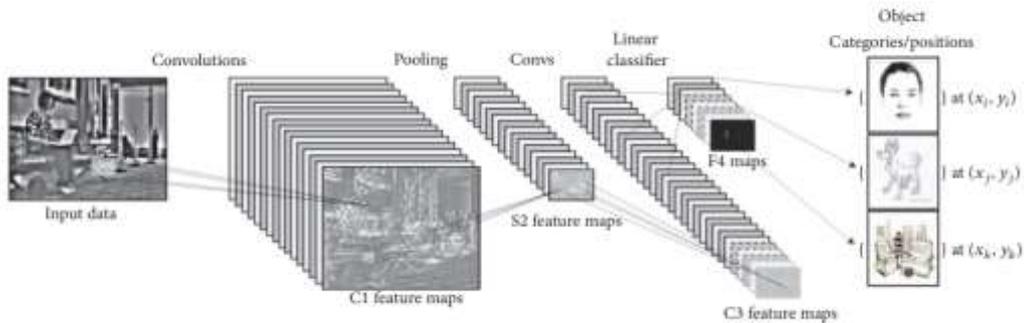
Las Redes Neuronales Convoluciones, en adelante CNN (*Convolutional Neural Networks*), contienen varias capas ocultas especializadas y jerarquizadas: esto significa que las primeras capas detectan propiedades o formas básicas y se van especializando hasta llegar a capas más profundas capaces de reconocer formas complejas como un rostro o una silueta (Moreno, 2019)<sup>14</sup>.

Una CNN comprende tres tipos principales de capas neuronales: (i)convolucionales, (ii)de agrupación o “*pooling*”, y (iii)las llamadas capas totalmente conectadas. La entrada se produce a través de la recepción de una imagen, tomando los píxeles de esta como *inputs* de la red. Las dimensiones de la imagen podrían resumirse como  $m \times n \times r$ , donde  $m$

---

<sup>13</sup> K. Fukushima, “Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

es la altura,  $n$  es el ancho de la imagen y  $r$  es el número de canales; que indica el color de la imagen, siendo  $r=1$  en caso de se trabaje en una escala de grises y  $r=3$  para las imágenes en color, indicando la presencia de tres canales distintos (rojo, verde y azul).



**Figura 11:** Ejemplo de la arquitectura de una CNN para el reconocimiento de imágenes. Fuente: Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

#### A) Capas convolucionales

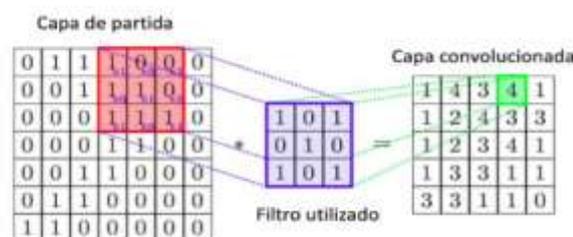
Una vez se produce la entrada en la red, las capas convolucionales se encargan de reducir la carga de trabajo mediante el proceso de convolución. A través de este, cada neurona se conecta únicamente a una región de la imagen y no a la imagen completa.

Durante esta etapa de la red se lleva a cabo una solución muy simple: se restringe el número de conexiones posibles entre las neuronas de la capa oculta y los elementos de la imagen de entrada. De esta manera, cada neurona oculta solo estará conectada con un pequeño subconjunto de elementos de la imagen total. Esta idea está inspirada en cómo funciona el sistema visual biológico, debido a que las neuronas que trabajan en el córtex visual poseen una serie de campos receptivos que responden sólo ante unos estímulos localizados en una región o área específica. (Durán, 2017)

Cada una de estas capas convolucionales contiene filtros, más conocidos como *kernels*, y es a través de estos que se lleva a cabo la convolución. Estos filtros se encargan de convertir los datos de entrada en “mapas de rasgos o características”, que facilitan el tratamiento de la imagen dentro de la red. La operación es simple: se multiplican los elementos del filtro por los elementos de la región, posición por posición y finalmente se suman.

No se aplica un único filtro sino que existe un conjunto de ellos, que da lugar a lo que recibe el nombre de *feature mapping*. La composición de este feature mapping es determinada por el programador dependiendo del número de filtros que éste decida incluir. El número de filtros no es el único parámetro controlable, sino que también lo son los llamados “*stride*” (número de píxeles que se desplazará el filtro sobre la imagen) y “*padding*” (referido al tratamiento del borde de la imagen)<sup>15</sup>.

Cada uno de estos filtros recorre la imagen; de izquierda a derecha y de arriba hacia abajo, y genera una nueva matriz de salida, que será la nueva capa de neuronas ocultas, y que también se conoce como matriz de activación.



**Figura 12:** Uso de un filtro de dimensiones 3x3. Fuente: Calvo, D (20 de julio de 2019). Red Neuronal Convolutacional. CNN <http://www.diegocalvo.es/red-neuronal-convolutacional>

La operación de convolución consiste en superponer un filtro sobre una imagen e ir desplazando/deslizándolo el filtro sobre la misma hasta haber recorrido así todas las regiones de la imagen (todos los píxeles de la imagen han de haber sido ‘visitados’ por el filtro en algún momento durante el proceso). (Erroz, 2019) Una de las principales características que aporta la capa convolutacional es que permite extraer rasgos comunes de la imagen en distintas zonas de la misma. Dado que se aplica el mismo filtro en todas las zonas de la imagen y, al mismo tiempo, teniendo en cuenta la localización de cada una de ellas. Esta nueva imagen, acompañada por un sesgo independiente, y tras haber pasado por una función de linealidad - tal y como se explicó en las redes convencionales- pasará a la siguiente capa.

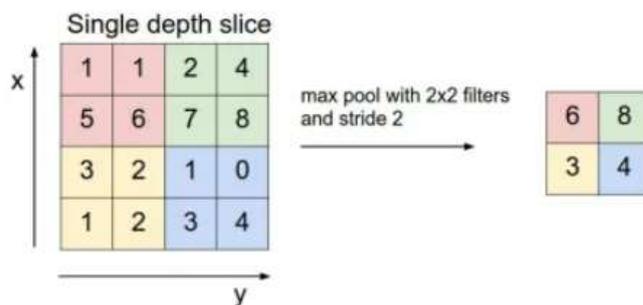
## B) Capas de submuestreo

<sup>15</sup>Los bordes de la imagen pueden ser ignorados o incluidos en la convolución en forma de ceros.

A continuación, las capas de submuestreo o agrupamiento se encargan de reducir las dimensiones espaciales (anchura  $\times$  altura) o (“ $m \times n$ ”) para la siguiente capa, que será de nuevo convolucional.

La operación realizada en esta capa se denomina submuestreo pues, aunque la reducción de tamaño conlleva una pérdida simultánea de información, esta pérdida es beneficiosa para la red. La disminución del tamaño conlleva una menor sobrecarga computacional para las siguientes capas de la red, y también actúa contra el *overfitting*<sup>16</sup> del modelo. (Voulodimos et al., 2017).

Los dos métodos más utilizados para llevar a cabo el “*pooling*” de la red son “*average pooling*” y “*max pooling*”. En el caso del *average pooling* o muestreo medio, se toma el promedio de los elementos que forman cada región, mientras que en el *max pooling*, se toma el valor máximo de cada una de ellas.



**Figura 13:** Ejemplo aplicación de submuestreo mediante el método max pooling. Fuente: datscientest.com

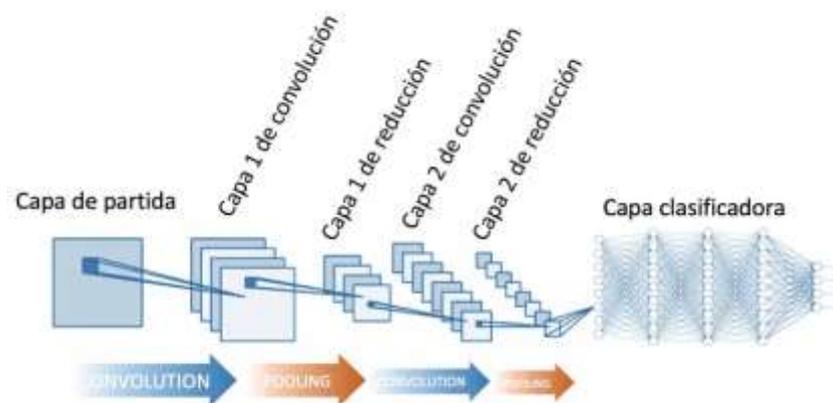
### C) Capas “fully connected”

Este tipo de capa se comporta como las redes neuronales convencionales, pues todas las neuronas están conectadas con todas las de la capa anterior; que será de tipo convolucional. Dado el ya explicado funcionamiento de las capas convolucionales, las neuronas de la capa *fully connected* se conectan a los mapas de características creados tras una o varias convoluciones y submuestreos. Así, la red neuronal deja de estar conectada por zonas o regiones.

<sup>16</sup> Sobre adaptación del modelo a los datos de entrenamiento

Tras varias capas convolucionales y de agrupación, el razonamiento de alto nivel en la red neuronal se realiza a través de capas totalmente conectadas. Las neuronas de una capa totalmente conectada tienen conexiones completas con todas las activaciones de la capa anterior, como su nombre indica. Su activación puede calcularse con una multiplicación matricial seguida por un sesgo de conjunto. Las capas totalmente conectadas acaban convirtiendo los mapas de características 2D en un vector de características 1D (Voulodimos, 2017).

La función principal de las capas completamente conectadas es llevar a cabo una especie de agrupación de la información que se ha obtenido hasta ese momento, que servirá en cálculos posteriores para la clasificación final. En general, en las redes de neuronas convolucionales, se suele utilizar más de una capa completamente conectada en serie y la última de ellas tendrá el parámetro K que es el número de clases que se encuentran presentes en el conjunto de datos. Los valores finales de K serán alimentados a la capa de salida, que a través de una cierta función probabilística realizará la clasificación. (Bonilla, 2020)



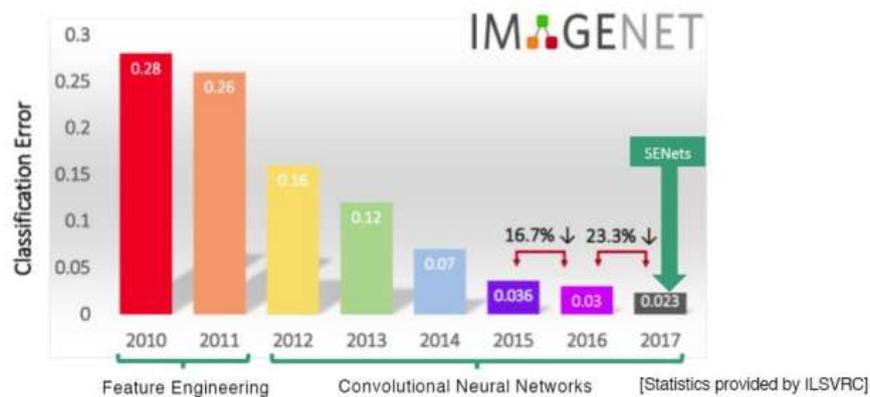
**Figura 14:** Arquitectura de una CNN. .Fuente: Calvo, D (20 de julio de 2019). Red Neuronal Convolucional. CNN <http://www.diegocalvo.es/red-neuronal-convolucional>

Recapitulando todo lo explicado hasta este punto podemos hacernos una idea de cómo se estructuran las CNN: A partir de una imagen de la que se toman los datos de entrada de la red, cada una de las neuronas de la capa convolucional se encarga de una determinada zona de la misma, recibiendo datos únicamente de esta. En estas primeras capas convolucionales se extraerán patrones más sencillos como curvas o esquinas, pero a

medida que se van aplicando las capas *pooling* o submuestreo que simplifican y reducen la red, las capas convolucionales serán capaces de diferenciar cada vez formas más complejas. La extracción de patrones en distintas zonas de la imagen a través de la aplicación del mismo filtro es elemental para comprender la organización y utilidad de las CNN. Finalmente, la información vuelve a unificarse gracias a las capas *fully connected* que envían la información a la capa de salida.

### 3.2-Arquitectura de Redes Convoluciones

Aun así, existen en la actualidad distintas posibilidades a la hora de construir una red de neuronas convolucionales. Los avances en cuanto a la evolución en la arquitectura de las CNN y, por ende, la mejora del reconocimiento de imágenes, se lo debemos en gran medida al desafío anual ILSVRC (*ImageNet Large Scale Visual Recognition*). Se trata de una competición que busca premiar el mejor modelo de clasificación del mayor conjunto de datos categorizados existente hasta el momento: *ImageNet*. Desde 2012, las CNN han protagonizado el desafío disminuyendo considerablemente la tasa de error generada por los modelos anteriores.



**Figura 15:** Evolución de la tasa de error en el desafío ILSVRC; Fuente: Tsang, 2018b

Se exponen a continuación algunas de las arquitecturas CNN ganadoras de este desafío y que se consideran los grandes hitos del estado del arte desde 2012 hasta la actualidad.

**Tabla 1 - Arquitecturas de Redes Neuronales Convolucionales Profundas**

Algoritmos	Nº de Capas	Índice de Error	Posición en Competencia ILSVRC	Año
AlexNet	8	15.3 %	Primer Lugar	2012
VGGNet	19	7.3 %	Segundo Lugar	2014
GoogleNet	22	6.67 %	Primer Lugar	2014
ResNet	152	3.57 %	Primer Lugar	2015

**Figura 16:** Resumen principales arquitecturas Redes Neuronales Artificiales profundas. Fuente: Chanampe, H., Aciar, S., Vega, M. D. L., Molinari Sotomayor, J. L., Carrascosa, G., & Lorefice, A. (2019). Modelo de redes neuronales convolucionales profundas para la clasificación de lesiones en ecografías mamarias.

-AlexNet: En 2012 AlexNet es la primera red convolucional en ganar la competición, tal y como se puede ver en la figura 16, el error respecto a los modelos anteriores se reduce considerablemente. En concreto, el modelo estaba compuesto por 5 capas convolucionales, 2 capas de submuestreo y 3 capas totalmente conectadas dispuestas de la siguiente forma:

La primera capa de convolución tiene 96 filtros de tamaño 11x11x3. Las dos primeras capas van seguidas de una capa de max-pooling. La tercera, cuarta y quinta capa convolucional, están directamente conectadas, mientras que la quinta capa le sigue una de *max-pooling*. Luego vienen las tres capas *fully connected*, donde la última capa aplica la función *softmax* (clasificadora) para realizar la predicción. Después de cada capa de convolución y capa totalmente conectada se aplica la función no lineal ReLU. (Moreno, 2020)

-VGGNet: Red finalista en 2014 que ha ganado una gran importancia, pues agrupa bajo su nombre distintas variantes muy usadas. Se trata de una red similar a AlexNet, pero con una profundidad mucho mayor que la de las arquitecturas anteriores. En concreto está formada por 19 capas convolucionales con filtros de tamaño 3x3.

-GoogLeNet: La red propuesta por Google fue la ganadora de 2014 e introduce una novedad en el estado del arte: la inclusión de módulos Inception para realizar las operaciones de convolución y filtrado en paralelo y no de manera secuencial. Además, esta arquitectura, a diferencia de sus predecesoras, prescinde de la capa totalmente conectada.

-ResNet: Microsoft se hizo con la victoria en 2015 gracias a su diseño de las *residual networks* o redes residuales. Estas proponen una solución a la degradación, que es el efecto negativo que puede surgir en las redes demasiado profundas. Se diseñan una serie de bloques residuales capaces de saltar algunas capas, de manera que lo que aprenden las capas intermedias es una función residual, en otras palabras, solo una parte de los datos que entran a la red. Las capas intermedias ajustan sus pesos con valores próximos a 0 si no existen características suficientemente novedosas y pasan a ser lo que se ha denominado como “capa identidad”.

## 4. Redes neuronales recurrentes

El siguiente tipo de red neuronal que merece la pena destacar en un trabajo como este son las Redes Neuronales Recurrentes. Aunque la aplicación de este tipo de redes al campo de la salud no sea tan espectacular como el de las Redes Convoluciones, debemos destacar la importancia de estas para el análisis de textos y el tratamiento de datos secuenciales. Ambas características, tal y como se verá en el siguiente apartado, han sido también aplicados para el tratamiento de datos sanitarios.

Siguiendo con nuestra línea comparativa de las posibilidades que ofrece el *DeepLearning* y las capacidades humanas, podríamos decir que las Redes Neuronales Recurrentes aportan un valor añadido pues tienen en cuenta lo que podemos llamar factor tiempo.

El concepto de memoria no es ajeno a los sistemas tradicionales, pero es gracias a la secuencialidad que la memoria no queda almacenada en *containers* ajenos los unos de los otros, sino que, a través de ella, se consigue extraer relaciones entre los datos recibidos. Cuando los seres humanos leemos somos capaces de enlazar las palabras de una misma frase para concluir el sentido de la misma, o incluso datos más alejados cuando seguimos la trama de una novela. Las RNN cuentan con infinidad de posibles aplicaciones, pero resulta especialmente interesante su uso en tareas relacionadas con el lenguaje; frecuentemente se usa para la clasificación de textos, traducciones o *sentiment analysis*<sup>17</sup>.

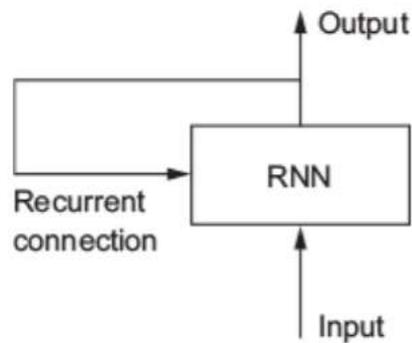
Así, de aquí en adelante nos referiremos a las redes *feedforward* para hablar de las redes neuronales tratadas hasta este punto del estudio. Sin embargo, en lo que respecta a las Redes Recurrentes, debemos tener en cuenta un elemento que hasta ahora no había intervenido: el *time-step* o instante del tiempo. En cada *time-step* cada capa recibe dos entradas: la entrada de la capa anterior -tal y como sucede en las *feedforward*- pero también la salida del instante anterior de la misma capa (t-1).

Las RNN también reciben una serie de *inputs* y generan, una vez tratados, un *output*, con la especial y esencial diferencia de que, en las neuronas de estas, el *output* de un *time-step* se pasa al siguiente, creando un bucle que permite retener información pasada. En otras palabras, una RNN se retroalimenta mediante la detección de dependencias en

---

<sup>17</sup> Procesamiento del lenguaje natural para la extracción la intención subjetiva

variables de gran longitud gracias a la conservación de la información. El *output* del modelo no sólo es ahora una función de unos determinados inputs sino también de la memoria de la red en si misma.



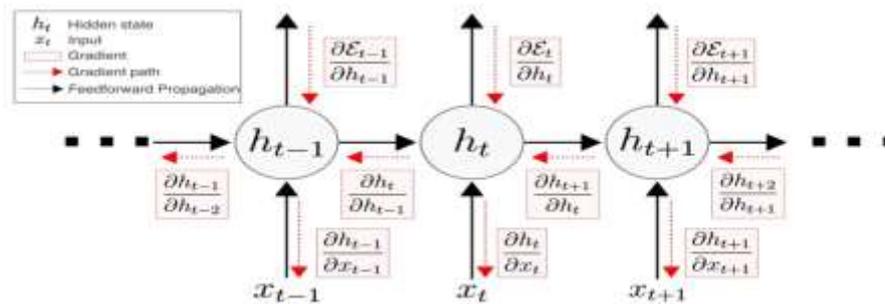
**Figura 17:** Esquema simplificado funcionamiento RNN. Fuente: François Chollet. (2018). DeepLearning with R. United States of America: Manning Publications.

Tal y como podemos observar en la figura 17, se trata de lo que algunos expertos han denominado “bucle interno” que permite la retroalimentación. El estado de la RNN (*Recurrent Neural Network*) se restablece entre el procesamiento de dos secuencias diferentes e independientes, por lo que se sigue considerando una secuencia como un único punto de datos: una única entrada a la red. Lo que cambia es que este punto de datos ya no se procesa en un solo paso, sino que la red hace un bucle interno sobre los elementos de la secuencia (Chollet, 2018).

#### ***4.1 Backpropagation through time***

El método de aprendizaje *backpropagation* fue ya introducido en este trabajo como el método de aprendizaje de las redes multicapa que, tiene en cuenta el error que cada una de las neuronas de capas anteriores aportan a la red. Tal y como fue explicado, durante el entrenamiento de estas se usa el algoritmo del descenso del gradiente para, a través de las derivadas parciales del error, modificar los pesos de la red. Sin embargo, nos encontrábamos ante lo que llamaremos ahora *feed-forward networks*. En las redes neuronales recurrentes debemos tener en cuenta el factor tiempo, puesto que las neuronas no solo se ven afectadas por la neurona anterior, sino que la pérdida de un determinado *time-step* depende del *time-step* precedente.

Se introduce así el método de BPTT (*Back Propagation Through Time*) donde la propagación del error se lleva a cabo no solo desde la pérdida total a cada neurona, sino también desde cada *time-step* hasta el principio, calculando la función de pérdida de cada instante de tiempo para ajustar los pesos.



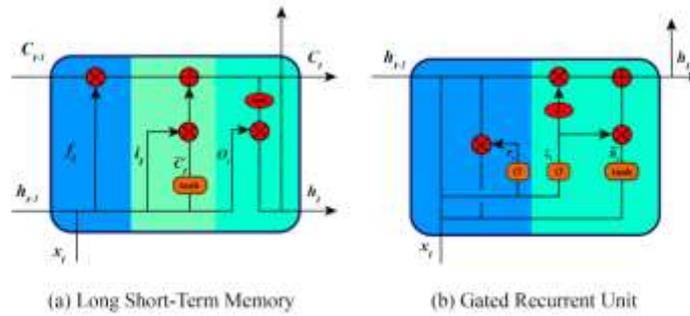
**Figura 18:** *Funcionamiento del backpropagation through time (BPTT)* Fuente: Backpropagation through time and the brain, Por T.P. Lillicrap y A. Santoro, 2019, Current Opinion in Neurobiology, Volumen 55, pág 83

## 4.2 Gated cells

Las redes neuronales recurrentes, como toda red neuronal profunda, cuentan con problemas a la hora de computar los gradientes<sup>18</sup> para el *backpropagation*. Este problema se puede afrontar de distintas maneras, pero la forma más eficiente es el uso de las llamadas *gated cells*.

En definitiva, se trata de conservar únicamente la información relevante, para así permitir que se vaya incorporando la información nueva. Dos son los principales modelos mediante los que se desarrolla el método de *gated cells*: LSTM (*Long Short Term Memory*) y GRU (*Gated Recurrent Unit*).

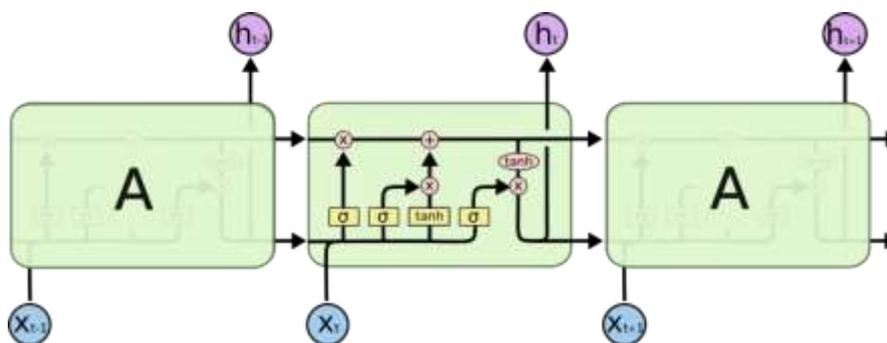
<sup>18</sup> Los dos principales problemas son el desvanecimiento del gradiente y la explosión del gradiente



**Figura 19:** Arquitectura de LSTM (a) y GRU (b). Fuente: L. Zhang, M. Wang, M. Liu, and D. Zhang, “A survey on deep learning for neuroimaging-based brain disorder analysis,” arXiv preprint arXiv:2005.04573, 2020.

A diferencia de las redes neuronales recurrentes al uso, en LSTM (*Long Short Term Memory Networks*) cada neurona no cuenta únicamente con las entradas *–input* en el instante  $t$  y memoria pasada– y una función de activación, sino que cuentan con tres puertas: *Input Gate*, *Forget Gate* y *Output Gate*. Estas puertas, con activaciones de tipo sigmoide, eliminan la información irrelevante de *time-steps* previos, enviando al siguiente *time-step* únicamente aquella información que el modelo estime relevante.

La *Input Gate* trabaja con la información actual y la memoria a corto plazo que llega del *time-step* previo. La *Forget Gate* filtra la información de la memoria a largo plazo. Y finalmente es la *Output Gate* la que tomará el *input* actual, la memoria a corto plazo y la parte de la memoria a largo plazo que ha sido seleccionada para volver a producir la memoria a corto plazo que, ahora sí, pasa al siguiente *time-step*.



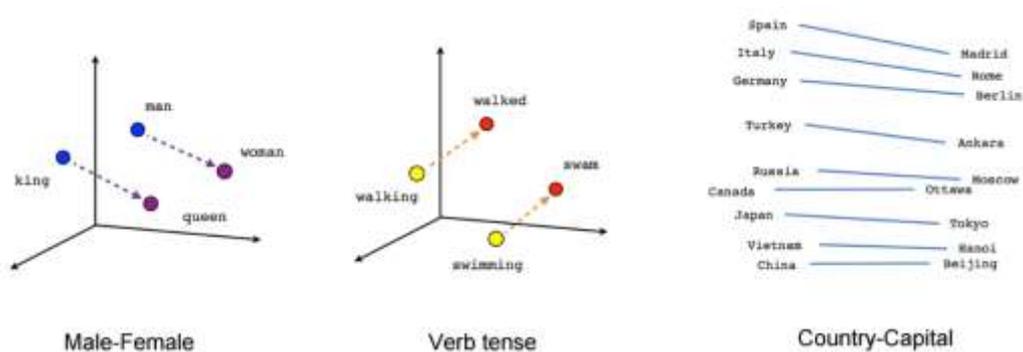
**Figura 20:** Funcionamiento del algoritmo LSTM. Fuente: Colah’s Blog. *Understanding LSTM Networks*.

Por otro lado, existe el concepto de GRU o *Gated Recurrent Unit*, en este modelo no se incorporan las llamadas *memory cells*, sino que el control de la información se realiza a través de dos tipos de puertas: La *update gate* o puerta de actualización, que selecciona la información pasada relevante -la que continuará retroalimentando al modelo. Y, al contrario, la *reset gate*, que decide qué parte de la información pasada sale del modelo o se olvida. A diferencia que en el caso de LSTM, la *reset gate* se aplica directamente en la información que llega de la capa anterior.

### 4.3 Codificación del lenguaje

Las principales aplicaciones de este tipo de redes se encuentran en el procesamiento del lenguaje, para ello es necesaria una labor previa de codificación, que no es más que la transformación de los textos a código numérico, para que el modelo pueda llevar a cabo el tratamiento la información.

Para ello, es necesario vectorizar los datos, siendo las dos formas más comunes de vectorización el *One-hot Encoding* y el *Word Embedding*. El *One-hot Encoding*, transforma las frases en vectores binarios, asignando el valor 1 o 0 dependiendo de si la frase en cuestión cuenta con palabras de un vocabulario N. Los modelos *Word Embedding*, aunque conllevan una mayor carga computacional, vinculan cada palabra a un vector, y estos estarán más o menos próximos según su significado.



**Figura 21:** Relaciones creadas a partir del modelo GloVe2 (modelo de *Word Embedding* desarrollado por la Universidad de Stanford). Fuente: *tensorflow.org*

## 4.4 Tipos de redes neuronales recurrentes

La clasificación de tipos de RNN se realiza según cómo se estructuren los *inputs* y *outputs* de estas. Son cuatro los modelos más conocidos en el actual estado del arte y que serán brevemente explicados en esta revisión.

El llamado *one to many*, cuenta con un único input y múltiples outputs. En este caso el input no tiene componente temporal. La estructura *one to many* se suele dar en tareas de descripción de imágenes. (Soleimany y Amini, 2021) El ejemplo paradigmático de *one to many* es el de generación de canciones, capaz de armar una canción siendo el input una única nota.

En el caso del *many to one*, múltiples inputs generan un único output. Este es el tipo de estructura común en los ya mencionados modelos de *sentiment analysis*.

Por último, en los modelos *many to many* múltiples inputs generan a su vez múltiples outputs. Un ejemplo paradigmático de este tipo de estructura serían los modelos de traducción, siendo los inputs las palabras de una frase en el idioma A y los outputs su traducción en el idioma B. (Soleimany y Amini, 2021)

## **5.-Aplicaciones actuales y futuro cercano en el campo de la salud**

Las dos técnicas principales de redes neuronales, las convolucionales y las recurrentes (CNN y RNN, respectivamente), han sido hasta este punto del estudio explicadas. Una vez introducida una perspectiva básica de su funcionamiento, doy paso a lo que podría considerarse la segunda gran sección, o subobjetivo de este Trabajo de Fin de Grado: el análisis de las aplicaciones de la Inteligencia Artificial -concretamente del *DeepLearning*- en el campo de la medicina.

La Inteligencia Artificial se creó para estar al servicio del hombre, y cada día, a través del desarrollo de nuevas aplicaciones, se trata de un instrumento más importante y presente en nuestras vidas. En este apartado del trabajo, pasaré a explicar algunas de las aplicaciones que a día de hoy –pues se espera que lleguen muchas más- buscan una mejora de la asistencia sanitaria a través del desarrollo de RNA.

La importancia del uso del DL para la salud es indudable. En los métodos tradicionales de ML, la necesidad de tratamiento previo de los datos ha venido dificultando hasta la actualidad los avances en la construcción de modelos predictivos en un sector de importancia tal como es este. Pero también el DL se ha servido de la medicina para llevar a cabo grandes avances. Ya se introdujo que, los sistemas multicapa necesitan de una gran cantidad de datos para obtener buenos resultados, y estos se sirven de los mismos a través del sistema sanitario, no solo por la gran cantidad de datos que genera este sector, sino también por la heterogeneidad de los mismos. Estos datos van desde algunos de tipo texto, como por ejemplo los antecedentes familiares de un paciente, numéricos como los de niveles de glucosa en sangre, u otros tan distintos como las imágenes 3D o las ondas ultrasonido.

Los datos médicos a gran escala (en forma de bases de datos estructuradas y no estructuradas) si son apropiadamente adquiridos e interpretados pueden generar grandes beneficios al reducir los costos del servicio de salud (lo cual ya en la actualidad se está utilizando), pero también podrían servir para predecir epidemias, mejorar los esquemas terapéuticos, asesorar a médicos en lugares remotos y mejorar la calidad de vida de pacientes. (Arias et al., 2019)

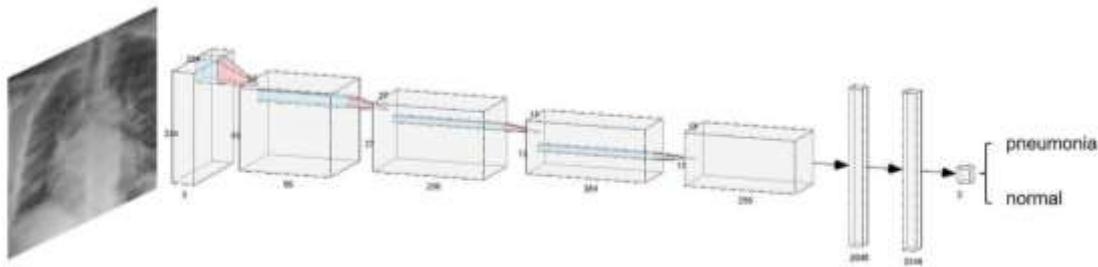
Sus propiedades como reconocedores de patrones altamente tolerantes a errores permiten combinar las cualidades propias del razonamiento humano con la lógica precisa y la gran capacidad de memoria que poseen los computadores, de esta manera resultan de gran utilidad en medicina como sistemas de apoyo a la toma de decisiones clínicas. La toma de decisiones es un tema importante en la práctica médica, tanto en el proceso diagnóstico como en el terapéutico. (Trujillano et al., 1995)

A continuación, serán brevemente explicadas algunas de las aplicaciones más exitosas tanto del reconocimiento de imágenes, como del procesamiento de lenguaje, en el campo del *Healthcare*. A día de hoy, podemos decir que el DL es la técnica más fructífera para mejorar la salud humana a través de la tecnología e intentaré exponer de la forma más clara posible ejemplos justificativos de esta afirmación.

### **5.1- Procesamiento de imágenes: diagnóstico y predicción**

Las primeras aplicaciones de las RNA en datos clínicos se realizaron a través del procesamiento de imágenes. Se han obtenido grandes éxitos a través de tareas de clasificación de imagen, *target recognition* y *target segmentation*. En estas tareas intervienen muchos aspectos, como la detección de la retinopatía, la edad de los huesos, la identificación del cáncer de piel, etc. El aprendizaje profundo alcanza el nivel de experto en estas tareas y la red neuronal convolucional es un potente método de aprendizaje profundo. (Yang et al., 2021)

Las CNN comenzaron a implementarse para el reconocimiento de imágenes a través de las tareas de clasificación, y de hecho sigue siendo esta su principal aplicación. Se trata de modelos que utilizan como inputs imágenes clínicas para la obtención de un determinado diagnóstico que podría resumirse básicamente en clasificar si la patología objeto de estudio está presente o no en el paciente del que se toma la imagen. En el ejemplo de la figura 22, se aplica una red convolucional a una radiografía de tórax, y una vez todas las capas han sido aplicadas, el modelo clasifica si el paciente padece o no neumonía.



**Figura 22:** Red Neuronal Convolutiva para la detección de neumonía. Fuente: Yang S, Zhu F, Ling X, Liu Q, Zhao P. Intelligent Health Care: Applications of Deep Learning in Computational Medicine. Front Genet. 2021 Apr 12;12:607471. doi: 10.3389/fgene.2021.607471. PMID: 33912213; PMCID: PMC8075004.

Los primeros trabajos que aplican estas técnicas para clasificación de imágenes aparecieron en 2013 y se centraron en la neuroimagen. (Litjnes et al., 2017) Uno de los modelos pioneros y paradigmáticos es el análisis de resonancias magnéticas para el diagnóstico de Alzheimer, más conocido como MRI (*Magnetic Resonance Imaging*).

### A. Detección de lesiones cutáneas

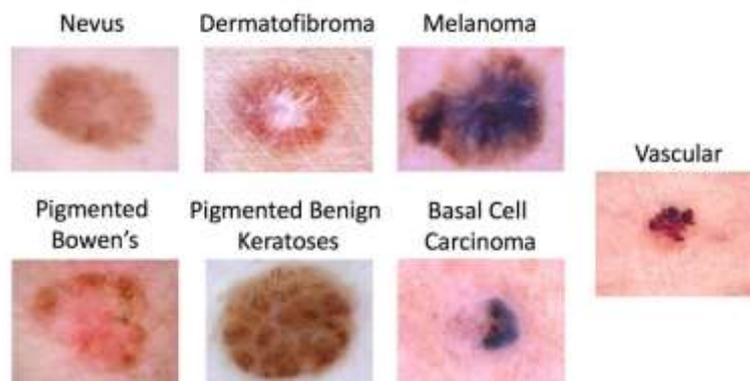
El cáncer de piel es la afección maligna más común en el ser humano y se diagnostica principalmente de forma visual, comenzando con una exploración clínica inicial y siguiendo potencialmente con un análisis dermatoscópico, una biopsia y un examen histopatológico. (Esteva et al., 2017) Es por ello que, el desarrollo de un programa de reconocimiento de imágenes efectivo es tan importante. Se muestran aquí un ejemplo de modelo desarrollado con tal finalidad.

La clasificación automatizada de las lesiones cutáneas a partir de imágenes es una tarea difícil debido a la casi inapreciable variabilidad del aspecto de las lesiones cutáneas, dificultando la distinción entre las benignas y las cancerosas. Sin embargo, las CNN han demostrado un fuerte potencial para este tipo de tareas.

El estudio en concreto del que se va aquí a hablar<sup>19</sup> demostró la exactitud en la clasificación de lesiones cutáneas mediante el uso de una única CNN. En concreto se utilizaron 129.450 imágenes clínicas, que contenían 2.032 enfermedades diferentes etiquetadas.

<sup>19</sup>Dermatologist-level classification of skin cancer with deep neural networks (2017).

La finalidad era la distinción por medio del modelo de carcinomas de queratinocitos (forma más común de cáncer de piel) frente a queratosis seborreicas (que se trata de neoplasias cutáneas benignas muy frecuentes a partir de una determinada edad). Así como de melanomas malignos (cáncer de piel con mayor tasa de mortalidad) frente a nevus benignos (otro tipo de neoplasia no cancerígena).



**Figura 23:** Imágenes reales de lesiones cutáneas, que tal y como puede apreciarse carecen de diferencias fácilmente apreciables por el ojo humano. Fuente:Dataart.com

El motivo para elegir este estudio y no otro, es que las imágenes que se toman no son imágenes que deban ser tomadas en un laboratorio, sino que las conclusiones del estudio preveían que en un futuro muy próximo se podría incluso llegar a detectar la presencia o no de este tipo de lesiones cutáneas a través del uso de teléfonos móviles inteligentes, lo que podría dar lugar tanto a una reducción de diagnósticos tardíos como al acceso universal a un tratamiento de calidad.

### **B. Camelyon 16 y detección de metástasis en cáncer de mama**

El Camelyon (*Cancer Metastases in Lymph Nodes Challenge*) es un concurso organizado por El Simposio Internacional sobre Imágenes Biomédicas (ISBI), donde se reta a grupos de investigadores de cualquier país a crear la mejor solución automatizada para la detección de metástasis en cáncer de mama a través de imágenes de biopsias de ganglios linfáticos centinela, en adelante SLN (*Sentinel Lymph Node*)<sup>20</sup>. A continuación, en este apartado se explicará tanto el modelo como la *performance* del ganador de la edición Camelyon16.

---

<sup>20</sup> En el cáncer de mama, la metástasis se produce principalmente a través del sistema linfático. Los ganglios linfáticos más cercanos a un tumor se denominan ganglios linfáticos centinela.

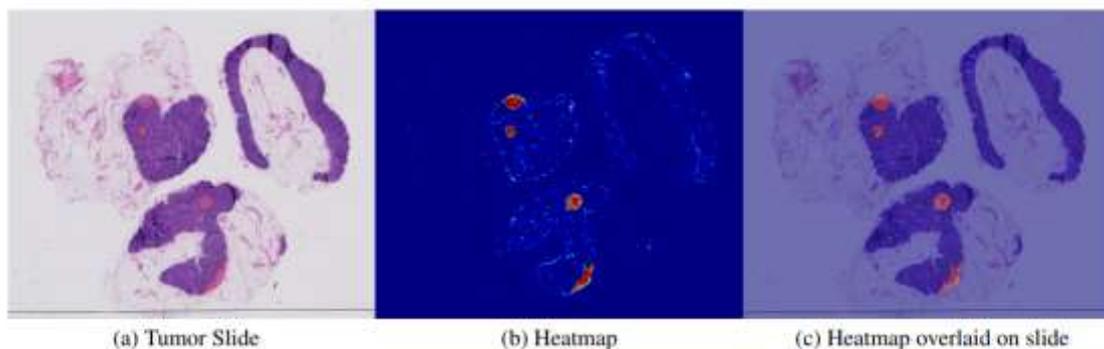
El dataset de Camelyon16 constaba de un total de 400 imágenes, tomadas de pacientes que pertenecían a dos hospitales distintos de Holanda. De estas, 270 debían destinarse al entrenamiento del modelo, dejando las 130 restantes para medir la performance conseguida.

Respecto al objetivo de la métrica para determinar qué modelo realizaba las mejores predicciones, se fijaron dos evaluaciones: (i) “*Slide-based Evaluation*”, que consistía en realizar una clasificación que diferenciara aquellas imágenes que contenían metástasis de, de las que no y (ii) “*Lesion-based Evaluation*”, en la que debía establecerse la localización de la lesión cancerígena en la imagen.

De los 32 algoritmos presentados, 25 utilizaban redes neuronales convolucionales profundas y, de hecho, los mejores algoritmos habían utilizado todos este tipo de redes. El algoritmo que obtuvo los mejores resultados en ambas métricas fue el creado por el equipo formado por investigadores de Harvard y del MIT (*Massachusetts Institute of Technology*).

El modelo ganador estaba creado a partir de 2 etapas principales: la clasificación basada en parches y la posterior aplicación de mapas de calor.

Durante el entrenamiento, esta etapa utiliza como entrada parches de 256x256 píxeles y entrena un modelo de clasificación para discriminar entre los parches positivos y negativos. La arquitectura de la red de aprendizaje adoptada fue GoogLeNet. (Wang et. al., 2016) Una vez realizada la misma, se aplicaba un mapa de calor (como fase de post procesamiento) para destacar las regiones con cáncer.



**Figura 23:** Visualización de la detección de una región tumoral. Fuente: Wang, D., Khosla, A., Gargeya, R., Irshad, H., & Beck, A. H. (2016). Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*.

La etapa de clasificación basada en parches toma como entrada las imágenes, indicando la ubicación de las regiones que contienen cáncer metastásico. Durante la fase de entrenamiento, se extrajeron aleatoriamente como *input* millones de pequeños parches, de 256x256 píxeles, para entrenar un modelo de clasificación supervisado que discriminara entre dos clases de parches (positivos y negativos). De forma que, si el parche se encontraba en una región tumoral (positivo), se etiquetaba con 1 y, de lo contrario, en un parche normal (sin metástasis) tomaba el valor 0.

Una vez realizado, todos los resultados de la predicción se integraron en un mapa de calor. Para generar mapas térmicos de probabilidad de tumores se utilizó la arquitectura GoogLeNet en todo el conjunto de datos de entrenamiento.

Según los desarrolladores, el motivo de la mejor performance de su modelo respecto al resto fue gracias a una modificación realizada en el modelo de entrenamiento. Se observó que una proporción significativa de errores se debía a la clasificación de falsos positivos y para mejorar el rendimiento, se extrajeron datos de entrenamiento adicionales. El modelo se volvió a entrenar, pero con un conjunto de entrenamiento con más carga de negativos, en lugar de una selección totalmente aleatoria como se hizo al principio.

Respecto a los resultados obtenidos, se compararon las predicciones del algoritmo con las realizadas por un grupo de expertos formado por 11 especialistas en la detección de este tipo de cáncer. El algoritmo mostró una mejor clasificación que el grupo de expertos con un AUC (*Area Under the Curve*)<sup>21</sup> de 0.994 (algoritmo) vs 0.844 (mejor experto). (Wang et al., 2016)

Sin embargo, cabe añadir y así lo incluyen los ganadores de la competición en su informe que “es muy importante destacar que no existía correlación entre los errores cometidos por el sistema de Deep Learning y los humanos especialistas”. Esto debería hacernos pensar como de momento el DL es una herramienta complementaria que podría actualmente ser la clave no sólo para la detección precoz del cáncer de mama – enfermedad que padecen 1 de cada 4 españolas- sino también para mejorar el tratamiento del paciente y reducir la realización de pruebas tan invasivas como las biopsias.

---

<sup>21</sup> El AUC es el área bajo la curva ROC (*Receiver Operating Characteristic*) y nos da una idea de la performance de un modelo de clasificación binaria

## **5.2-Procesamiento de lenguaje: *electronic health records***

Aunque tal y como se introdujo en el apartado anterior las Redes Neuronales Convolucionales son las más usadas en el campo de la medicina, existen otros tipos de redes como las recurrentes que también se utilizan en la investigación médica. La tecnología de procesamiento del lenguaje natural también se utiliza en el campo de la medicina, que utiliza estas redes neuronales para procesar los registros médicos electrónicos.

En los últimos años, la historia clínica electrónica ha recibido cada vez más atención, pues toda la información del tratamiento de los pacientes se almacena ahora en formato electrónico. La información incluye datos demográficos estructurados, información sobre el diagnóstico, información sobre los medicamentos, información sobre el proceso de operación, resultados de pruebas experimentales y texto clínico no estructurado. (Jensen et al., 2012)

Entre una variedad de métodos de aprendizaje profundo, las redes neuronales recurrentes han demostrado su eficacia en tareas de procesamiento de lenguaje natural porque tienen la capacidad de capturar información secuencial que es inherente al lenguaje humano. Las redes neuronales recurrentes, al contrario que las convolucionales, tienen en cuenta el factor tiempo y la importancia de esto para el tratamiento del paciente está siendo cada vez más tenida en consideración. Por ejemplo, proporciona un tratamiento oportuno de los pacientes mediante la extracción de datos de la historia clínica que contienen incluso los fármacos que el paciente ha tomado anteriormente, pudiendo ser esto causante de otras patologías posteriores.

### **A. *Encoding***

Tal y como se explicó en el epígrafe dedica a las RNN, el análisis de datos en formato texto conlleva una labor previa de codificación. En el caso de los datos médicos, el problema no es solo que algunos de estos datos no estén codificados, sino que los distintos lenguajes de codificación utilizados hacen que sea casi imposible relacionar unos con otros, dificultando el tratamiento lógico de los mismos.

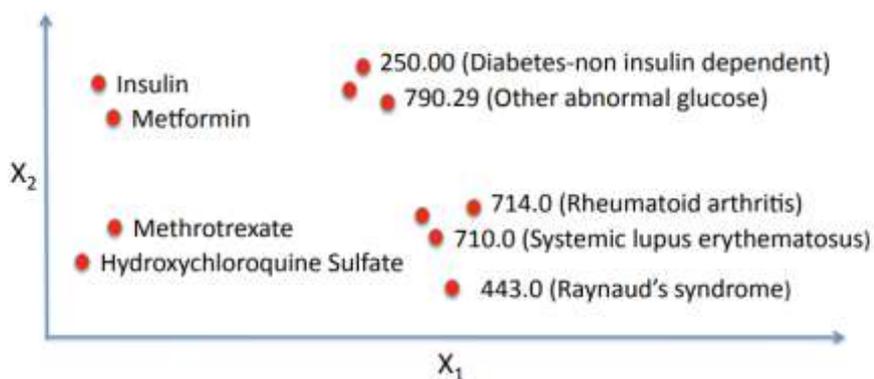
Por tanto, son muchos los especialistas que han llevado a cabo una labor de lo que han venido desarrollar modelos de “*Low Dimension Representation*”, también conocido como *Word Embedding*. Buscan pasar los códigos de los principales métodos de codificación

utilizados en el campo de la salud a un método lógico que facilite el tratamiento unitario a posteriori de los mismos. El aprendizaje de representaciones distribuidas o *embedding* ha demostrado ser especialmente útil en este tipo de tareas.

Así, el aprendizaje automático para predecir el inicio o la progresión de la enfermedad, o el reingreso en el hospital, puede utilizar un vector de características derivado de la agrupación de los códigos de la historia clínica de un paciente, que tendría una dimensión de cientos en lugar de decenas de miles. (Krompa et al., 2015)

La idea clave es que las palabras que pertenecen a contextos similares deben tener significados similares. Por ejemplo, si vemos las dos frases "el paciente se quejó de síntomas gripales" y "el paciente informó de síntomas gripales", podríamos deducir que "se quejó" significa lo mismo que "informó". En consecuencia, estas dos palabras deberían estar cerca en el espacio de representación. (Choi et al., 2016)

En EEUU los principales lenguajes de codificación son: ICD9 (*International Classification of Diseases*); compuesto por códigos de diagnóstico y procedimiento usados principalmente en reclamaciones e investigación médica, LOINC (*Logical Observation Identifiers Names and Codes*); que incluye los códigos más usados en los laboratorios y NDC (*National Drug Code*); identificador universal de medicamentos destinados al consumo humano. Sin duda alguna, de todos ellos se pueden sacar conclusiones y relaciones entre parámetros que pueden ser usados por la Inteligencia Artificial. Así, en 2016, un grupo de investigadores de la *New York University* comenzaron un proyecto que tenía como objetivo unificar esta codificación mediante *word embedding*.



**Figura 24:** *Low-dimensional representation o word embedding* de 2 dimensiones que muestra como conceptos médicos relacionados aparecen representados próximos los unos a los otros. Fuente: Choi Y, Chiu CY, Sontag D. Learning Low-Dimensional Representations of Medical Concepts. *AMIA Jt Summits Transl Sci Proc.* 2016;2016:41-50. Published 2016

## B. Redes Neuronales Recurrentes y datos médicos

Pero las RNN no son importantes únicamente para las tareas de procesamiento de lenguaje, sino que el sector sanitario se ha servido también de las grandes posibilidades que ofrecen estas gracias a la posibilidad que ofrecen de un tratamiento secuencial de los datos.

La arquitectura LSTM fue ya explicada en el apartado de este proyecto dedicado en particular a las RNN. Como ya se enunció entonces, utiliza las llamadas *embedded gates* para capturar únicamente aquella información que resulta relevante para el modelo. La selección de información que realiza el LSTM ha sido de gran importancia para la modelización de programas que beben de bases de datos que capturan información recopilada lo largo de un margen amplio de tiempo.

En 2016<sup>22</sup>, un grupo de investigadores desarrollaron, con motivo de la Conferencia Internacional EMBC (*Engineering in Medicine and Biology Society*), un modelo predictivo que buscaba medir el rango de dolor del paciente tras la administración de analgésicos en el post operatorio. En concreto, se cogió la dosis que normalmente suele aplicarse (10 mg de oxicodona por vía oral). La arquitectura de la red utilizaba LSTM y ReLU como función de activación. La razón por la que se usó en este estudio un Red Neuronal Recurrente (que normalmente analiza datos de un margen muy amplio de

<sup>22</sup>El estudio en concreto se publicó bajo el nombre de "Deep neural network architectures for forecasting analgesic response" (P. Nickerson, P. Tighe, B. Shickel y P. Rashidi)

tiempo) es muy interesante. Pues, debemos pensar que este tipo de redes no solo sirve para el tratamiento de datos de un amplio horizonte temporal, sino también de datos secuenciales. Los pacientes hospitalizados (como lo están aquellos que se encuentran en fase postoperatoria) están constantemente monitorizados (frecuencia cardíaca, respiratoria...) creándose, en márgenes muy pequeños de tiempo, muchos datos de marcado carácter temporal.

También se han desarrollado modelos que utilizan las RNN para el tratamiento de datos históricos, como es el paradigmático ejemplo de Doctor AI<sup>23</sup>. Se trata de un modelo predictivo genérico que cubre tanto dolencias médicas observadas, como medicación aplicada. Doctor AI es un modelo temporal que utiliza Redes Neuronales Recurrentes y fue desarrollado y aplicado a datos de Electronic Health Record (EHR) longitudinales para predecir diagnósticos y medicación.

---

<sup>23</sup> Choi et. al., 2016

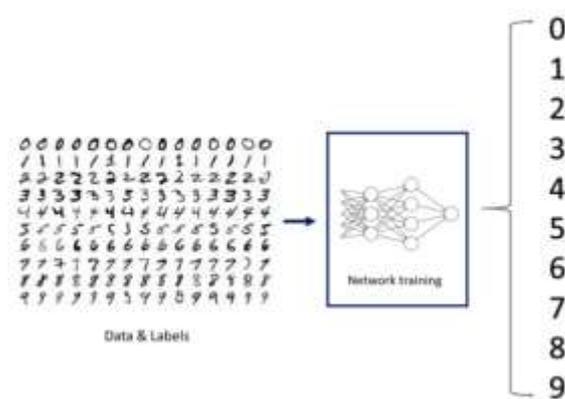
## 6.-Desarrollo de una Red Neuronal Convolutacional en Python

Se muestra a continuación un código sencillo de desarrollo de una Red Neuronal Convolutacional (*Introduction to CNN Keras – Acc 0.997 (top 8%), Yassine Ghouzam*). Este código sencillo, aun no siendo de elaboración propia, se añade aquí a modo ilustrativo junto a comentarios de mi propia autoría.

Se busca así mostrar un ejemplo sencillo de algoritmo de reconocimiento de imágenes, que sería la base utilizada para implementar otras RNA más complejas como las de reconocimiento de imágenes clínicas que han sido objeto de estudio de este trabajo. Para su desarrollo se utiliza Keras, que es la biblioteca de redes neuronales de código abierto escrita en Python.

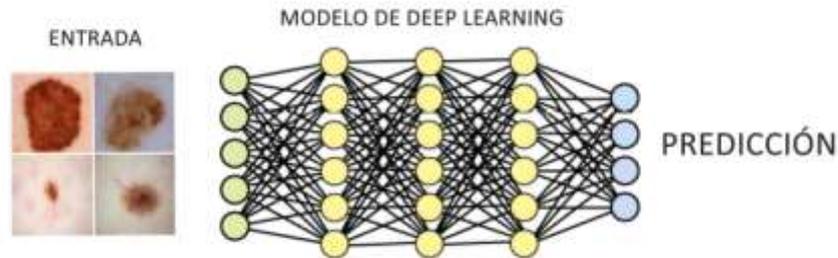
El código en cuestión es un clasificador de dígitos (modelo paradigmático en el reconocimiento de imágenes). El modelo se entrena sobre la base de datos MNIST (*Modified National Institute of Standards and Technology*). Este database contiene imágenes en las que aparecen, sobre un fondo blanco, una serie de dígitos (del 1 al 10) escritos a mano en color negro. Cuenta con 60.000 imágenes de entrenamiento y 10.000 para test, todas ellas de 28 x 28 píxeles.

La tarea consiste en clasificar qué dígito se encuentra dibujado en cada una de las imágenes.



**Figura 25:** Imagen ilustrativa del modelo de clasificación de dígitos a través de una Red Neuronal Convolutacional. Fuente: ichi.pro

El comportamiento de la red que se muestra a continuación utiliza un funcionamiento, aunque más básico, muy similar al utilizado por los modelos de clasificación de imágenes clínicas.



**Figura 26:** Representación del funcionamiento de una red neuronal Convolucional de clasificación de imágenes de lesiones cutáneas. Fuente: Pérez Lorenzo, C. (2019). *Detección precoz de cáncer de piel en imágenes basado en redes convolucionales*.

DATASET:

[cnn\\_mnist\\_datagen.pdf](#)

CÓDIGO:

1) Preparación de los datos

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import seaborn as sns
%matplotlib inline

np.random.seed(2)

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import itertools

from keras.utils.np_utils import to_categorical # convert to one-hot-encoding
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPool2D
from keras.optimizers import RMSprop
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ReduceLRonPlateau

sns.set(style='white', context='notebook', palette='deep')
train = pd.read_csv("../input/train.csv")
test = pd.read_csv("../input/test.csv")
```

```

In [3]:
linkcode
Y_train = train["label"]

# Drop 'label' column
X_train = train.drop(labels = ["label"],axis = 1)

# free some space
del train

g = sns.countplot(Y_train)

Y_train.value_counts()

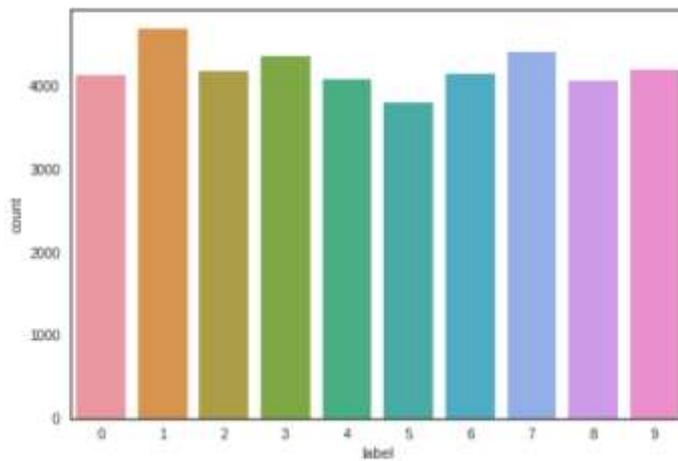
```

Output:

```

1    4684
7    4481
3    4351
9    4188
2    4177
6    4137
0    4132
4    4072
8    4063
5    3795
Name: label, dtype: int64

```



- Una vez preparados los datos, se produce el etiquetado mediante Hot Encoding (son 10 dígitos):

```
Y_train = to_categorical(Y_train, num_classes = 10)
```

- Se separan los conjuntos de Train y Validation

```

random_seed = 2

In [10]:
linkcode
# Split the train and the validation set for the fitting
X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, test_size
= 0.1, random_state=random_seed)

```

## 2) Definición del modelo

- Este código utiliza la Keras Sequential API, que permite que se vayan añadiendo las capas una a una. En primer lugar, se define la capa convolucional, en concreto se trata de 2 capas (de 32 y 64 filtros respectivamente) de 2 dimensiones cada una. Para la capa de *pooling* se utiliza la función Maxpool. La función activación utilizada es la ReLu. Respecto de las capas *fully connected*, se utiliza la función “softmax” para realizar la clasificación final.

```

model = Sequential()

model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
activation = 'relu', input_shape = (28,28,1)))
model.add(Conv2D(filters = 32, kernel_size = (5,5),padding = 'Same',
activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'Same',
activation = 'relu'))
model.add(Conv2D(filters = 64, kernel_size = (3,3),padding = 'Same',
activation = 'relu'))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(256, activation = "relu"))
model.add(Dropout(0.5))
model.add(Dense(10, activation = "softmax"))

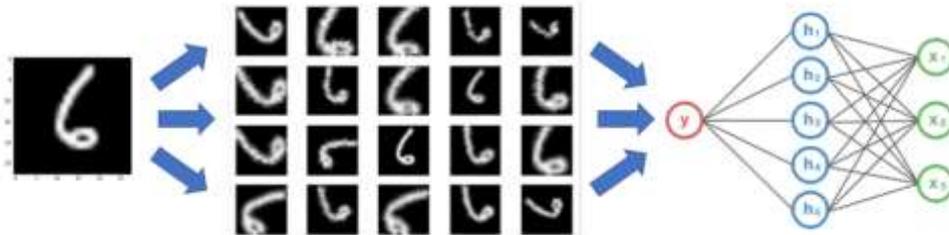
```

- Una vez creadas las capas se fijan 3 funciones de performance: *Score function*, *Loss function* (se utiliza una forma específica de las técnicas de clasificación: "categorical\_crossentropy") y *Optimisation algorithm* (se escogió RMSprop con valores por defecto en lugar del algoritmo del descenso gradiente que ha sido el explicado en este proyecto).

```
# Define the optimizer
```

```
optimizer = RMSprop(lr=0.001, rho=0.9, epsilon=1e-08, decay=0.0)
# Compile the model
model.compile(optimizer = optimizer , loss = "categorical_crossentropy",
metrics=["accuracy"])
```

- Se realiza un aumento del número de datos para mejorar la precisión del modelo. La idea no es añadir datos nuevos, sino alterar aquellos de los que ya se dispone. En este código se llevaron a cabo, sobre algunas imágenes escogidas de manera aleatoria, las siguientes alteraciones:
  - Se rotaron algunas imágenes 10 grados.
  - Otras se giraron en su ángulo horizontal un 10%
  - Algunas lo hicieron de manera vertical también un 10%



**Figura 27:** Imagen ilustrativa del funcionamiento del “data augmentation”. Fuente: Pérez Lorenzo, C. (2019). *Detección precoz de cáncer de piel en imágenes basado en redes convolucionales*.

```
# With data augmentation to prevent overfitting (accuracy 0.99286)

datagen = ImageDataGenerator(
    featurewise_center=False, # set input mean to 0 over the dataset
    samplewise_center=False, # set each sample mean to 0
    featurewise_std_normalization=False, # divide inputs by std of the
dataset
    samplewise_std_normalization=False, # divide each input by its std
    zca_whitening=False, # apply ZCA whitening
    rotation_range=10, # randomly rotate images in the range (degrees, 0 to 180)
    zoom_range = 0.1, # Randomly zoom image
    width_shift_range=0.1, # randomly shift images horizontally (fraction of total width)
    height_shift_range=0.1, # randomly shift images vertically (fraction of total height)
    horizontal_flip=False, # randomly flip images
    vertical_flip=False) # randomly flip images

datagen.fit(X_train)
```

- Se ajusta el modelo de entrenamiento:

```
# Fit the model
history = model.fit_generator(datagen.flow(X_train,Y_train, batch_size=batch_size),
                             epochs = epochs, validation_data = (X_val,Y_val),
                             verbose = 2, steps_per_epoch=X_train.shape[0] // batch_size,
                             callbacks=[learning_rate_reduction])
```

### 3) Evaluación del modelo:

- Confusion matrix:

```
# Look at confusion matrix

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix. Normalization
    can be applied by setting `normalize=True`.
    """
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

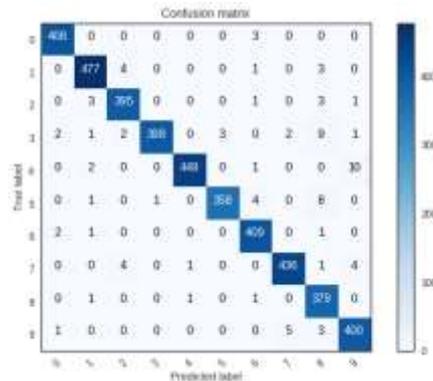
    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')

# Predict the values from the validation dataset
Y_pred = model.predict(X_val)
# Convert predictions classes to one hot vectors
Y_pred_classes = np.argmax(Y_pred,axis = 1)
# Convert validation observations to one hot vectors
Y_true = np.argmax(Y_val,axis = 1)
# compute the confusion matrix
```

```

confusion_mtx = confusion_matrix(Y_true, Y_pred_classes)
# plot the confusion matrix
plot_confusion_matrix(confusion_mtx, classes = range(10))

```



La matriz de confusión muestra algunos errores como es la mala clasificación en varias ocasiones del número 4, que el modelo identifica como un 9.

- Investigación de los errores:

```

# Display some error results

# Errors are difference between predicted labels and true labels
errors = (Y_pred_classes - Y_true != 0)

Y_pred_classes_errors = Y_pred_classes[errors]
Y_pred_errors = Y_pred[errors]
Y_true_errors = Y_true[errors]
X_val_errors = X_val[errors]

def display_errors(errors_index,img_errors,pred_errors, obs_errors):
    """ This function shows 6 images with their predicted and real labels"""
    n = 0
    nrows = 2
    ncols = 3
    fig, ax = plt.subplots(nrows,ncols,sharex=True,sharey=True)
    for row in range(nrows):
        for col in range(ncols):
            error = errors_index[n]
            ax[row,col].imshow((img_errors[error]).reshape((28,28)))
            ax[row,col].set_title("Predicted label :{}\nTrue label :{}".format(pred_errors[error],obs_errors[error]))
            n += 1

# Probabilities of the wrong predicted numbers
Y_pred_errors_prob = np.max(Y_pred_errors,axis = 1)

# Predicted probabilities of the true values in the error set
true_prob_errors = np.diagonal(np.take(Y_pred_errors, Y_true_errors, axis=1))

# Difference between the probability of the predicted label and the true label
L

```

```

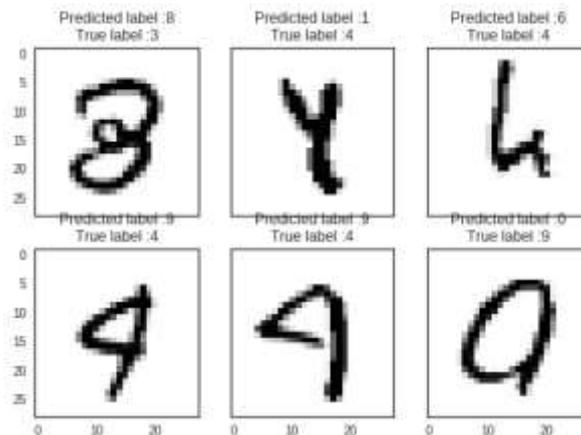
delta_pred_true_errors = Y_pred_errors_prob - true_prob_errors

# Sorted list of the delta prob errors
sorted_delta_errors = np.argsort(delta_pred_true_errors)

# Top 6 errors
most_important_errors = sorted_delta_errors[-6:]

# Show the top 6 errors
display_errors(most_important_errors, X_val_errors, Y_pred_classes_errors, Y_true_errors)

```



Se muestran los 6 errores más comunes del modelo. Por ejemplo, si nos fijamos en el primero de ellos, el modelo clasifica el 3 como un 8. Sin embargo, la *accuracy* del modelo es significativamente alta (0,99286) y este tipo de errores carecen de importancia, pues incluso para el ojo humano se trata de una imagen difícil de clasificar.

```

# predict results
results = model.predict(test)

# select the index with the maximum probability
results = np.argmax(results,axis = 1)

results = pd.Series(results,name="Label")

In [24]:
linkcode
submission = pd.concat([pd.Series(range(1,28001),name = "ImageId"),results],axis = 1)

submission.to_csv("cnn_mnist_datagen.csv",index=False)

```

## **7.-Dilemas éticos: AI y DeepLearning aplicados al healthcare.**

En este apartado nos alejamos de la parte técnica para centrarnos en el debate que inevitablemente surge a raíz de la incorporación de la tecnología de la AI al campo de salud.

Brevemente y en primer lugar, por ser el tema que más preocupa a los escépticos del avance de la AI, no se está defendiendo aquí una sustitución del personal sanitario, sino un uso en paralelo que ayude a confirmar los diagnósticos y facilite el trabajo de estos, repercutiendo notablemente en un mejor tratamiento del paciente. De hecho, tal y como se argumenta en el epígrafe dedicado a la clasificación de imágenes clínicas, los errores cometidos por los modelos y los errores humanos eran mayoritariamente distintos. Lo que debe llevarnos a pensar en la tremenda utilidad del uso de este tipo de herramientas por profesionales de la salud. Se trata de compenetración, y en ningún caso de sustitución.

Pero, ¿podría la AI afectar las decisiones de los especialistas?, ¿podrían estos dejarse llevar por los diagnósticos automatizados dejando a un lado los suyos propios, que en muchas ocasiones serán los acertados? Esta es una cuestión que en mi opinión si deberíamos plantearnos, pero, que por supuesto la respuesta dependerá de la aplicación específica que se les dé a estas tecnologías. Aunque se creen sistemas capaces de funcionar de manera autónoma, la decisión del profesional humano debe prevalecer por encima del criterio de la máquina. Así, más que compenetración, podríamos hablar de una suerte de jerarquía a través de la cual el especialista se sirve de estos modelos como punto de apoyo, pero teniendo siempre la última palabra.

En mi opinión, es el tratamiento de los datos de los pacientes el tema más controvertido y sobre el que debemos acentuar nuestra atención. En toda Revolución Industrial, encontramos un elemento motriz o motor precursor del cambio, como lo fue la locomotora en su día. En la actualidad, se habla del resurgimiento de una IV Revolución Industrial, y se identifica al Big Data como impulsor del cambio de paradigma.

La proliferación masiva de datos crece exponencialmente y no sólo a través de las redes sociales, que, aunque juegan un papel importante, no son la única causa de este

crecimiento masivo. Tan solo durante el año 2017, los datos de la industria de la salud y los seguros tuvieron un crecimiento medio de casi el 97% comparado con el año anterior. Hemos escuchado hablar del valor que tienen los datos a día de hoy para las compañías, y de cómo se identifica a estos con “el nuevo petróleo”, y en mi opinión no falta razón alguna a los defensores de esta idea. La recolección y el uso de los datos sanitarios es un tema muy delicado, por lo que deberíamos considerar qué tipos de datos recogen por ejemplo las historias clínicas y quién puede acceder a ellos.

Tal y como hemos visto en secciones anteriores, las RNA se han ido desarrollando de manera que necesitan una inmensa cantidad de datos, para evitar sesgos y funcionar con la mayor precisión posible. Pero, no es suficiente con incluir muchos datos, sino que debe tratarse de datos de calidad. Por ejemplo, para un modelo que intenta predecir si un paciente padece o no diabetes, no es suficiente contar con la cantidad de insulina en sangre (dato anónimo), sino que se debe recopilar otro tipo de información en relación a sus hábitos de vida, antecedentes familiares, edad, sexo, raza... todos datos de carácter personal y de una gran sensibilidad.

Con el desarrollo de aplicaciones para evitar los contagios por Covid, los desarrolladores de estos sistemas, y también los gobiernos como principales impulsores, permitieron el acceso a información personal de los individuos que en mi opinión se salta los límites de lo estrictamente necesario; como es el caso de los datos del entorno del paciente.

La Organización Mundial de la Salud (OMS) alerta del peligro adicional de que las instituciones encargadas de prestar los servicios de salud recojan más datos de carácter personal que los que requiere el algoritmo para funcionar, y que ese “superávit de datos”, sea compartido con terceros, por ejemplo, agencias gubernamentales, para llevar a cabo acciones que puedan vulnerar los derechos humanos, como pueden ser la vigilancia y el control de la población, o la aplicación de castigos individuales. La OMS pone de ejemplo de esto la aplicación para rastrear contactos de COVID-19 que puso en marcha el Gobierno de Singapur, cuyos datos podían ser cedidos en el marco de una investigación criminal. (Canfrnac, 2022)

Pero, ¿existe alguna legislación que proteja el uso de estos datos? En otros ámbitos los datos pueden ser totalmente anonimizados para su tratamiento, pero en el ámbito sanitario, aunque se elimine el nombre, existen elementos que imposibilitan la anonimización.

Actualmente, la Unión Europea es la institución que presta mayor atención a la protección de la privacidad de sus ciudadanos. Entre las iniciativas llevadas a cabo destacan: la creación por parte de la Comisión de un grupo de expertos en IA, el lanzamiento de la Alianza Europea, y además, grupos de trabajo como el Grupo Europeo sobre la Ética en la Ciencia y las Nuevas Tecnologías y la Agencia Europea de Derechos Fundamentales, que, junto con la comisión de expertos en IA elaboran lo que se ha venido a denominar “ethics guidelines”. La comisaria de Economía Digital y Sociedad, Mariya Gabriel, se pronunció respecto de la importancia del trabajo de estos grupos para la construcción del futuro de Europa: “La inteligencia artificial trae consigo enormes beneficios, pero también desafíos, y por ello es esencial involucrar a todos los actores, incluidos los del mundo académico, las empresas y la sociedad civil. Confío en que, juntos, nos aseguraremos de que los sistemas de IA se desarrollen para bien y para todos, respetando nuestros valores y derechos fundamentales.”

En abril de 2021, la Comisión Europea crea la llamada *AI strategy*, para fijar objetivos claros del enfoque que la Unión Europea quería dar al futuro de la IA en nuestro continente. Se centra en 2 áreas principales. Por una parte, busca la mejora de los sistemas desarrollados dentro de la Unión, se habla de la búsqueda de la “excelencia”, y por otro, de la salvaguarda tanto del mercado, como del sector público y por supuesto, de la seguridad de los ciudadanos mediante la protección de sus Derechos Fundamentales.

La herramienta perfecta para el objetivo proteccionista es la creación de una legislación unificada que no permita que el tratamiento de los datos de los mismos individuos pueda diferir de un país a otro. En abril del pasado año 2020, la Comisión propuso la realización de un Reglamento sobre Inteligencia Artificial para abordar los riesgos de los usos específicos de la IA. Este Reglamento entraría en juego sólo en cuestiones no tratadas a nivel nacional, pero nos sitúa en una situación muy cercana a la tan buscada unidad en el ámbito de protección de datos. A nivel internacional, la propuesta de la Comisión es el primer marco jurídico sobre IA, lo que sitúa a Europa en una posición líder. Aunque en mi opinión esta es la dirección que debemos seguir para un futuro donde la IA cada vez estará más presente en nuestro día a día, no es suficiente con una legislación a nivel europeo. Los datos son valores fácilmente transferibles y se puede atentar contra los derechos del paciente desde países extraeuropeos sin que exista entonces este tipo de control del riesgo.

Aunque el debate queda aquí servido, y sin duda es de obligado planteamiento, pues la Inteligencia Artificial no son solo luces, sino también tiene sus sombras. Las aplicaciones de esta pueden mejorar nuestra sanidad, calidad de vida, economía y muchos otros factores más, pero son nuestras instituciones las responsables de proveernos de una protección lo suficientemente fuerte para hacer que la sociedad no se sienta enfrentada a estas nuevas tecnologías.

## 8.-Conclusión

El sector de la salud es uno de los motores más importantes de toda sociedad, tanto económica como socialmente. Pero no siempre hemos sido conscientes de la importancia de disponer de un sistema sanitario potente.

En 2020, con la llegada de la pandemia y la expansión de la COVID-19 por todo el mundo, los países entraron en una especie de carrera al tratar de conseguir llevar la mejor gestión posible de la pandemia. Sin duda esta no era tarea fácil: la saturación de los hospitales, el desabastecimiento de personal y material sanitario, la incertidumbre de desconocer los efectos de la enfermedad y la eterna duda de cuándo podríamos volver a la llamada “nueva normalidad”. En resumen, supuso un ataque directo a los sistemas sanitarios que se vieron, a nivel mundial, superados por no estar lo suficientemente preparados.

Así, nos vimos obligados a buscar soluciones nuevas, para problemas que no se habían visto antes, y qué mejor forma de hacerlo que invirtiendo en futuro, en tecnología. Un diagnóstico precoz de la enfermedad, así como la predicción del número potencial de personas que necesitarían de asistencia sanitaria hubieran sido los dos puntos clave para haberse enfrentado a la pandemia, y no son datos de los que hablemos utópicamente, sino que tenemos al alcance de nuestra mano métodos para conseguirlos.

Se desarrollaron debido a las exigencias de la pandemia aplicaciones para detectar posibles contagios, para diagnosticar la enfermedad a través de la voz, gestión de recursos médicos e incluso algunos estudios sobre la viabilidad de cada paciente para superar o no la enfermedad. Sin duda, se trata de aplicaciones de la Inteligencia Artificial increíbles y muy útiles, pero ¿es necesario que llegue una pandemia para que comencemos a tomarnos en serio la incorporación de nuevos métodos de diagnóstico en el ámbito sanitario?

## REFERENCIAS:

1. Acevedo, E., Serna, A., & Serna, E. (2017). Principios y características de las redes neuronales artificiales. *Desarrollo e innovación en ingeniería*, 173.
2. Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python.
3. Bonilla Carrión, C. (2020). *Redes Convolucionales*. (Trabajo Fin de Grado Inédito). Universidad de Sevilla, Sevilla.
4. Cuevas-Tello, J. C. (2018). Apuntes de Redes Neuronales Artificiales. arXiv preprint arXiv:1806.05298.
5. Chanampe, H., Aciar, S., Vega, M. D. L., Molinari Sotomayor, J. L., Carrascosa, G., & Lorefice, A. (2019). Modelo de redes neuronales convolucionales profundas para la clasificación de lesiones en ecografías mamarias.
6. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, pp. 106–154, 1962
7. Durán Suárez, J. (2017). *Redes neuronales convolucionales en R: Reconocimiento de caracteres escritos a mano*.
8. Erroz Arroyo, D. (2019). Visualizando neuronas en redes neuronales convolucionales.
9. F. Rosenblatt. The perceptron—A perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory, 1957.
10. Flórez, R., & Fernández, J. M. (2008). *Las redes neuronales artificiales, fundamentos teoricos y aplicaciones practicas*. La Coruña, Oleiros, España: Netbiblo.
11. François Chollet. (2018). *DeepLearning with R*. United States of America: Manning Publications.
12. Fukushima, K., & Miyake, S. (1980). Neocognitron: Self-organizing network capable of position-invariant recognition of patterns. In *Proc. 5th Int. Conf. Pattern Recognition* (Vol. 1, pp. 459-461).
13. Moreno-Díaz-Alejo, L. (2020). Análisis comparativo de arquitecturas de redes neuronales para la clasificación de imágenes (Master's thesis).
14. P., Inza, I., & Moujahid, A. (1997). Tema 8. redes neuronales. *Redes Neuronales*, U. del P. Vasco, 12, 17.
15. Pouliakis et al.(2016) Artificial Neural Networks as Decision Support Tools in Cytopathology: Past, Present, and Future. *Biomedical Engineering and Computational Biology* 2016:7 1–18 doi:10.4137/BECB.S31601.
16. Serna, E. (2018). *Desarrollo e innovación en ingeniería*. ANTIOQUIA: INSTITUTO ANTIOQUEÑO DE INVESTIGACION. Recuperado el, 12.
17. Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

18. Arias, V., Salazar, J., Garicano, C., Contreras, J., Chacón, G., Chacín-González, M., ... & Bermúdez-Pirela, V. (2019). Una introducción a las aplicaciones de la inteligencia artificial en Medicina: Aspectos históricos. *Revista Latinoamericana de Hipertensión*, 14(5), 590-600.
19. Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A., Ciompi, F., Ghafoorian, M., van der Laak, J., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60–88. <https://doi.org/10.1016/j.media.2017.07.005>
20. Wang, D., Khosla, A., Gargeya, R., Irshad, H., & Beck, A. H. (2016). Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*.
21. Griffin J, Treanor D. Digital pathology in clinical use: where are we now and what is holding us back? *Histopathology*. 2017;70(1):134-145.
22. P. Nickerson, P. Tighe, B. Shickel and P. Rashidi, "Deep neural network architectures for forecasting analgesic response," 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016, pp. 2966-2969, doi: 10.1109/EMBC.2016.7591352.
23. Krompaß D, Esteban C, Tresp V, Sedlmayr M, Ganslandt T. Exploiting Latent Embeddings of Nominal Clinical Data for Predicting Hospital Readmission. KI. 2015
24. Choi Y, Chiu CY, Sontag D. Learning Low-Dimensional Representations of Medical Concepts. *AMIA Jt Summits Transl Sci Proc*. 2016;2016:41-50. Published 2016 Jul 20.
25. Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., & Sun, J. (2016, December). Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine learning for healthcare conference* (pp. 301-318). PMLR.
26. Blanco, R. L., Labrador-Espinosa, M. Á., Mir, P., & Matarazzo, M. (2021). Inteligencia Artificial y Machine Learning en trastornos del movimiento. *MANUAL SEN DE*.
27. Esteva, A., Kuprel, B., Novoa, R. *et al*. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542, 115–118 (2017).
28. Rodríguez Canfrac, P. (21 de marzo de 2022). Los límites éticos de la inteligencia artificial en la sanidad. *Fundación telefónica* <http://telos.fundaciontelefonica.com/la-cofa/los-limites-de-la-inteligencia-artificial-en-la-sanidad/>
29. Cabrol, M., González, N. A., Pombo, C., & Sánchez, R. A. (2020). fAIr LAC: adopción ética y responsable de la inteligencia artificial en América Latina y el Caribe. *bid*, 1.
30. Google AI Blog: Assisting Pathologists in Detecting Cancer with Deep Learnng (March 3, 2017). Martin Stumpe, Technical Lead, and Lily Peng, Product Manager
31. Official website of the European Union. <https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence>

32. *Introduction to CNN Keras – Acc 0.997 (top 8%), Yassine Ghouzam.* kaggle  
kernels pull code/yassineghouzam/introduction-to-cnn-keras-0-997-top-6