

Hacia un Sistema Inteligente de ayuda al diagnóstico de la COVID-19 mediante imágenes de Radiografías y/o Tomografías por Computador de la zona pulmonar y vías respiratorias de los pacientes



Memoria TFT - ULPGC

Autor: Samuel Arrocha Quevedo

Tutora: Carmen Paz Suárez Araujo

Co-tutor: Pablo Carmelo Fernández López



Abstract

Desde que la COVID-19 se convirtió en pandemia, supuso una emergencia sanitaria mundial. Urge detectarla tempranamente. Proponemos Sistemas Inteligentes (SI), basados en Redes Neuronales Convolucionales (CNN), para detección temprana de la COVID-19 mediante Radiografías (RX) y Tomografías Computerizadas (TC). Para la experimentación se aplican dos datasets, "covid-chestxray-dataset" [1] para las RX y "COVID-CT" para las TC [2]. Se desarrollan y analizan dos modelos. El modelo RX alcanza una precisión de 95% y una sensibilidad de 96%, mientras que para el desarrollado con TC, la precisión lograda es de 96% y la sensibilidad de 96%. Se concluye la capacidad de las CNN en la ayuda al diagnóstico COVID-19 y las RX y TC como pruebas diagnósticas.

Índice

Capítulo 1: Introducción	5
1.1 Estado actual del proyecto	5
1.2 Objetivos	6
1.3 Competencias involucradas	6
1.4 Herramientas utilizadas	8
Capítulo 2: Fundamentos teóricos	9
2.1 Fundamentos de la patología	9
2.1.1 Definición	9
2.1.2 Medios de contagio	11
2.1.3 Sintomatología	12
2.1.4 Identificación del virus en pruebas médicas	15
2.2 Inteligencia artificial	20
2.2.1 Aplicaciones	21
2.3 Red neuronal artificial	21
2.3.1 La neurona artificial	22
2.3.2 Funciones de activación básicas	23
2.3.3 El modelo y las capas	29
2.3.4 Tipos de aprendizaje	32
2.3.5 Backpropagation	33
2.3.6 Tipos de redes	36
2.4 Red Neuronal Convolutiva (CNN)	39
2.4.1 Convolución	40
2.4.2 Subsampling	42
2.4.3 Análisis de la precisión y la pérdida del modelo	43
2.4.4 Técnicas de preprocesado de imágenes	45
2.4.4.1 Padding	45
2.4.4.2 Matriz de confusión	47
2.4.4.3 Curva ROC	48

Capítulo 3: Sistema inteligente basado en redes neuronales profundas de ayuda al diagnóstico de la COVID-19	49
3.1 Conjuntos de datos	49
3.1.1 Radiografía (RX)	50
3.1.2 Tomografía computarizada (TC)	50
3.1.3 Descripción de los datasets	51
3.1.4 Procesamiento de las imágenes	53
3.2 Metodología y análisis de los modelos desarrollados	56
3.2.1 Modelo CNN de las RX	57
3.2.2 Modelo CNN de las TC	66
3.3 Resultados	74
Capítulo 4: Conclusiones y trabajos futuros	79
Bibliografía	81
Anexos	87

Capítulo 1: Introducción

La pandemia por el COVID-19 está suponiendo un cambio en la vida tal y como la vivíamos. La detección temprana de la patología es crucial para intentar curar la enfermedad y evitar las posibles secuelas que esta pueda ocasionar en los pacientes.

Como consecuencia, no es solo necesaria la movilización de los expertos sanitarios e investigadores sino también la de todas aquellas entidades que puedan contribuir con el fin de ayudar a combatir el virus y disminuir lo máximo posible los efectos que está teniendo en la sociedad. La tecnología es una de las herramientas más importantes en el ámbito de la salud y esta está muy ligada a la informática y la programación. Es por ello que la principal motivación de este proyecto es la de ayudar a los expertos sanitarios a detectar la patología y actuar en consecuencia. Este proyecto se basa en la creación de una Red Neuronal Artificial (RNA) capaz de clasificar pruebas médicas de origen respiratorio señalando si estas muestran signos de infección por el virus o, por el contrario, se encuentran libres de este.

1.1 Estado actual del proyecto

Gracias a los avances en el ámbito de la medicina se han desarrollado una variedad de vacunas que son efectivas para la protección y disminución de la infección por COVID-19 en la población del planeta [1] [2]. Sin embargo, a pesar del gran aumento del porcentaje de población vacunada, el virus sigue amenazando la vida tal y como la vivíamos antes de esta pandemia global y los números de infectados y fallecidos aumentan con el paso de los días [3]. La batalla continúa y con ello, la investigación tanto médica como tecnológica está en auge. Son muchos los estudios que, al igual que este, han demostrado la solidez y fiabilidad que las herramientas basadas en RNA pueden proporcionar a los especialistas, no

brindando solo diagnósticos muy precisos sino añadiendo valor a estas valoraciones y al sector de investigadores del mundo. Dados los resultados de las investigaciones es cuestión de tiempo que la aplicación de las Redes Neuronales Convolucionales (CNN) como metodología de apoyo a la detección y diagnóstico de enfermedades como la COVID-19 se haga efectiva y sea un recurso real en nuestros hospitales [4] [5] [6] [7].

1.2 Objetivos

Los objetivos del proyecto son los siguientes:

- Analizar y describir los conjuntos de datos que se utilizarán para el aprendizaje de la RNA.
- Conocer el potencial de las tecnologías basadas en inteligencia artificial para la detección de COVID-19.
- Investigar las capacidades de las redes CNN tanto para Radiografías (RX) como para Tomografías Computarizadas (TC).
- Ampliar mis conocimientos sobre redes neuronales, análisis de resultados y utilización de herramientas relacionadas con el campo como Keras, TensorFlow, Scikit-Learn, OpenCV, etc.
- Desarrollar un software expandible y extensible, que sea capaz de recibir cambios y mejorar con ellos.

1.3 Competencias involucradas

A continuación se explicará cómo el desarrollo de este proyecto ha conseguido aplicar las competencias citadas en el documento TFT01:

- **Competencias generales:**
 - G1, G2, G3, G4 y G5: se ha hecho uso de los conocimientos adquiridos durante el estudio del grado tanto mediante el desarrollo de

los modelos neuronales artificiales como de la escritura de esta memoria con un objetivo tanto científico como personal. Se han aplicado tanto metodologías muy específicas como generales y además se ha argumentado con el objetivo de llegar a un público general e intentar que estos entiendan el objetivo del proyecto y el alcance de este.

- **Competencias ULPGC:**

- N1, N2 y N3: dada la naturaleza del problema a resolver se ha optado por la comunicación con expertos en el campo de la IA encaminandome hacia la obtención de un proyecto bien formado y sólido. Tanto su implicación en el desarrollo de los modelos como en la escritura de esta memoria han hecho del proyecto un proyecto que habla por sí solo.
- N4 y N5: este proyecto nace de la naturaleza solidaria propiamente humana. Su objetivo no es solo el de demostrar las capacidades de la IA en la sociedad actual si no también el de ayudar con ella de forma activa, en este caso, en el campo de la medicina radiológica.

- **Competencias de la rama de computación:**

- CP01, CP04, CP05 y CP07: el desarrollo de este proyecto ha requerido tanto de un conocimiento teórico previo sobre la materia como de conocimiento biológico y médico. La aplicación de las técnicas adquiridas mediante la programación como la manipulación de datos y la generación de diferentes resultados según una amplia gama de variantes demuestran que se han llegado a cumplir con éxito todos los objetivos expuestos en las competencias de la rama de computación de la universidad.

1.4 Herramientas utilizadas

Para el desarrollo del modelo *Deep Learning* (DL) que se pretende desarrollar se hará uso de las siguientes herramientas y tecnologías:

- Lenguajes:
 - Python: lenguaje de programación en el que se desarrollan las dos RNA que se exponen.
 - Markdown: lenguaje de documentación para los repositorios de GitLab.

- Librerías:
 - TensorFlow y keras: librería de soporte al desarrollo de RNA.
 - Scikit-Learn: librería para análisis de datos.
 - OpenCV: librería de administración de imágenes.
 - NumPy: librería de operaciones matemáticas basada en vectores ζ , matrices, etc.
 - Seaborn: librería de representación de datos en gráficos.
 - Matplotlib: librería de representación de datos en gráficos.
 - Os: librería de operaciones con ficheros ubicados en el sistema.
 - Csv: librería de manejo de ficheros CSV.
 - Io: librería de lectura y escritura de ficheros.
 - Datetime: librería de fechas.
 - Pandas: librería de representación de datos en tablas.
 - Threading: librería para la creación de hilos en Python.
 - Albumentation: librería para alteración de imágenes.

- Plataformas de desarrollo:

- Colab: plataforma de desarrollo de Google que proporciona recursos hardware para proyectos basados en Python.
- Google Docs: plataforma para el desarrollo de la documentación del proyecto.
- Google Sheets: plataforma para el almacenamiento y análisis de las pruebas.
- Libre Office: conjunto de programas para escritura y análisis de los acontecimientos sucedidos durante el desarrollo del proyecto.
- Visual Studio Code: librería para desarrollo de código.
- Anaconda: administrador de paquetes de Python.

- Control de versiones:
 - Git: programa para control de versiones.
 - GitLab: plataforma en la nube para control de versiones.

- Almacenamiento:
 - Google Drive: plataforma de almacenamiento en la nube.

Capítulo 2: Fundamentos teóricos

2.1 Fundamentos de la patología

2.1.1 Definición

Los coronavirus pertenecen a una familia vírica zoonótica, lo que quiere decir que son susceptibles de infectar tanto a humanos como animales. En este caso no se ha demostrado la capacidad de infección en toda especie pero sí se ha detectado

la patología en diferentes especies mamíferas, como algunos felinos. Según el artículo científico oficial del ministerio de sanidad de España las diferentes variantes del virus *“...pueden transmitirse de los animales a los humanos. Los coronavirus que afectan al ser humano (HCoV) pueden producir cuadros clínicos que van desde el resfriado común con patrón estacional en invierno hasta otros más graves como los producidos por los virus del Síndrome Respiratorio Agudo Grave (por sus siglas en inglés, SARS) y del Síndrome Respiratorio de Oriente Próximo (MERS-CoV)”* [8].

El foco de la pandemia se originó en China, para luego expandirse por una larga lista de países a lo largo del tiempo. Debido al considerable grado de infección en el planeta, el nacimiento de nuevas cepas comenzó a surgir, cada una con sus características particulares que complican aún más el tratamiento de la enfermedad. Estas cepas se reproducen debido a los contagios masivos, lo cual produce la replicación del genoma que ocasiona errores y mutaciones. Actualmente las principales cepas son la surgida en Reino Unido, en Sudáfrica y en Brasil. Según Matthew Binnicker, director de un laboratorio especializado en virología clínica *“Hay tres variantes o mutaciones principales del SARS-CoV-2. La primera fue identificada por primera vez en el Reino Unido, la segunda en Sudáfrica y la tercera en Brasil. “La variante del Reino Unido tiene un mayor potencial para transmitirse, por lo que se propagará más rápido y con mayor facilidad entre las personas. No creemos aún que tenga resultados clínicos peores, por lo que no produce una enfermedad más grave ni lleva a una tasa más elevada de hospitalizaciones. La variante sudafricana también parece tener tasas más elevadas de transmisión. Luego, la variante de Brasil es muy nueva todavía y estamos aprendiendo sobre ella... Los virus no pueden mutar si no se replican. Entonces, todos esos pasos (vacunación, uso de mascarilla, distanciamiento físico) serán muy importantes para contribuir a la reducción del número de infecciones, lo que ralentizará el surgimiento de nuevas variantes y mutaciones.”* [9].

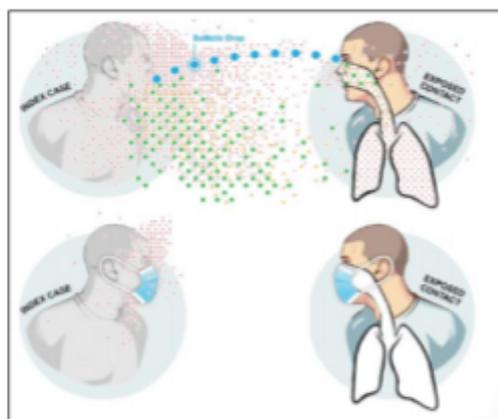


2.1.2 Medios de contagio

Los medios de contagio de los coronavirus de humano a humano se basan en las secreciones orales o nasales del individuo infectado. Si este habla, tose, estornuda o realiza cualquier acción que pueda liberar partículas de saliva al aire y estas entran en contacto con una persona sana, ya sea a través de alguna de las vías respiratorias o a través de los ojos, es muy probable que la persona sana quede infectada por el virus. Según el informe científico del ministerio de sanidad de España sobre el virus *“Con la evidencia científica acumulada, se considera que SARS-CoV-2 puede transmitirse de persona a persona por diferentes vías, siendo la principal mediante el contacto y la inhalación de las gotas y aerosoles respiratorios emitidos por un enfermo hasta las vías respiratorias superiores e inferiores de una persona susceptible. También se puede producir el contagio por contacto indirecto a través de las manos u objetos contaminados las secreciones respiratorias del enfermo con las mucosas de las vías respiratorias y la conjuntiva del susceptible. La transmisión vertical a través de la placenta también es posible, aunque poco frecuente. Otras vías de transmisión son muy improbables.”* [8] Para que la transmisión y la infección del individuo se completen deben suceder estos tres puntos sin excepción:

1. Que los aerosoles generados contengan microorganismos viables
2. Que los microorganismos contenidos en los aerosoles estén en cantidad suficiente y demuestren su capacidad de generar infección
3. Que los tejidos diana a estos microorganismos sean accesibles

Es debido a este contexto de transmisión que las mascarillas son las principales herramientas para evitar la expansión de los contagios (figura 1 [8]).



Puntos azules: gotas balísticas ($\geq 100 \mu\text{m}$); puntos verdes: aerosoles grandes ($>15 \mu\text{m}$ hasta $100 \mu\text{m}$); puntos naranjas: aerosoles intermedios ($>5 \mu\text{m}$ hasta $15 \mu\text{m}$); puntos rojos: aerosoles de pequeño tamaño ($\leq 5 \mu\text{m}$)

Figura 1: representación de un escenario de un posible contagio con y sin mascarilla

2.1.3 Sintomatología

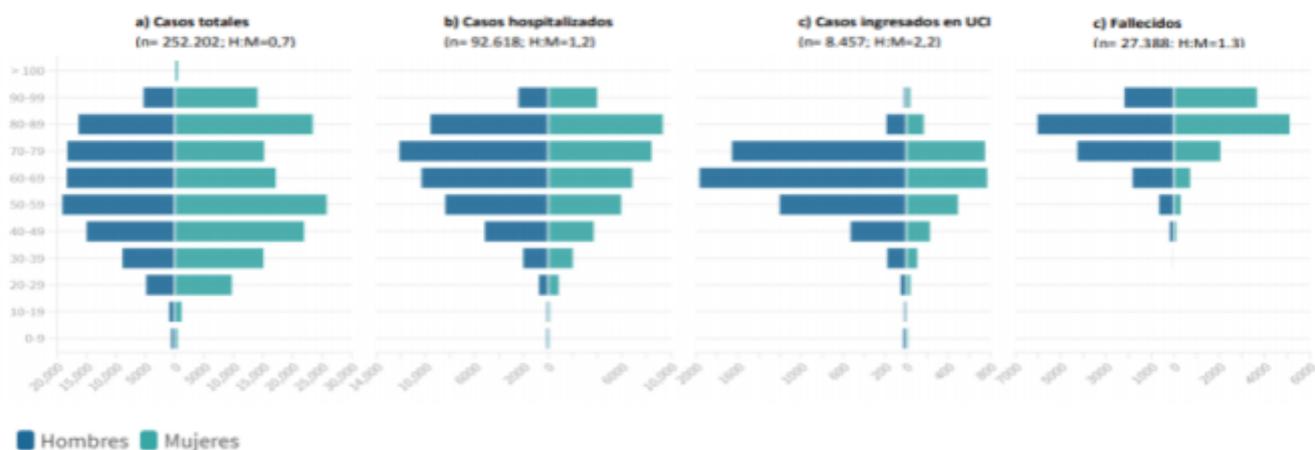
La sintomatología ocasionada por el COVID-19 es variada y depende en gran medida del estado de la infección y del estado de salud previo del paciente. Entre los síntomas más comunes encontramos:

- Hipertermia
- Tos seca
- Astenia
- Expectoración
- Disnea
- Dolor de garganta
- Cefalea
- Mialgia o artralgia
- Escalofríos
- Náuseas y vómitos

- Congestión nasal
- Diarrea
- Hemoptisis
- Congestión conjuntival

La relación entre la edad y el sexo de la personas infectadas con el virus es importante. En el caso del género biológico, se ha demostrado no ser un factor importante a la hora de alcanzar sintomatologías más graves o diferentes, aunque sí es cierto que los hombres tienden a ser hospitalizados en mayor proporción que las mujeres (figura 2) [8]. El caso de la edad, sí conviene tenerlo en cuenta ya que un importante porcentaje de los enfermos hospitalizados están comprendidos entre los 50 hacia delante. A pesar de todo, la enfermedad ha afectado a diferentes sectores de edad a lo largo del tiempo, comenzando primeramente con el origen de una mayor cantidad de personas de la tercera edad para continuar con un aumento importante de infectados jóvenes cuyas edades comprenden entre los 20 y 29 años. *“Se ha observado un cambio en la distribución por edad de los casos a lo largo de los últimos meses. En el periodo enero-mayo, el 18% de los casos tenían entre 50 y 59 años y el 69% eran mayores de 50 años. En un segundo periodo, entre junio y agosto, ha habido una disminución en la edad de los casos, siendo el grupo de edad entre 20 y 29 el más afectado con un 20% del total de casos notificados, seguido del grupo entre 30 y 39 años con un 18% de los casos. La edad de los pacientes ingresados en hospital y en UCI también ha disminuido si comparamos el primer periodo con el segundo. Sin embargo, entre los fallecidos se observan menos diferencias entre los dos periodos”* [8].

Periodo 1. enero-mayo 2020



Periodo 2. junio-agosto 2020

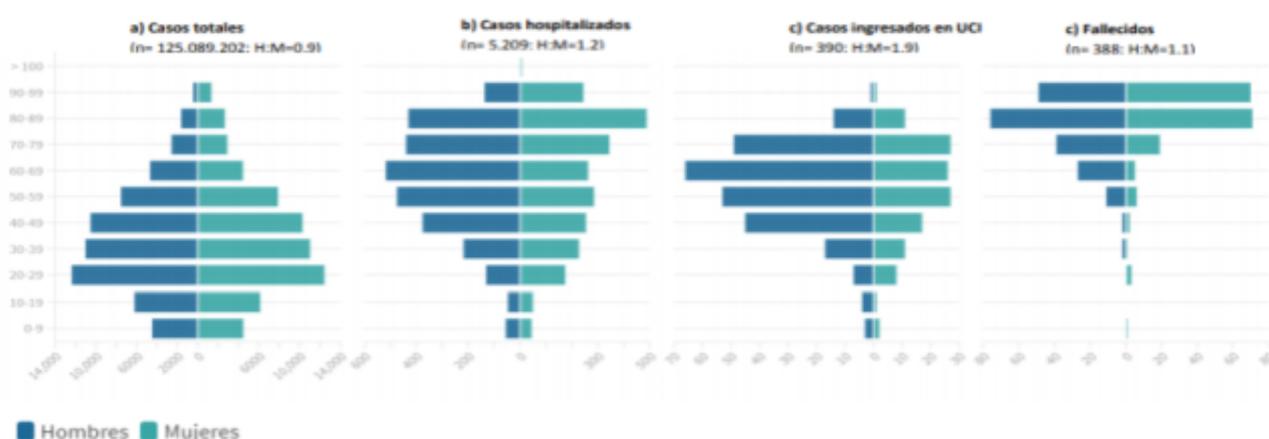


Figura 2: diagramas de afectados por coronavirus según edad

La duración de la enfermedad depende en gran medida del grado de infección entre otros factores, y este a su vez depende de la susceptibilidad del afectado y la virulencia a la que se vea expuesto entre otros:

- En casos de infección leve: dos semanas.
- En casos de infección media o grave: entre tres y seis semanas.

A pesar de ello, esta no es una norma general y muchas veces hay casos excepcionales, "... se han constatado multitud de casos de personas que refieren síntomas prolongados y recurrentes, durante semanas o meses, y que empiezan a adquirir una entidad propia que en algunos contextos se ha denominado COVID-19 persistente o "Long COVID" (77,78). Estas formas clínicas se describen en el apartado 3.3 de este informe." [8].

2.1.4 Identificación del virus en pruebas médicas

Generalmente, las pruebas más utilizadas para la detección de la patología son el test de antígenos y la conocida RT-PCR, siendo esta última muy eficaz en casos en los cuales la infección no es demasiado reciente. Sin embargo esto no relega las pruebas radiográficas y tomográficas a otro plano, ya que no solo sirven para la detección, sino también para la evaluación posterior por parte de los profesionales del campo de la salud.

A simple vista, la identificación del coronavirus en una prueba de tipo radiográfica puede parecer complicada debido a su similitud con otras afecciones pulmonares, y de hecho lo es, no por la identificación de la afección en una prueba ideal en sí, sino precisamente por las complicaciones que pueden surgir. Por ejemplo, un exceso de grasa en la zona o una postura inadecuada durante la realización de la prueba pueden dar lugar a una prueba que puede ser

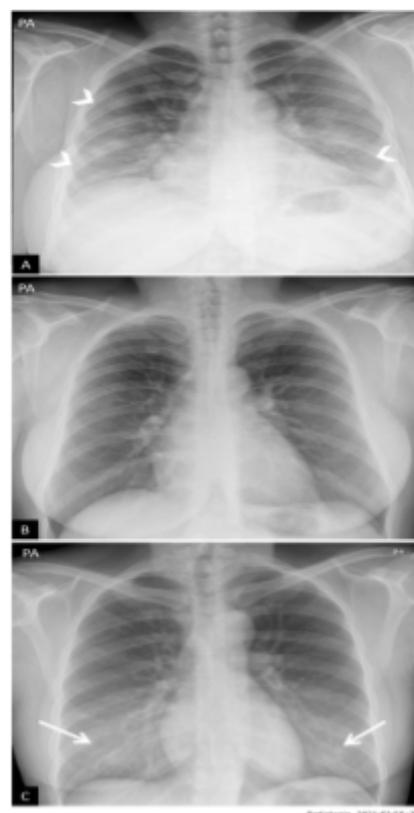


Figura 3: falsos positivos en coronavirus debido a tejidos adiposos

un falso positivo, no solo por parte de un sistema inteligente sino incluso por un especialista humano. Sin embargo esta afección cuenta con cierta diferenciación que hace que sea más fácil detectar que incluso otras enfermedades pulmonares. Según un especialista del Hospital Universitario Clínico de San Cecilio, en el cual están investigando el uso de la inteligencia artificial para la detección del virus en pacientes supuestamente infectados, distinguir una RX de una persona sana de una RX de una persona afectada por el virus se conforma de un proceso en el cual se verifica que hay una infección bilateral, es decir, que ambos pulmones muestran signos de infección, y la presencia de unas masas blancas, denominadas opacidades, que afectan prácticamente a todo el volumen de los pulmones. En la figura 3 [10] podemos ver dos casos concretos de falsos positivos:

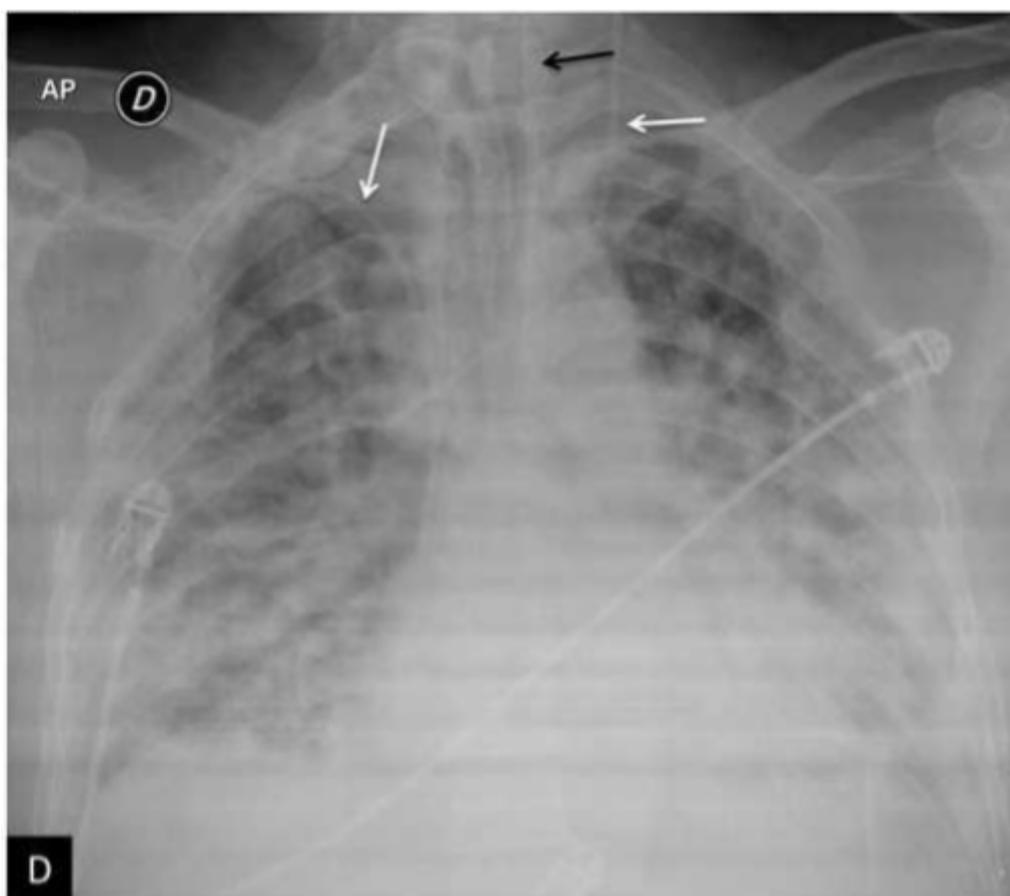
- Las dos primeras RX fueron tomadas para una mujer de 38 años con síntomas de infección por coronavirus. El resultado fue falso positivo debido a los pechos que creaban una ilusión de opacidad que en realidad no era infección. Podemos apreciar el cambio entre la primera RX y la segunda en la cual se posiciona mejor y se logra obtener la ausencia de líquido o pus en los pulmones.
- Lo mismo ocurre con la tercera RX tomada para una mujer de 18 años. Nuevamente los tejidos adiposos mamarios crean una sección de opacidad en las zonas inferiores de los pulmones pudieron dar lugar a un diagnóstico de falso COVID-19.

A continuación veamos casos en los cuales la presencia de coronavirus fue notoria. Si observamos la figura 4 [10] podemos apreciar la evolución de la infección de una mujer de 47 años. La primera RX fue tomada solo tres días antes de la segunda. Puede verse como la infección afecta a ambos laterales, estando el foco predominante en el pulmón derecho. Lo mismo ocurre con la figura 5 [10], en la cual podemos observar claros signos de infección por coronavirus en un paciente de 45

años. Podemos apreciar que las opacidades en ambos pulmones llegan a ocupar gran parte de los mismos, lo que da lugar a concluir en un estado avanzado de la patología [10] [11].



Figura 4: evolución de un verdadero positivo



Radiologia. 2021;63:56-73

Figura 5: muestra de un paciente con infección por coronavirus avanzada

Por su parte, las TC son también grandes protagonistas en la detección del virus en personas infectadas. Poseen una precisión y sensibilidad muy elevadas aunque son algo controversiales en algunas instituciones debido a su baja especificidad (25%), motivo por el cual se ha relegado a una segunda opción en muchos países y regiones. Por regla general, y según la Sociedad Española de Radiología Médica [10], se recomienda la realización de pruebas tomográficas en casos como los siguientes:

- Pacientes con muchos de los síntomas propios de la infección por coronavirus que muestran grandes signos de enfermedad.
- Pacientes con difícil acceso a RX o a RT-PCR.
- Pacientes con infección por COVID-19 confirmada y con sospechas de empeoramiento.
- Pacientes con otras enfermedades que muestren signos de infección por coronavirus y para las que esta pueda ser de mayor gravedad.

El análisis de una TC es muy similar al de una RX. Las infecciones en su fase más temprana suelen ser unilaterales y unifocales, para, con el transcurso del tiempo, volverse bilaterales y multifocales. De nuevo la clave está en las opacidades, es decir, las zonas grisáceas de las TC, las cuales, por norma general, suelen dejar entrever los vasos sanguíneos detrás, aunque no siempre, si el caso es muy concentrado y más severo. Un ejemplo de TC de un paciente con infección por coronavirus se puede ver en la figura 6 [10].

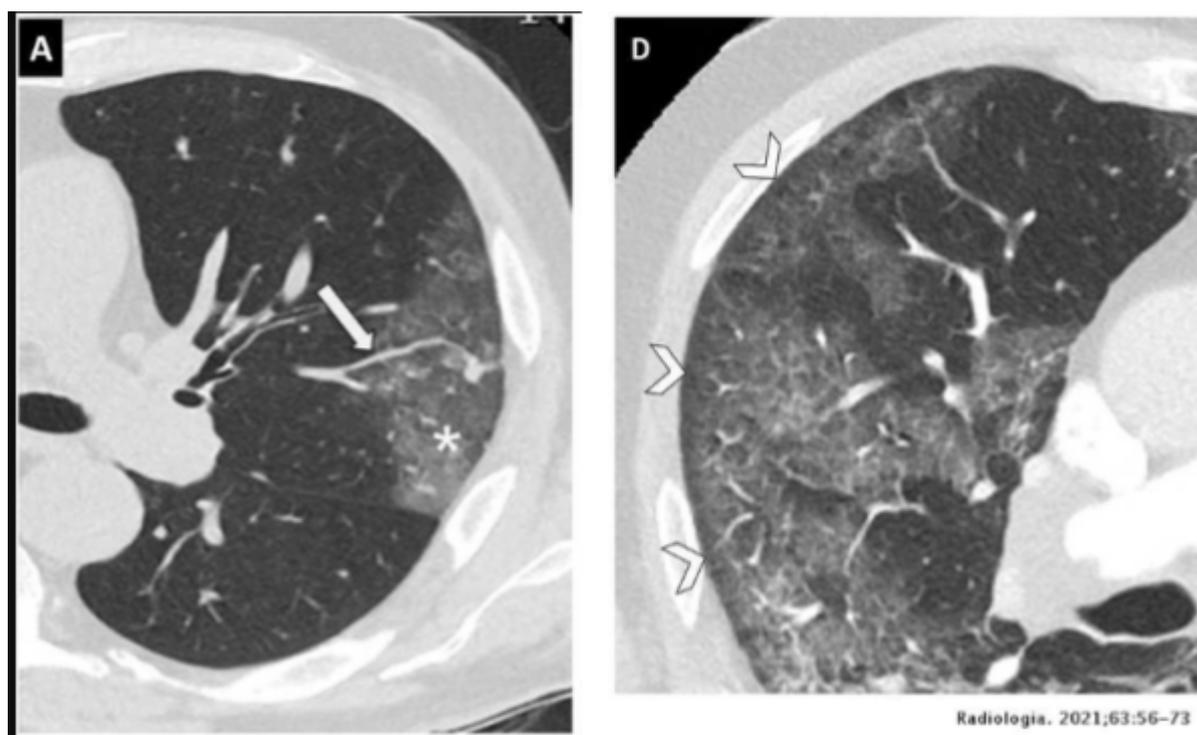


Figura 6: muestras de tomografías con signos de infección por COVID-19

2.2 Inteligencia artificial

El concepto de Inteligencia Artificial (IA) no es una novedad, se trata de un término investigado desde hace muchos años. Los recursos y la tecnología de la que dispone el ser humano en la actualidad han hecho posible el desarrollo de muchos tipos de inteligencias artificiales que imitan de forma muy convincente el comportamiento y pensamiento humano.

En palabras de Russell y Norvig “...el término *inteligencia artificial* se aplica cuando una máquina imita las funciones «cognitivas» que los humanos asocian con otras mentes humanas, como por ejemplo: «percibir», «razonar», «aprender» y «resolver problemas»” [12].

2.2.1 Aplicaciones

Las aplicaciones de la IA en la sociedad son considerables actualmente. Por regla general son sometidas a un proceso de investigación y pruebas antes de incorporarse como herramientas utilizables por la sociedad. Los principales campos en los que tiene importante influencia son [13]:

- Lingüística computacional
- Minería de datos (Data Mining)
- Industria
- Medicina
- Mundos virtuales
- Procesamiento de lenguaje natural (Natural Language Processing)
- Robótica
- Mecatrónica
- Sistemas de apoyo a la decisión
- Videojuegos
- Prototipos informáticos
- Análisis de sistemas dinámicos
- Simulación de multitudes
- Sistemas Operativos
- Automoción

2.3 Red neuronal artificial

Las Redes Neuronales Artificiales (RNA), conocidas también como Sistemas Conexionistas, nacen del deseo de crear capacidad de aprender y elegir de forma matemática. Para ello se hace uso de la computación y de los recursos que esta nos brinda.

Para la construcción de su arquitectura los expertos llevan décadas analizando el comportamiento neurológico del cerebro y las neuronas, humanas con el fin de desarrollar una inteligencia no humana que se asemeje a esta en varias de sus cualidades. Con el fin de entender mejor cómo funciona y cómo se inspira en nuestra biología debemos comenzar por la mínima unidad inteligente, la neurona artificial.

2.3.1 La neurona artificial

La neurona artificial es la mínima unidad inteligente que compone una RNA. Su funcionamiento matemático se basa en el *modus operandi* de su concepto biológico y consta de varias partes (figura 7 [14]). En palabras de la wikipedia inglesa las partes más importantes de una neurona son *“Dendrites – In a biological neuron, the dendrites act as the input vector. These dendrites allow the cell to receive signals from a large (>1000) number of neighboring neurons. As in the above mathematical treatment, each dendrite is able to perform “multiplication” by that dendrite’s “weight value.” The multiplication is accomplished by increasing or decreasing the ratio of synaptic neurotransmitters to signal chemicals introduced into the dendrite in response to the synaptic neurotransmitter... Soma – In a biological neuron, the soma acts as the summation function, seen in the above mathematical description... the positive and negative ions are effectively added in summation, by simple virtue of being mixed together in the solution inside the cell’s body.*

Axon – The axon gets its signal from the summation behavior which occurs inside the soma. The opening to the axon essentially samples the electrical potential of the solution inside the soma. Once the soma reaches a certain potential, the axon will transmit an all-in signal pulse down its length. In this regard, the axon behaves as the ability for us to connect our artificial neuron to other artificial neurons. [Dendritas: en un modelo biológico, las dendritas actúan como un vector de entradas. Estas

dendritas permiten a las neuronas obtener información de más de 1000 neuronas ajenas. Como se vio en la representación matemática superior, cada dendrita es capaz de realizar una “multiplicación” por el peso asignado a la propia dendrita. La multiplicación se realiza gracias al incremento o decremento del grado de neurotransmisores sinápticos... Soma: en una neurona biológica, el soma actúa como la función sumatoria, como se vio en la descripción matemática superior... los iones negativos y positivos se añade a la función sumatorio para estar entremezclados en la solución dentro del cuerpo de la célula... Axón: el axón obtiene su señal de la función sumatoria que ocurre dentro del soma. Una vez el soma obtiene cierto grado de carga transmite la información en forma de pulso eléctrico, en este aspecto el axón sería el elemento de salida que conectan con otras dendritas.]” [15].

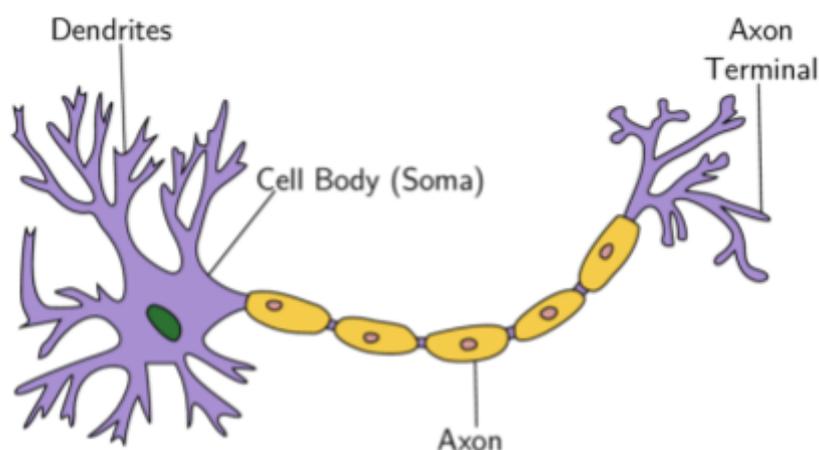


Figura 7: partes de una neurona

2.3.2 Funciones de activación básicas

Las funciones de activación son aquellas reglas matemáticas o funciones que dictaminan cuando una unidad se activará o no. En el caso de activación los valores

pueden variar dependiendo de la función de activación que se haya aplicado. Cada una de ellas tiene sus ventajas e inconvenientes, por lo que no hay una función de activación simplemente mejor para todos los casos, sino que cada una puede destacar en un problema determinado mientras que para otro no funciona tan bien como cabría esperar. A continuación definiremos algunas de las funciones de activación más utilizadas en el mundo del DL [16] [17]:

- Función sigmoide o logística (figura 8 [17]):
 - Valores de salida: entre 0 y 1
 - Descripción: *“La función sigmoide transforma los valores introducidos a una escala (0,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0.”* [16].
 - Características:
 - Saturación del gradiente.
 - Convergencia considerablemente lenta.
 - No centrada en el 0.
 - Acotada entre 0 y 1.
 - Rendimiento óptimo en la última capa de la RNA.
 - Fórmula: $f(x) = \frac{1}{(1+e^{-x})}$

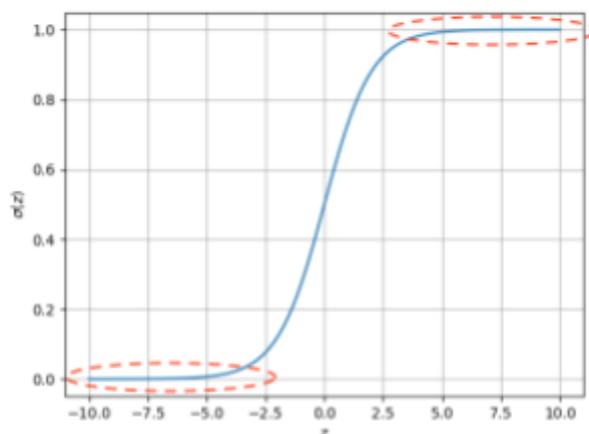


Figura 8: representación del resultado de una función sigmoide

- Función Tanh o tangente hiperbólica (figura 9 [17]):
 - Valores de salida: entre -1 y 1
 - Descripción: *“La función tangente hiperbólica transforma los valores introducidos a una escala (-1,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a -1.”* [16].
 - Características:
 - Saturación del gradiente.
 - Lenta convergencia.
 - Centrada en 0.
 - Está acotada entre -1 y 1.
 - Muy buenas para predicciones entre dos casos.
 - Muy útiles en redes neuronales artificiales recurrentes.
 - Fórmula: $f(x) = \frac{2}{1+e^{-2x}} - 1$

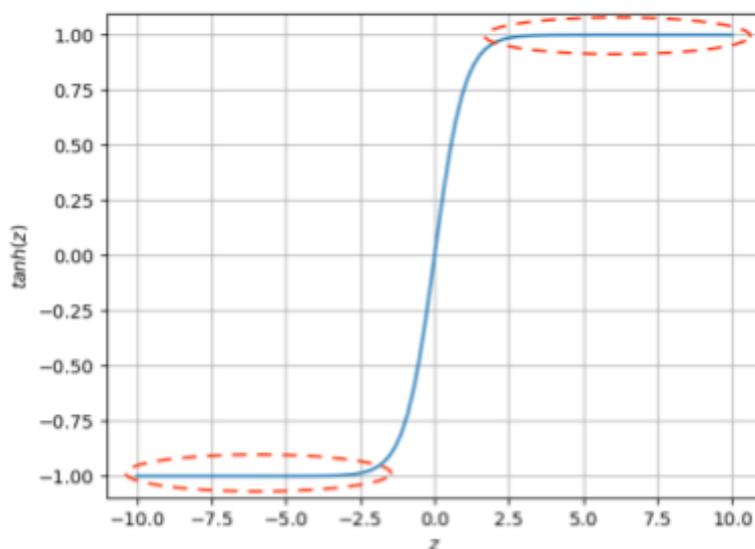


Figura 9: representación de una función Tanh

- Función ReLU (figura 10 [17]):
 - Valores de salida: 0 o x , dependiendo de la entrada.
 - Descripción: “La función ReLU transforma los valores introducidos anulando los valores negativos y dejando los positivos tal y como entran.” [16].
 - Características:
 - Activación solo con números mayores o iguales a 0.
 - Sin acotación
 - Buena para procesamiento de imágenes, por lo que son una buena opción para redes del tipo convolucional
 - Fórmula: $f(x) = \max(0, x) = \{0 \text{ para } x < 0; x \text{ para } x \geq 0$

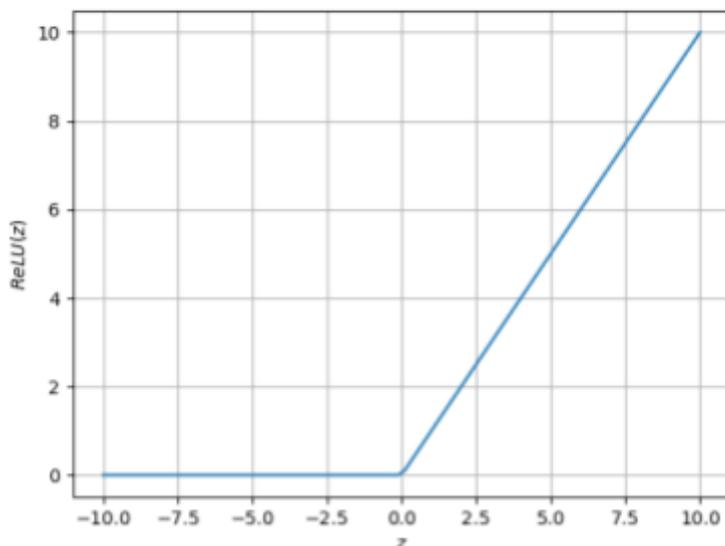


Figura 10: representación de una función ReLU

- Función *Leaky Relu* (figura 11 [18]):
 - Valores de salida: 0 o $a \cdot x$, dependiendo de la entrada.
 - Descripción: “La función *Leaky ReLU* transforma los valores introducidos multiplicando los negativos por un coeficiente rectificativo y dejando los positivos según entran.” [16].
 - Características:
 - Añade una penalización a los números menores que 0 mediante la aplicación de un coeficiente rectificador.
 - Sin acotación
 - Buena opción en el caso de las redes convolucionales
 - Fórmula: $f(x) = \{0 \text{ para } x < 0; a \cdot x \text{ para } x \geq 0\}$

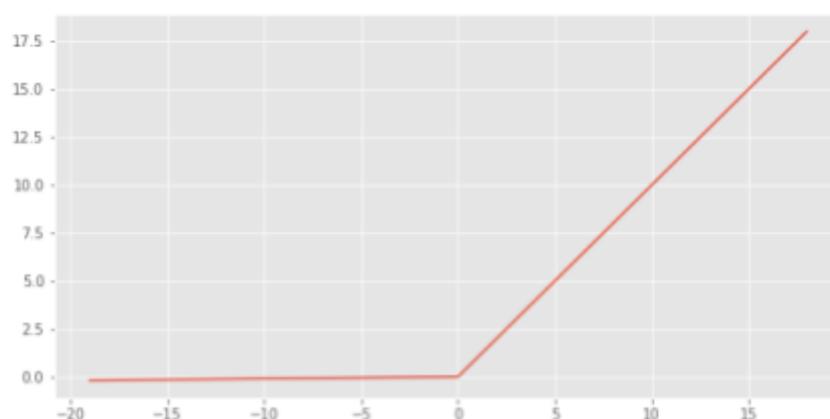


Figura 11: representación de una función Leaky ReLU

- Función softmax (figura 12 [19]) [60]:
 - Valores de salida: entre 0 y 1, representando probabilidades.
 - Descripción: *“La función Softmax transforma las salidas a una representación en forma de probabilidades, de tal manera que el sumatorio de todas las probabilidades de las salidas de 1.”* [16].
 - Características:
 - Utilizada para obtener dos o más probabilidades
 - Acotación entre 0 y 1
 - Utilizada cuando se trabaja con varios tipos posibles de salidas, es decir, en problemas con múltiples clases.
 - Rendimiento muy bueno en las capas finales.
 - Fórmula: $f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^k e^{z_k}}$

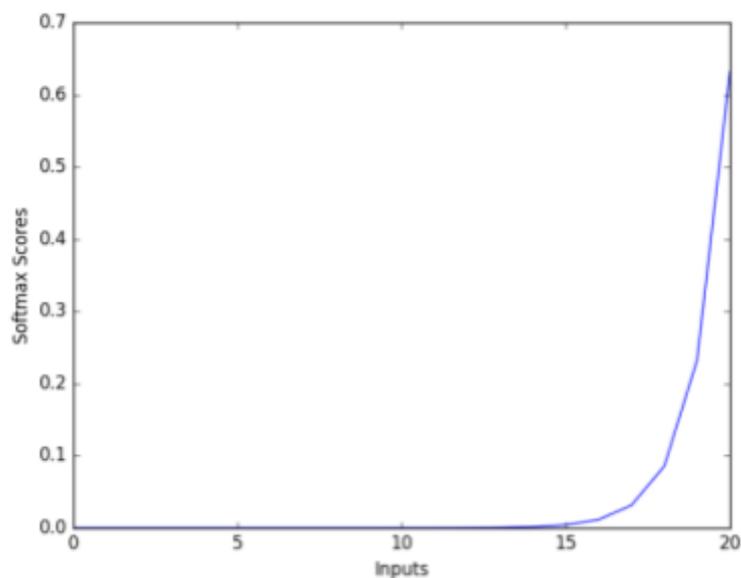


Figura 12: representación de una función softmax

2.3.3 El modelo y las capas

Según IBM *“Las redes neuronales son modelos simples del funcionamiento del sistema nervioso. Las unidades básicas son las neuronas, que generalmente se organizan en capas... Una red neuronal es un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas.”* [21].

Generalmente un modelo se compone de capas, compuestas de neuronas, las cuales pueden pertenecer a tres niveles diferentes:

- Capa de entrada: compuesta de las neuronas que reciben las entradas sin alteración previa obra de la propia RNA.

- Capas intermedias u ocultas: conformadas por los grupos de neuronas que manipulan la información con el objetivo de realizar la mejor clasificación o predicción posible para cada una de las entradas.
- Capa de salida: conformada por una o varias neuronas representativas de los tipos de salida que pueden darse en el problema.

En la figura 13 [20] se puede apreciar un ejemplo de una modelo de RNA sencilla aplicada a la tarea de recetar un tipo de medicamento. En él se obtienen diversas entradas que será procesadas por la capa de entrada y las capas ocultas para finalmente conducir las salidas a la capa de salida, la cual determinará qué medicamento es el mejor para el paciente definido por las entradas [20].

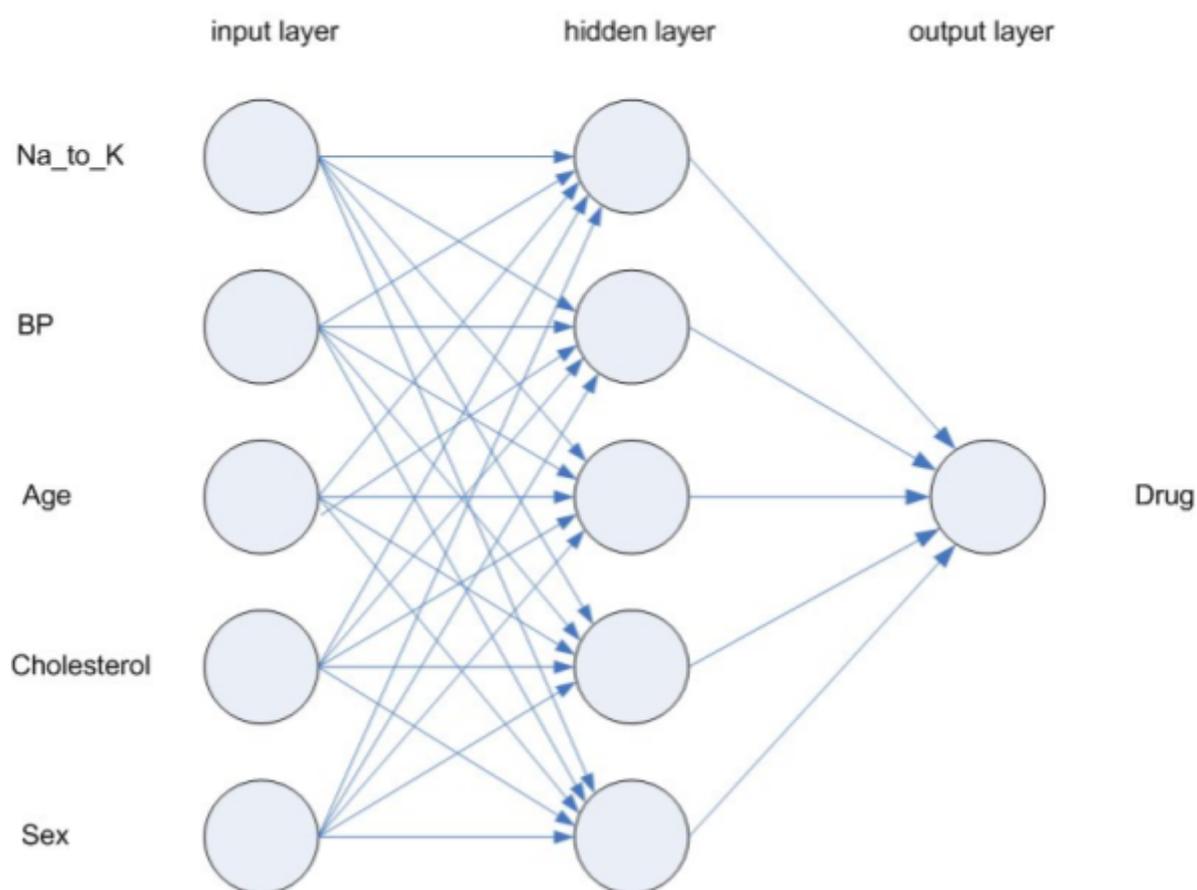


Figura 13: representación de un modelo con 3 capas. Su funcionamiento se basa en la predicción de un medicamento según diferentes entradas.

Como mencionamos anteriormente, las RNA pretenden imitar las características de aprendizaje y elección biológicas por lo que un modelo de RNA ideal debería cumplir las siguientes condiciones [21]:

- Estructura de procesamiento de carácter distribuido y paralelo.
- Conectividad alta entre las unidades básicas de las RNA, es decir, las neuronas.
- Conexiones mutables en base a la experiencia obtenida.
- Proceso de aprendizaje continuo y, si es posible, no supervisado.

- Robustez y fiabilidad en casos en los que algunas de las neuronas de la RNA sea eliminada del modelo.

2.3.4 Tipos de aprendizaje

Existe variedad en cuanto a las metodologías aplicables para lograr que una RNA aprenda por sí sola. Entre los tipos de entrenamiento sobresalen dos importantes y sus subtipos [22]:

- Aprendizaje Supervisado (AS): se realiza en base a un conocimiento sobre las entradas y salidas deseadas. La red va automodificando sus parámetros mediante la comparativa de las salidas de varias iteraciones de entrenamiento intentando acercarse lo máximo posible a los resultados ya proporcionados en la supervisión. El AS se subdivide a su vez en varios subtipos de aprendizajes:
 - Aprendizaje estructural: un tipo de aprendizaje supervisado no demasiado dependiente de las iteraciones de entrenamiento previas. En palabras de DialNet “se refiere a la búsqueda de la mejor conexión o afinidad posible entrada/salida para cada pareja de patrones individuales. La mayor parte de los algoritmos de aprendizaje... tiene un enfoque estructural”
 - Aprendizaje temporal: se basa en la obtención de patrones de aprendizaje en base a las entradas de la iteración y a los resultados generados en iteraciones anteriores. Al contrario que el aprendizaje estructural, este posee una dependencia estrecha con la experiencia adquirida con anterioridad en el entrenamiento.
- Aprendizaje No Supervisado (ANS): se trata de un tipo de aprendizaje sin supervisión alguna que se basa en la información local. Se trata de una

metodología de entrenamiento más cercana a la del modelo biológico donde solo hay información de entrada. La RNA que aplique ANS actualizará los parámetros durante el entrenamiento basándose en generar salidas consistentes, es decir, agrupa las salidas en grupos (clases) analizando las entradas con características comunes. Para ello no solo se fija en las similitudes, ya que es muy probable que para un modelo de RNA el conjunto de muestras tenga similitudes, sino que también es capaz de ignorar aquellas características de las entradas que son comunes a todas o, al menos, una gran parte de las entradas. En definitiva, su funcionamiento se basa en la clasificación mediante la concentración en características comunes de grupos y la abstracción de características comunes a todo el *dataset*.

2.3.5 Backpropagation

El principal problema de las RNA con más de una capa oculta consiste en aplicar una metodología mediante la cual se actualizan los pesos en base al grado del error cometido por cada una de las neuronas que conforman las capas ocultas del modelo. Estas pueden conformarse de más de una neurona artificial, lo cual hace que el cálculo no se base solo en una de las neuronas que conforman la capa sino en todas sus neuronas hermanas a las que se conectan.

La Regla Delta Generalizada, más conocida en inglés como

$$y = (3x - 11)^2$$

$$u = 3x - 11$$

$$y = u^2$$

$$\frac{dy}{dx} = \frac{dy}{du} \times \frac{du}{dx}$$

$$\frac{dy}{du} = 2u = 2(3x - 11)$$

$$\frac{du}{dx} = \lim_{h \rightarrow 0} \frac{3(x+h) - 11 - (3x - 11)}{h}$$

$$\frac{du}{dx} = \lim_{h \rightarrow 0} \frac{3x + 3h - 11 - 3x + 11}{h}$$

Figura 15: representación matemática del proceso de la Regla de la cadena de dos funciones Y y U.



Backpropagation, es una técnica aplicada en las RNA multicapa que se basa en el análisis de la relación entre la función de coste y los pesos y *bias* del modelo (figura 14 [23]).

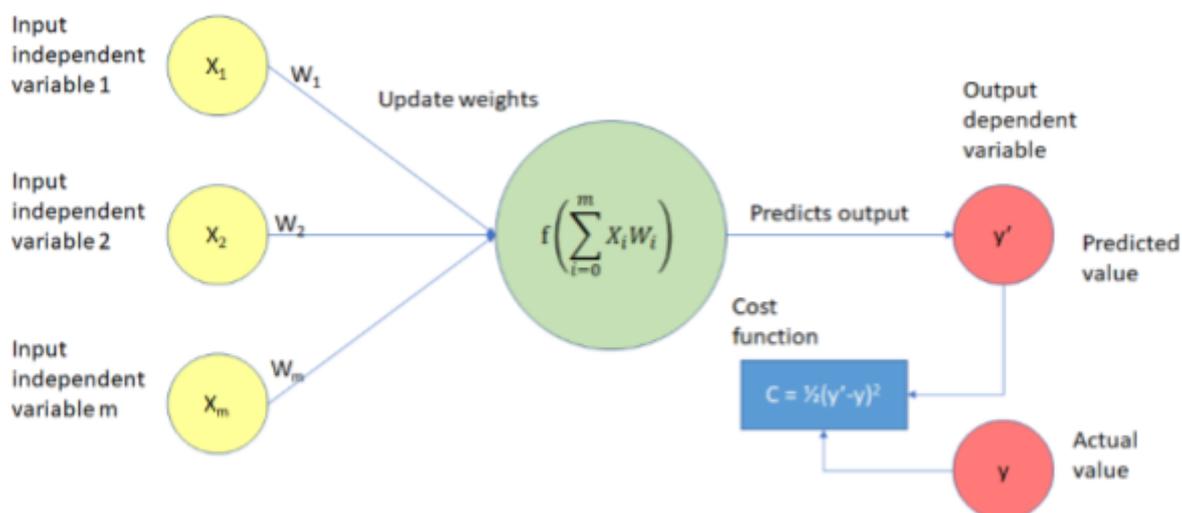


Figura 14: representación gráfica del cálculo de la función de coste dado un grupo de neuronas X_m , un valor de salida predicho por X_m denominado Y' y el valor de la salida deseada Y'

Mediante la aplicación de varias derivadas parciales se calcula el gradiente de la función, el cual informa sobre cuánto deben actualizarse los parámetros de la red para minimizar las diferencias entre la salida actual de la capa y la salida deseada en la última capa del modelo. Para ello hacen uso de lo que se conoce como la Regla de la cadena (Figura 15 [24]), que consiste en mirar atrás en la red y comparar con el fin de actualizar los pesos y *bias*. Cada vez que el peso de una determinada neurona tenga relación con otras neuronas de la RNA se hará uso de la Regla de la cadena que implicará el producto de las derivadas parciales con el fin de obtener el gradiente, que informará cómo debe ajustarse la RNA con el fin de obtener un resultado más prometedor mediante la observación continua de la función de coste que determinará la diferencia entre la salida actual del modelo y la salida deseada (figura 16 [25]). El proceso es diferente para la capas entrada/ocultas y la capa de salida debido a la predisposición sobre la salida

deseada para la muestra que se ha procesado. En el caso de la capa de salida pueden ocurrir dos escenarios diferentes dependiendo de su estructura:

- Se compone de una sola neurona: basta con calcular el error que ha cometido esta con respecto al resultado deseado.
- Se compone de más de una neurona: se calcula la sumatoria de los cuadrados de los errores de todas las unidades y se actualiza la capa

Luego, la capa de salida, enviará dicha información a su capa inmediatamente anterior. Esta capa oculta anterior hará los cálculos que hemos explicado anteriormente no solo actualizándose a sí misma sino también enviando información a la capa directamente anterior ya sea una oculta o la de entrada. Es por ello que la capa de entrada y las ocultas poseen un grado de dependencia considerable de la capa de salida y el conocimiento previo que se tiene sobre el conjunto de muestras ya que, la técnica *Backpropagation*, se basa en un proceso que comienza en la capa de salida y va propagándose hacia atrás hasta actualizar los hiperparámetros pertinentes de todas las neuronas que conforman la RNA. [23] [24] [25] [26] [27] [61].

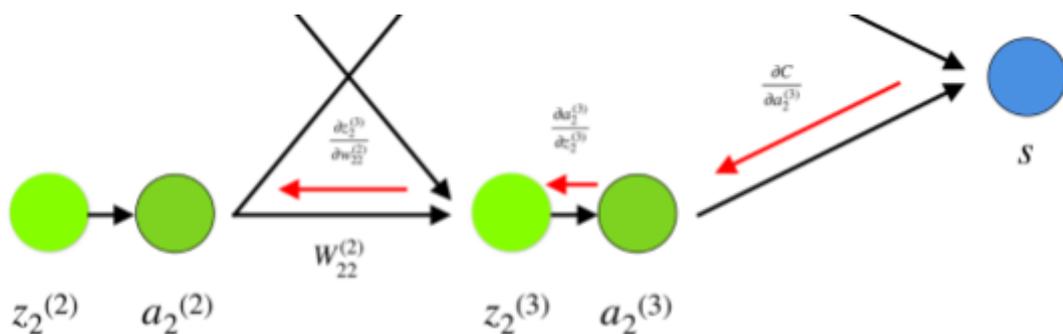


Figura 16: representación de la necesidad de la aplicación de la Regla de la cadena dada la relación entre un peso W y las unidades del modelo A y Z .

2.3.6 Tipos de redes

Las RNA se clasifican según las características que la definen y sus usos principales. Los tipos de principales son:

- Red neuronal monocapa: compuesta de dos capas simples, una de entrada y una de salida, cada una con un número de unidades determinadas (Figura 17 [28]).

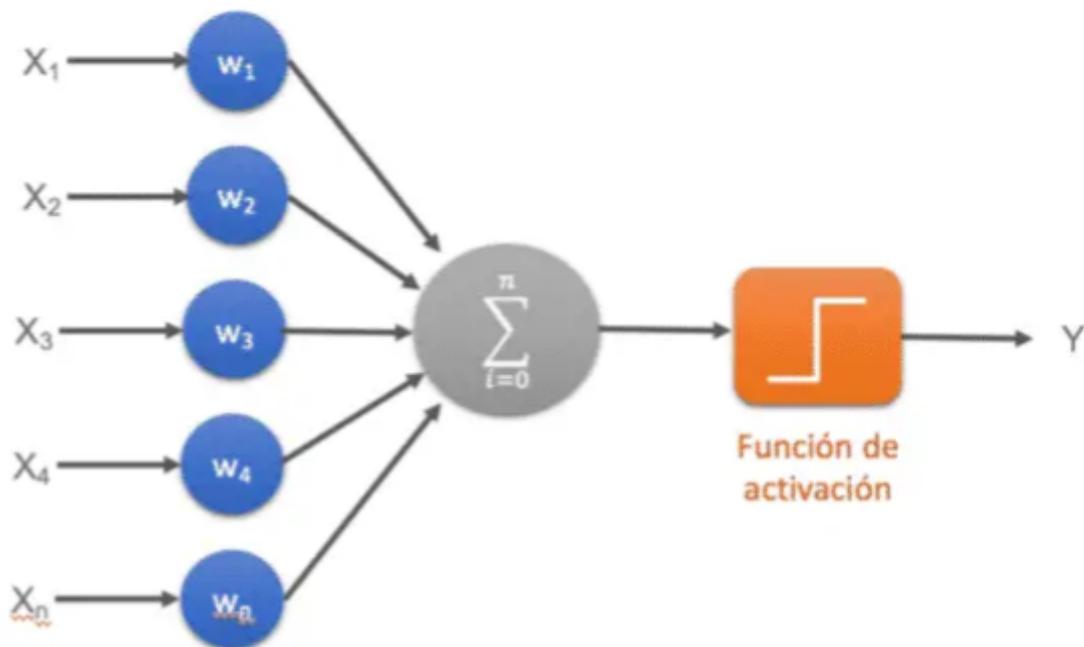


Figura 17: representación de una red neuronal monocapa

- Red neuronal multicapa (Perceptrón multicapa): se trata de una red compuesta de una capa de entrada, varias capas intermedias conocidas como capas ocultas y una capa de salida que designará el resultado final. El número de asociaciones entre las unidades de la red determina si la RNA es total o parcialmente conectada (Figura 18 [28])

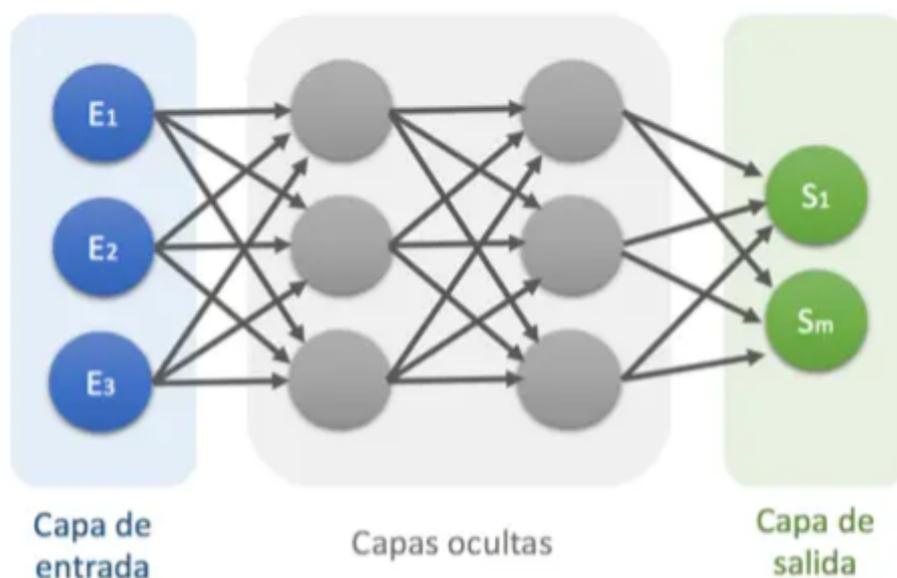


Figura 18: representación de un modelo de RNA multicapa

- Red Neuronal Convolutiva (CNN): se trata del tipo de RNA que se ha usado en este proyecto. Se define como un perceptrón multicapa con capas ocultas de tipo convolutivas y de submuestreo. Se especializan en el análisis de imágenes y son capaces de reconocer características en ellas. Su funcionamiento se basa en el análisis de la matriz de píxeles que conforman la imagen y la consecuente simplificación de esta a través de la RNA debido a que cada paso por cada capa convolutiva supone un aumento considerable de información en la salida, un defecto que se encargan de corregir las capas *subsampling* que simplifican e indican aquella información más relevante de la imagen. Durante este proceso la CNN recoge las características más importantes de las imágenes y las aprende con el fin de diferenciar un tipo de imagen de otra (Figura 19 [28]) [29].

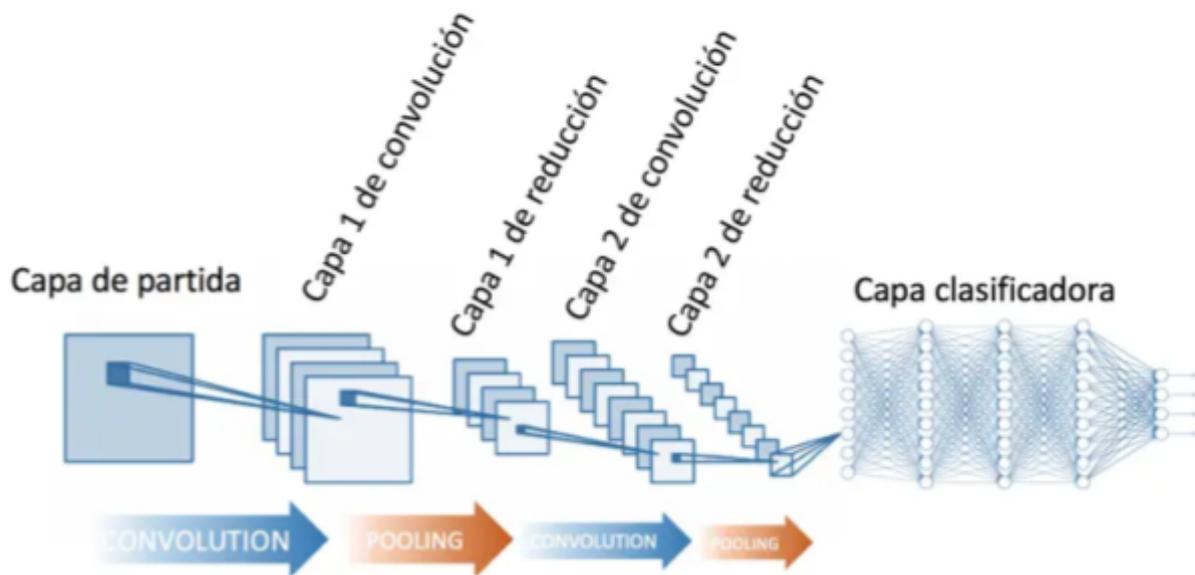


Figura 19: representación simplificada del modelo de una CNN

- Red neuronal recurrente (RNN): se trata de una red conocida por su capacidad de memorizar las ocurrencias sucedidas a lo largo del tiempo. No posee una estructura típica formada por capas sino que está permitida la conexión arbitraria entre neuronas permitiendo incluso ciclos (Figura 20 [28]).

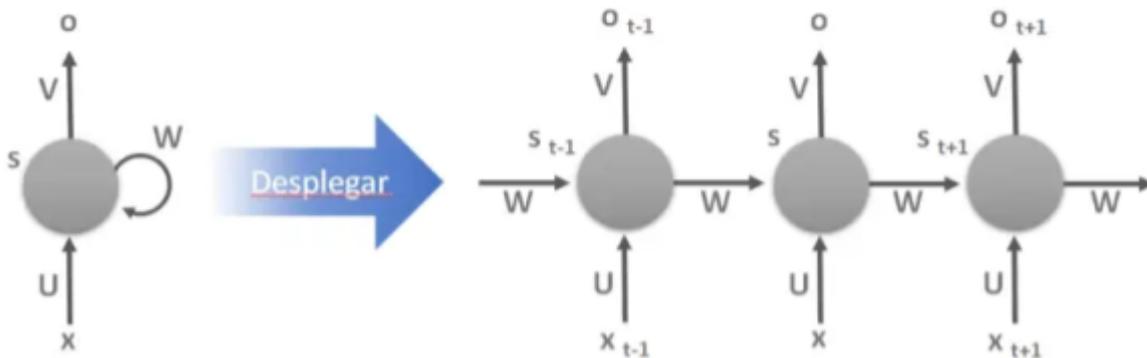


Figura 20: representación de un modelo de RNN

- Red neuronal de base radial (RBF): se trata de una RNA de conexión total que se caracteriza por el cálculo en función de la distancia a un centro. No se basa en capas sino en neuronas interconectadas y la salida proviene del resultado de una combinación lineal de varias funciones de activación radiales (Figura 21 [28]) [28].

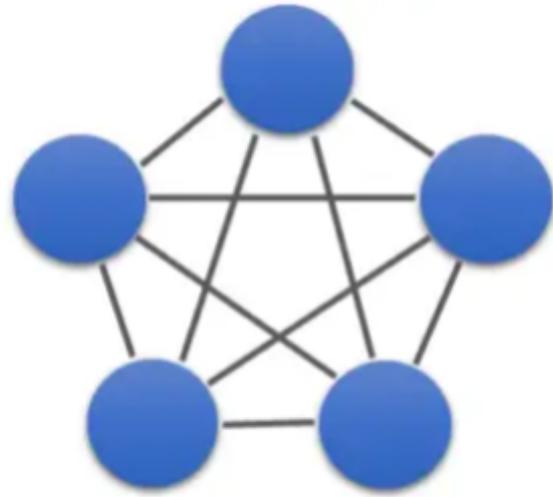


Figura 21: representación simplificada de una RBF

2.4 Red Neuronal Convolutiva (CNN)

Dentro de la variedad de subtipos de redes neuronales artificiales existe una en concreto que se especializa en la detección de patrones específicos que identifican una imagen. Este tipo de RNA es conocida como Red Neuronal Convolutiva (CNN en inglés) debido al nombre que identifica al proceso por el cual se detectan estos patrones de los que hablábamos anteriormente, la convolución.

“La CNN es un tipo de Red Neuronal Artificial con aprendizaje supervisado que procesa sus capas imitando al cortex visual del ojo humano para identificar distintas características en las entradas que en definitiva hacen que pueda identificar objetos y “ver”. Para ello, la CNN contiene varias capas ocultas

especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal.” [30]

2.4.1 Convolución

Las CNN basan su funcionamiento en el uso de procesos convolutivos. Estos procesos se llevan a cabo en las capas convolucionales ubicadas en la sección de las capas ocultas de la RNA. Gracias a la aplicación de una o varias convoluciones, el modelo es capaz de identificar, en principio, pequeños patrones como líneas y demás figuras simples. Con el paso de varias convoluciones estos patrones se vuelven más

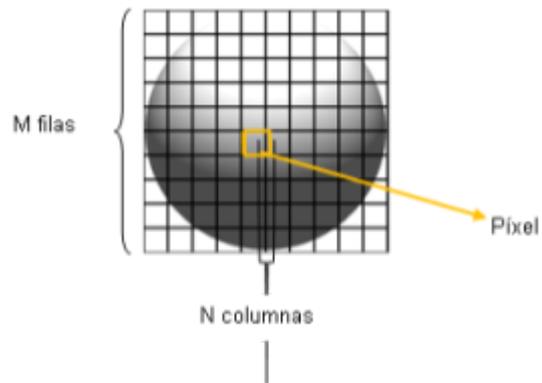


Figura 22: representación de un píxel

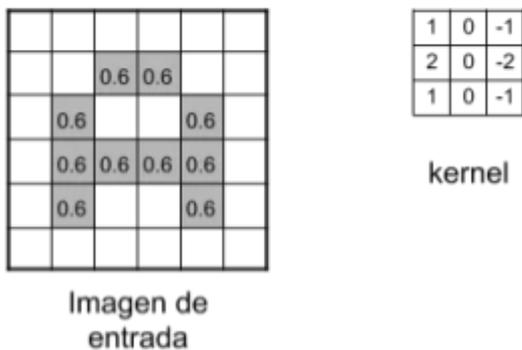


Figura 23: representación de una ventana de convolución y un filtro

complejos llegan a identificar elementos como ojos, manos o cualquier otro aspecto que pueda resultar representativo para la clasificación que se pretende.

Generalmente la entrada de este tipo de redes neuronales consiste en imágenes, las cuales, vistas desde un punto de vista matemático, son representadas por matrices numéricas que indican el valor almacenado en

cada píxel (Figura 22 [31]).

Las CNN no analizan la totalidad de los píxeles de manera individual sino que hacen uso de ventanas, que no son más que matrices de tamaño menor a la matriz de la imagen que representan una pequeña sección de esta. Estas ventanas van recorriendo la imagen, analizándola y descubriendo estos patrones que identifican a la representación de la entrada. El proceso es algo más complejo ya que no solo interviene esta ventana sino otra matriz del mismo tamaño denominada filtro o *kernel* (Figura 23 [30]). Para lograr la obtención de los patrones se realiza el producto escalar de la ventana y el *kernel*, obteniendo como resultado una matriz de valores diferente. En palabras de Baganto J. I. la convolución consiste en “...tomar *“grupos de píxeles cercanos” de la imagen de entrada e ir operando matemáticamente (producto escalar) contra una pequeña matriz que se llama kernel. Ese kernel supongamos de tamaño 3×3 pixels “recorre” todas las neuronas de entrada (de izquierda-derecha, de arriba-abajo) y genera una nueva matriz de salida, que en definitiva será nuestra nueva capa de neuronas ocultas.*” [31].

El número de *kernels* de las capas se puede especificar y está además estrechamente ligado a la salida de las capas ocultas convolucionales ya que se obtendrán tantas nuevas matrices de píxeles generadas como *kernels* se apliquen [30], es decir, si analizamos una capa convolucional que tenga implementada una cantidad de 32 filtros obtendremos 32 nuevas imágenes procesadas por convolución. Cada una de estas convoluciones no exporta una matriz sin procesar, sino que sus valores pasan por una función de activación. Por norma general, la función de activación más común en capas convolucionales es la función ReLU debido a que proporciona resultados más certeros en lo que al análisis pixelar se refiere ya que las imágenes por naturaleza no contienen características lineales y ReLU tiende a intensificar esta no linealidad, lo cual hace más sencilla la búsqueda de patrones identificadores [32].

La aplicación de varias convoluciones sin simplificar conlleva un aumento considerable de la dimensión de los datos debido al número de *kernels* aplicados.

Este aumento de tamaño hace que la RNA consuma una cantidad cuantiosamente mayor de los recursos. Es por ello que es necesario aplicar una extracción y simplificación de los datos mediante un proceso conocido como *subsampling* [30].

2.4.2 Subsampling

Subsampling es una técnica de simplificación de muestras de tipo imagen. Está estrechamente relacionada con las CNN debido a, como mencionamos anteriormente, el crecimiento de las dimensiones de las salidas de las capas convolucionales de la RNA. En palabras de Richmond Alake “*Sub-sampling is a technique that has been devised to reduce the reliance of precise positioning within feature maps that are produced by convolutional layers within a CNN.*

CNN internals contains kernels/filters of fixed dimensions, and these are referred to as feature detectors. Once features from an image are detected, the information in regards to the position of the feature within the image can actually be disregarded, and there are benefits to this.

It turns out that specific feature positioning reliance is a disadvantage to building and developing a network that can perform relatively well on input data that have undergone some form of an affine transformation. We mostly don't want the weights within the networks learning patterns that are too specific to the training data.” [33].

Existe una variedad cuantiosa de tipos de *subsamplings*, sin embargo en este informe me limitaré a la explicación del *Max Pooling*, ya que es la que usa la CNN desarrollada. Para la aplicación de cualquier *pooling* es necesaria la definición de un tamaño de ventana o matriz de *pooling*, la cual delimitará el área de simplificación. Dicha ventana irá recorriendo los mapas de características originados por la capa convolucional más reciente e irá simplificando la información de diferentes maneras. En el caso de *Max pooling*, la capa *subsampling* simplificará todos los valores a uno solo que será el valor mayor de todos los presentes en la

matriz de *pooling*. Una vez finalice el procesamiento del *Max pooling*, el resultado será una nueva matriz de píxeles representando una versión simplificada del mapa de características [33] [34]. En la figura 24 [34] puede verse un ejemplo ilustrativo de cómo sería el proceso.

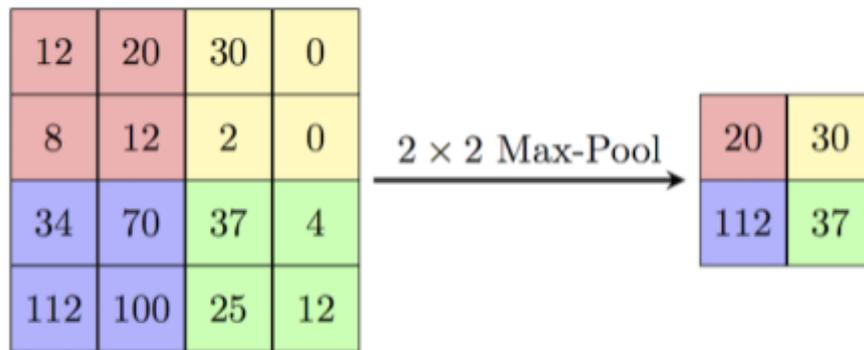


Figura 24: representación de un subsampling del tipo Max pooling y la matriz resultante

2.4.3 Análisis de la precisión y la pérdida del modelo

Toda RNA genera ciertos datos sobre su aprendizaje una vez acaba el entrenamiento que se le ha definido. Estos datos pueden ser utilizados y representados con el fin de analizar el comportamiento de la red dadas las entradas, los hiperparámetros y las capas implementadas. Generalmente la mejor forma para observar qué tan bueno es un modelo de una red es representar la precisión, conocida como *accuracy*, y la función de pérdida o pérdida a secas, conocida como *loss*. Para cada una de ellas surgirán dos vertientes diferentes:

- Datos de entrenamiento: son aquellos datos (80% del total) sobre precisión y pérdida que nacen del propio conjunto de entrenamiento.
- Datos de validación: son aquellos datos (20% del total) sobre precisión y pérdida que nacen del conjunto de validación conformado por entradas nunca vistas por la red.

En el caso de la representación del *loss* lo ideal es que esta empiece en un punto alto de la gráfica durante las primeras épocas y vaya bajando conforme avanza en las iteraciones (Figura 25 [35]). Ocurre lo contrario en el caso de la gráfica de *accuracy* debiendo empezar en un punto bajo en la gráfica para ir ascendiendo poco a poco hasta estabilizarse (Figura 25 [35]).

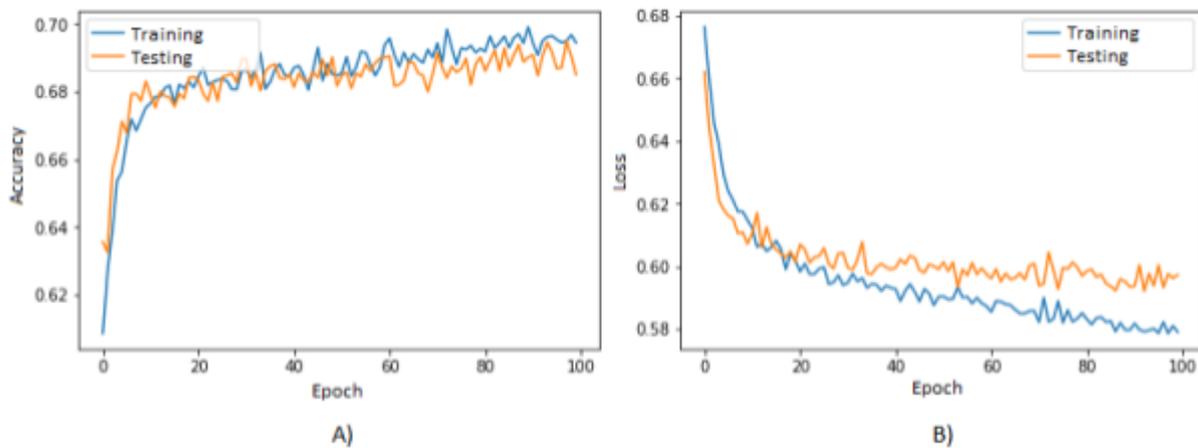


Figura 25: ejemplos de gráficas ideales de la precisión y la pérdida de un modelo de red neuronal

Gracias a la representación de estos gráficos es sencillo identificar posibles problemas propios del entrenamiento de las redes neuronales. Los más frecuentes son:

- *Overfitting*: sucede cuando la red lleva a cabo un entrenamiento demasiado específico y funciona de manera excepcionalmente con el conjunto de entrenamiento pero de manera cuestionable con entradas que no pertenecen a este. Este problema es uno a evitar ya que como desarrolladores se

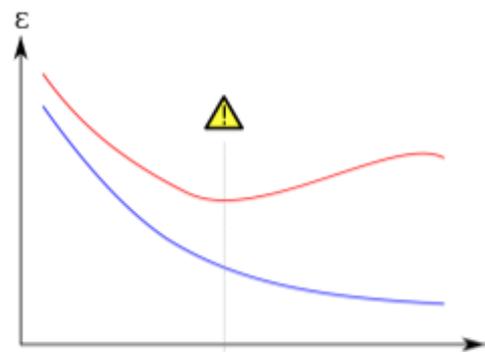


Figura 26: ejemplo de gráfica con overfitting

pretende construir un modelo que funcione bien tanto para el conjunto de entrenamiento como para cualquier otra entrada ajena a este [36] [37] (Figura 26 [37])

- *Underfitting*: sucede cuando el modelo generado no es útil para el problema que se intenta solventar ya que no da buenos resultados ni para el conjunto de entrenamiento ni para el de la validación (Figura 27 [38]) [36] [37].

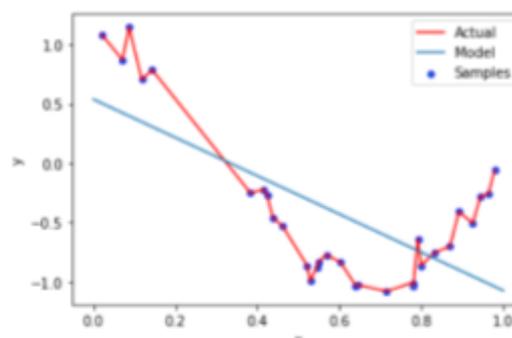


Figura 27: ejemplo de gráfica con underfitting

2.4.4 Técnicas de preprocesado de imágenes

En muchas ocasiones el preprocesamiento de las imágenes antes de introducirlas en la CNN da a lugar a mejores resultados en el proceso de aprendizaje. Es por ello que es usual aplicar técnicas como las que se explican a continuación.

2.4.4.1 Padding

En algunos casos las CNN tienden a clasificar mejor cuando se les añade “relleno” o acolchado en las imágenes tanto en los procesos convolutivos como en los de submuestreo. La técnica mediante la cual se añade el acolchado a las imágenes durante su transcurso por los modelos de CNN se conoce como *Padding* y es útil en aquellos casos en los cuales los extremos superior, inferior y laterales de la imagen tienen características notables en lo que a clasificación respecta, por lo que puede resultar en un cambio considerable en el resultado final del aprendizaje

de la RNA. El porqué de esta mejora se debe principalmente a que la ventana de convolución no analiza el mismo número de veces los píxeles más extremos de la imagen, por lo que la posibilidad de que opte por tenerlos en cuenta como información destacada se reduce. En las figuras 28 [39] y 29 [40] podemos apreciar la imagen resultante tras la aplicación de *Padding*. [39]

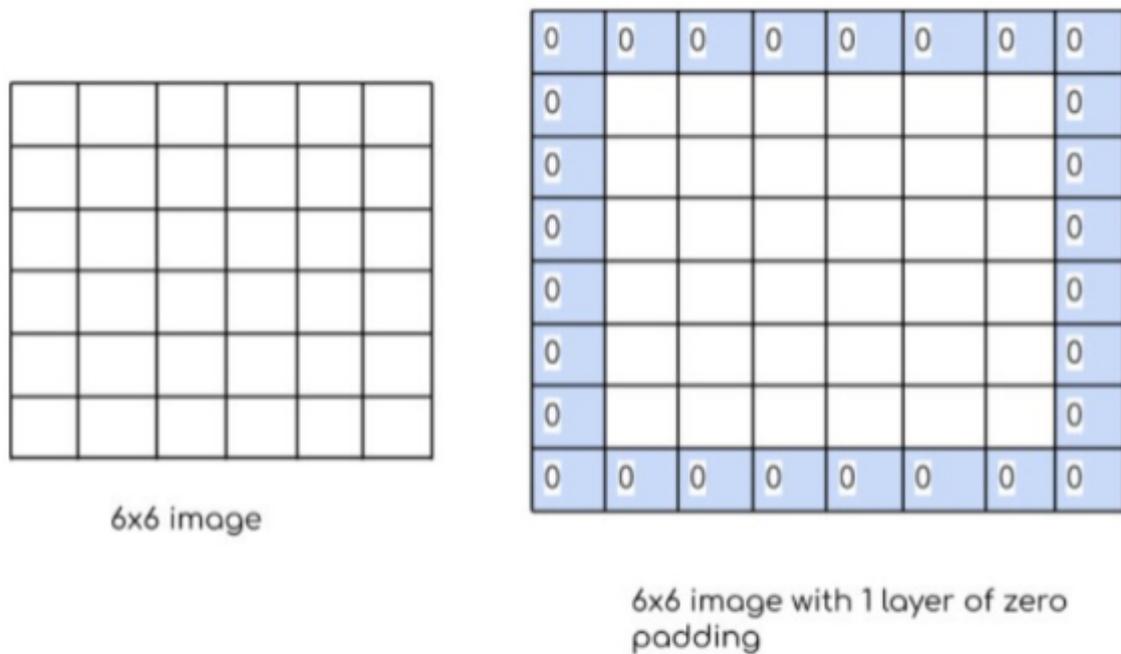


Figura 28: representación matricial de una imagen sin padding (matriz izquierda) y una imagen con padding (matriz derecha)

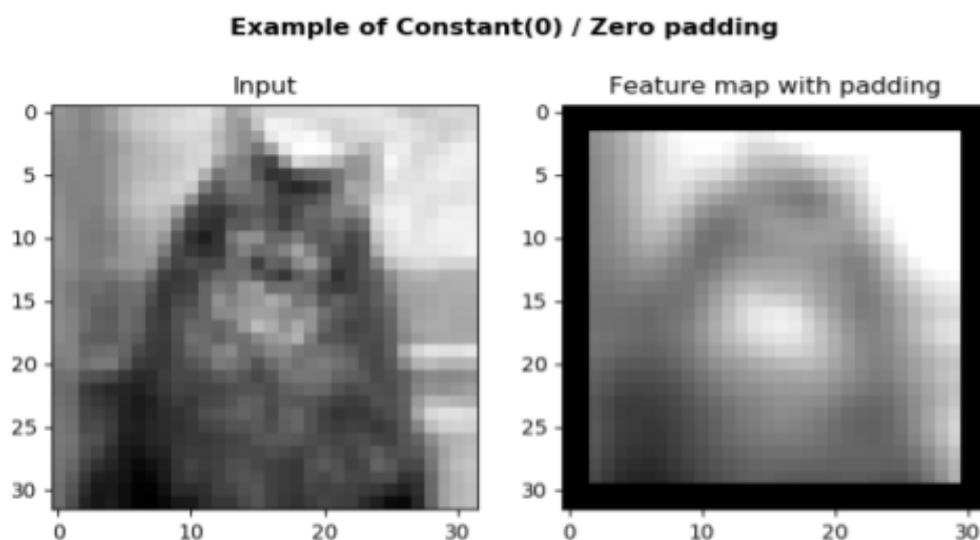


Figura 29: representación del proceso de padding en una imagen

2.4.4.2 Matriz de confusión

Los valores de precisión y pérdida que nos suministran los modelos son idóneos para la verificación de diversos problemas del entrenamiento como el *overfitting*, sin embargo, no debemos confiar en que los resultados obtenidos reflejan la precisión real del modelo. Existen métodos con los cuales podemos verificar verazmente que nuestro modelo hace predicciones correctas y precisas o, al menos, verificar el grado de exactitud real de estas y hacer modificaciones según las necesidades de la RNA. Uno de estos métodos es el conocido como Matriz de confusión que consiste en la detección de los verdaderos y falsos positivos y negativos de la predicción del modelo. Las magnitudes medibles para una clase A y otra B son cuatro [58]:

- **Exactitud o verdaderos positivos:** representa la cercanía de un valor predicho con el valor real del mismo objeto, o lo que es lo mismo, la cantidad de predicciones que fueron correctas.
- **Precisión:** porcentaje de predicciones de una clase A detectadas por el modelo.
- **Sensibilidad o tasa de verdaderos positivos (Sensitivity):** proporción de predicciones correctas para una clase A que fueron correctamente identificadas.
- **Especificidad o tasa de verdaderos negativos (Specificity):** proporción de predicciones correctas para una clase B que fueron correctamente identificadas.

$$\text{Exactitud} = \frac{VP + VN}{\text{Total}}$$

$$\text{Precisión} = \frac{VP}{VP + FP}$$

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

Matriz de confusión

Verdaderos positivos	Falsos positivos
Falsos negativos	Verdaderos negativos

Figura 30: formulación de las áreas de una matriz de confusión

2.4.4.3 Curva ROC

La técnica conocida como Curva ROC (*Receiving Operating Characteristic Curve*) tiene la capacidad de ilustrar de forma simple cuánto de sensible y específico son los resultados de un modelo de RNA determinado. Se compone de una gráfica con ejes X e Y, representando en el eje X la sensibilidad del modelo (capacidad de clasificar verdaderos positivos correctamente) y en el eje Y representando 1 - especificidad (grado de falsos positivos obtenidos), aunque la representación de

estos ejes varía según el objetivo con el cual son generadas. La forma de interpretar este tipo de gráfica es sencilla, se trata de hacer que el área bajo la curva representada sea la mayor posible, lo cual indicaría que nuestro modelo presenta un comportamiento ideal. Si analizamos la figura 31 [41] observamos un conjunto de curvas ROC que representan cómo de bueno es un modelo de RNA según el área bajo la curva [42] [43] [44] [59].

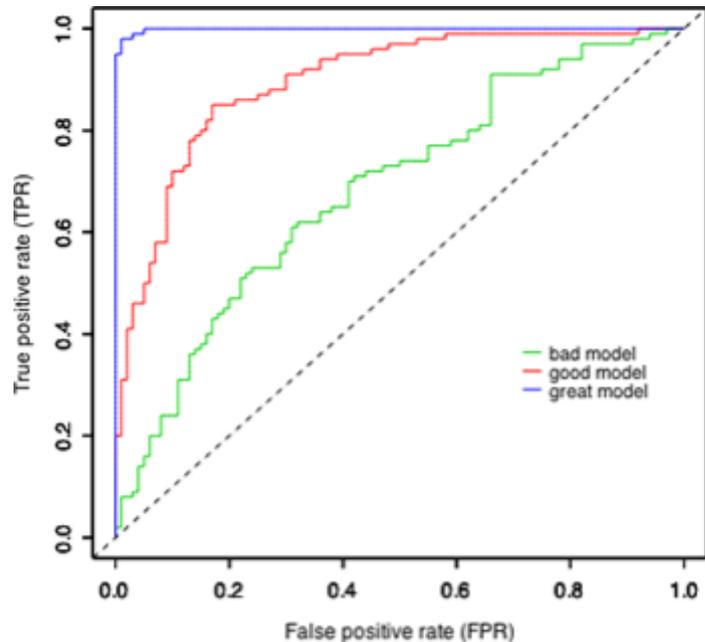


Figura 31: representación de varias curvas ROC y cómo de prometedor sería el modelo al que están sujetas

Capítulo 3: Sistema inteligente basado en redes neuronales profundas de ayuda al diagnóstico de la COVID-19

3.1 Conjuntos de datos

Para el aprendizaje de la CNN se hizo uso de 2 conjuntos de entrenamiento diferentes. Uno de ellos se compone de pruebas médicas de tipo RX mientras que el último se conforma de TC.

3.1.1 Radiografía (RX)

Una radiografía es una prueba médica que hace uso de rayos-X con el fin de detectar problemas en aquellas zonas o componentes más densos de la anatomía del paciente. Este tipo de pruebas son especialmente útiles para identificar incidencias o patologías óseas, pero también son eficaces para la detección de afecciones de algunos órganos dependiendo de su ubicación como son los pulmones, el corazón, los vasos sanguíneos, etc. Para la correcta interpretación de la información se tiene especial interés en las tonalidades de la prueba ya que aquellas áreas con tonos blancos son aquellas pertenecientes a aquellos elementos con mayor densidad dentro del espectro mientras que aquellos con tonalidades más oscuras son aquellos elementos con densidad baja [45] (Figura 32 [45]), entre ellos el aire que se representa en color negro.

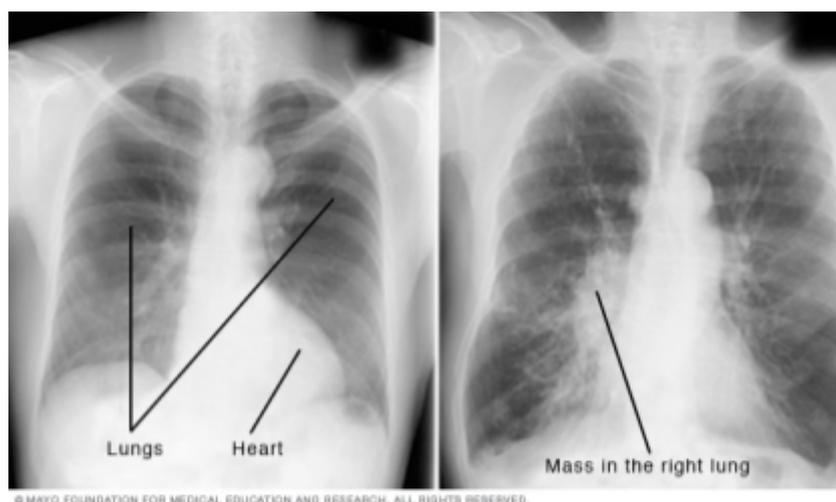


Figura 32: ejemplos de radiografías

3.1.2 Tomografía computarizada (TC)

Las pruebas tomográficas son pruebas realizadas con el fin de detectar o controlar afecciones óseas o de algunas otras zonas de la anatomía del sujeto (tumores, patologías musculares, patologías óseas, pulmones, corazón, etc). Su

funcionamiento se basa en la combinación del uso de rayos-X y la obtención de varias instantáneas, las cuales se combinan formando una única imagen resultante (Figura 33 [46]). Su precisión y claridad son considerablemente mayores al de las RX, es por ello que, en algunos casos, los expertos deciden realizar TC en lugar de RX dependiendo de la sintomatología o de si el paciente está en algún tipo de tratamiento [46] [47].

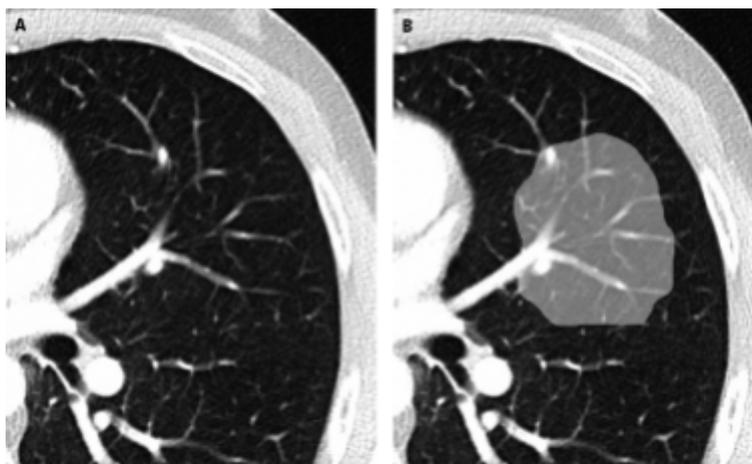


Figura 33: tomografía con signos de infección por COVID-19

3.1.3 Descripción de los *datasets*

Como se describió anteriormente, se cuenta con 2 *datasets* diferentes. A continuación los definiremos individualmente:

- **Dataset 1:** creado con fines de investigación, se compone de radiografías y se encuentra ya clasificado en pruebas con signos de infección por COVID-19 y pruebas libres de esta [48]. Está respaldado por el *University of Montreal's Ethics Committee* y nace de la necesidad de datos para el desarrollo específico de aplicaciones de ayuda al diagnóstico de COVID-19 y detección de las posibles complicaciones que este produce. Se trata de un proyecto público al cual muchas entidades pueden contribuir de diferentes maneras, respetando siempre un protocolo que garantiza la veracidad de los

datos [49] [50]. El objetivo principal del proyecto es apoyar la creación de herramientas que ayuden a los expertos en la detección temprana de estas patologías y sirvan como segunda opinión en casos difusos o de complejidad considerable [51] [52] [53]. Actualmente las muestras contienen las patologías definidas en la Figura 34 [51].

Type	Genus or Species	Image Count
Viral	COVID-19 (SARSr-CoV-2)	468
	SARS (SARSr-CoV-1)	16
	MERS-CoV	10
	Varicella	5
	Influenza	4
	Herpes	3
Bacterial	<i>Streptococcus</i> spp.	13
	<i>Klebsiella</i> spp.	9
	<i>Escherichia coli</i>	4
	<i>Nocardia</i> spp.	4
	<i>Mycoplasma</i> spp.	5
	<i>Legionella</i> spp.	7
	Unknown	2
	<i>Chlamydomphila</i> spp.	1
Fungal	<i>Pneumocystis</i> spp.	24
	<i>Aspergillus</i> spp.	2
Lipoid	Not applicable	8
Aspiration	Not applicable	1
Unknown	Unknown	59

Figura 34: tabla de contenido de las pruebas del Dataset 1.1

- Dataset 2:** este *dataset* está formado por pruebas de tipo tomográficas recopiladas de una variedad cuantiosa de *papers* relacionados con la patología (*medRxiv*, *NEJM*, *JAMA*, *Lancet*, etc). El proyecto ha sido apoyado por un experto radiólogo del hospital de Yuhan tras varios problemas acontecidos en los orígenes debido a la supuesta influencia de la calidad de las imágenes en los sistemas que se pretende desarrollar con los datos. Finalmente el experto afirmó que la calidad de las pruebas es adecuada tanto para el análisis proveniente de un experto humano como para el análisis de un sistema inteligente [54] [55].

3.1.4 Procesamiento de las imágenes

Tanto en el *dataset* 1 como en el 2 las imágenes habían sido preprocesadas desde la fuente desde la que se obtuvieron, sin embargo existen algunas diferencias entre ambos:

- *Dataset* 1: como mencionamos anteriormente se compone de imágenes modificadas mediante una técnica conocida como *Data augmentation* (DA) o Aumento de datos [Figura 35]. Gracias a ello el volumen de datos del *dataset* aumenta en gran cantidad y ha sido suficiente para la realización del entrenamiento, validación y pruebas. Debido a la preparación tan completa de este conjunto de muestras no ha sido necesario procesar las imágenes más allá de redimensionarlas a unas dimensiones de 200x200 y transformar los valores RGB para definirlos en un rango de números entre 0 y 1 para un mejor análisis por parte de la CNN.

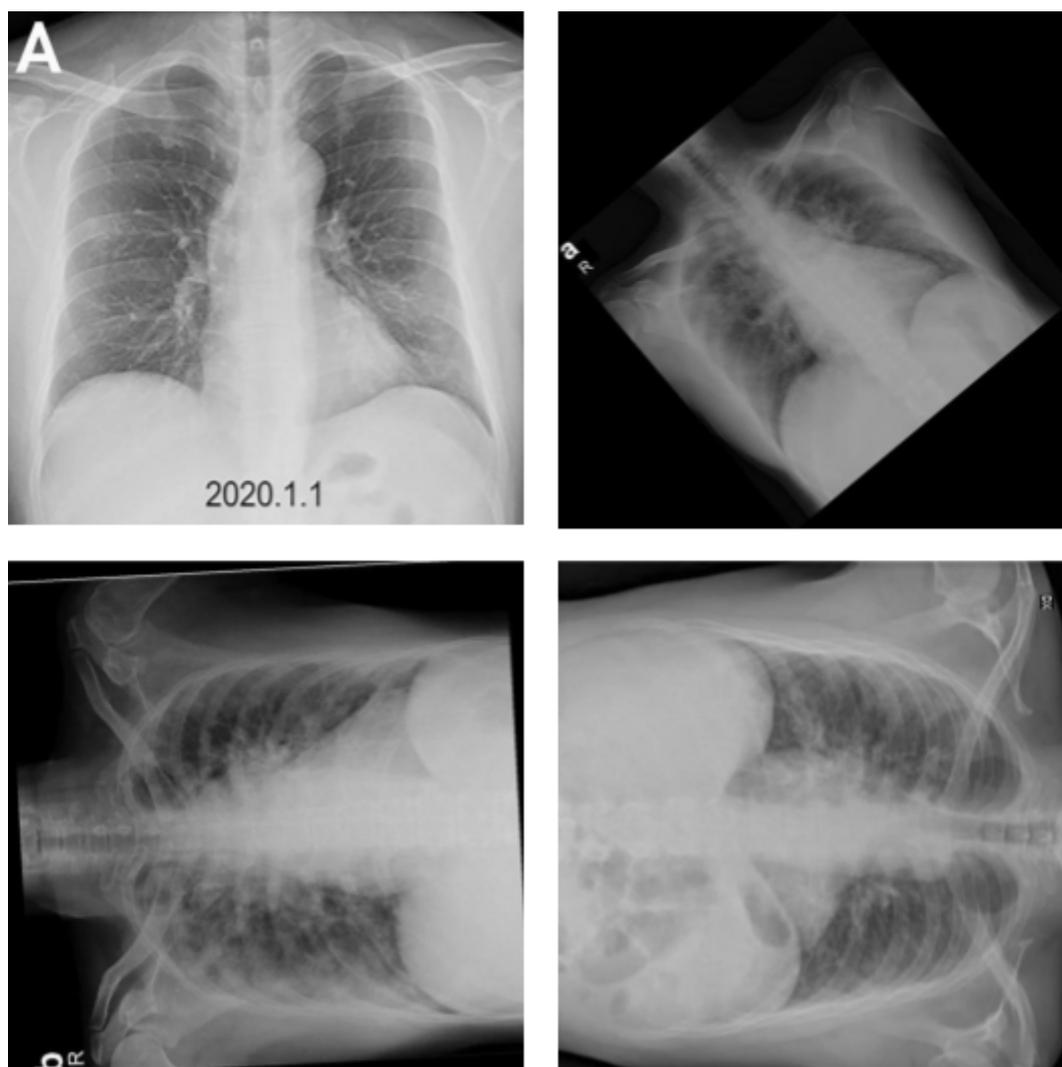


Figura 35: ejemplo de muestras de RX provenientes del dataset 1

- *Dataset 2*: el caso de este conjunto de muestras ha sido diferente. Debido a la escasez de muestras con las que contaba no era posible obtener resultados certeros y concluyentes sobre la capacidad del modelo de clasificar correctamente las imágenes de las TC. Como consecuencia se desarrolló un proyecto paralelo [56] con el fin de obtener una nuevo *dataset* a

partir de técnicas de DA consistentes en la modificación del brillo y contraste, el cual consiguió impulsar el potencial del modelo de las TC considerablemente. La decisión de incidir en estas dos características se basó en el análisis de las imágenes al observar que algunas de ellas eran más oscuras o más brillosas que otras. La CNN no las clasificaba tan bien como las otras debido posiblemente a la escasez de muestras de dicha índole. La variación tanto del brillo como del contraste se basó en el aumento o disminución del contraste y brillo en un rango comprendido entre un -50% y un 50% [Figura 36]. Se trató también de aplicar rotaciones y recortes aleatorios dentro de unos rangos determinados y controlados, sin embargo, la CNN no proporcionó resultados mejores con la inserción de este tipo de muestras aumentadas.

Además de las modificaciones aleatorias, se llevaron a cabo redimensionamientos de 200x200 y la normalización de los valores RGB de los píxeles entre 0 y 1.

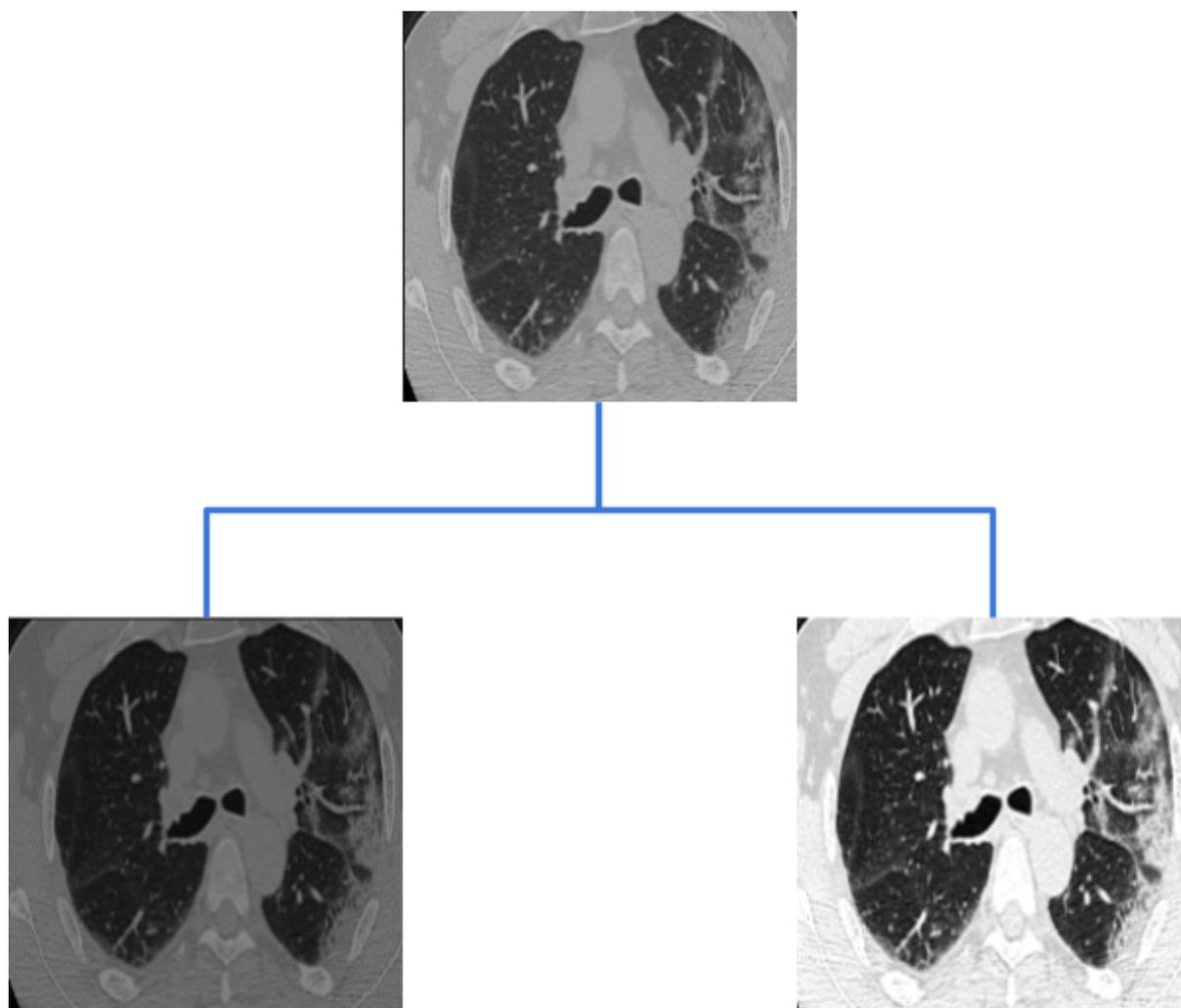


Figura 36: comparación entre la TC original (imagen superior) y las dos TC obtenidas mediante técnicas de DA (imágenes inferiores)

3.2 Metodología y análisis de los modelos desarrollados

Como ya citamos anteriormente se han desarrollado 2 modelos diferentes dependiendo del tipo de muestras utilizadas, RX o TC. A pesar de tener una

arquitectura de capas común, el *modus operandi* de estas es diferente en cada modelo.

Ambas CNN han sido entrenadas para un número de épocas máximo de 60 para el aprendizaje, sin embargo, tienen implementados funciones de tipo *callback* con diversos objetivos [57]:

- *Callback* de parada de aprendizaje: tiene la capacidad de concluir el entrenamiento en caso de que no haya mejora a lo largo de un número de 10 épocas. Por norma general, las sesiones de aprendizaje de ambas CNN no superan las 20 épocas ya que una vez superado este número de épocas el aprendizaje tiende a volverse poco o nada fructífero.
- *Callback* de alteración del *Learning Rate* (LR): se encarga de ir reduciendo el ratio de aprendizaje a lo largo del proceso de entrenamiento de la CNN con el fin de ayudar al modelo a realizar clasificaciones más precisas. Si llega un momento en el cual el LR deja de ser reducible sin que afecte negativamente al modelo la reducción cesará.
- *Callback* de parada por *NaN*: previene la continuación del entrenamiento en el caso en el que el modelo llegue a obtener valores de pérdida *NaN*.

3.2.1 Modelo CNN de las RX

El modelo desarrollado para las RX se basa en una CNN conformada por una capa de entrada seguida de 3 pares de capas convolutivas y reductoras (*subsampling*) para terminar una capa de tipo aplanador (*flatten*) y una última capa de salida con una función de activación sigmoide, debido a la binariedad de las clases que presenta el problema (muestra de individuo sano o muestra infectada por COVID-19). En la figura 37 es posible observar un esquema del modelo que se ha implementado. Las capas convolutivas reciben una imagen de determinadas dimensiones y crean una muestra con mucha más información, es por ello que se genera la necesidad de la aplicación de capas de tipo *Max Pooling Subsampling*

(MPS) debido a que estas son capaces de simplificar la información y enfocar las características más relevantes.

Con esto se logra reducir las dimensiones de la imagen y enviarla a la siguiente capa convolutiva o a la capa *flatten* conectada a la de salida. En un principio la imagen será de dimensiones 200 píxeles de ancho por 200 píxeles de largo con 3 canales correspondientes a los canales RGB de colores. A medida que la imagen es analizada por la red, el número de unidades que analizan la imagen aumenta mientras que las dimensiones disminuyen de acuerdo al *subsampling* aplicado. Una vez finalizado todo el proceso de análisis de

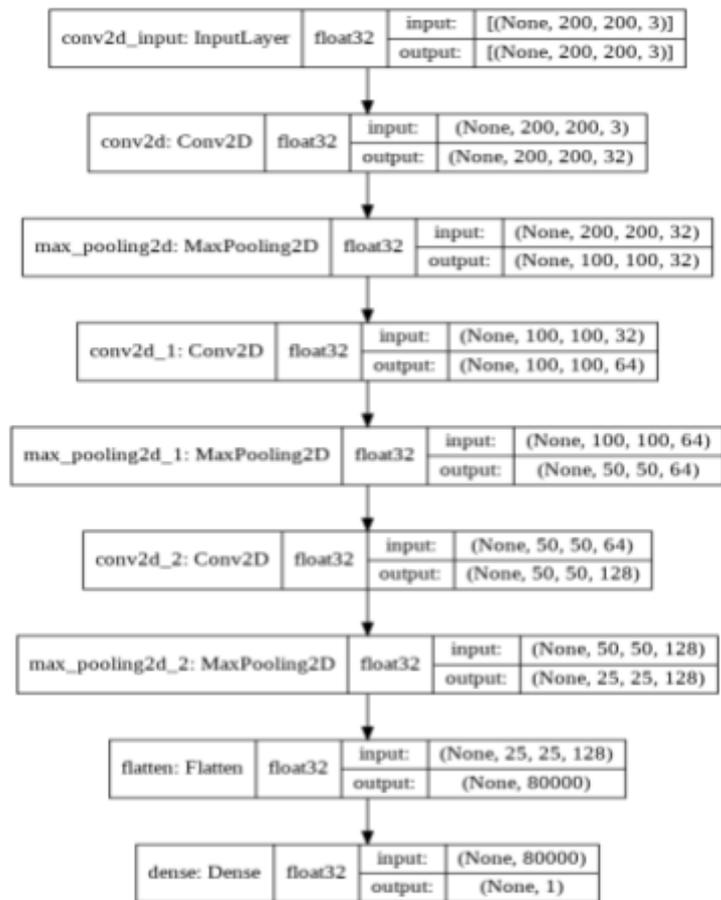


Figura 37: representación del modelo de CNN para las RX

cada una de las muestras del conjunto la información necesita ser representada en forma vectorial en lugar de matricial con el fin de poder ser finalmente tratada por la capa de salida que se encargará de clasificar la muestra en no infectada o infectada.

Las características de cada tipo de capa son las siguientes:

- Capas convolucionales:
 - Número de neuronas, que comienza en 32 para la primera capa convolucional para acabar en una última conformada por 128.

- Ventana de convolución de tamaño 5x5 píxeles.
- Dimensiones de la entrada (solo en el caso del contacto con la primera capa convolucional) 200x200.
- Tipo de *Padding same*, dadas las pruebas que demostraron la superioridad de este frente a la ausencia de *padding* (Figura 38).

No padding (valid) y Zeros padding (same)

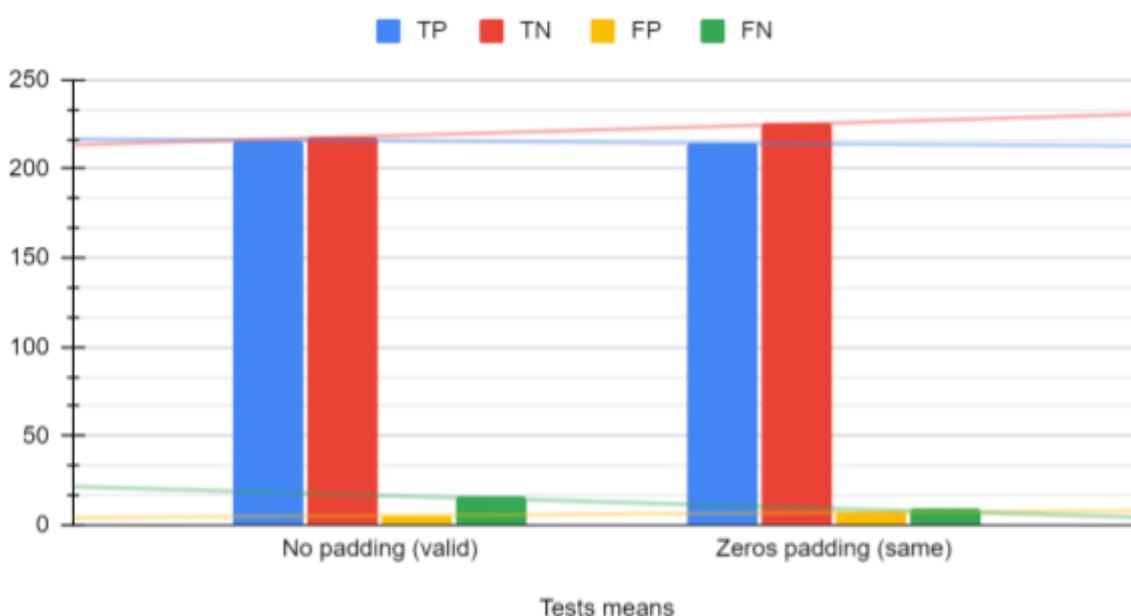


Figura 38: representación de los VP, VN, FP, y FN según el padding utilizado en las capas convolutivas y subsampling. Las líneas de tendencia demuestran que el padding Same reduce los FN.

- El inicializador de pesos aplicado es el conocido como *Truncated normal*. La decisión de aplicar este inicializador se basó en una serie de pruebas que dieron lugar a la elección (Figura 39, 40, 41). Los inicializadores de peso determinan los valores con los que comienzan los pesos de las unidades de las capas. Esto influye de forma notable en muchas ocasiones en la obtención de resultados prometedores por parte de las RNA.

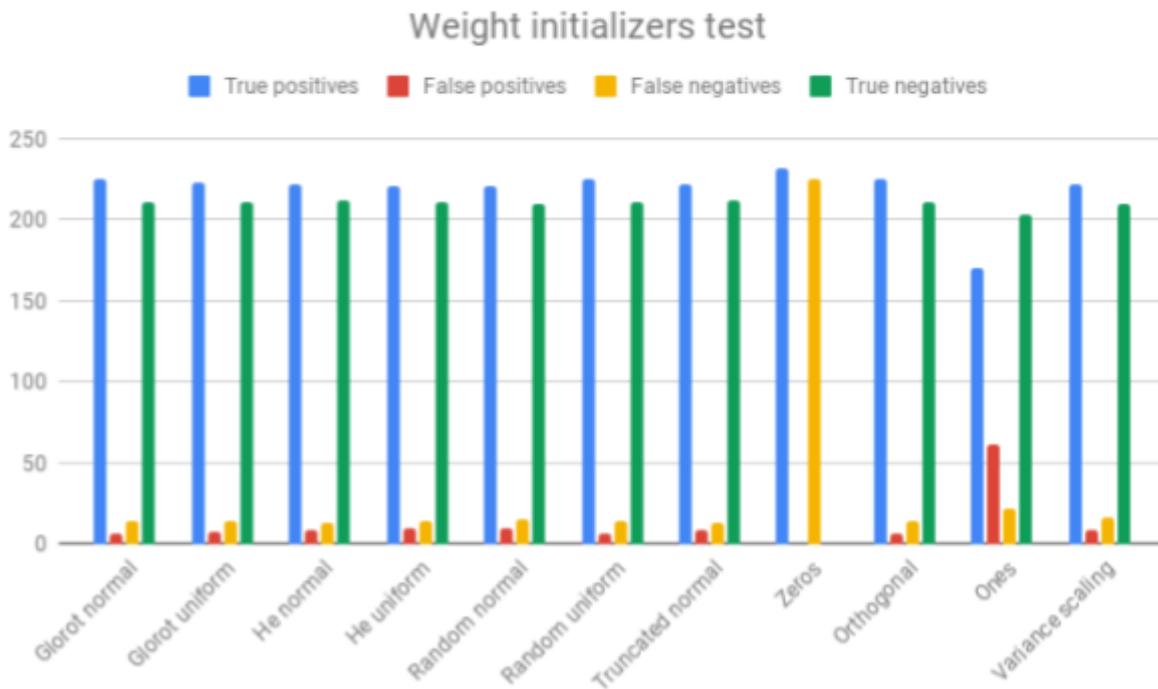


Figura 39: diagrama de barras sobre los VP, VN, FP y FN obtenidos mediante los diferentes inicializadores de pesos



Figura 40: diagrama de barras sobre los FN según el inicializador de pesos utilizado

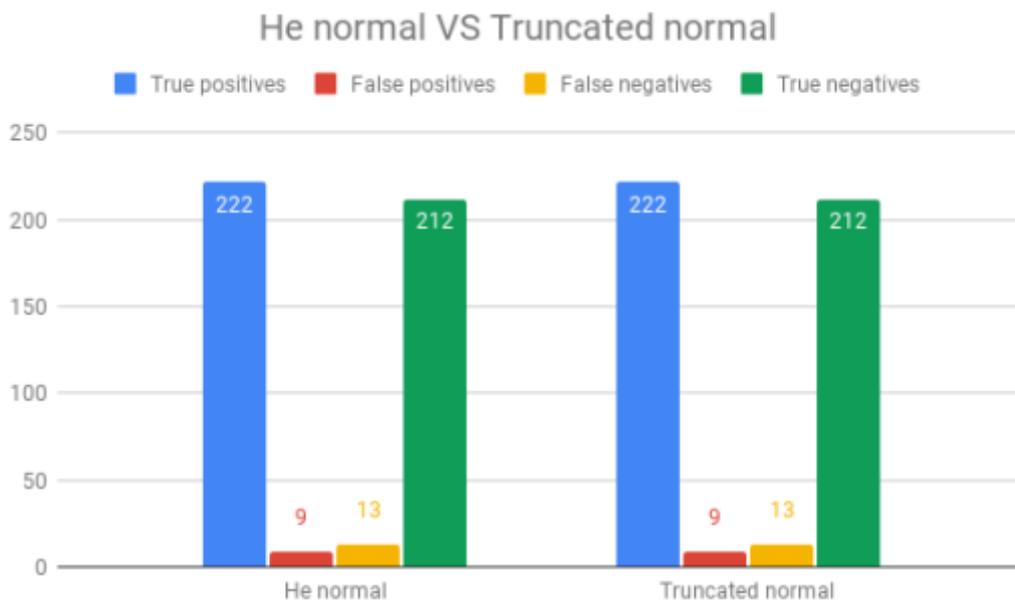


Figura 41: diagrama de barras sobre VP, VN, FN, FP de los inicializadores de pesos He normal y Truncated normal



Figura 42: diagrama de barras representativo de las inestabilidad de resultados del uso del inicializador de pesos Ones

- PReLU como función de activación dados los resultados obtenidos en las pruebas con el modelo (Figura 43 y 44).

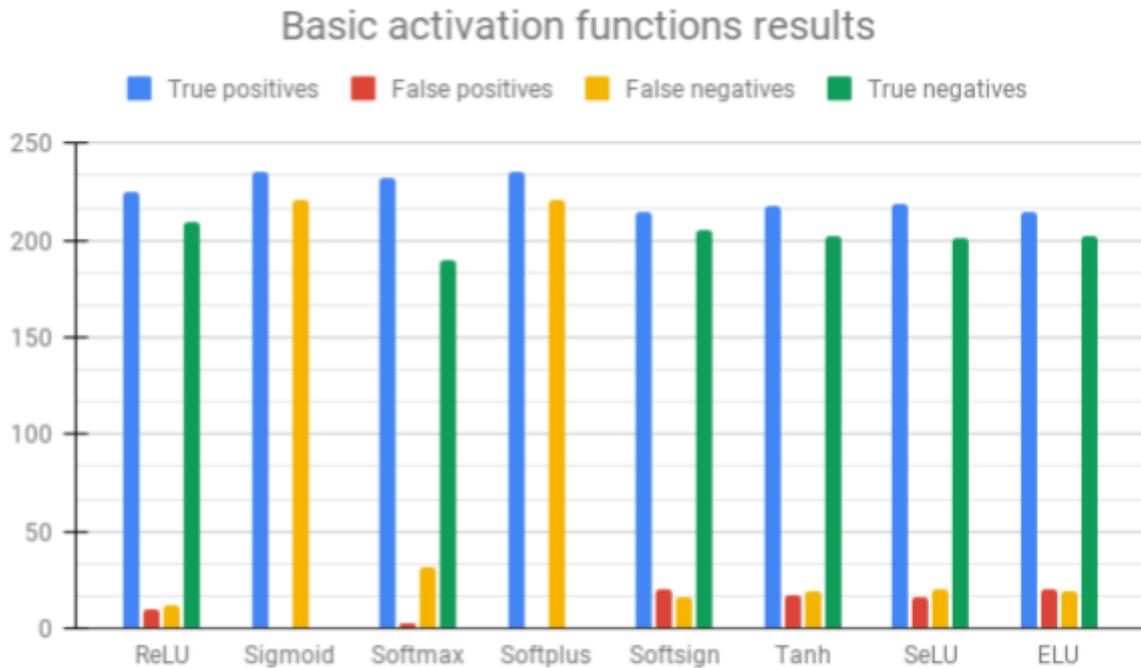


Figura 43: diagrama de barras representativo de los resultados obtenidos en las pruebas con las diferentes funciones de activación básicas para el modelo RX

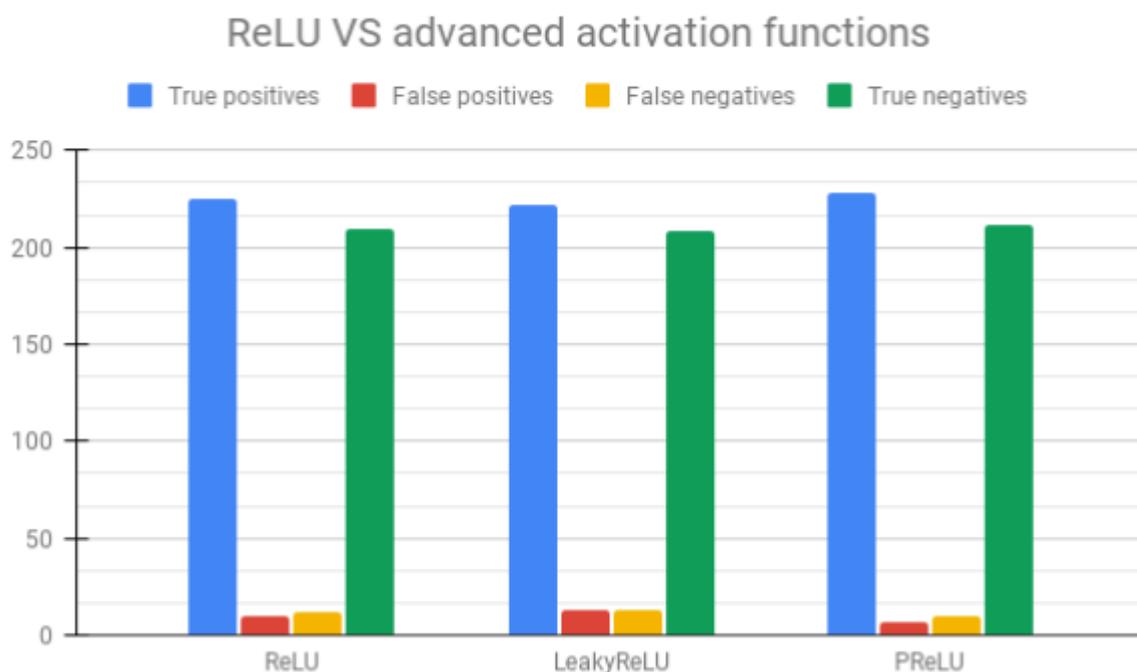


Figura 44: diagrama de barras sobre los resultados obtenidos mediante las funciones de activación ReLU y sus hermanas más avanzadas

- Capas MPS:
 - Tamaño de ventana de *pooling* de 2x2.
 - Tipo de *padding* *same*.
- Capa *flatten*: ninguna característica especial. Sirve como traductor de la información para la capa de salida.
- Capa de salida:
 - Una neurona.
 - Aplicación de la función de activación sigmoide para la clasificación de las muestras.

Una vez el modelo está construido compila con una función de pérdida de tipo *Binary crossentropy* y usando el optimizador Adam, el cual dio los mejores resultados entre todos los probados para la reducción de los falsos negativos, que era la característica a reducir viendo los resultados obtenidos en un principio (Figura 45).

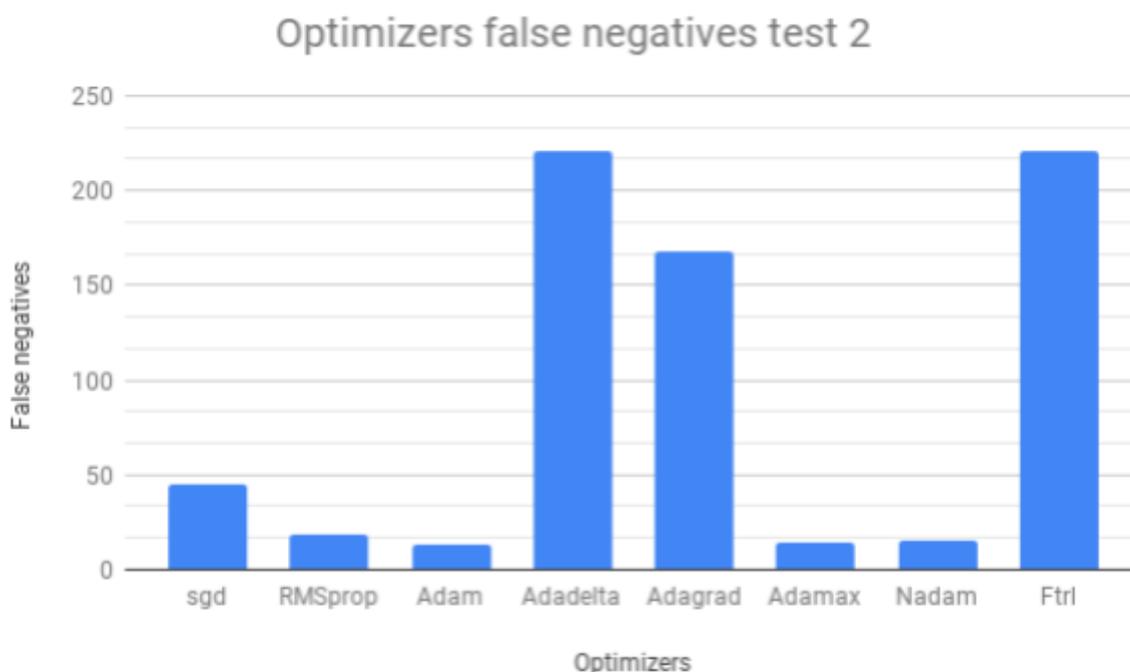


Figura 45: diagrama de barras representativo de una de las pruebas que se realizaron sobre los optimizadores

En definitiva, a continuación se expone en forma de listado las características que definen el modelo:

- Épocas: máximas 60, modificables dinámicamente gracias a los *callbacks*.
- Ventana convolucional de 5x5 píxeles.
- Ventana de filtro de 2x2 píxeles.

- *Padding* tanto de convolución como de *subsampling* de tipo *Same* de acuerdo a las herramientas suministradas por Keras.
- Tamaño de lote de 50.
- Optimizador de modelo Adam.
- Función de pérdida *Binary Crossentropy*.
- Porcentaje del *dataset* para la validación de un 20% del total de este.
- Función de activación PReLU para las capas convolutivas.
- Función de activación Sigmoide para la capa de salida.
- Inicializador de pesos *Truncated Normal*.
- Dimensiones de las entradas de 200x200 por 3 canales representando los RGB.
- Número de unidades que componen las capas convolutivas (en orden):
 - 32 neuronas.
 - 64 neuronas.
 - 128 neuronas.

3.2.2 Modelo CNN de las TC

El modelo para las TC es muy parecido al anterior. La arquitectura es semejante, una capa de entrada seguidas de tres pares de capas de convolucionales y MPS para finalmente acabar en una capa *flatten* conectada a la de salida (Figura 46).

Las diferencias entre ambas CNN radican únicamente en las características internas de las capas convolucionales:

- Número de neuronas ascendentes desde 32 hasta 128.
- Ventana de convolución de 5x5.
- Dimensiones de la entrada (solo en el caso del contacto con la primera capa convolucional) 200x200.

- Tipo de *padding same*, debido a las pruebas que se realizaron que demostraban que este tipo de *padding* lograba reducir el número de FN (Figura 47).

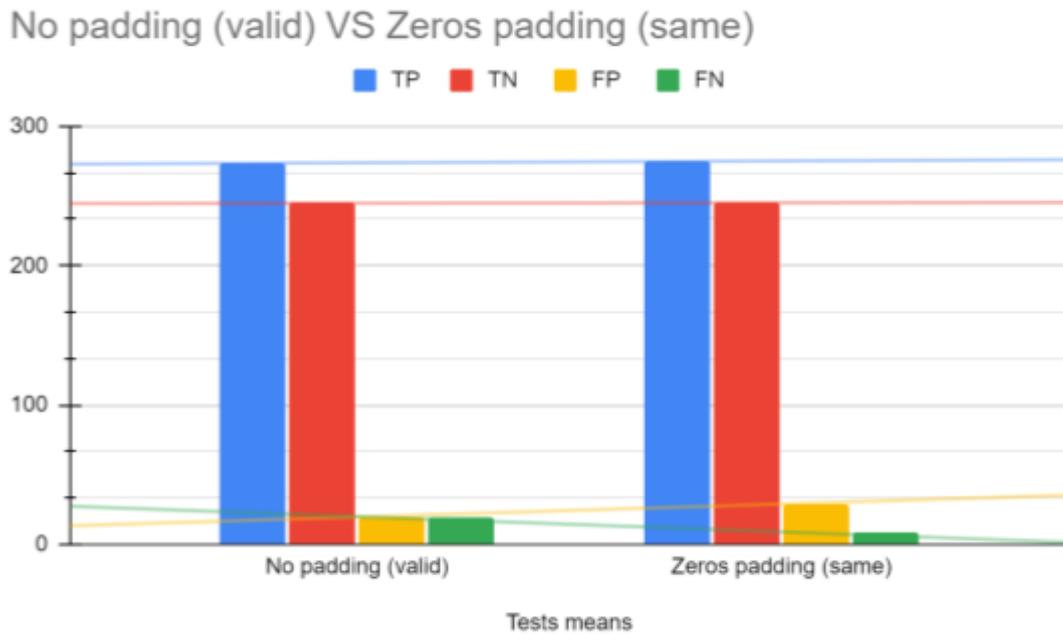


Figura 47: representación de los VP, VN, FP, y FN según el padding utilizado en las capas convolutivas y subsampling. Las líneas de tendencia demuestran que el padding Same reduce los FN.

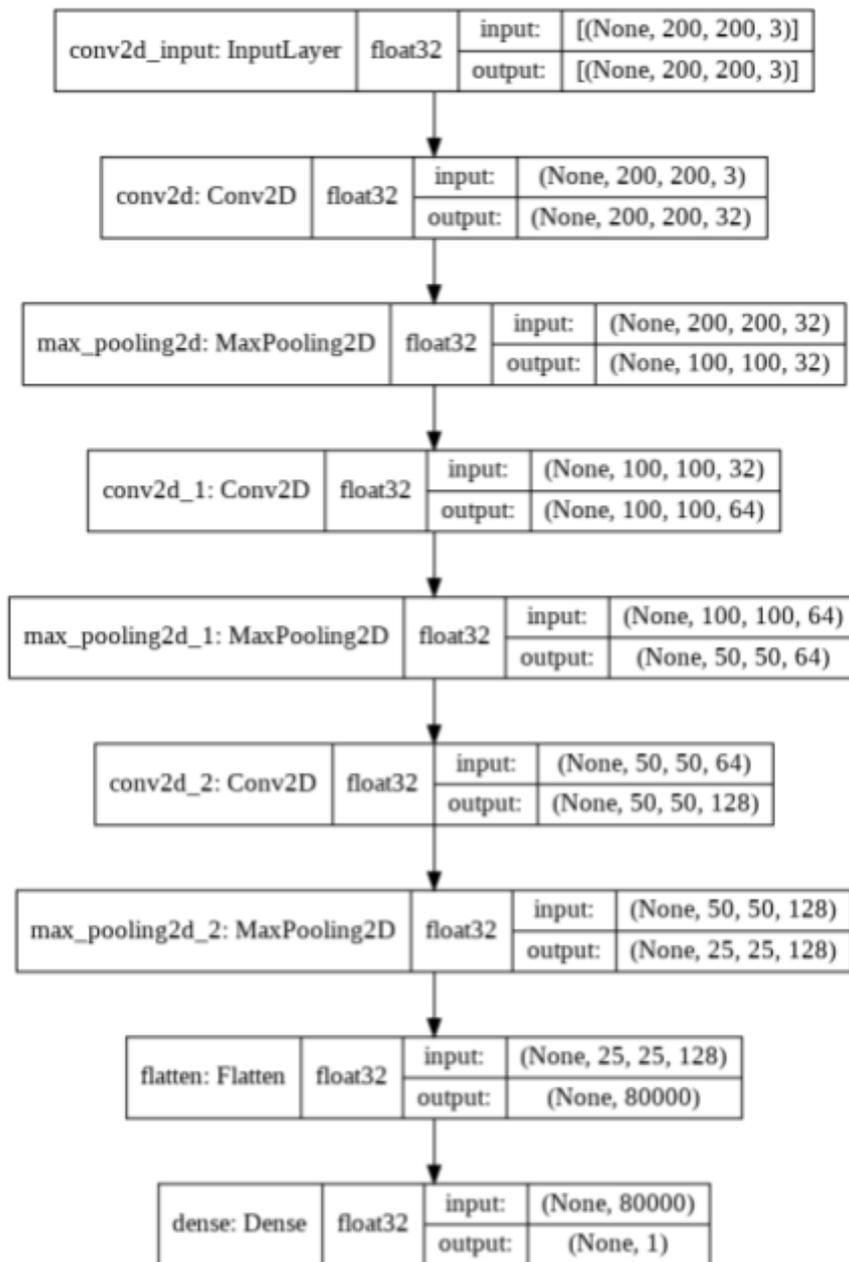


Figura 46: representación del modelo de CNN para las TC

- El inicializador de pesos aplicado es el conocido como *Glorot normal* vistas las pruebas realizadas sobre el modelo TC (Figura 48, 49 y 50).



Figura 48: Resultados de las pruebas realizadas con los inicializadores de pesos disponibles

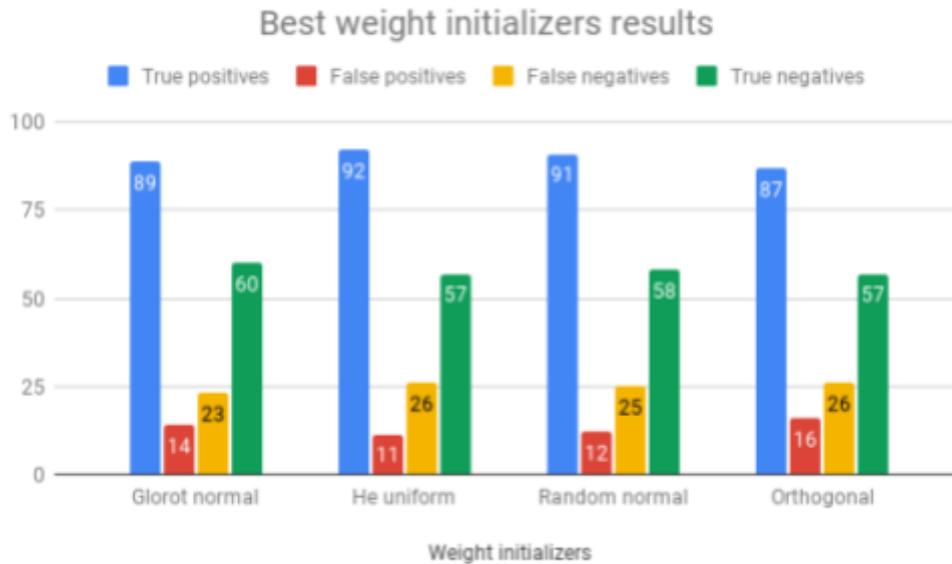


Figura 49: representación simplificada de los mejores optimizadores de pesos para el modelo TC



Figura 50: diagrama de barras representativo de las inestabilidad de resultados obtenidos mediante el inicializador de pesos Ones

- ReLU como función de activación dados los resultados obtenidos en las pruebas (Figura 51 y 52).

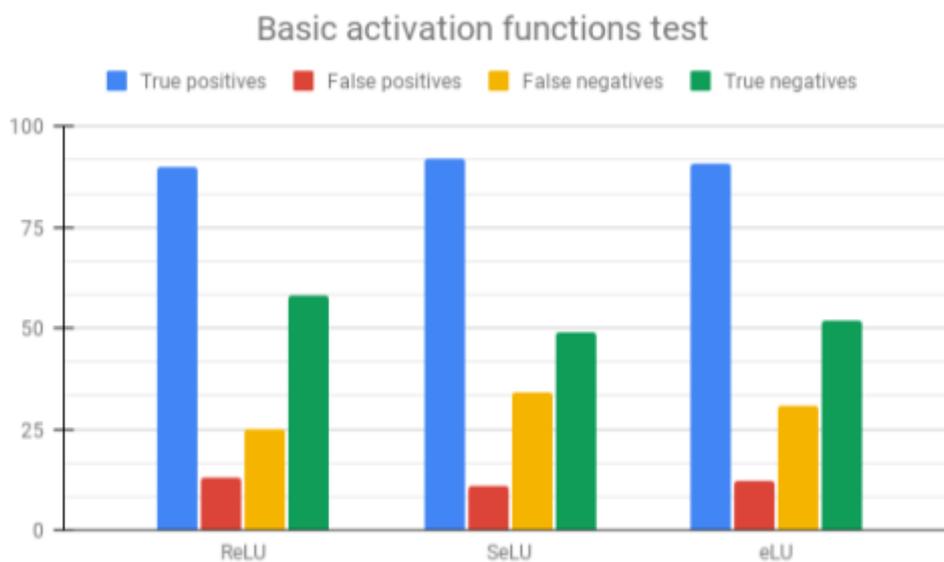


Figura 51: diagrama de barras representando los resultados obtenidos con las funciones de activación de la familia ReLU

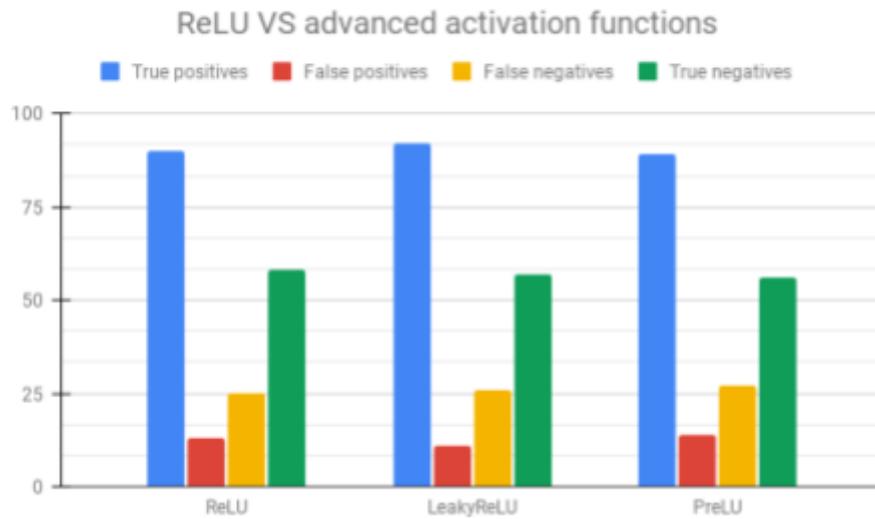


Figura 52: diagrama de barras representativo de las ventajas de ReLU frente a sus funciones hermanas más avanzadas para el modelo TC

Con el modelo ya construido se compila con una función de pérdida *Binary crossentropy* y el optimizador *Adam* dados los resultados obtenidos durante las pruebas realizadas sobre el modelo (Figura 53).

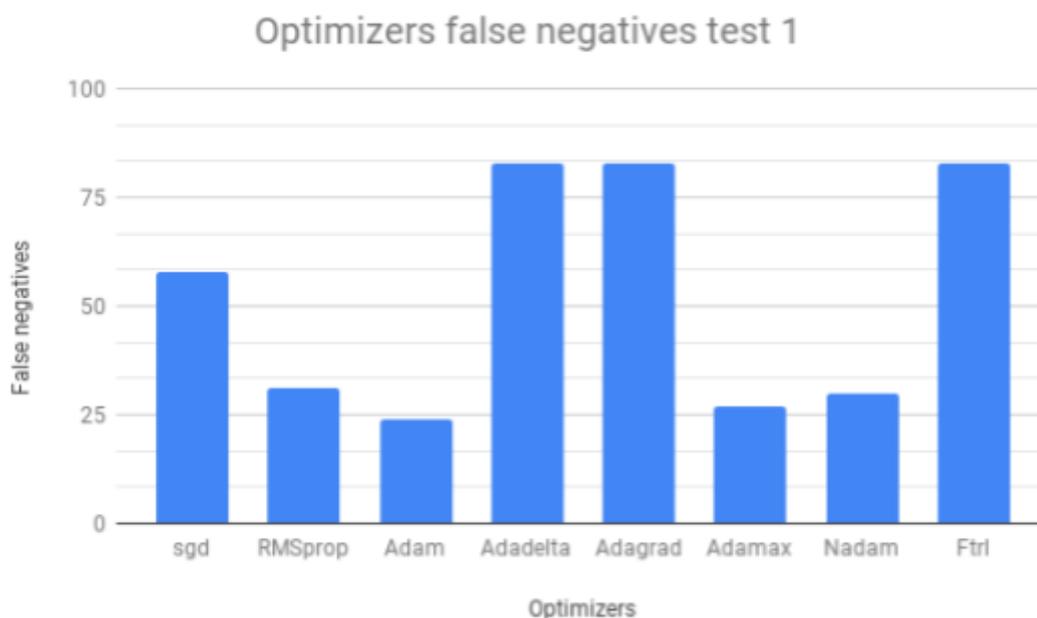


Figura 53: muestra de uno de los diagramas de barras de los falsos negativos obtenidos con los diferentes optimizadores disponibles

En definitiva, a continuación se expone en forma de listado las características que definen el modelo:

- Épocas: máximas 60, modificables dinámicamente gracias a los *callbacks*.
- Ventana convolucional de 5x5 píxeles.
- Ventana de filtro de 2x2 píxeles.
- *Padding* tanto de convolución como de *subsampling* de tipo *Same* de acuerdo a las herramientas suministradas por Keras.
- Tamaño de lote de 50.
- Optimizador de modelo Adam.
- Función de pérdida *Binary Crossentropy*.
- Porcentaje del *dataset* para la validación de un 20% del total de este.
- Función de activación ReLU para las capas convolutivas.

- Función de activación Sigmoide para la capa de salida.
- Inicializador de pesos *Glorot Normal*.
- Dimensiones de las entradas de 200x200 con 3 canales representando los RGB.
- Número de unidades que componen las capas convolutivas (en orden):
 - 32 neuronas.
 - 64 neuronas.
 - 128 neuronas.

3.3 Resultados

El desarrollo de este proyecto ha conducido a una serie de resultados prometedores en lo que a las capacidades de las RNA en nuestra sociedad se refiere, concretamente, en el ámbito de la medicina. El aprendizaje de los modelos desarrollados y las consecutivas pruebas demuestran con creces lo útiles que podrían ser herramientas que hagan uso de redes CNN con el fin de ayudar al diagnóstico de enfermedades respiratorias como es el caso de la COVID-19. La rama de la salud y la rama tecnológica siempre han ido de la mano en la prevención y curación de las enfermedades, esta investigación demuestra que es hora de incluir inteligencia artificial en el movimiento y conducir a un camino de mayor seguridad diagnóstica.

Primeramente me gustaría hablar sobre las capacidades propias de la capas convolutivas y las capas *subsampling* (principales tipos que componen las CNN) en la búsqueda de características diferenciadoras en las imágenes así como la precisa clasificación que confieren. Se ha demostrado en varias ocasiones en diferentes experimentos el poder y lo lejos que pueden llegar este tipo de RNA. Su aplicación en la salud podría revolucionar la prevención y detección de afecciones aportando

no solo un nuevo punto de vista al ya dado por los especialistas en sí sino seguridad y solidez en los diagnósticos. Imaginemos por ejemplo una situación en la cual los especialistas duden de si el paciente sufre de una patología determinada basándose en unas pruebas realizadas sobre el paciente. Este tipo de herramientas podrían suponer un impulso significativo en los diagnósticos gracias a una mayor rapidez de actuación y un ahorro de recursos considerable a coste 0 en infraestructura más allá de lo que la CNN necesita.

Otro punto a tratar, sobre todo en el caso de las CNN, es la fuerte relación entre el estado, cantidad y calidad de los conjuntos de muestra en la obtención de un modelo de RNA fiable y sólido. Conseguir pruebas como RX o TC es más complicado que conseguir otros tipos de pruebas, más siendo de una calidad confiable y en una cantidad suficiente. El caso de los modelos de RX y TC no ha sido diferente. En el modelo de la CNN para RX la cantidad (1824 muestras, una mitad de COVID-19 y otra de muestras sanas) y la calidad fueron más que suficientes para conseguir resultados muy prometedores con un modelo de arquitectura sencilla (figura 54) debido no solo a que las muestras habían sido procesadas con anterioridad sino que se había aplicado DA sobre el *dataset* 1 haciendo que el volumen de datos fuera aún mayor. El caso de la CNN para TC fue más complejo debido a que el volumen de datos era muy reducido. Esto dió lugar a que el modelo no supiera estimar y clasificar con seguridad las muestras y los resultados de las primeras fases de las pruebas no fueron del todo precisos. Si observamos la figura 55 podemos apreciar los resultados obtenidos con el modelo final de CNN y el *dataset* 2 por defecto. Con el fin de comprobar y verificar que el volumen de datos es crucial para un aprendizaje fiable de una RNA me propuse desarrollar un proyecto paralelo con el fin de generar una versión aumentada del *dataset* 2 mediante técnicas de DA. Viendo que la CNN fallaba mayormente en casos de TC oscuras o más brillantes que el resto, me propuse generar una nueva

dataset que se conformara de réplicas de los datos pero modificados en contraste y brillo en un porcentaje aleatorio entre un -50% y un 50%. De esta manera se logró no solo aumentar el volumen de datos, sino preparar a la CNN de TC para pruebas no tan claras o más claras de lo normal. En la figura 56 podemos apreciar los resultados obtenidos con el mismo modelo para un *dataset 2* aumentado, no solo solo mejorando los resultados de precisión y pérdida en los procesos de entrenamiento y validación sino que el área bajo la Curva ROC es considerablemente mayor y los valores resultantes de la Matriz de Confusión son mucho mejores. Si calculamos porcentajes la mejora es llamativa (véase tabla a continuación).

	Dataset 2	Dataset 2 mejorado
Verdaderos positivos	37.6 %	45.5 %
Verdaderos negativos	37.1 %	44.6 %
Falsos positivos	10.2 %	9 %
Falsos negativos	15.1 %	0.9 %

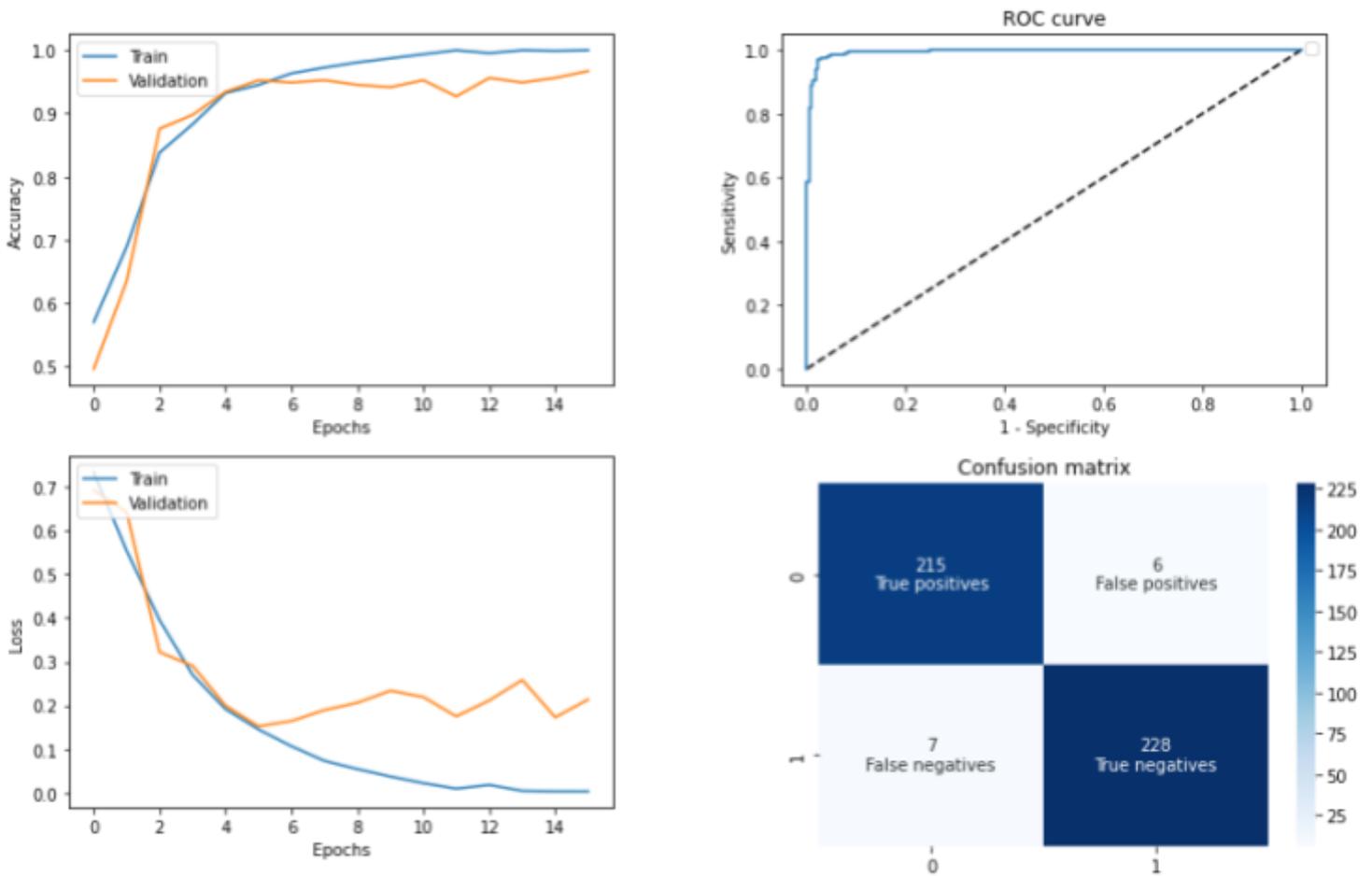


Figura 54: representación de la precisión, la pérdida, Curva ROC y Matriz de Confusión de la fase final del modelo de CNN para las RX

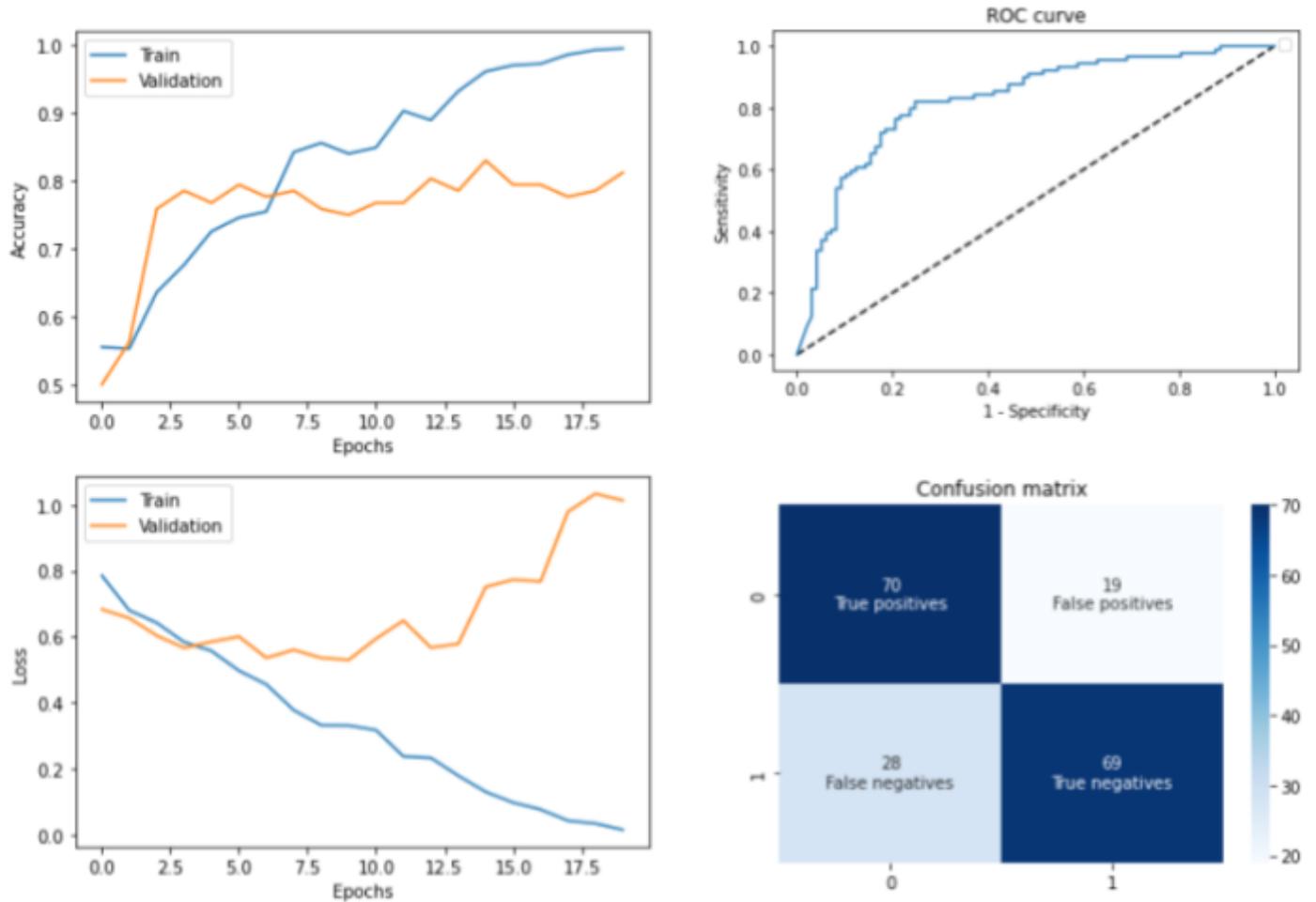


Figura 55: gráficas de los resultados obtenidos con el modelo TC sin la aplicación de técnicas de DA

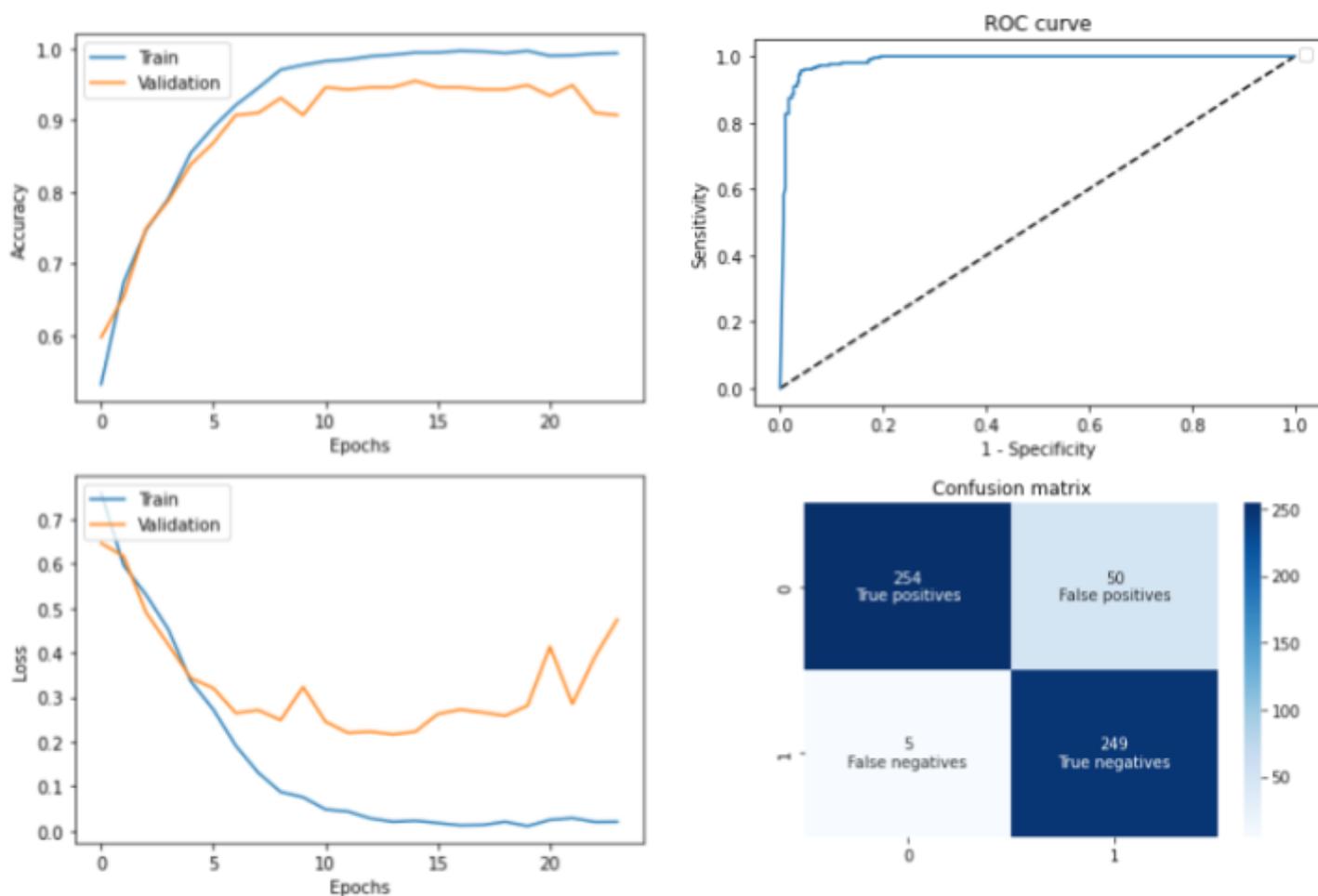


Figura 56: gráficas de los resultados obtenidos con el modelo TC y la aplicación de técnicas de DA

Capítulo 4: Conclusiones y trabajos futuros

Para concluir, me gustaría destacar cómo de importante es la buena construcción y las pruebas sobre los modelos de cualquier RNA para la obtención de los mejores resultados posibles. Este proyecto en sí ha demostrado el alcance de este tipo de IA para la detección de infección por la COVID-19, añadiendo valor a

una larga lista de investigaciones similares en el ámbito de la computación. La realización de pruebas, informes y recopilación de datos es crucial para la construcción de CNN de valor así como las características que la componen, ya sean inicializadores de pesos, funciones de activación, modificadores de LR dinámicas, DA, composición de las capas y sus tipos, etc. Existe cierto paradigma en cuanto a la construcción de modelos de los tipos de RNA más populares en la comunidad científica dando por hecho que ciertas variables o ciertas funciones funcionan mejor que otras para RNA específicas como la CNN o la RNN. El mundo en sí presenta un amplio abanico de problemas posiblemente resolubles mediante IA, y cada problema es a su vez un mundo. Es por ello que la investigación y sus paradigmas son importantes, pero no debemos perder de vista que la investigación consiste en un probar y ver resultados, lo que hace de la recopilación de datos y la realización de *test* un deber en la comunidad científica.

El trabajo desarrollado ha permitido alcanzar los objetivos propuestos en el TFG. Se ha desarrollado un sistema Inteligente basado en CNN de ayuda al diagnóstico de la COVID-19 con resultados altamente prometedores. Se han desarrollado dos modelos con variantes en las CNN para diagnóstico usando RX y usando CT. Se ha visto que los resultados alcanzados con ambas pruebas complementarias son de una gran precisión y sensibilidad, entre el 95 y 96%. Se concluye que con pruebas complementarias no altamente invasivas, como las RX, puede darse un diagnóstico de alta precisión y sensibilidad

En cuanto al estado de los modelos desarrollados, se seguirá intentando mejorarlos y además convertirlos en una herramienta útil a nivel de usuario además de científico mediante la creación de varias plataformas interfaz mediante la cual se puedan realizar una comunicación especialista-CNN. Además se intentarán aplicar técnicas más complejas de clasificación e incluso hacer que la RNA señale aquellas

características que hacen que una imagen sea clasificada de una forma u otra. Comprender el funcionamiento de las CNN es más sencillo si se observa en qué basa su comportamiento y sobre todo en saber cómo ha aprendido.

Bibliografía

Imagen de la portada recuperada de

<https://fineartamerica.com/featured/anatomy-of-a-typical-human-neuron-erzebet-s.html>

[1] Datos sobre la evolución actual de la pandemia, recuperado de <https://datosmacro.expansion.com/otros/coronavirus-vacuna>

[2] Ana Ordáz, Victoria Oliveres y Raúl Sánchez (2021), Así avanza la vacunación: mapas y gráficos sobre su evolución en España y el mundo, recuperado de https://www.eldiario.es/sociedad/vacuna-covid-mapas-graficos-proceso-vacunacion-espana-mundo-octubre-29_1_6782953.html

[3] Datos sobre la vacunación en el mundo, recuperado de <https://datosmacro.expansion.com/otros/coronavirus>

[4] Abdulkader Helwan, Mohammad Khaleel, Hani Hamdan, Dilber Uzun, Ozum Tuncyurek (2021) - Radiologists versus Deep Convolutional Neural Networks: A comparative study for diagnosing COVID-19.

[5] Joseph Paul Cohen, Paul Morrison, Lan Dao (2020) - COVID-19 Image Data Collection

[6] Jennifer Abbasi (2021) - Biotech Innovations

[7] Amira Ehtioui, Wassim Zouch, Mohamed Ghrobel, Chrokri Mhiri, Habib Hamam (2020) - Detection Methods of COVID-19.



- [8] Ministerio de sanidad de España, Información específica técnica sobre la COVID-19. Recuperado de <https://www.msbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/ITCoronavirus.pdf>
- [9] Matthew Binnicker (2021), ¿Qué es una cepa variante de la COVID-19?, recuperado de <https://www.mayoclinic.org/es-es/what-is-a-covid-19-variant-strain/vid-20508903>
- [10] E. Martínez, A. Díez, L. Ibáñez, S. Ossaba y S. Borrueal (2020), Radiologic diagnosis of patients with COVID-19, recuperado de <https://www.elsevier.es/en-revista-radiologia-english-edition--419-articulo-radiologic-diagnosis-patients-with-covid-19-S2173510721000033>
- [11] Heraldo TV (2020), Una simple radiografía, nuevo método de detección de la covid, recuperado de <https://www.heraldo.es/multimedia/videos/nacional/una-simple-radiografia-nuevo-metodo-de-deteccion-de-la-covid/>
- [12] Russell, Stuart J.; Norvig, Peter Norvig (2009). Artificial intelligence: a modern approach (en inglés) (3.ª edición). Recuperado de https://es.wikipedia.org/wiki/Inteligencia_artificial#CITAREFRussellNorvig2009
- [13] Aplicaciones de la inteligencia artificial. Recuperado de https://es.wikipedia.org/wiki/Inteligencia_artificial#Aplicaciones_de_la_inteligencia_artificial
- [14] Representación de una neurona, recuperado de https://www.researchgate.net/figure/Illustration-of-the-main-parts-of-a-neuron_fig1_267411352
- [15] Artificial Neuron, recuperado de https://en.wikipedia.org/wiki/Artificial_neuron
- [16] D. Calvo (2018), Función de activación redes neuronales, recuperado de <https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>
- [17] (2018), La función de activación, recuperado de <https://www.codificandobits.com/blog/funcion-de-activacion/>



- [18] H. Mujtaba (2020) , What is Rectified Linear Unit (ReLU)? | Introduction to ReLU Activation Function, recuperado de <https://www.mygreatlearning.com/blog/relu-activation-function/>
- [19] Himanshu S. (2019), Activation Functions : Sigmoid, tanh, ReLU, Leaky ReLU, PReLU, ELU, Threshold ReLU and Softmax basics for Neural Networks and Deep Learning, recuperado de <https://himanshuxd.medium.com/activation-functions-sigmoid-relu-leaky-relu-and-softmax-basics-for-neural-networks-and-deep-8d9c70eed91e>
- [20] IBM (2021), El modelo de redes neuronales, recuperado de <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model>
- [21] Recuperado de <http://www.nib.fmed.edu.uy/Seminario%202009/Monografias%20seminario%202009/Nacell-Redes%20NeuronalesImplementacion.pdf>
- [22] F.J. Gómez, M.A. Fernández, M.T. López y M.A. Díaz, Aprendizaje con redes neuronales artificiales, recuperado de <https://dialnet.unirioja.es/descarga/articulo/2281678.pdf>
- [23] M. Zeeshan (2020), Cost, Activation, Loss Function|| Neural Network|| Deep Learning. What are these?, recuperado de <https://medium.com/@zeeshanmulla/cost-activation-loss-function-neural-network-deep-learning-what-are-these-91167825a4de>
- [24] G. Westreicher, Regla de la cadena, recuperado de <https://economipedia.com/definiciones/regla-de-la-cadena.html>
- [25] S. Kostadinov (2019), Understanding Backpropagation Algorithm, recuperado de <https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd>
- [26] J. McGonagle, G. Shalkouski y C. Williams (2021), Backpropagation, recuperado de <https://brilliant.org/wiki/backpropagation/>
- [27] M. Nielsen (2019), How the backpropagation algorithm works, recuperado de <http://neuralnetworksanddeeplearning.com/chap2.html>



- [28] D. Calvo (2017), Clasificación de redes neuronales artificiales, recuperado de <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>
- [29] (2018), Convolutional Neural Network, recuperado de <https://searchenterpiseai.techtarget.com/definition/convolutional-neural-network>
- [30] Bagnato, J.I. (2020), Aprende machine learning. Recuperado de <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-network-s-vision-por-ordenador/>
- [31] 2016, Recuperado de <http://procesamientodigitaldeimagenes2016.blogspot.com/2016/12/texto-argumentativo.html>
- [32] SuperDataScience team (2018), Convolutional Neural Networks (CNN): Step 1(b) - Relu Layer. Recuperado de <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer/>
- [33] Richmond Alake (2020), (You Should) Understanding Sub-Sampling Layers Within Deep Learning. Recuperado de <https://towardsdatascience.com/you-should-understand-sub-sampling-layers-within-deep-learning-b51016acd551>
- [34] *Max pooling / pooling*. Recuperado de https://computersciencewiki.org/index.php/Max-pooling/_/_Pooling
- [35] Recuperado de https://www.researchgate.net/figure/Graphs-of-the-performance-of-the-accuracy-A-and-the-loss-function-B-of-the-ANN-in_fig5_325829191
- [36] Brownlee J. (2019), Overfitting and Underfitting With Machine Learning Algorithms. Recuperado de



<https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/#:~:text=Overfitting%3A%20Good%20performance%20on%20the,poor%20generalizati on%20to%20other%20data>

[37] Figura overfitting. Recuperado de https://en.wikipedia.org/wiki/File:Overfitting_svg.svg

[38] Tripathi, M. (2020), Underfitting and overfitting in Machine Learning. Recuperado de <https://datascience.foundation/sciencewhitepaper/underfitting-and-overfitting-in-machine-learning>

[39] Recuperado de <https://ichi.pro/es/como-ayuda-padding-en-cnn-186958315035>

[40] Recuperado de <https://www.machinecurve.com/wp-content/uploads/2020/02/reflection-710x210.png>

[41] Recuperado de <https://qph.fs.quoracdn.net/main-qimg-86023a8070b2c3e00d086b718e8a5e67>

[42] J. Cerda y L. Cifuentes (2012), Uso de curvas ROC en investigación científica, recuperado de https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0716-10182012000200003

[43] Curvas ROC, recuperado de http://www.hrc.es/bioest/roc_1.html

[44] G. López y P. Hernández (2001), Curvas ROC, recuperado de https://www.fisterra.com/mbe/investiga/curvas_roc/curvas_roc2.pdf

[45] Personal de Mayo Clinic (2020), Radiografía. Recuperado de <https://www.mayoclinic.org/es-es/tests-procedures/x-ray/about/pac-20395303>

- [46] Dobranowski J, Sabri A. COVID-19: Computed Tomography (CT). McMaster Textbook of Internal Medicine. Kraków: Medycyna Praktyczna. Recuperado de <https://empendium.com/mcmtextbook/chapter/B31.1269.1>.
- [47] Personal de Mayo Clinic (2020), Tomografía computarizada. Recuperado de <https://www.mayoclinic.org/es-es/tests-procedures/ct-scan/about/pac-20393675>
- [48] Ali Mohammad Alqudah, Shoroq Qazan (2020), Augmented COVID-19 X-ray Images Dataset. Recuperado de <https://data.mendeley.com/datasets/2fxz4px6d8/4>
- [49] Research Protocol - COVID-19 image data collection - Public. Recuperado de <https://docs.google.com/document/d/14b7cou98YhYcJ2jwOKznChtn5y6-mi9bgjeFv2DxOt0/edit>
- [50] Research Ethics Review Committee (ERC). Recuperado de <https://www.who.int/groups/research-ethics-review-committee/recommended-format-for-a-research-protocol/>
- [51] Cohen J. P. (2020), covid-chestxray-dataset. Recuperado de <https://github.com/ieee8023/covid-chestxray-dataset>
- [52] Josheph P. C. (2021), COVID-19 image data collection. Recuperado de <https://www.youtube.com/watch?v=ineWmqfelEQ>
- [53] Joseph P. C. (2021) COVID-19 image data collection slices. Recuperado de https://docs.google.com/presentation/d/1VNaMvjWuFNYPoe9XIWVV1Pg4x5cs1Xu4gykhwLkKXZE/edit#slide=id.g73a18ac1fa_0_0
- [54] COVID-CT. Recuperado de <https://github.com/UCSD-AI4H/COVID-CT>

[55] Xingyi Yang, Xuehai He, Jinyu Zhao, Yichen Zhang, Shanghang Zhang, Pengtao Xie (2020), COVID-CT-Dataset: A CT Image Dataset about COVID-19. Recuperado de <https://arxiv.org/pdf/2003.13865.pdf>

[56] Enlace de Gitlab al proyecto personal de DA. Recuperado de https://gitlab.com/Samuel-AQ/data_augmentation_dataset_generator

[57] Listado oficial de funciones *Callback* de Keras. Recuperado de <https://keras.io/api/callbacks/>

[58] Barrios Arce, J. I. (2019), La matriz de confusión y sus métricas. Recuperado de <https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>

[59] López de Ullibarri Galparsoro I, Píta Fernández, S. (2001), Curvas ROC. Recuperado de https://www.fisterra.com/mbe/investiga/curvas_roc/curvas_roc.asp

[60] ¿Qué es la función softmax? Recuperado de <https://ichi.pro/es/que-es-la-funcion-softmax-explica-el-adolescente-111365796021048>

[61] M. González Penedo, Perceptrón Multicapa. Recuperado de <http://www.varpa.org/~mgpenedo/cursos/scx/Tema4/nodo4-1.html>

Anexos

1) Entregables del TFT:

- a) Código del modelo de la CNN para las RX.
- b) Código del modelo de la CNN para las TC.
- c) Código del proyecto DA.
- d) CSV con los diferentes resultados obtenidos durante la investigación para cada uno de los *datasets* utilizados.



- e) Gráficas que demuestran varias de las pruebas realizadas con el fin de llegar al modelo de CNN final.