



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO DE GRADUACIÓN

**IMPLEMENTACIÓN DE REDES NEURONALES PARA DIAGNÓSTICOS AUTOMATIZADOS DE
IMÁGENES MÉDICAS: TUMORES CEREBRALES, ALZHEIMER, NEUMONÍA Y TUBERCULOSIS**

PREVIO A LA OBTENCIÓN DEL TÍTULO

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTADO POR:

11811045 ANDRÉS ENRIQUE NUILA DOMÍNGUEZ

ASESOR: ING. ROMÁN ARTURO PINEDA SOTO

CAMPUS TEGUCIGALPA

AUTORIZACIÓN

AUTORIZACIÓN DEL AUTOR(ES) PARA LA CONSULTA, LA REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO DE TESIS DE GRADO.

Señores:

CENTRO DE RECURSOS PARA EL APRENDIZAJE Y LA INVESTIGACION (CRAI)
Tegucigalpa

Estimados Señores:

La presentación del documento de tesis forma parte de los requerimientos y procesos establecidos de graduación para alumnos de pregrado de UNITEC.

Yo, Andrés Enrique Nuila Domínguez, de Tegucigalpa, autor(es) del trabajo de grado titulado: Implementación de Redes Neuronales para Diagnósticos Automatizados de Imágenes Médicas: Tumores Cerebrales, Alzheimer, Neumonía y Tuberculosis, presentado y aprobado en el año XXXX, como requisito para optar al título de Profesional de Ingeniero En Sistemas Computacionales, autorizo a:

Las Bibliotecas de los Centros de Recursos para el Aprendizaje y la Investigación (CRAI) de la Universidad Tecnológica Centroamericana (UNITEC), para que, con fines académicos, pueda libremente registrar, copiar y usar la información contenida en él, con fines educativos, investigativos o sociales de la siguiente manera:

Los usuarios puedan consultar el contenido de este trabajo de grado en las salas de estudio de la biblioteca y la página Web de la universidad.

Permita la consulta y la reproducción, a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato CD o digital desde Internet, Intranet, etc., y en general para cualquier formato conocido o por conocer.

De conformidad con lo establecido en el artículo 19 de la Ley de Derechos de Autor y de los Derechos Conexos; los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

Es entendido que cualquier copia o reproducción del presente documento con fines de lucro no está permitida sin previa autorización por escrito de parte de los principales autores.

En fe de lo cual, se suscribe la presente acta en la ciudad de Tegucigalpa a los XX días del mes de XX de XXXX.

Andrés Enrique Nuila Domínguez

11811045

HOJA DE FIRMAS

Los abajo firmantes damos fe, en nuestra posición de miembro de Terna, Asesor y/o Jefe Académico y en el marco de nuestras responsabilidades adquiridas, que el presente documento cumple con los lineamientos exigidos por la Facultad de Ingeniería y los requerimientos académicos que la Universidad dispone dentro de los procesos de graduación.

PhD(c). Román Arturo Pineda Soto

Asesor UNITEC

Ing. X	Ing. X.	Ing. X.
Coordinador de Terna	Miembro de Terna	Miembro de Terna

Ing. Reynod Marcial Bocanegra

Jefe Académico de Ingeniería en Sistemas Computacionales | UNITEC

RESUMEN EJECUTIVO

El objetivo principal de esta investigación es desarrollar un enfoque innovador que aproveche las capacidades de las redes neuronales en el análisis de imágenes médicas, específicamente resonancias magnéticas y radiografías, para mejorar la precisión y eficiencia del diagnóstico de las mismas y acortar tiempos de respuesta. El estudio se basa en la premisa de que las imágenes médicas modernas proporcionan una gran cantidad de información valiosa, pero a menudo compleja, que puede ser desafiante para los médicos interpretar con precisión. Mediante el uso de redes neuronales, un tipo de inteligencia artificial inspirado en la estructura y función del cerebro humano, esta investigación busca desarrollar modelos capaces de reconocer patrones sutiles y características distintivas en las imágenes médicas. El proceso de investigación involucra la recopilación de un conjunto de datos diverso y representativo que consta de resonancias magnéticas y radiografías de pacientes con diagnósticos confirmados de tumores cerebrales, Alzheimer, neumonía y tuberculosis. Estos datos se utilizan para entrenar y ajustar las redes neuronales, permitiéndoles aprender a identificar signos tempranos y específicos de estas enfermedades en las imágenes. Ésta investigación destaca la utilidad y el impacto potencial de las redes neuronales en el campo de la medicina, especialmente en la detección temprana de enfermedades cerebrales y pulmonares a través de imágenes médicas como resonancias magnéticas y radiografías. Si bien se requiere más investigación y validación clínica, los resultados hasta ahora indican que este enfoque podría revolucionar la forma en que se abordan estas enfermedades, mejorando la vida de los pacientes y contribuyendo al avance de la atención médica.

ABSTRACT

The main goal of this research is to develop an innovative approach taking advantage of neural networks capabilities in the analysis of medical images, specifically magnetic resonance imaging (MRI) and x-rays (RX), to improve the accuracy and efficiency of diagnosis based on these images and shorten response and diagnostic times. The research is based upon the premise that modern medical images provide a lot of valuable but complex information; which translates in medical diagnosis being a demanding and meticulous task for physicians in order to give a correct prognosis. By using neural networks, a kind of artificial intelligence (AI) that uses the human brain structure and functioning as inspiration, this research sought to set in motion neural networks models that can potentially recognize subtle patterns and distinctive traits from medical images as magnetic resonance imaging and x-rays. The research process involves the collection of a diverse and representative data set consisting of MRIs and X-rays of patients with confirmed diagnoses of brain tumors, Alzheimer's, pneumonia, and tuberculosis. This data is used to train and tune neural networks, allowing them to learn to identify early and specific signs of these diseases in images. This research highlights the usefulness and potential impact of neural networks in the field of medicine, especially in the early detection of brain and lung diseases through medical imaging such as MRIs and X-rays. While further research and clinical validation are required, the results so far indicate that this approach could revolutionize the way these diseases are addressed, improving the lives of patients and advancing healthcare.

LISTA DE SIGLAS

AI	Artificial Intelligence
BCH	Banco Central Hondureño
CEL	Cross-Entropy Loss
COMCAEDA	Confederación Multidisciplinaria de Centroamérica de Alzheimer
IA	Inteligencia Artificial
ML	Machine Learning
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NHLBI	National Heart, Lung and Blood Institute
NIBIB	National Institute of Biomedical Imaging and Bioengineering
OMS	Organización Mundial de Salud

Neurona artificial:	Es una unidad de cálculo que intenta modelar el comportamiento de una neurona "natural", similares a las que constituyen del cerebro humano.
Red Neuronal:	Consiste en un conjunto de neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida.
Accuracy/Exactitud:	El ratio entre las predicciones correctas (suma de verdaderos positivos y verdaderos negativos) y las predicciones totales.
Precisión:	La precisión es el ratio entre el número de documentos relevantes recuperados entre el número de documentos recuperados.
Recall:	También conocido como exhaustividad. Es el porcentaje de muestras de datos que un modelo de aprendizaje automático identifica correctamente como pertenecientes a una clase de interés del total de muestras de esa clase.
F-1 Score:	Es la media armónica que combina los valores de la precisión y de la recall.

TABLA DE ILUSTRACIONES

Ilustración 1: Distintas funciones pueden abarcar el mismo conjunto de datos.....	11
Ilustración 2: Diagrama básico de una red neuronal.....	14
Ilustración 3: Neurona Artificial Primitiva.....	15
Ilustración 4: Ejemplo gráfico de función de activación lineal.....	17
Ilustración 5: Recta para la función lógica OR	17
Ilustración 6: Tabla de verdad función lógica OR.	18
Ilustración 7: Tabla de verdad función lógica XNOR.....	18
Ilustración 8: Comparación gráfica entre combinación de varias neuronas con funciones de activación lineales y una sola neurona con una función de activación no lineal sobre la función lógica XNOR.....	19
Ilustración 9: Ejemplo de localización de ejes usando convoluciones.....	23
Ilustración 10: Pre procesamiento de imágenes.	32
Ilustración 11: Primer diseño de redes neuronales.	33
Ilustración 12: Diseño y estructura final de redes neuronales convolucionales.....	34
Ilustración 13: Código usado en el entrenamiento y validación.....	35
Ilustración 14: Script para la validación final de los modelos.	35
Ilustración 15: Código para mostrar la pantalla principal del GUI.....	36
Ilustración 16: Código usado para desplegar la pantalla de selección de modelo de predicción a utilizar.	36
Ilustración 17: Código usado para la selección de la imagen a analizar.....	37
Ilustración 18: Bloque de código encargado de seleccionar el modelo de diagnóstico.	37
Ilustración 19: Código encargado de cargar el modelo para clasificación previa de imágenes....	38
Ilustración 20: Diagrama con el cronograma de actividades.	39

Ilustración 21: Bitácora de Actividades.	39
Ilustración 22: Evolución de la exactitud del modelo de detección de Alzheimer.....	40
Ilustración 23: Evolución del valor de la función de pérdida del modelo de detección de Alzheimer.	40
Ilustración 24: Reporte de clasificación de modelo para detección y clasificación de Alzheimer.	41
Ilustración 25: Matriz de confusión del modelo de detección y clasificación de Alzheimer.....	41
Ilustración 26: Evolución de la exactitud del modelo de detección y clasificación de tumores cerebrales.....	43
Ilustración 27: Evolución del valor de la función de pérdida del modelo de detección y clasificación de tumores cerebrales.....	43
Ilustración 28: Reporte de clasificación del modelo de detección y clasificación de tumores cerebrales.....	44
Ilustración 29: Matriz de confusión del modelo de detección y clasificación de tumores cerebrales.	44
Ilustración 30: Evolución de la exactitud del modelo de detección y clasificación de enfermedades pulmonares.....	46
Ilustración 31: Evolución del valor de la función de pérdida del modelo de detección y clasificación de enfermedades pulmonares.	46
Ilustración 32: Reporte de clasificación para el modelo de detección y clasificación de enfermedades pulmonares.....	47
Ilustración 33: Matriz de confusión para el modelo de detección y clasificación de enfermedades pulmonares.....	47
Ilustración 34: Evolución de la exactitud del modelo de clasificación previa de imágenes.....	49
Ilustración 35: Evolución del valor de la función de pérdida del modelo de clasificación previa de imágenes.	49

Ilustración 36: Reporte de clasificación del modelo de clasificación previa de imágenes.....	50
Ilustración 37: Matriz de confusión del modelo de clasificación previa de imágenes.....	50
Ilustración 38: Pantalla principal del programa.	51
Ilustración 39: Pantalla de selección de un modelo de diagnóstico específico.	51
Ilustración 40: Pantalla de selección de imagen a cargar.	52
Ilustración 41: Retroalimentación dirigida al usuario de la imagen y el modelo cargado.....	52
Ilustración 42: Ejemplo de diagnóstico final de imagen cargada.....	52

TABLA DE ECUACIONES

Ecuación 1: Generalización de la función de salida de una neurona artificial	15
Ecuación 2: Mean Squared Error o error de la suma de cuadrados.	20
Ecuación 3: Cross-Entropy Loss o Pérdida de Entropía Cruzada en clasificación binaria.....	21
Ecuación 4: Cross-Entropy Loss o Pérdida de Entropía Cruzada en clasificación multiclase.	21

ÍNDICE DE CONTENIDO

I	Introducción.....	1
II	Planteamiento del Problema.....	3
2.1	Precedentes del Problema	3
2.2	Definición del Problema.....	5
2.3	Justificación.....	5
2.4	Preguntas de Investigación.....	6
2.5	Objetivos	7
2.5.1	Objetivo General.....	7
2.5.2	Objetivos Específicos.....	7
III	Marco Teórico	8
3.1	Inteligencia Artificial	8
3.1.1	Conceptos Preliminares.....	8
3.1.2	Agentes Inteligentes.....	9
3.1.3	Aprendizaje Automático	9
3.1.3.1	Aprendizaje Supervisado.....	10
3.1.3.2	Aprendizaje No Supervisado	12
3.2	Redes Neuronales.....	13
3.2.1	Estructura de Una Red Neuronal	14
3.2.2	Neurona Artificial.....	15
3.2.3	Funciones de Activación	16
3.2.4	Funciones de Pérdida.....	20
3.2.5	Redes Neuronales Convolucionales	22

3.2.5.1	Proceso de Convolución.....	23
3.3	Enfermedades	24
3.3.1	Tumores Cerebrales.....	24
3.3.2	Alzheimer.....	26
3.3.3	Tuberculosis.....	26
3.3.4	Neumonía.....	27
3.4	Imágenes Médicas.....	27
3.4.1	Resonancias Magnéticas.....	27
3.4.2	Radiografías	28
IV	Metodología	28
4.1	Enfoque	28
4.2	Variables de Investigación.....	29
4.3	Técnicas e Instrumentos Aplicados.....	29
4.4	Materiales	30
4.4.1	Hardware.....	30
4.4.2	Software:	30
4.4.3	Toolkits y Herramientas de Programación:.....	30
4.5	Metodología de Estudio.....	31
4.6	Cronograma de Actividades	39
4.6.1	Bitácora de Actividades	39
V	Resultados y Análisis.....	40
5.1	Redes Neuronales.....	40
5.1.1	Detección de Alzheimer:.....	40

5.1.2	Detección y Clasificación de Tumores Cerebrales.....	43
5.1.3	Detección y Clasificación de Enfermedades Pulmonares.....	46
5.1.4	Clasificación Previa de Imágenes Médicas.....	49
5.2	Interfaz Gráfica de Usuario.....	51
VI	Conclusiones.....	53
VII	Recomendaciones.....	54
VIII	Bibliografía.....	55

I INTRODUCCIÓN

En el vasto panorama de la medicina y la tecnología, la detección temprana y precisa de enfermedades representa un pilar fundamental para mejorar los resultados clínicos y la calidad de vida de los pacientes. En este contexto, las imágenes médicas, como las resonancias magnéticas (RM) y las radiografías, han demostrado ser herramientas invaluable para diagnosticar una variedad de condiciones médicas, desde trastornos neurológicos hasta enfermedades pulmonares. Sin embargo, la interpretación de estas imágenes a menudo complejas y ricas en información, puede ser un desafío incluso para los profesionales médicos más experimentados.

La evolución de la tecnología médica y la inteligencia artificial presenta una oportunidad única para abordar estos desafíos. Las redes neuronales, algoritmos que imitan las conexiones neuronales en el cerebro humano, han demostrado un poder sin precedentes en el análisis de datos complejos, incluidas las imágenes médicas. Al aprovechar la capacidad de estas redes para reconocer patrones en grandes conjuntos de datos, podemos aspirar a una detección más temprana y precisa de enfermedades, lo que podría resultar en un impacto significativo en la salud pública y la atención individualizada.

Esta investigación se sumergirá en la intersección de la medicina y la inteligencia artificial, con el propósito de mejorar la precisión y eficiencia en la detección de enfermedades cerebrales y pulmonares. En particular, se centrará en el uso de redes neuronales convolucionales, un paradigma de aprendizaje automático inspirado en la estructura y función del cerebro humano, para analizar imágenes médicas con un nivel de detalle y sensibilidad que trasciende las capacidades humanas.

El enfoque de esta investigación será ambicioso pero vital. Se buscará no solo desarrollar modelos de detección automatizada, sino también comprender cómo las redes neuronales podrán discernir patrones sutiles y características distintivas en imágenes médicas. El objetivo será potenciar la detección temprana de enfermedades que afectan el cerebro, como tumores cerebrales y Alzheimer, así como afecciones pulmonares críticas como neumonía y tuberculosis.

Para lograr estos objetivos, se planteará un proceso de investigación exhaustivo. Esto incluirá la recopilación de un conjunto de datos diversificado y representativo, compuesto por resonancias magnéticas y radiografías de pacientes con diagnósticos confirmados. Estos datos serán la base sobre la cual se entrenarán y ajustarán las redes neuronales, permitiéndoles internalizar patrones que hasta ahora han sido difíciles de detectar.

Este estudio no solo se tratará de tecnología; será un esfuerzo colaborativo que une la experiencia médica con la innovación tecnológica. Al mejorar la capacidad de interpretación de las imágenes médicas, se espera no solo contribuir al conocimiento científico, sino también brindar una herramienta que impacte positivamente la práctica clínica y la atención al paciente.

Esta investigación es un llamado a explorar nuevos horizontes en la convergencia entre la medicina y la inteligencia artificial. En última instancia, se aspira a contribuir a la mejora de la atención médica y a la lucha contra enfermedades que afectan la vida de millones de personas en todo el mundo.

II PLANTEAMIENTO DEL PROBLEMA

2.1 PRECEDENTES DEL PROBLEMA

Conforme a la información en el sitio oficial del Colegio Médico (Colegio Médico de Honduras, 2018) en un censo de médicos realizado para el 6 de febrero de 2018, en el país habían 12,855 médicos colegiados en total. Por otro lado, en el censo publicado en 2021 por el Banco Central (Banco Central de Honduras, 2021) nos confirma que para finales de 2018 la cifra subió a 13,792, dando también la cantidad de cuántos dentro de este conjunto son especialistas, siendo un total de 3,928 (un 28.48% sobre el total de médicos colegiados) siendo este último dato constituido por médicos especializados en cualquiera de las ramas de la medicina moderna (cardiología, neumología, neurología, radiología, etc.). Estos números fueron creciendo paulatinamente de modo que, para 2019, habían 15,280 y para 2020 16,083 médicos colegiados; en contraste el aumento en los especialistas en 2019 pasó de ser 3,928 a 3,936 y en 2020 3,941, lo que implica un crecimiento de apenas un 0.20% entre 2018 y 2019, un 0.12% entre 2019 y 2020 y de un 0.33% entre 2018 y 2020.

Respecto a enfermedades, en Honduras en los últimos años han incrementado los casos de tumores cerebrales, siendo esta una problemática a nivel mundial y generacional (Mayo Clinic, 2023). Así mismo, el Alzheimer y la demencia, según el trabajo de Cáceres y Marín, son otras dos enfermedades que han venido en aumento a nivel mundial estimándose para el 2015 un total de 46 millones de enfermos, con una proyección para el año 2030 de 130 millones de enfermos (Cáceres & Marín, 2019); y siguiendo lo mencionado por COMCAEDA, tomando mayor énfasis en que Honduras, al ser un país centroamericano, también reflejaría en su número de afectados un aumento más notable al ser el mismo proyectado en países con el perfil de envejecimiento que muestran los de su región. De igual manera, se presume que en regiones similares a la del país, un 90% de los casos no se diagnostican de manera temprana (Acosta, y otros, 2013) dando como resultado una atención terapéutica tardía en el paciente. Dentro de esta problemática también es notable resaltar el aumento de casos desde la pandemia por SARS-Cov-2 de neumonía y tuberculosis dando cifras cercanas a 450 millones de casos y aproximadamente 4 millones de muertes anuales por neumonía; mientras que por tuberculosis hubieron cifras cercanas a 10.6

millones de casos y 1.6 millones de muertes (World Health Organization, 2023) al año según informes brindados por la OMS.

Profundizando en lo anterior, cabe definir que un tumor cerebral es una masa o crecimiento anormal de células en el cerebro. El cráneo es muy rígido y el crecimiento de células en esta área tan reducida puede causar problemas graves y es por esto que su detección temprana es de suma importancia para salvar vidas (Mayo Clinic, 2023).

Respecto al Alzheimer y la demencia lo más notable a resaltar es su origen siendo causadas ambas por un rápido deterioro en la masa cerebral el cual los científicos creen que, en la mayoría de los casos, es consecuencia de una combinación de factores genéticos, ambientales y del estilo de vida que afectan el cerebro a lo largo del tiempo (Mayo Clinic, 2023). El daño inicialmente parece ocurrir en el hipocampo y la corteza entorrinal, partes del cerebro que son esenciales para la formación de los recuerdos. A medida que mueren más neuronas, más partes del cerebro se ven afectadas y comienzan a encogerse. En la etapa final de la enfermedad de Alzheimer, el daño es generalizado y el tejido cerebral se ha reducido significativamente, lo cual, de misma manera que con los tumores, es de vital importancia su rápida detección para poder comenzar el tratamiento terapéutico cuánto antes al ser esta una enfermedad incurable, pero como mencionan Cáceres y Marín en su trabajo su afección en el paciente puede verse disminuida usando terapia ocupacional (Cáceres & Marín, 2019), donde un 80% de los estudios que las autoras tomaron en cuenta mostraron resultados positivos.

Sobre otros puntos relevantes en esta investigación. la neumonía es una enfermedades grave que afecta mayormente los pulmones ocasionadas por bacterias adquiridas debido a la baja de las defensas en el cuerpo humano (Center for Disease Control And Prevention, 2022) y de manera similar lo hace la tuberculosis (Center for Disease Control And Prevention, 2023); y al ser estas dos enfermedades que afectan las vías respiratorias, su urgencia en la detección y tratamiento es primordial para la efectiva recuperación del paciente ya que como mencionan López Marín et al., en los países latinoamericanos, el control de esta enfermedad parece lejos de nuestro alcance: siendo la tuberculosis humana responsable de más de cien muertes diarias en esta región (López Marín, Díaz Otero, Vallecillo Maza, Esquivel Solís, & Gutierrez Pabello, 2006).

Es también notable mencionar que en los últimos años, el aprendizaje profundo (deep learning) ha sido usado exhaustivamente en varios campos de investigación como visión artificial, procesamiento de lenguaje natural, procesamiento de imágenes, clasificación y predicción de salidas, entre otros (Goodfellow, Bengio, & Courville, 2016). Esto se debe a que el deep learning tiene como propósito ser usado en áreas donde se necesite analizar una cantidad amplia de información con inteligencia que se asemeje a la humana. Utilizar deep learning como una forma de machine learning para el reconocimiento de patrones, se está convirtiendo en un aspecto de importancia en el campo de análisis de imágenes para propósitos médicos (Greenspan, van Ginneken, & Summers, 2016).

2.2 DEFINICIÓN DEL PROBLEMA

Actualmente el proceso de análisis de imágenes médicas para la detección enfermedades y sus respectivos diagnósticos es un proceso tardado como tal debido a la complejidad del cuerpo humano y también debido a que en Honduras hay una corta cantidad de médicos especialistas en relación a la población general del país, lo cual puede traducirse en tiempo desaprovechado en el tratamiento que podría afectar negativamente al paciente.

2.3 JUSTIFICACIÓN

La relevancia de esta investigación se acentúa en un mundo donde la carga de enfermedades crónicas y degenerativas está en constante aumento. Las enfermedades cerebrales como los tumores y el Alzheimer, así como las afecciones pulmonares como la neumonía y la tuberculosis, tienen un impacto significativo en la calidad de vida de los pacientes y representan un desafío considerable para los sistemas de atención médica. La detección temprana y precisa es esencial para brindar tratamientos efectivos y mejorar los pronósticos. Sin embargo, a menudo las señales iniciales de estas enfermedades pueden ser sutiles y difíciles de discernir, lo que destaca la importancia de herramientas de diagnóstico más sofisticadas y precisas. Este análisis de imágenes automatizado busca apoyar a radiólogos y médicos en hacer que su proceso de diagnóstico sea más eficiente haciendo énfasis en las enfermedades mencionadas con anterioridad.

Uno de los puntos a tomar en cuenta partiendo de las cifras mencionadas anteriormente, es lo desproporcionada que está la distribución de especialistas por habitantes dentro del país. Según el censo del BCH consultado para fines de este trabajo (Banco Central de Honduras, 2021), en 2020 habían 9, 304,400 habitantes; lo que deja a 1 especialista por cada 2,361 habitantes. El número limitado de radiólogos especializados en el análisis de imágenes médicas no puede satisfacer la creciente demanda de servicios, lo que lleva a una mayor demora en la obtención de resultados. Optimizar el proceso de análisis permitiría una utilización más eficiente de los recursos humanos disponibles y aceleraría el diagnóstico. También cabe mencionar que la tardanza en el análisis puede conducir a errores de diagnóstico o interpretación, lo que puede tener consecuencias graves para la salud del paciente. Reducir los tiempos de procesamiento de las imágenes médicas puede aumentar la precisión y la confiabilidad del diagnóstico, lo que a su vez mejorará la calidad general de la atención médica como también la detección temprana de tumores cerebrales, neumonía o tuberculosis es crucial para aumentar las tasas de supervivencia y mejorar el pronóstico de los pacientes. La demora en el análisis de las imágenes puede retrasar el inicio del tratamiento y afectar negativamente los resultados clínicos, lo que subraya la necesidad de reducir los tiempos de procesamiento de estas imágenes médicas.

2.4 PREGUNTAS DE INVESTIGACIÓN

- ¿Qué tan correcto puede llegar a ser el diagnóstico brindado por el prototipo para el diagnóstico correcto de enfermedades en comparación al criterio de un especialista?
- ¿Con cuántos médicos será preciso consultar los diagnósticos durante el entrenamiento y desarrollo para evitar falsos positivos/negativos?
- ¿Con cuántos datos (imágenes) se podrá obtener un resultado confiable en el entrenamiento de la red neuronal?

2.5 OBJETIVOS

2.5.1 OBJETIVO GENERAL

- Desarrollar un programa capaz de brindar diagnósticos médicos con base en imágenes médicas y su previo análisis acortando los tiempos de respuesta para el paciente.

2.5.2 OBJETIVOS ESPECÍFICOS

- Construir el set de datos necesario para el entrenamiento y ajuste de la red neuronal recolectando resonancias magnéticas de cráneo con diagnósticos de tumores cerebrales o Alzheimer y radiografías de pecho con diagnósticos de neumonía o tuberculosos.
- Programar un analizador y preprocesador de imágenes para su posterior uso en el entrenamiento de una red neuronal convolucional.
- Diseñar la arquitectura de las redes neuronales que permita clasificar las imágenes por parte del cuerpo y brinde un diagnóstico respecto a lo presentado en la resonancia magnética o radiografía.
- Desarrollar redes neuronales convolucionales que una vez clasificada y etiquetada la imagen de entrada, analice la misma y encuentre anomalías, detectando enfermedades o patologías entre tumores cerebrales para resonancias magnéticas de cráneo y neumonía o tuberculosos a partir de radiografías de pecho.
- Desarrollar una interfaz gráfica de usuario para el uso amigable del prototipo que implemente las redes diseñadas.

III MARCO TEÓRICO

A continuación, se explorarán las bases conceptuales y teóricas que respaldan esta investigación, examinando tanto los fundamentos de las redes neuronales como el conocimiento existente sobre las enfermedades objeto de estudio. Este análisis crítico sienta las bases para comprender cómo las redes neuronales pueden ser aprovechadas para transformar la forma en que se enfrentan estos desafíos médicos cruciales, marcando así un hito en la intersección de la tecnología y la salud.

3.1 INTELIGENCIA ARTIFICIAL

3.1.1 CONCEPTOS PRELIMINARES

Para poder comprender el medio que se utilizará en el presente trabajo, primero se debe definir qué es la inteligencia artificial.

Este concepto ha evolucionado significativamente a lo largo del tiempo, desde los desarrollos tecnológicos primitivos como los primeros programas de ajedrez hasta los sistemas actuales que pueden realizar tareas complejas que antes solo podían ser realizadas por humanos, siendo un ejemplo los sistemas de conducción automática. Algunos hitos importantes en la evolución de la inteligencia artificial incluyen el desarrollo de los sistemas de procesamiento del lenguaje natural, robots industriales, sistemas de aprendizaje automático y profundo. Mencionado lo anterior, es posible plantear la idea que la inteligencia artificial es un campo de la informática que se enfoca en la creación de sistemas y programas que pueden realizar tareas que, según Cazana Cachís, requieren inteligencia humana, como el aprendizaje, la toma de decisiones y la resolución de problemas (Cazana Cachís, 2017).

Como tal, la inteligencia artificial es un concepto muy amplio y que actualmente, se ha convertido en una de las ramas de la informática más discutidas debido a sus múltiples aplicaciones tales como el razonamiento automático, la demostración de teoremas, los sistemas expertos, el procesamiento de lenguaje natural, robótica, lenguajes y ambientes de IA, aprendizaje, redes neuronales y algoritmos genéticos, por mencionar solo algunas (López Takeyas, 2007). También se debe mencionar que ésta cuenta con un gran campo de definiciones pero que las mismas

convergen en su mayoría a una sola proposición como se mencionó anteriormente: emular la inteligencia humana; dándole comportamiento inteligente a las máquinas (Nilsson, 2001), siendo su principal finalidad el poder resolver problemas y que esta solución sea lo más cercana a la solución que un humano daría sin la necesidad de que sea la persona quien haga el proceso de toma de decisiones.

3.1.2 AGENTES INTELIGENTES

Para poder lograr lo anterior se hace uso de lo que llamamos agentes inteligentes, entidades que son capaces de percibir su entorno, teniendo una parcial representación del mismo y siendo capaz de actuar sobre éste dependiendo de las decisiones que el agente tome basándose en su aprendizaje previo. Los agentes inteligentes en inteligencia artificial funcionan como entidades autónomas que observan y actúan sobre un entorno dirigiendo su actividad hacia el logro de objetivos (Valencia, Valdovinos Rosas, Marcial Romero, & Eleuterio, 2022). Para que un agente inteligente pueda implementar su inteligencia artificial o al menos tener un mínimo grado de la misma y lograr los objetivos planteados, debe aprender respecto al entorno en el que éste se desenvolverá, siendo su toma de decisión realizada en consecuencia a este mismo aprendizaje. Respecto a este último punto, se puede decir que un agente está aprendiendo si mejora su desempeño en tareas futuras después de hacer observaciones sobre el mundo o el entorno en el que este mismo se desenvuelve.

3.1.3 APRENDIZAJE AUTOMÁTICO

En informática, el aprendizaje, mejor conocido por el término *aprendizaje de máquina* (machine learning), se refiere todos aquellos programas y algoritmos computacionales cuya función es optimizar toda la información recibida como parámetros de un modelo matemático-computacional (Goodfellow, Bengio, & Courville, 2016), siendo que los modelos pueden ser descriptivos o inductivos, dependiendo de si permiten generar conocimiento a partir de datos previos o si permiten realizar predicciones futuras. Indistintamente, para poder aprender, todos los tipos de modelos necesitan grandes cantidades de datos y estos deben ser particionados en dos subconjuntos siendo denominados *conjunto de entrenamiento* y *conjunto de prueba*, con el primero usado para el proceso de aprendizaje y entrenamiento y el segundo para validar si el

modelo aprendió correctamente, y gracias a los cuales mediante teoría estadística y cálculos matemáticos se pueden realizar inferencias a partir de estas muestras (Ponce, y otros, 2014) para así brindar soluciones y respuestas sin necesidad de ser explícitamente programadas por parte del humano (Mahesh, 2018). De esta premisa nace el siguiente dilema: Si bien es el programador quien diseña y planifica el entrenamiento y los algoritmos de aprendizaje, ¿Por qué queremos que un agente o una máquina aprendan? Si mediante el aprendizaje, un agente inteligente mejora su desempeño ¿No sería más práctico que sea el programador quien programe directamente este modelo optimizado? La respuesta es no. Esto debido a que un humano no puede anticiparse a absolutamente todos los escenarios posibles que el agente pueda encontrarse. Una persona tampoco puede anticiparse a todos los cambios que ocurrirán con el paso del tiempo y aún más importante, un humano no siempre podrá programar o siquiera sabrá la solución al problema planteado o la situación que el agente se encuentre.

Retomando lo menciona Mahesh en su trabajo, existen varios tipos de algoritmos partiendo de esta premisa como lo son los árboles de decisión, bosques aleatorios, redes neuronales artificiales, algoritmos genéticos y los algoritmos de agrupamiento, por mencionar los más conocidos actualmente. Cada uno de estos tiende a recaer en una de las dos principales categorías de aprendizaje que existen hoy en día: no supervisado y supervisado, de los cuales se estará hablando a continuación.

3.1.3.1 *Aprendizaje Supervisado*

Matemáticamente a grandes rasgos, el aprendizaje supervisado se ve de la siguiente manera:

Dado un set de entrenamiento de N pares entrada – salida de la forma:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$$

donde cada y_i fue generado por una función desconocida $y = f(x)$,

encuentre una función H que se aproxime a la función f .

Donde las variables **(x, y)** de cada par no necesariamente tiene que ser números si no que pueden tomar cualquier valor. La función **H** es una hipótesis que el modelo genera tratando de acercarse a la función **f** que nos brinda la respuesta real y correcta (Russel & Norvig, 2010).

Siguiendo lo mencionado por Mahesh, y añadiendo lo encontrado en el libro de Russell y Norvig, se puede afirmar que este tipo de aprendizaje es en el que más presencia del humano encontramos, debido a que es la persona quien va mapeando cada entrada con su salida esperada (Russel & Norvig, 2010), tratando este tipo de aprendizaje de apegarse a la definición matemática.

Hablando de la misma definición y rescatando lo mencionado en el mismo libro de Russell y Norvig, siendo la función ***H*** nuestra hipótesis nacen preguntas tales como *¿De qué manera aprende un modelo mediante este tipo de aprendizaje?*

En este caso en particular, decimos que el aprendizaje es una búsqueda de una hipótesis a través del conjunto de todas aquellas posibles que funcione bien, incluso en nuevos ejemplos más allá del conjunto de entrenamiento. Para medir la precisión de una hipótesis, le damos un conjunto de pruebas de ejemplos distintos del conjunto de entrenamiento, siendo este el conjunto de prueba del que se habla en secciones anteriores de este trabajo. Finalmente, Russell y Norvig concluyen que una hipótesis generaliza bien si predice correctamente el valor de ***y*** para ejemplos nuevos. Es notable mencionar que a veces la función ***f*** es estocástica, o dicho en otras palabras, no es estrictamente una función de ***x*** y en este caso lo que el modelo tiene que aprender es una distribución de probabilidad condicional tal que $P(Y|x)$.

Retomando la idea planteada respecto a la manera que el modelo encuentra la respuesta as adecuada a su escenario dentro el conjunto de todas las posibles hipótesis, matemáticamente y gráficamente es posible plantear el siguiente ejemplo, similar a lo mencionado por Russell y Norvig:

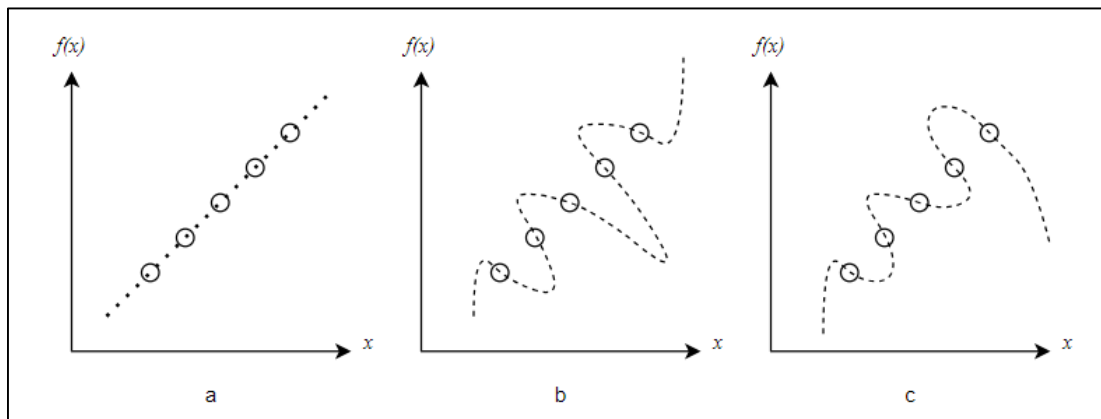


Ilustración 1: Distintas funciones pueden abarcar el mismo conjunto de datos

Podemos ver que para el mismo conjunto de datos hay tres funciones matemáticas que pueden contenerlos, sirviendo como ejemplos una función lineal (*ejemplo a*), un polinomio grado cinco (*ejemplo b*) o un polinomio grado seis (*ejemplo c*). Los tres casos son intrínsecamente soluciones para la definición matemática del aprendizaje automático, pero dependerá del contexto y del tipo de problema que se quiera solucionar la elección de cuál de las tres funciones el agente deberá utilizar para desenvolverse.

En el caso del aprendizaje supervisado es el desarrollador quien, viendo la retroalimentación brindada por el agente inteligente, decide los parámetros que se necesitan afinar para acercarse a la función que mejores resultados brinde a su criterio. Dentro de estos parámetros podemos encontrar el radio de aprendizaje (learning rate) en el caso de las redes neuronales artificiales o la cantidad de árboles de decisión y la profundidad de cada uno en el caso de los bosques aleatorios (random forests).

En la práctica, este tipo de aprendizaje es implementado en problemas de clasificación y regresión. Cuando hablamos de clasificación, el resultado esperado dado un dato x es una respuesta y que se encuentre dentro de un grupo finito de posibilidades (Sandoval, 2018), como por ejemplo decir si un cliente x ira al restaurante y que sea de la franquicia A, B o C; si el día de hoy lloverá o hará frío; o si en una imagen dada hay la foto de un perro o un gato; mientras que un problema de regresión la salida esperada es un número el cual es relacionado a un predicción ya sea predecir por cuanto se puede vender una propiedad, estimar el tiempo de llegada del punto A al punto B o predecir el tiempo que un empleado permanecerá en la empresa X.

3.1.3.2 *Aprendizaje No Supervisado*

Dado que, en estos casos, no hay explícitamente respuestas correctas ni un supervisor o "maestro" que enseñe al modelo, a este tipo de aprendizaje se le conoce como aprendizaje no supervisado, en contraste con el aprendizaje supervisado que se vio con anterioridad. La estructuración de los datos tanto de entrenamiento como de prueba queda por completo a manos del modelo y este es dejado a su suerte para que pueda examinar a detalle esta estructura y sacar sus propias conclusiones, dándole al modelo solamente las características que cada dato de entrada; esperando que sea este quien pueda clasificar y agrupar los datos que sigan estas características.

Con esto mencionado anteriormente, es posible deducir que este tipo de aprendizaje es utilizado en problemas de agrupamiento y etiquetado o reducción de características.

3.2 REDES NEURONALES

Una vez definidos los conceptos preliminares, se puede ahondar en el término red neuronal: esta es un agente inteligente que utiliza un modelo de aprendizaje supervisado y busca implementar la inteligencia artificial tratando de replicar el funcionamiento del cerebro humano, siendo desarrolladas con referencia a un modelo simplificado del sistema nervioso humano (Russel R. , 2018).

En este contexto y profundizando más en el concepto, las redes neuronales artificiales son sistemas de neuronas sintéticas interconectadas entre sí, tratando de simular las conexiones neuronales naturales que existen en el cerebro humano tal que, como menciona Mahesh, usando una serie de algoritmos intentan reconocer relaciones subyacentes en un conjunto de datos a través de un proceso que imita la forma en que opera el sistema nervioso central de las personas (Mahesh, 2018).

Seguido a esta idea y haciendo mención a lo planteado por Russell y Norvig, el funcionamiento de las redes neuronales artificiales comenzó a ser desarrollado bajo la hipótesis (hoy en día confirmada) que la actividad cerebral y mental consistía principalmente en envío de señales electroquímicas (sinapsis) entre células cerebrales conocidas como neuronas (Russel & Norvig, 2010), siendo esta línea de pensamiento continuada por Russell en su trabajo (Russel R. , 2018).

En contraste al cerebro biológico y siguiendo lo planteado por los autores de los trabajos mencionados en la sección anterior, las redes neuronales artificiales usan funciones matemáticas para simular el envío la sinapsis entre neuronas mediante el uso de funciones de activación las cuales, dependiendo del resultado de salida que tengan, si sobrepasan un umbral se "activan", enviando el resultado al elemento inmediatamente conectado, siendo esta acción sinónima al envío de señales entre las redes de neuronas naturales que hay en cada ser humano. En la siguiente sección se entrará a detalle en la estructura de este tipo de modelo de aprendizaje automático y el funcionamiento de cada componente.

3.2.1 ESTRUCTURA DE UNA RED NEURONAL

Al tratar de replicar una estructura tan compleja como el cerebro humano, una red neuronal consta no solo de muchos cálculos matemáticos, necesita de una estructura ordenada y detallada para lograr su cometido de una manera óptima y ágil. Como componente principal en el cerebro humano están las neuronas y, de manera similar a una red neuronal biológica, una red neuronal artificial también consta de neuronas que reciben información de fuentes externas o de otras, siendo estas neuronas artificiales. En el sistema nervioso humano, las fuentes externas son los cinco sentidos, y de manera análoga, en una red neuronal artificial, la fuente externa es una capa de perceptrones encargadas de recibir los datos y empezar la propagación de salidas. Después de sopesar y agregar las entradas, el resultado se transforma en salida mediante una función de transferencia de la que se estará hablando más a detalle en secciones posteriores.

A grandes rasgos, una red neuronal artificial cuenta con, mínimo, tres capas de neurona interconectadas entre sí, siendo la primera capa la de entrada que es la que replica el funcionamiento de la vista, el tacto o la audición. Luego tenemos una capa oculta de neuronas que es donde se realizan las operaciones matemáticas que simulan el proceso de sinapsis entre neuronas y una capa final que es la capa de salida que es la que devuelve el resultado final, como nos muestran Sharma, Rai y Dev en su trabajo con el siguiente diagrama (Sharma, Rai, & Dev, 2012) el cual también sigue la línea de trabajo implementada por Suzuki en su libro (Suzuki, 2011):

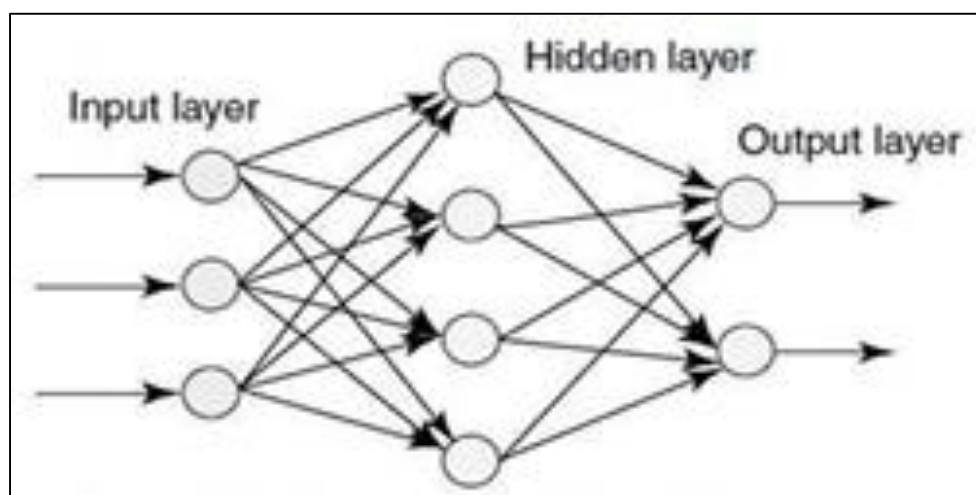


Ilustración 2: Diagrama básico de una red neuronal

3.2.2 NEURONA ARTIFICIAL

La complejidad de las neuronas reales queda muy abstraída al modelar neuronas artificiales. Esencialmente, como afirma Suzuki en su libro, éstas se componen de entradas, que se multiplican por pesos (la intensidad de las señales correspondientes) y luego se calculan utilizando una función matemática que decide cuándo se activará la neurona (Suzuki, 2011) siendo este resultado (si fue activada o no) enviado a las n salidas a las que esta neurona se encuentre conectada y enlazada.

Siguiendo este concepto, en el libro de Russell podemos encontrar el siguiente diagrama que ejemplifica la estructura de una neurona artificial primitiva el cual fue planteado originalmente por McCulloch y Pitts en 1943 (Russel & Norvig, 2010), diseño que hoy en día sigue siendo el pilar a diseñar perceptrones/neuronas artificiales:

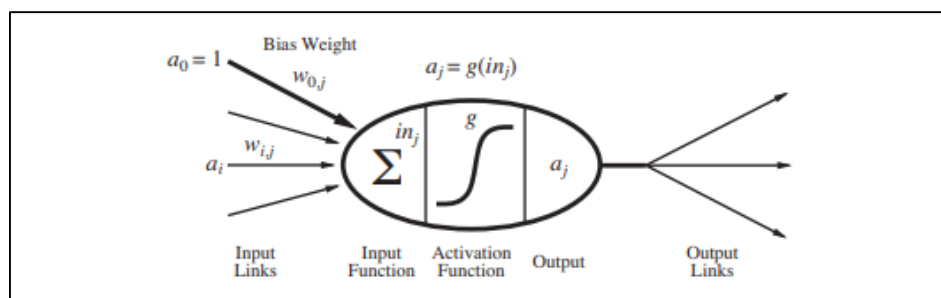


Ilustración 3: Neurona Artificial Primitiva

En este diagrama la función de activación de salida, que es un cálculo matemático que el programador elige, el cual devuelve un resultado que es evaluado sobre un umbral de activación propio de la neurona para saber si esta es activada o no y se estará entrando a detalle sobre este cálculo en secciones posteriores, es la función g la cual, siguiendo lo planteado por Suzuki, es la evaluada en la sumatoria del producto entre los valores de entrada a_i por los pesos de cada conexión precedente $w_{i,j}$ y es asignado en el valor a_j de tal manera que el valor final se ve de la forma:

$$a_j = g\left(\sum_{i=0}^n w_{i,j} a_i\right)$$

Ecuación 1: Generalización de la función de salida de una neurona artificial

Como se menciona anteriormente, cada neurona está conectada a, mínimo, una entrada y una salida, y es gracias estas conexiones que, comprobando y uniendo las neuronas que se han activado, que la red logra “pensar” e intenta predecir usando su modelo de aprendizaje; planteando una réplica al sistema nervioso central del ser humano de la que se habló anteriormente, siendo el resultado a_i evaluado en el umbral la sinapsis entre neuronas. En la sección siguiente a esta se estará explicando cómo funciona esta operación matemática y su finalidad.

3.2.3 FUNCIONES DE ACTIVACIÓN

Uno de los componentes clave de las redes neuronales artificiales, son las funciones de activación que utiliza cada neurona de la red para agregar no linealidad al modelo y brindar salidas mejor procesadas (Russel R. , 2018). Con esto es posible afirmar el hecho que para una red logre aprender y representar relaciones o patrones complejos en los datos, estas funciones que determinan la salida de una neurona en función de su entrada, son esenciales porque sin ellas, la red sería simplemente una combinación lineal de entradas, lo que limitaría su capacidad para aprender y representar relaciones no lineales en los datos (Bishop, 1995), de igual manera que las funciones de pérdida sobre las que se estará entrando a detalle en la siguiente sección del presente trabajo.

Para comprender mejor el concepto y uso de las funciones de activación, García Serrano en su libro (García Serrano, 2016) plantea como ejemplo un banco que necesita de un programa que sea capaz de clasificar a sus clientes para ver si, en caso que el cliente lo solicite, se le conceda una hipoteca teniendo en cuenta dos factores: el sueldo bruto del cliente x_1 y su deuda neta x_2 . Para este ejemplo se utiliza una neurona simple, que después de un entrenamiento e realizado, se consigue tener como pesos $w_1 = -0.2$, $w_2 = 0.5$ (uno por cada valor de entrada) y quedando el umbral de salida como $\theta = 0.6$, con lo cual siguiendo lo planteado en la *Ecuación 1* encontrada en la página 15 en la sección 3.2.2 más atrás debe ser asignado al valor a_j , lo cual desarrollando la sumatoria y sustituyendo en la formula queda de la forma: $a_j = -0.2x_1 + 0.5x_2$ y el resultado obtenido evaluando los dos valores de entrada debería ser mayor al umbral $\theta = 0.6$, lo que reescrito siguiendo la línea de pensamiento planeada por Suzuki en su trabajo (Suzuki, 2011)

puede ser asignado a una función g , fungiendo esta como la función de activación y quedando para este ejemplo en específico de la forma: $g = -0.2x_1 + 0.5x_2 - 0.6$, siendo esta una ecuación que podría ser representada gráficamente como una recta graficada usando Desmos, viéndose de la siguiente manera haciendo alusión a lo postulado por García Serrano:

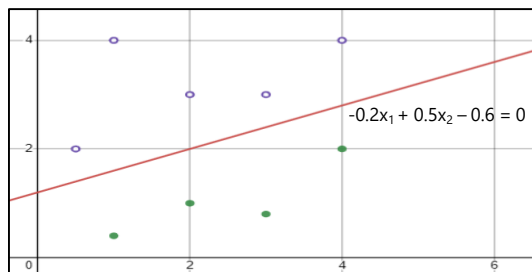


Ilustración 4: Ejemplo gráfico de función de activación lineal

Siendo la recta la que representa la función de activación denotando que si el valor de salida de la función de activación evaluando en los valores de entrada x_1 y x_2 es mayor o igual a, en este caso, 0.6 significa que la neurona se activó y entonces esos valores de entrada pertenecen a un cliente que si aplica a una solicitud positiva de hipoteca, siendo representados por los puntos sin rellenar. Caso contrario los puntos rellenos, siendo estos aquellos que dados sus valores de entrada x_1 y x_2 son menores a 0.6 y lo que indica que no se activó dicha neurona, y en caso de solicitar una hipoteca, recibirían una respuesta negativa. Independientemente del tipo o la cantidad de valores de entrada, siempre que el problema sea uno de regresión se espera que la función de activación se comporte de manera similar.

De manera análoga, es posible decir que este tipo de problemas se asemejan al comportamiento de las compuertas lógicas OR, teniendo este tipo de compuertas solamente dos tipos de salida, o 1 o 0 que pueden ser interpretadas en un enfoque gráfico de la siguiente manera como nos muestra García Serrano:

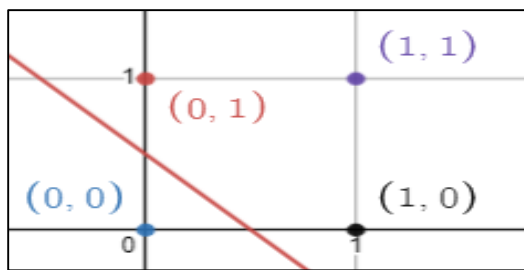


Ilustración 5: Recta para la función lógica OR

Es posible ver lo sencillo que es separar los valores, pues al tener al menos un valor de entrada 1, sin importar el valor de la otra entrada, el punto siempre estará sobre la recta, caso contrario si ambos valores son 0, el valor estará debajo de la recta y esto es algo que es posible comprobar revisando la tabla de verdad para la operación lógica homónima.

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

Ilustración 6: Tabla de verdad función lógica OR.

Pero, ¿funcionaría sin errores un acercamiento similar en los demás tipos de compuertas? La respuesta es que depende. Fausset en su trabajo compartiendo su línea de pensamiento con García Serrano, nos afirma que para las compuertas AND y NOT, si es posible atacar el problema y el entrenamiento de la red neuronal de una manera similar replicando la estrategia utilizada con la compuerta OR, debido a que estas compuertas resuelven problema linealmente separables y no mutuamente excluyentes (Fausett, 1994); caso contrario, por ejemplo, la compuerta XNOR, siendo esta compuerta una que involucra eventos mutuamente excluyente por definición y a su vez, no linealmente separables; para reforzar esta idea es necesario visualizar la tabla de verdad de la función XNOR:

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

Ilustración 7: Tabla de verdad función lógica XNOR

Fausset afirma que después de experimentar con la función XNOR usando la misma estrategia que implemento para la compuerta OR, lo que logró fue meramente un ciclo infinito en el entrenamiento de la red (Fausett, 1994). Tomando dos enfoques distintos intento dar con la solución, conectando más neuronas en la capa interna conocida como capa oculta, lo cual si bien funciona, termina siendo un resultado engorroso y muy ineficiente al tener que realizar más de una sola evaluación de función con los valores de entrada, lo que la llevo a tomar el segundo enfoque desprendiéndose de la linealidad en la función de activación, usando funciones de orden n con $n \geq 2$, algo que gráficamente lo refuerza García Serrano en su trabajo con diagramas como los mostrados a continuación (García Serrano, 2016):

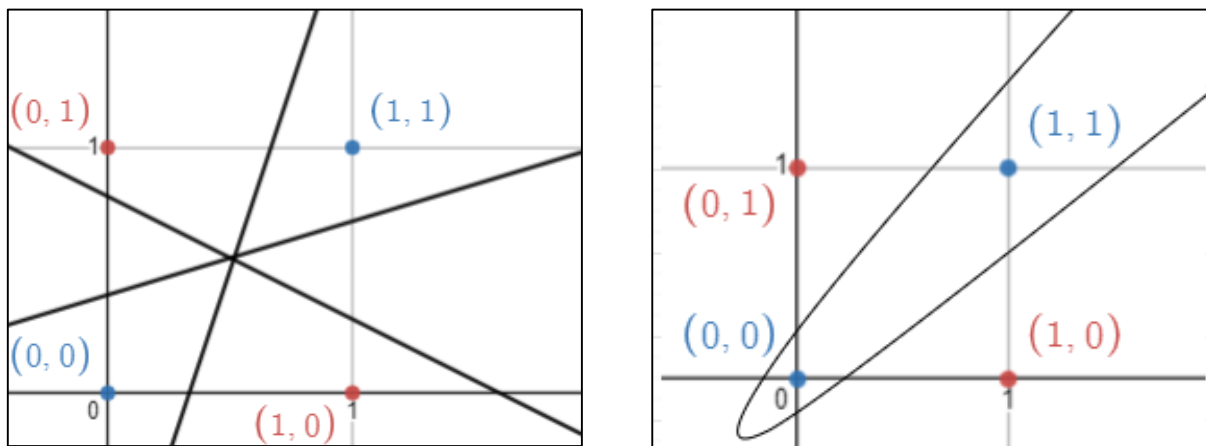


Ilustración 8: Comparación gráfica entre combinación de varias neuronas con funciones de activación lineales y una sola neurona con una función de activación no lineal sobre la función lógica XNOR.

Siguiendo esta lógica, nació la necesidad de tener otro tipo de funciones de activación para así conseguir distintos niveles de complejidad en la red neuronal combinando funciones no lineales junto a la conexión de varias neuronas entre si al momento de intentar partir el plano al separar y clasificar las salidas, y entre las funciones más utilizadas se encuentra las funciones sigmoideas (ecuaciones cuya gráfica tiene forma de "S") como la tangente hiperbólica es lo que mencionan Fausset, Bishop y García Serrano respectivamente en sus libros; siendo así debido a que suelen estar acotadas entre -1 y 1 y añadido a esto, tienen un rápido crecimiento hacia 1 cuando se trata de valores positivos o un rápido decrecimiento a -1 para valores negativos. Esto le permite a la red comprender y tener salida que no sean valores discretos entre 0 y 1 si no, valores reales.

3.2.4 FUNCIONES DE PÉRDIDA

La importancia de las funciones de pérdida en el proceso de entrenamiento de una red neuronal surge de su capacidad para medir la discrepancia entre los valores predichos y los valores reales. La minimización de esta función de pérdida es el objetivo del entrenamiento, lo que hace que la red neuronal se adapte y mejore en sus predicciones. La naturaleza de la tarea determinará la mejor función de pérdida a utilizar, ya que diversas cuestiones requieren diversas formas de medir la diferencia entre las predicciones y las etiquetas reales. Como postula Bishop en su libro (Bishop, 1995), en la resolución de este tipo de problemas, la función de pérdida predilecta es el **error de la suma de cuadrados** (Mean Squared Error), la cual implementa en el conjunto de datos objetivo normalmente (más no obligatoriamente) una distribución Gaussiana. En términos matemáticos, y como muestran Hodson, Over y Foks en su investigación, la función se ve de la siguiente manera (Hodson, Over, & Foks, 2021):

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$$

Ecuación 2: Mean Squared Error o error de la suma de cuadrados.

Donde:

- \hat{y}_i es la predicción dada por el modelo para x_i de la muestra (x_i, y_i) .
- y_i es la salida real para x_i dada la muestra (x_i, y_i) .
- N es el total de muestras.

En lo que respecta a los problemas de clasificación, Bishop menciona que se sigue teniendo una condición de las variables de entrada sobre las de salida con la diferencia que este tipo de problemas buscan catalogar cada entrada en un conjunto finito de clases por lo que, si bien el error de la suma de cuadrados puede ser útil no suele ser la mejor opción para esto debido a que en esta categoría de problemas la red neuronal busca la presencia o ausencia de atributos que no son necesariamente mutuamente excluyentes como si es el caso de los problemas de regresión, siendo la **función de entropía cruzada** (Cross-Entropy Loss) la mejor opción para estos problemas (Bishop, 1995), enfocándose en su implementación en la comparación de dos o más distribuciones de probabilidad (una por cada clase) a diferencia de la suma de error de los

cuadrados, pues esta se enfoca en solamente una distribución de probabilidad (Hodsom, Over, & Foks, 2021).

Como se menciona anteriormente, la función de entropía cruzada es usada dentro de los problemas de clasificación y estos pueden ser de dos categorías: clasificación binaria o multiclase. Cuando se habla de clasificación binaria, es cuando la salida es o positiva o negativa; un ejemplo puede ser el diferenciar entre perros y gatos de una fotografía mientras que al hablar de clasificación multiclase es cuando la clasificación de la entrada cae dentro de 3 o más clases dentro de un conjunto finito de posibilidades como lo podría ser determinar a qué tipo de tumor (glioma, meningioma, pituitario, etc.) pertenece el encontrado en una resonancia magnética de cerebro.

En término matemáticos, cuando el problema planteado es sobre clasificación binaria, la función de pérdida de entropía cruzada se ve de la siguiente manera como muestra Brownlee en su trabajo (Brownlee, 2019):

$$CEL = -[y \log(p) + (1 - y) \log(1 - p)]$$

Ecuación 3: Cross-Entropy Loss o Pérdida de Entropía Cruzada en clasificación binaria.

Donde:

- y es la salida (1 o 0) real del valor de entrada x de la muestra (x, y) del conjunto de datos.
- p es la probabilidad predicha por el modelo sobre el valor de entrada x de la muestra (x, y) del conjunto de datos.

Siendo la función mostrada anteriormente el desarrollo para dos términos de la sumatoria que constituye a la función en la clasificación multiclase que se ve de la siguiente manera:

$$CEL = - \sum_{i=1}^C y_i \log(p_i)$$

Ecuación 4: Cross-Entropy Loss o Pérdida de Entropía Cruzada en clasificación multiclase.

Donde:

- y_i es la salida del valor de entrada x_i de la muestra (x_i, y_i) del conjunto de datos.
- p_i es la probabilidad predicha por el modelo para la clase i .
- C es la cantidad de clases en la muestra.

Para mejorar la capacidad del modelo de red neuronal de realizar clasificaciones precisas, el objetivo del proceso de entrenamiento es minimizar esta función de pérdida lo más cercano a 0 (Bishop, 1995). No tiene que ser una de estas dos opciones forzosamente la que el programador encargado de diseñar la red debe de elegir, existen muchas otras opciones como lo son la función de pérdida hinge que se utiliza en problemas de clasificación de márgenes suaves y está diseñada para maximizar el margen entre las clases y como nos menciona Varma en su trabajo, es útil para problemas de clasificación binaria (Varma, 2018). Se encuentra también la regresión de Poisson que se utiliza en problemas de regresión cuando se asume que los datos siguen, como su nombre sugiere, una distribución de Poisson (Li & Lu, 2023) o la divergencia de Kullback-Leiber la cual utiliza en problemas de divergencia entre distribuciones de probabilidad siendo comúnmente utilizada en problemas de aprendizaje profundo generativo (Togami, Masuyama, Komatsu, & Nakagome, 2020) por mencionar otro conjunto de funciones de pérdida utilizadas en sistemas de aprendizaje automático y aprendizaje profundo.

3.2.5 REDES NEURONALES CONVOLUCIONALES

Este tipo de red neuronal es posible considerarla como la que más se acerca al funcionamiento del cerebro humano pues como plantea Artola Moreno, en las redes neuronales convolucionales las neuronas de la red corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico (Artola Moreno, 2019) dando así un mejor acercamiento y ataque para solucionar problemas que requieran de visión artificial ya que, las computadoras no ven las imágenes igual que un ser humano. Siguiendo lo mencionado por Stockman y Shapiro en su libro, para un hombre es simple, si en la imagen hay píxeles contiguos que cambian drásticamente de color o valor, una persona sana puede decir con confianza que es un eje o un borde mientras que las computadoras lo que ven es una serie de píxeles y cada píxel se conforma de valores numéricos que indican sus niveles de color (Stockman & Shapiro, 2001) ya sea en formato RGB (**R**ed-**G**reen-**B**lue) o CYMB (**C**yan-**Y**ellow-**M**agenta-**B**lack), sin saber distinguir exactamente si es o no un borde o eje algún pixel en específico.

La primera solución a la que muchos programadores principiantes suelen llegar es recorrer toda la matriz de pixeles y comenzar a comparar uno por uno, verificando si hay cambios agresivos en

las tonalidades o sus valores de color, siendo esta una aproximación no tan alejada al funcionamiento real de una red neuronal convolucional, específicamente cuando se realiza el proceso de convolución del cual se estará entrando a detalle en secciones posteriores. Continuando con la detección de bordes y ejes en una imagen no es posible hacerlo viendo los píxeles de manera individual, sino considerar el contexto del valor de un pixel y comparándolo contra los píxeles que tiene alrededor, usando una herramienta llamada núcleo o kernel (Artola Moreno, 2019).

3.2.5.1 Proceso de Convolución

Lo que se hace con el núcleo explicado en la sección anterior es que sea este el que itere la matriz de píxeles tomando, en vez de solamente un píxel a la vez, sino un conjunto de $n \times m$ píxeles, definiendo de ese conjunto un solo pixel principal que es el que se estará revisando, el cual será el foco principal a tener en cuenta, pues, siguiendo lo mencionado por autores anteriores cada pixel es multiplicado por un mismo valor exceptuando el pixel principal del núcleo que será operado con un valor distinto, para luego sumar los valores obtenidos siendo colocado en la imagen de salida, lo que resultará en haciendo una especie de filtro mediante este proceso, lo que a su vez también permite eliminar el ruido dentro de los datos y así enfocarse en, solamente, los ejes y bordes de las imágenes por mencionar usos elementales de este proceso. Un ejemplo simple para encontrar ejes sería multiplicar el valor principal por un valor alto (9 por ejemplo) y los demás del núcleo por -1 (Dávila, 2022) lo que nos dejaría una imagen parecida a la siguiente:



Ilustración 9: Ejemplo de localización de ejes usando convoluciones

Así como este, existen también filtros de desenfoque, de cambio de tonalidades entre otros, los cuales implementan el mismo concepto: multiplicar los valores del núcleo por x o y valor dependiendo de la importancia o relevancia que se les quiera brindar.

Entrando de lleno en lo que realiza una red neuronal convolucional, es realizar la aplicación de varios filtros de los explicados anteriormente de manera simultánea en todos los núcleos por cada iteración de manera que la máquina pueda tener una vista más compleja de la imagen.

Cabe aclarar que este proceso de convolución explicado con anterioridad se enfoca en imágenes en blanco y negro. Para imágenes a color se debe tomar en cuenta que existen de tres (imágenes RGB) a cuatro (imágenes CYMB) canales de pixeles, lo que implica que por cada iteración habrá que realizar el proceso de tres o cuatro veces pues en cada iteración se encuentra esa cantidad de núcleos.

3.3 ENFERMEDADES

3.3.1 TUMORES CEREBRALES

Los tumores cerebrales son una de las afecciones que involucran la generación de nuevo tejido de manera anormal que más afectan a la población mundial superado en cifras por aquellos tumores específicos por género: tumores mamarios en las mujeres y tumores prostáticos en los hombres, dato brindado por Sánchez Gatica en su estudio (Sánchez Gatica, 2002), siendo este dato reforzado por Gómez-Vega, Ocampo y Lee en su estudio añadiendo también que entre los más comunes en la población adulta existen 3 tipos de tumores siendo estos: meningioma, glioma y pituitarios (Gómez-Vega, Ocampo Navia, & Lee, 2019).

A grandes rasgos, los tumores cerebrales se diferencian de tumores en otras zonas corporales debido a la complejidad del cerebro, y no siempre es posible tener un diagnóstico correcto entre tumor benigno o maligno por lo difícil que es visualizar una zona tan comprometida como lo suele ser el sistema nervioso central, añadiendo también complejidad en el tratamiento al tener una extracción quirúrgica comprometida al manipular el centro principal del funcionamiento del cuerpo humano.

En lo que respecta a la clasificación y diferenciación de los distintos tipos de tumores, Gómez-Vega, Ocampo y Lee mencionan que los meningiomas suelen ser tumores redondos en mayor parte y comúnmente solitarios, siendo originados a partir una de las tres capas de las meninges, la capa aracnoides, siendo las otras dos la piamadre y la duramadre. En términos generales, la aracnoides es la capa que rodea la parte externa del cerebro teniendo como función protegerlo y brindarle nutrientes para que se puedan transmitir correctamente los impulsos eléctricos entre neuronas. En términos de cifras, estos suelen ser los más encontrados en la población adulta respecto a los otros dos tipos de tumores mencionados con anterioridad, teniendo hasta un 37% más de incidencia como mencionan Gómez-Vega, Ocampo y Lee, siendo también sustentado en cifras por lo encontrado en la base de datos principal de Mayo Clinic (Mayo Clinic, 2022).

Luego se encuentran los gliomas los cuales son una multiplicación de células que comienza en el cerebro o la médula espinal. Las células de los gliomas se parecen a las neuronas cerebrales sanas, llamadas células gliales, células que rodean a las células nerviosas y las ayudan a funcionar (Mayo Clinic, 2023) y entre sus causas la más probable es una variación en el ADN de las células. Este tipo de tumor corresponde al 24% casos encontrados siguiendo lo dicho por Gómez-Vega, Ocampo y Lee.

Cerrando con la clasificación de los tumores más comunes se encuentran los pituitarios, correspondiendo a una incidencia de 3,47 de cada 100,000 pacientes siguiendo las mismas cifras planteadas por Gómez-Vega, Ocampo y Lee. Los tumores pituitarios son masas anormales que se desarrollan como su nombre hace mención, en la glándula pituitaria; un órgano que se encuentra detrás de la nariz, en la base del cerebro (Mayo Clinic, 2023). La causa del crecimiento celular descontrolado en la glándula pituitaria que provoca un tumor, sigue siendo desconocida. Es posible, aunque en muy raras ocasiones, que los tumores pituitarios sean causados por genes heredados como mencionan Gómez-Vega, Ocampo y Lee.

3.3.2 ALZHEIMER

La enfermedad de Alzheimer es un trastorno cerebral que empeora con el tiempo. Se caracteriza por cambios en el cerebro que provocan depósitos de determinadas proteínas, ocasionada en su mayoría de casos conforme avanza la edad en el paciente, aunque también existe la posibilidad que una persona adquiera esta enfermedad por factores genéticos hereditarios. La enfermedad de Alzheimer hace que el cerebro se encoja y las células cerebrales eventualmente mueran, siendo a su vez esta enfermedad la causa más común de demencia (Donoso, 2003). Como nos menciona Mayo Clinic y la OMS, el olvido de eventos o conversaciones recientes es uno de los primeros signos de la enfermedad (World Health Organization, 2023). Con el tiempo, el paciente desarrolla problemas de memoria graves y pierde la capacidad de realizar las tareas diarias e incluso olvidar acciones que realizó hace un instante (Mayo Clinic, 2023).

La necesidad de ayuda con el cuidado personal aumenta a medida que avanza la enfermedad. Los pacientes con demencia pueden tener problemas para moverse, tener problemas para comer y beber y experimentar cambios de comportamiento como agresión hacia sus cuidadores o a sí mismo.

3.3.3 TUBERCULOSIS

Como Bermejo et al. y López Marín et al. mencionan, actualmente, la tuberculosis se considera un gran problema de salud pública, ya que es la causa de muerte más común por agentes infecciosos y representa aproximadamente la cuarta parte de la mortalidad evitable en los países en desarrollo, donde se registran la mayoría de los casos y fallecimientos, siendo los países en vías de desarrollo los más afectados por esta enfermedad sin importar un rango de edad, mientras que en los países más desarrollados, esta suele verse en pacientes de la tercera edad (Bermejo, Clavera, Michel de la Rosa, & Marín, 2007).

La tuberculosis es una infección pulmonar causada por la bacteria *Mycobacterium tuberculosis*. Aunque las bacterias de la tuberculosis pueden infectar cualquier parte del cuerpo, como los riñones, la columna vertebral y el cerebro, los pulmones son el área más común donde se desarrollan (López Marín, Díaz Otero, Vallecillo Maza, Esquível Solís, & Gutierrez Pabello, 2006) (World Health Organization, 2023).

3.3.4 NEUMONÍA

Siguiendo lo planteado por el NHLBI, la neumonía es una infección que afecta uno o ambos pulmones. Hace que los sacos de aire o alvéolos de los pulmones se llenen de líquido o pus y a diferencia de la tuberculosis, la neumonía puede ser causada por bacterias (*Streptococcus pneumoniae*), virus (virus sincital respiratorio) u hongos. Los síntomas pueden variare entre tos con o sin moco (una sustancia viscosa), fiebre, escalofríos y dificultad para respirar. La gravedad de su neumonía depende de la edad, salud general y causa de su infección del paciente (National Heart, Lung, and Blood Institute, 2022).

3.4 IMÁGENES MÉDICAS

Para la correcta detección de las enfermedades planteadas anteriormente, es normal que los especialistas receten a sus pacientes tomar exámenes ambulatorios como también tener un muestreo en imagen de la zona a estudiar, entre los más usados se encuentran las resonancias magnéticas para el caso de las enfermedades cerebrales y las radiografías para las enfermedades pulmonares siguiendo los datos planteados por los sitios principales tanto de la OMS, Mayo Clinic, la CDC y el NHBLI.

3.4.1 RESONANCIAS MAGNÉTICAS

Sell plantea que el propósito principal de una resonancia magnética es realizar una localización anatómica de un proceso funcional (Sell, 2007), y siguiendo este concepto encontramos lo planteado por Zaragoza-Velasco y Fernández-Tapia, quienes afirman que la resonancia magnética es un método bastante útil para el análisis del tejido blando del cuerpo humano (Zaragoza-Velasco & Fernández-Tapia, 2013).

Como definición, es posible decir que una resonancia magnética es un examen imagenológico que utiliza imanes y ondas de radio potentes para crear imágenes del cuerpo en el cual no se emplea radiación ionizante (rayos X) (MedlinePlus, 2022).

3.4.2 RADIOGRAFÍAS

Estas son un tipo de imágenes médicas que utilizan los rayos X, un tipo de radiación electromagnética ionizante de la misma naturaleza de las ondas de radio, los rayos infrarrojos, la luz visible, los rayos ultravioleta y los rayos gamma (Cullity, 1956).

Siguiendo lo mencionado por el NIBIB, los rayos X médicos se utilizan para generar imágenes de los tejidos y las estructuras dentro del cuerpo. Si los rayos X que viajan a través del cuerpo también pasan a través de un detector de rayos X al otro lado del paciente, se formará una imagen que representa las siluetas formadas por los objetos dentro del cuerpo (National Institute of Biomedical Imaging and Bioengineering, 2022). Añadiendo a lo mencionado anteriormente y siguiendo lo planteado por el NIBIB, es posible concluir que los rayos X muestran el interior del cuerpo en diferentes tonos de blanco y negro debido a que diferentes tejidos absorben diferentes cantidades de radiación. El aire absorbe la menor cantidad, por lo que los pulmones se ven negros, lo que es muy conveniente al querer detectar algún acceso de líquido en los pulmones ocasionado posiblemente por alguna infección como tuberculosis o neumonía. El calcio en los huesos absorbe la mayoría de los rayos X, por lo que los huesos se ven blancos mientras que la grasa y otros tejidos blandos absorben menos, y se ven en distintas tonalidades de gris.

IV METODOLOGÍA

4.1 ENFOQUE

Al ser esta investigación una que depende tanto de resultados numéricos como de un análisis exhaustivo de los mismos es posible concluir que se tiene un enfoque mixto. Desde el punto de vista cuantitativo el proyecto realizado en esta investigación busca minimizar el valor de la función de pérdida mientras que, desde el enfoque cualitativo, el proyecto en cuestión busca dar como resultados distintos tipos de diagnósticos basados en los resultados brindados por la función de pérdida en cada tipo de red neuronal.

4.2 VARIABLES DE INVESTIGACIÓN

- Imágenes médicas y sus distintos resultados con las respectivas enfermedades diagnosticadas.
- Profundidad y complejidad de las redes neuronales

4.3 TÉCNICAS E INSTRUMENTOS APLICADOS

Se utilizaron mayormente técnicas propias al aprendizaje de máquina, un pilar fundamental para el funcionamiento y evolución de las inteligencias artificiales y a su vez técnicas de medición y métricas meramente estadísticas para la documentación de resultados del trabajo realizado.

Correspondiente al área de aprendizaje de máquina e IA se utilizaron redes neuronales convolucionales implementado Python y la librería Pytorch, un ambiente de trabajo dedicado a procesos que involucren aprendizaje de máquina desarrollado por la empresa MetaAI. Estas basaban su método de aprendizaje en el algoritmo de "backpropagation", resumido del inglés "propagación hacia atrás de los errores", el cual se enfoca en que por cada iteración hacia adelante en el aprendizaje de una neurona de la red, esta envía hacia sus predecesoras una señal y les avisa sobre qué cambios realizó en sus pesos y sesgos (weights and bias) para así utilizar estos valores en el cálculo del descenso del gradiente por cada capa neuronal, calculando el primer gradiente en la capa final y por último el de la capa inicial (Dávila, Redes Neuronales y Algoritmos de Aprendizaje de Máquina, 2022).

En lo que respecta al apartado estadístico del proyecto, se debieron recolectar un gran número de imágenes de muestra para convertirse en la población sobre la cual se iba a realizar el estudio, análisis y desarrollo del proyecto dividiéndolas en sets de entrenamiento y de validación para que con cada iteración completa de entrenamiento, la red pudiera lanzar una predicción y así medir la exactitud de los resultados (accuracy) para determinar, iteración a iteración, un porcentaje de que tan bien estaban realizando su trabajo que a su vez la idea principal es llegar esa métrica a un valor de 100% de exactitud o en su defecto un valor muy cercano (~98% o más) así como la precisión del modelo y su recall siguiendo el mismo principio mencionado anteriormente. Finalmente se utilizaron gráficos de línea para la visualización y documentación de los resultados

obtenidos y la librería SciKit Learn para la documentación de métricas relevantes en el proceso de validación como lo son el recall, puntaje F-1 y la precisión.

4.4 MATERIALES

4.4.1 HARDWARE

HP Pavilion Notebook 15-bc407TX 2018

- *Procesador:* Intel Core I5-8300H
- *RAM:* 8GB DDR4-2400 SDRAM
- *Batería:* 3 celdas de iones de litio.
- *Tarjeta gráfica:* NVidia GeForce GTX 1050

4.4.2 SOFTWARE:

- *Sistema Operativo:* Ubuntu 22.04 LTS Jammy Jellyfish Release.
- *IDE/Editores de texto utilizados:* Visual Studio Code 2019
- *Redacción de reporte:* Microsoft Office Word 2013.

4.4.3 TOOLKITS Y HERRAMIENTAS DE PROGRAMACIÓN:

- *Lenguaje de Programación:* Python v3.12.0.

Módulos nativos de Python:

Entrenamiento de Redes Neuronales: sys, time, numpy, argparse, os.

GUI: Tkinter.

Documentación gráfica de resultados: matplotlib.pyplot.

- *Framework para implementación de Aprendizaje de máquina:* Pytorch 2.1.

Librerías:

Arquitectura y Entrenamiento de Redes Neuronales: torch.nn, torch.nn.functional, torchvision.transforms, torchvision.dataset.ImageFolder, torch.utils.data.DataLoader.

4.5 METODOLOGÍA DE ESTUDIO

Comenzando el desarrollo del proyecto, era imperativo tener una gran cantidad de imágenes que pudieran nutrir a los modelos de aprendizaje profundo utilizados a lo largo del tiempo de trabajo; imágenes de las cuales no solo importaba la cantidad, también su calidad ya que no es lo mismo un set de datos de quince mil imágenes (15,000) de 5 x 5 píxeles que uno de la misma cantidad de imágenes, pero con una resolución más alta como lo podría ser 240 x 240 píxeles. Si bien en el proceso de entrenamiento estas imágenes debían ser normalizadas y estandarizadas, las imágenes de menor calidad suelen sufrir la tenencia de más ruido al tener que distorsionar tan pocos píxeles de modo que es muy complicado poder identificar características imprescindibles como lo pueden ser los bordes y filos de la misma.

En este proceso se utilizaron mayormente imágenes recolectadas de varios datasets públicos descargados desde la página Kaggle, llegando a tener una cantidad de aproximadamente ciento veinte mil imágenes (120,000); número que con el paso del tiempo fue reduciéndose debido al proceso de filtrado y revisión de las mismas, proceso que duro alrededor de 2 semanas por todos los datasets, dejando una cantidad final cerca de sesenta y ocho mil imágenes totales (68,000). Acto seguido se precedió a separarlas entre sus respectivas clases:

- Tumores cerebrales: Glioma, meningioma, pituitarios, cerebro sano.
- Alzheimer: Demencia moderada, demencia media, demencia muy media, cerebro sano.
- Enfermedades pulmonares: Tuberculosis, neumonía, pulmones sanos.

Una vez completado lo anterior, se realizó la partición de datos de entrenamiento y validación en una proporción aproximada de 80/20 para cada una de estas clases llevando a la construcción de los datasets que alimentarían a las redes neuronales de diagnósticos.

Añadido a este proceso, se realizó un set de datos aparte juntando todas las imágenes, pero separándolas en solamente las tres clases principales para así entrenar una red neuronal capaz de clasificar las imágenes según el uso que se le daría; detectar y clasificar tumores cerebrales, detectar Alzheimer y su avance, detectar y clasificar enfermedades pulmonares.

Es notable mencionar que para manipular las imágenes y que estas pudieran ser entendidas por las redes y manipulables por las mismas, se debió programar un pre procesador de imágenes que

estandarizara y normalizara cada imagen antes de que las redes pudieran trabajar con el set de datos. Esto fue necesario para evitar la creación de dependencias no deseadas entre datos, reducir el tiempo de lectura y procesamiento de los datos y optimizar el uso de memoria por parte de la red neuronal ya que, si bien los datos y la red eran enviados a la tarjeta gráfica dedicada, siempre es necesario optimizar al máximo el uso de la memoria para evitar trabajos y tiempos demasiado tardados.

El pre procesamiento de imágenes se realizó mediante el siguiente bloque de código:

```
images_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    # transforms.RandomHorizontalFlip(p=0.5),
    # transforms.RandomVerticalFlip(p=0.5),
    # transforms.GaussianBlur(kernel_size=(5, 9), sigma=(0.1, 5)),
    # transforms.RandomRotation(degrees=(30, 70)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]),
])
valid_transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]),
])
```

Ilustración 10: Pre procesamiento de imágenes.

Es notable mencionar que primeramente se comenzó trabajando con estrategias de data augmentation las cuales no fueron efectivas, generando overfit de la red muy rápido, habiendo pasado muchas veces no menos de 15 épocas, por lo que al final se acabó solamente normalizando y estandarizando las imágenes.

Una vez concluido el proceso de recolección, filtro y estructuración de la información, se trabajó en el diseño de la estructura para las redes neuronales. Como primera opción, se manejaba tener cuatro (4) redes neuronales convolucionales distintas, tres (3) para los diagnósticos de enfermedades y una para la clasificación previa, pero se terminó usando solamente dos (2), una genera para el entrenamiento de los tres modelos y una aparte para la clasificación previa, esto debido a que tras los primeros entrenamientos usando la estrategia de tres redes distintas estaban

dando problemas de overfitting dos de ellas, la red para detección de tumores y la red para la detección de enfermedades pulmonares y con cada corrección realizada en ambos diseños, hacía que la estructura de las tres redes fuera cada vez más similar entre sí. Añadiendo a este punto, cabe resaltar que, si bien se comenzó haciendo estrategia de data augmentation sobre los datos, los mejores resultados se obtuvieron al no aplicar esto sobre las imágenes, dejando así un diseño de red neuronal convolucional con cinco (5) capas de convolución conectadas a tres (3) capas lineales mientras que para la red de clasificación pre-diagnostico se usaron tres (3) capas de convolución y dos (2) lineales.

```
import torch.nn as nn
import torch.nn.functional as F

class image_classifier(nn.Module):
    def __init__(self, out):
        super(image_classifier, self).__init__()
        self.out = out
        self.conv_1 = nn.Conv2d(3, 32, 3)
        self.conv_2 = nn.Conv2d(32, 64, 3)
        self.conv_3 = nn.Conv2d(64, 128, 3)

        self.linear_1 = nn.Linear(128, 32)
        self.linear_2 = nn.Linear(32, self.out)

        self.pool = nn.MaxPool2d(2,2)
        self.dropout = nn.Dropout(0.5)

    def forward(self, x):
        x = self.pool(F.relu(self.conv_1(x)))
        x = self.pool(F.relu(self.conv_2(x)))
        x = self.pool(F.relu(self.conv_3(x)))

        bs, _, _, _ = x.shape
        x = F.adaptive_avg_pool2d(x, 1).reshape(bs, -1)
        x = self.dropout(F.relu(self.linear_1(x)))
        x = self.linear_2(x)
        return x
```

Ilustración 11: Primer diseño de redes neuronales.

Quedando la estructura y diseño final de las redes neuronales de la siguiente manera, teniendo a la izquierda a la red para diagnósticos y a la derecha la red para clasificación previa:

<pre>import torch.nn as nn import torch.nn.functional as F class image_classifier(nn.Module): def __init__(self, out): super(image_classifier, self).__init__() self.out = out self.conv_1 = nn.Conv2d(3, 32, 3) self.conv_2 = nn.Conv2d(32, 64, 3) self.conv_3 = nn.Conv2d(64, 128, 3) self.conv_4 = nn.Conv2d(128, 256, 3) self.conv_5 = nn.Conv2d(256, 512, 3) self.linear_1 = nn.Linear(512, 126) self.linear_2 = nn.Linear(126, 32) self.linear_3 = nn.Linear(32, self.out) self.pool = nn.MaxPool2d(2,2) self.dropout = nn.Dropout(0.5) def forward(self, x): x = self.pool(F.relu(self.conv_1(x))) x = self.pool(F.relu(self.conv_2(x))) x = self.pool(F.relu(self.conv_3(x))) x = self.pool(F.relu(self.conv_4(x))) x = self.pool(F.relu(self.conv_5(x))) bs, _, _, _ = x.shape x = F.adaptive_avg_pool2d(x, 1).reshape(bs, -1) x = self.dropout(F.relu(self.linear_1(x))) x = self.dropout(F.relu(self.linear_2(x))) x = self.linear_3(x) return x</pre>	<pre>import torch.nn as nn import torch.nn.functional as F class image_classifier(nn.Module): def __init__(self, out): super(image_classifier, self).__init__() self.out = out self.conv_1 = nn.Conv2d(3, 32, 3) self.conv_2 = nn.Conv2d(32, 64, 3) self.conv_3 = nn.Conv2d(64, 128, 3) self.linear_1 = nn.Linear(128, 32) self.linear_2 = nn.Linear(32, self.out) self.pool = nn.MaxPool2d(2,2) self.dropout = nn.Dropout(0.5) def forward(self, x): x = self.pool(F.relu(self.conv_1(x))) x = self.pool(F.relu(self.conv_2(x))) x = self.pool(F.relu(self.conv_3(x))) bs, _, _, _ = x.shape x = F.adaptive_avg_pool2d(x, 1).reshape(bs, -1) x = self.dropout(F.relu(self.linear_1(x))) x = self.linear_2(x) return x</pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Ilustración 12: Diseño y estructura final de redes neuronales convolucionales.

Teniendo la estructura de las redes lista, el siguiente paso para la realización del proyecto era el entrenamiento de las redes neuronales, el cual se realizó mediante iteraciones repetitivas que usaban el set de datos ya estructurado y dividido en las particiones explicadas anteriormente. En cada iteración se debía hacer una sesión de entrenamiento le debía de dar a la red una imagen dentro del set y a que clase pertenece para que con cada iteración la red trate de hacer los cálculos necesarios para que la salida sea la clase real a la que esta imagen pertenece. Una vez realizado el entrenamiento, se hacía una sesión de validación del modelo, el cual seguía un proceso similar al usado en la sesión de entrenamiento con la diferencia que aquí no se realizan cambios sobre el modelo, usando los siguientes métodos que retornaban el valor de perdida en esa corrida, el cual se buscaba minimizar hasta cero o lo más cercano posible:

```

def training(model, trainloader, device, criterion, optimizer):
    # begin training
    model.train()
    print("TRAINING")
    training_running_loss = 0.0
    training_running_correct = 0
    cont = 0
    for i, data in tqdm(enumerate(trainloader), total=len(trainloader)):
        cont = cont + 1
        image, labels = data
        image = image.to(device)
        labels = labels.to(device)
        optimizer.zero_grad()
        output = model(image)
        loss = criterion(output, labels)
        training_running_loss += loss.item()
        # calculate the accuracy
        _, preds = torch.max(output.data, 1)
        training_running_correct += (preds == labels).sum().item()
        # backpropagation
        loss.backward()
        # update the optimizer parameters
        optimizer.step()
    epoch_loss = training_running_loss / cont
    epoch_accuracy = 100.0 * (training_running_correct / len(trainloader.dataset))
    return epoch_loss, epoch_accuracy

def validate(model, validation_loader, criterion, device):
    model.eval()
    print("VALIDATING")
    validation_running_loss = 0.0
    validation_running_correct = 0
    cont = 0
    y_values, predictions = [], []
    with torch.no_grad():
        for i, data in tqdm(enumerate(validation_loader), total=len(validation_loader)):
            cont += 1
            image, labels = data
            image = image.to(device)
            labels = labels.to(device)
            # forward pass
            outputs = model(image)
            # calculate the loss
            loss = criterion(outputs, labels)
            validation_running_loss += loss.item()
            # calculate the accuracy
            _, preds = torch.max(outputs.data, 1)
            y_values.append(labels.cpu().numpy())
            predictions.append(preds.cpu().numpy())
            validation_running_correct += (preds == labels).sum().item()
    epoch_loss = validation_running_loss / cont
    epoch_accuracy = 100.0 * (
        validation_running_correct / len(validation_loader.dataset)
    )
    return epoch_loss, epoch_accuracy, y_values, predictions

```

Ilustración 13: Código usado en el entrenamiento y validación.

Cabe mencionar que cada iteración en la que se hacía un proceso de entrenamiento y uno de validación, se conoce como época y para poder obtener un modelo correcto, en este proyecto se implementaron estrategias de detención temprana (early stopping) con lo que se buscaba parar el entrenamiento una vez el valor de la función de pérdida dejaba de decrecer y, por el contrario, aumentaba generando así un estado de overfit en la red. Para comprobar los resultados obtenidos se usaron los siguientes scripts que recorrían todas las imágenes del set de datos y finalmente imprimían el reporte de clasificación y la matriz de confusión del proceso:

```

y_values, predictions = [], []
with torch.no_grad():
    for i, data in tqdm(
        enumerate(validation_dataset_loader), total=len(validation_dataset_loader)
    ):
        cont = 0
        cont += 1
        image, labels = data
        image = image.to(device)
        labels = labels.to(device)
        # forward pass
        outputs = red(image)
        # calculate the accuracy
        _, preds = torch.max(outputs.data, 1)
        y_values.append(labels.cpu().numpy())
        predictions.append(preds.cpu().numpy())
        # time.sleep(0.2)

y_values = [i[0] for i in y_values]
predictions = [i[0] for i in predictions]
print(
    classification_report(
        y_values, predictions, zero_division=0, target_names=classes
    )
)

# confusion matrix
cm = confusion_matrix(y_values, predictions)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=classes)
disp.plot()
plt.show()

```

Ilustración 14: Script para la validación final de los modelos.

Concluyendo el desarrollo del prototipo, era necesario tener una interfaz gráfica de usuario para el uso amigable de los modelos entrenados anteriormente. Esta librería fue desarrollada usando

una librería nativa propia de Python, TKinter. Se diseñaron distintas pantallas tanto para la selección de una imagen y que el programa clasificara directamente el tipo y luego su diagnóstico o si el usuario deseaba enviar directamente la imagen a un modelo en específico, usando los siguientes bloques de código:

```
def main():
    app = App()
    app.geometry("720x320")
    diverse_classification_button = tk.Button(
        app,
        text="Clasificar y diagnosticar una imagen",
        command=lambda: open_file_choser(4),
    )
    specific_classification_button = tk.Button(
        app, text="Hacer un diagnóstico en específico", command=specific_prognosis_frame
    )
    diverse_classification_button.place(x=240, y=100)
    specific_classification_button.place(x=240, y=200)
    app.mainloop()

if __name__ == "__main__":
    main()
```

Ilustración 15: Código para mostrar la pantalla principal del GUI.

Partiendo de las acciones de cada botón, el usuario es direccionado a uno u otro método dependiendo la acción seleccionada. Para el uso directamente de uno de los modelos siendo seleccionado el diagnóstico que el médico o usuario desea realizar utiliza el siguiente bloque de código del método `specific_prognosis_frame()`:

```
def specific_prognosis_frame():
    app = App()
    app.geometry("720x240")
    brain_tumor_detection = tk.Button(
        app,
        text="Detección y clasificación de tumores cerebrales",
        command=lambda: open_file_choser(1),
    )
    tuberculosis_pneumonia_detection = tk.Button(
        app,
        text="Detección y clasificación de tuberculosis y neumonia",
        command=lambda: open_file_choser(2),
    )
    alzheimer_detection = tk.Button(
        app,
        text="Detección y clasificación de Alzheimer",
        command=lambda: open_file_choser(0),
    )
    brain_tumor_detection.place(x=220, y=50)
    tuberculosis_pneumonia_detection.place(x=200, y=100)
    alzheimer_detection.place(x=230, y=150)
    app.mainloop()
```

Ilustración 16: Código usado para desplegar la pantalla de selección de modelo de predicción a utilizar.

Donde el usuario seleccionando uno de los botones, manda como parámetro un número del 0 al 2 al método `open_file_choser()` y cual representa que modelo es el que debe utilizar siendo:

- 0 - Detección de Alzheimer, 1 - Detección de Tumores Cerebrales y 2 – Detección de enfermedades pulmonares.

El código del método mencionado anteriormente es el siguiente:

```
def open_file_choser(disease):
    filename = askopenfilename()
    app_aux = App()
    app_aux.geometry("720x250")
    frame = ttk.Frame(app_aux)
    label = ttk.Label(
        frame,
        text="Model loaded and sent to CUDA\nImage opened.\nTransforming...\nImage transformed",
    )
    label.pack(padx=5)
    frame.pack(padx=10, pady=50, expand=True, fill=tk.BOTH)
    if disease == 4:
        diagnosis_number = image_classification(filename)
        predict_class = disease_prediction(filename, diagnosis_number)
    else:
        predict_class = disease_prediction(filename, disease)
    button = tk.Button(
        app_aux,
        text="ANALIZAR IMAGEN CARGADA",
        command=lambda: diagnosis_frame(predict_class),
    )
    button.place(x=250, y=200)
    app_aux.mainloop()
    predict_class = ""
```

Ilustración 17: Código usado para la selección de la imagen a analizar.

Cada botón realiza el llamado a un método que identifica el modelo que se deberá utilizar con este snippet corto de código el cual recibe como parámetro la ruta en disco de la imagen y el número mencionado anteriormente:

```
def disease_prediction(filename, disease):
    predict_class = ""
    if disease == 1:
        predict_class = brain_tumor_detection(filename)
    elif disease == 2:
        predict_class = tuberculosis_detection(filename)
    elif disease == 0:
        predict_class = alzheimer_detection(filename)
    return predict_class
```

Ilustración 18: Bloque de código encargado de seleccionar el modelo de diagnóstico.

Como se mencionó anteriormente, los métodos que cargan el modelo seleccionado y envían la imagen al mismo para que este realice la clasificación y brinde su diagnóstico. Se reutilizó el código utilizado en el proceso de evaluación de cada red y modelo sin la iteración exhaustiva sobre todas las imágenes, haciendo solamente una predicción.

En el caso de que el usuario decida enviar directamente una imagen al programa, es necesario primero determinar para que tipo de diagnóstico es útil la imagen, enviando primero al método *open_file_chooser()* un 4, validación realizada mediante un bloque de código if el cual una vez se entra a él, manda a llamar al método *image_classification()* lo cual se realiza carga el modelo de clasificación previa de la siguiente manera:

```
def image_classification(image_route):
    images_transform = transforms.Compose(
        [
            transforms.ToPILImage(),
            transforms.Resize((224, 224)),
            transforms.ToTensor(),
            transforms.Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5]),
        ]
    )
    model = torch.load(
        "../codes/models/second-training-session/prediagnosis-model.pth",
        map_location="cuda",
    )
    red = prediagnosis_classifier(3).to("cuda")
    red.load_state_dict(model["model_state_dict"])
    images = cv2.imread(image_route)
    images = cv2.cvtColor(images, cv2.COLOR_BGR2RGB)
    image_to_predict = images_transform(images)
    image_to_predict = torch.unsqueeze(image_to_predict, 0)
    red.eval()
    with torch.no_grad():
        output = red(image_to_predict.to("cuda"))
        output_label = torch.topk(output, 1)
        class_number = int(output_label.indices)
    return class_number
```

Ilustración 19: Código encargado de cargar el modelo para clasificación previa de imágenes.

4.6 CRONOGRAMA DE ACTIVIDADES

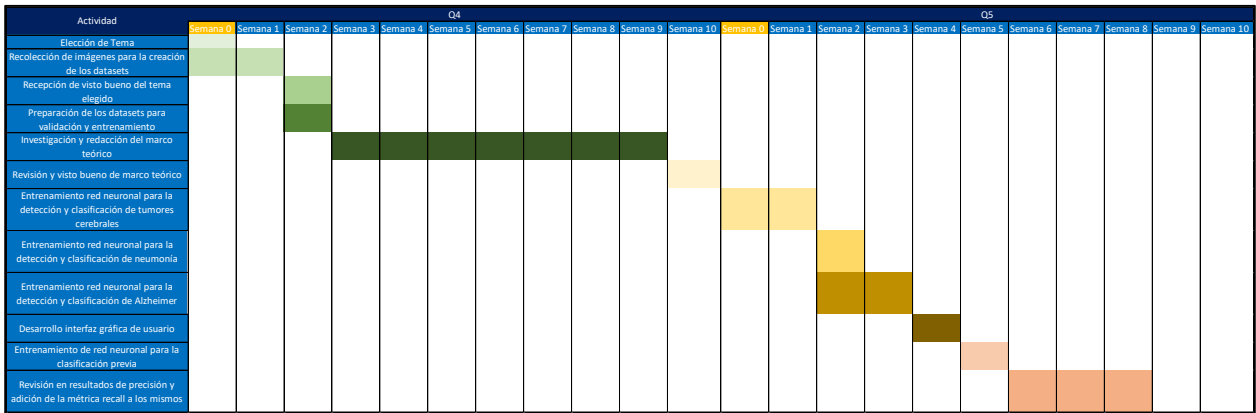


Ilustración 20: Diagrama con el cronograma de actividades.

4.6.1 BITÁCORA DE ACTIVIDADES

Actividad	Fecha Inicio	Fecha Fin	Duración
Elección de Tema	8/6/2023	12/6/2023	4 días
Recolección de imágenes para la creación de los datasets	15/6/2023	30/6/2023	15 días
Recepción de visto bueno del tema elegido	16/7/2023	23/7/2023	7 días
Preparación de los datasets para validación y entrenamiento	25/7/2023	28/7/2023	3 días
Investigación y redacción del marco teórico	1/8/2023	14/9/2023	45 días
Revisión y visto bueno de marco teórico	15/9/2023	22/9/2023	8 días
Entrenamiento red neuronal para la detección y clasificación de tumores cerebrales	30/8/2023	15/10/2023	15 días
Entrenamiento red neuronal para la detección y clasificación de neumonía	16/10/2023	18/10/2023	2 días
Entrenamiento red neuronal para la detección y clasificación de Alzheimer	19/10/2023	28/10/2023	9 días
Desarrollo interfaz gráfica de usuario	29/10/2023	3/11/2023	4 días
Entrenamiento de red neuronal para la clasificación de imágenes dependiendo el diagnóstico que se le de realizar	4/11/2023	8/11/2023	5 días
Revisión en resultados de precisión y adición de la métrica recall a los mismos y revisiones finales en reporte	10/11/2023	30/11/2023	20 días

Ilustración 21: Bitácora de Actividades.

V RESULTADOS Y ANÁLISIS

5.1 REDES NEURONALES

A continuación, serán mostrados los resultados obtenidos durante los entrenamientos de los modelos de las redes neuronales, enfocándose durante este proceso en los valores de exactitud (accuracy) y pérdida, siendo más importante este último, pues es el que media que tan bien estaba aprendiendo la red:

5.1.1 DETECCIÓN DE ALZHEIMER:

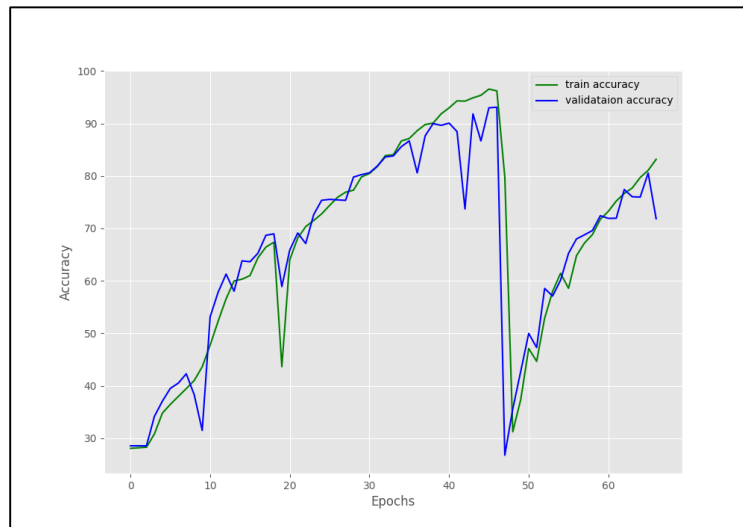


Ilustración 22: Evolución de la exactitud del modelo de detección de Alzheimer.

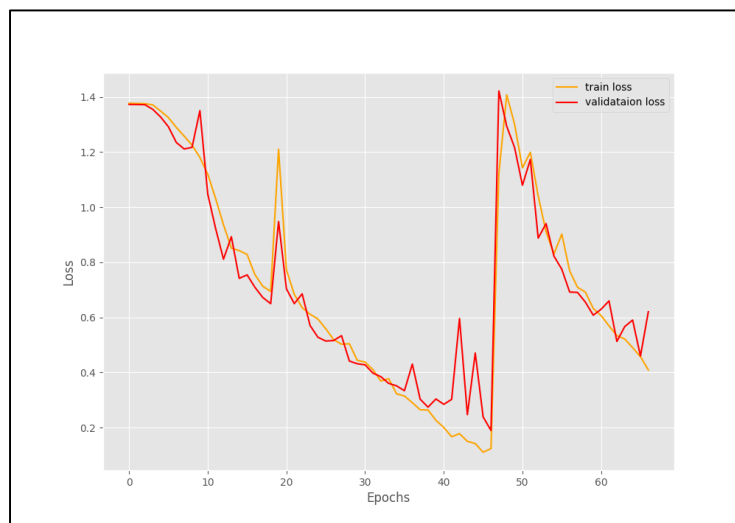


Ilustración 23: Evolución del valor de la función de pérdida del modelo de detección de Alzheimer.

Como se puede observar, el modelo logro superar el 95% de precisión tanto en el entrenamiento como en la validación, información que se ve reflejada en los valores devueltos por la función de pérdida, los cuales lograron alcanzar resultados menores a 0.2. Es notable mencionar que una vez se llega a estos valores, la red empezó a generar resultados variables siendo todos negativos, lo que indica el momento en que la red dejo de aprender nuevas características, siendo estos cambios negativos para el modelo, y es donde se ven reflejadas las estrategias de detención temprana, dejando guardado el último modelo con el mejor rendimiento.

Al momento de evaluar el rendimiento de la red, se le enviaron a la misma todas las imágenes propias del dataset de detección de Alzheimer tanto las de entrenamiento como las de validación y las imágenes descartadas para uno u otro set, para que la red las clasificara absolutamente todas, obteniendo las siguientes métricas junto a la matriz de confusión respectiva:

Model loaded and sent to CUDA
100%|

	precision	recall	f1-score	support
Mild	0.97	0.98	0.98	9856
Moderated	0.99	0.99	0.99	6528
Healthy	0.96	0.85	0.90	12800
Very Mild	0.84	0.94	0.89	11200
accuracy			0.93	40384
macro avg	0.94	0.94	0.94	40384
weighted avg	0.93	0.93	0.93	40384

Ilustración 24: Reporte de clasificación de modelo para detección y clasificación de Alzheimer.

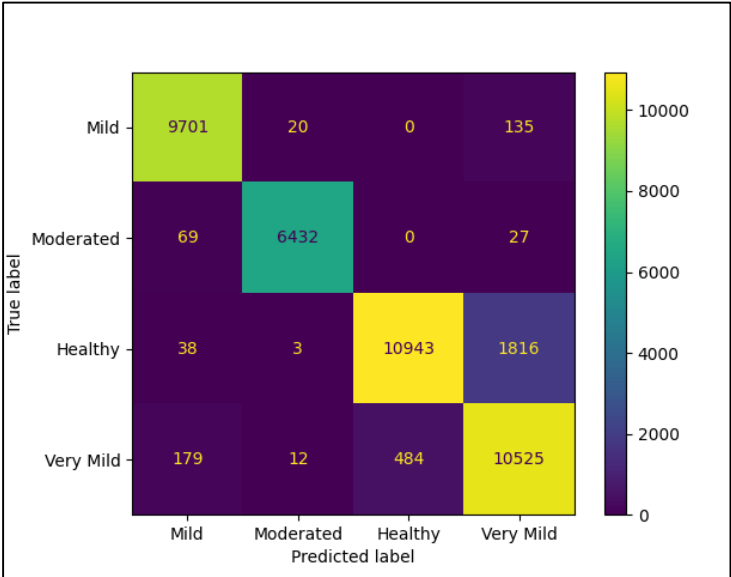


Ilustración 25: Matriz de confusión del modelo de detección y clasificación de Alzheimer.

De estos resultados es posible concluir que el modelo detecta y clasifica correctamente resonancias magnéticas de cerebros afectados con la enfermedad de Alzheimer con ciertas excepciones pues como se puede visualizar en el gráfico de la exactitud del modelo, el mismo no pudo llegar al 100%.

Un dato importante a tomar en cuenta con los falsos positivos encontrados, muy pocos fueron clasificados como cerebros enfermos cuando su etiqueta y clase real era "cerebro sano"; 484 de un total de 40,384 para ser exactos, una cifra que no representa ni siquiera el 1% del total de imágenes de muestra (en términos numéricos, el 0.012% aproximadamente). Por otro lado, otro número significativo en el proceso de validación es 1,816, una cifra que representa los falsos positivos de cerebros clasificados como enfermos cuando realmente eran sanos, un 0.045% del total de imágenes analizadas por la red. Teniendo en cuenta ambos resultados que son los más llamativos del proceso de evaluación de este modelo, si bien no es posible decir que el modelo es perfecto, se puede establecer que el modelo trabaja de manera efectiva para solucionar la problemática planteada puesto que en la práctica, un falso positivo de esta enfermedad conllevaría a la toma de más pruebas médicas por parte del paciente para confirmar o descartar completamente el diagnóstico, tomándose como una acción preventiva, en lugar de un caso donde a un paciente se le clasifico como cerebro sano cuando realmente podría tener etapas temprana de la afección.

Es notable mencionar el hecho que tanto la precisión como el recall y el f-1 score, las tres métricas tiene valores no menores a 0.80 (80%), teniendo números más cercanos a 1.0 que a 0.8 realmente, lo que implica una alta tasa de acierto y efectividad en las predicciones realizadas, dando así un modelo confiable, confirmando así solución a una de las problemáticas planteadas al comienzo del trabajo en cuestión.

5.1.2 DETECCIÓN Y CLASIFICACIÓN DE TUMORES CEREBRALES

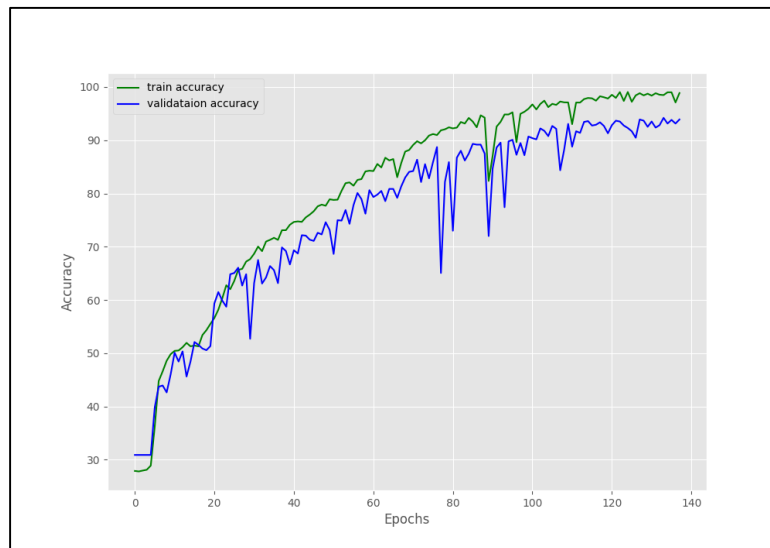


Ilustración 26: Evolución de la exactitud del modelo de detección y clasificación de tumores cerebrales.

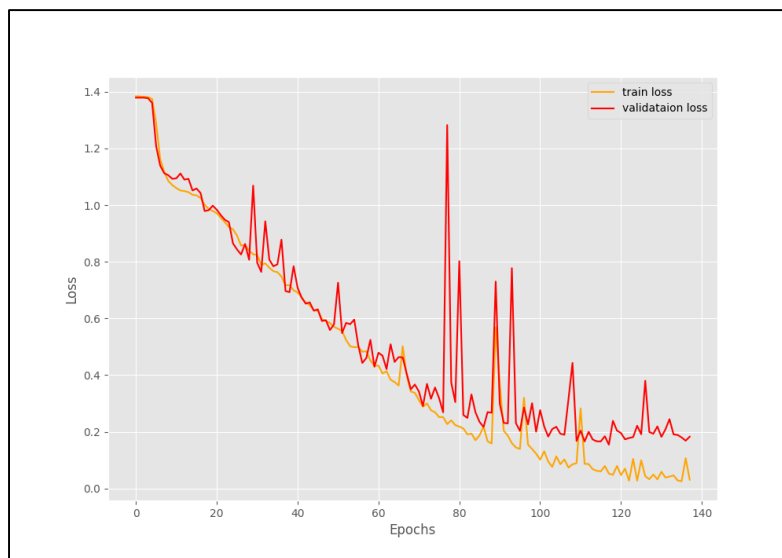


Ilustración 27: Evolución del valor de la función de pérdida del modelo de detección y clasificación de tumores cerebrales.

Durante el entrenamiento se puede apreciar una fluctuación en el valor devuelto por la función de pérdida, algo reflejado y afectando la exactitud en el proceso de validación. Esto en mayor parte es resultado de la similitud encontrada entre varias resonancias magnéticas al tener muchas una mancha que sobresale en la imagen siendo esta el tumor en la misma. Lo que clasifica si un tumor es glioma, meningioma o pituitario a grandes rasgos y a priori su posición y tamaño; una característica que al modelo le costaba trabajo entender y en ciertos momentos realizaba cambios

que no beneficiaban a una correcta predicción del tipo de tumor encontrado, y aunque si se logró obtener un modelo con una exactitud y precisión superior al 95%, este detalle ralentizó el proceso de aprendizaje del modelo en comparación a los tiempos de aprendizaje de otros modelos como podría ser el tiempo de aprendizaje del modelo para la detección y clasificación de Alzheimer, el cual tomo 47 épocas para llegar a su pico de exactitud y precisión, en contraste con las 118 épocas tomadas por el modelo usado para la detección y clasificación de tumores cerebrales. Cabe destacar que el proceso de early stopping o detención temprana no fue tan exagerado para este modelo.

Así como en el modelo anterior, se enviaron todas las imágenes recolectadas para tumores cerebrales para que el modelo brindara un diagnóstico para cada una dando como resultados el siguiente reporte de clasificación y matriz de confusión:

Model loaded and sent to CUDA
100%|

	precision	recall	f1-score	support
glioma	0.97	0.92	0.94	2547
meningioma	0.93	0.96	0.95	2582
no_tumor	0.97	0.98	0.98	2397
pituitary	0.97	0.99	0.98	2658
accuracy			0.96	10184
macro avg	0.96	0.96	0.96	10184
weighted avg	0.96	0.96	0.96	10184

Ilustración 28: Reporte de clasificación del modelo de detección y clasificación de tumores cerebrales.

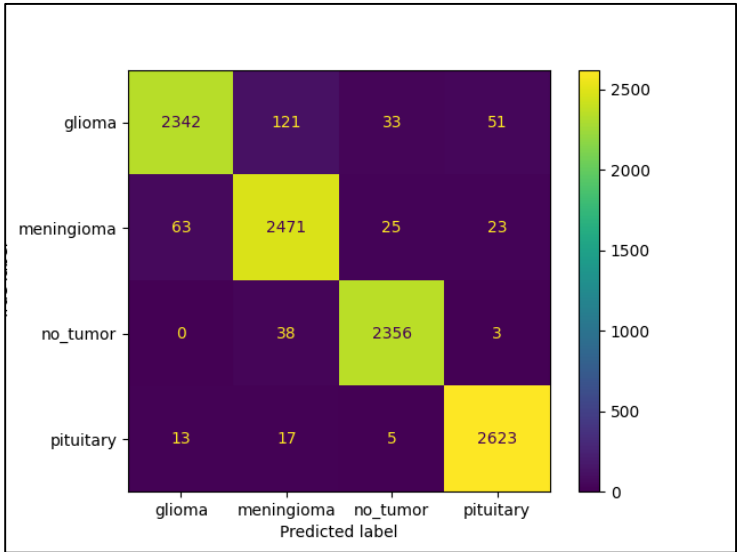


Ilustración 29: Matriz de confusión del modelo de detección y clasificación de tumores cerebrales.

Analizando los resultados obtenidos, es notable mencionar la baja cantidad de falsos positivos y negativos encontrados con este modelo, haciendo énfasis en la correcta predicción de imágenes sanas, dando como resultado un total de apenas 41 falsos negativos, siendo esto un 0.004%. De igual manera en la incorrecta predicción del tipo de tumor encontrado, la clase con mayor error son los tumores del tipo glioma con un total de 205 falsos positivos, una cifra que representa el 0.02% sobre el total de la muestra y dentro de la cual solamente se clasificaron 33 casos como enfermos cuando eran cerebros sanos (un 0.003%). También se puede añadir que solamente clasificó un total de 63 imágenes como falsos positivos de cerebros sanos, un 0.006%. En lo que respecta a las demás clases, los falsos negativos y positivos tampoco son cifras que superen siquiera el 0.5%.

Si bien es imperativo saber el tipo de tumor en específico afecta al paciente para poder brindar el mejor tratamiento posible, teniendo esta base, el médico puede realizar mejor su análisis y realizar pruebas más específicas, de modo que esto también representa una respuesta más rápida y efectiva para el tratamiento de la persona afectada por la enfermedad.

Así como en el modelo para la detección y clasificación del avance del Alzheimer, todas las métricas tienen valores no menores a 0.80 (80%), siendo en todo caso valores superiores a 0.90 (90%) lo que implica también una alta tasa de acierto y efectividad en las predicciones realizadas, dando así un modelo confiable, confirmando así solución a una de las problemáticas planteadas al comienzo del trabajo en cuestión.

5.1.3 DETECCIÓN Y CLASIFICACIÓN DE ENFERMEDADES PULMONARES

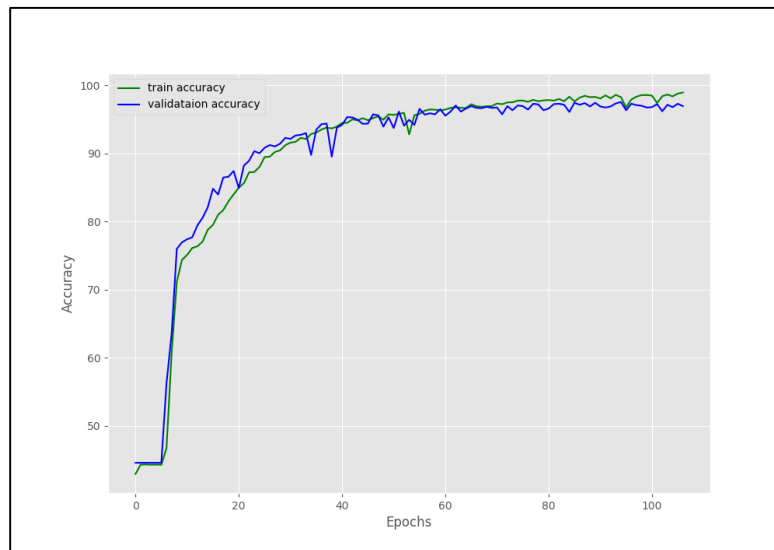


Ilustración 30: Evolución de la exactitud del modelo de detección y clasificación de enfermedades pulmonares.

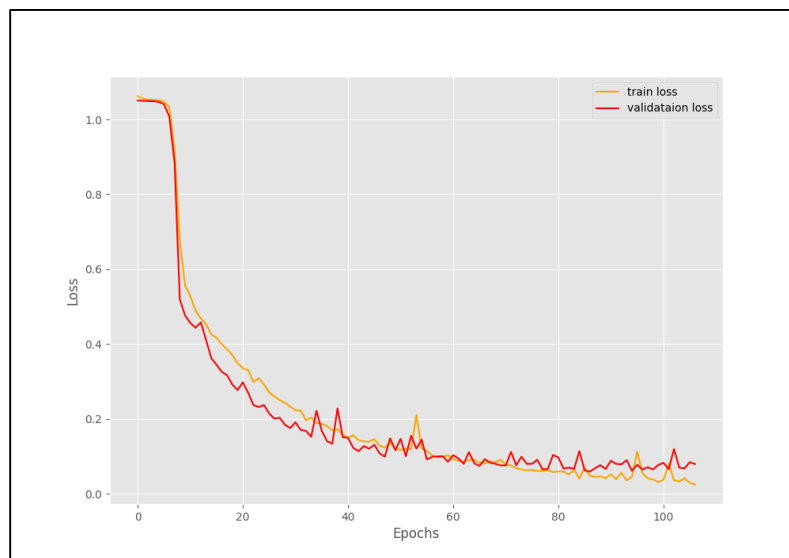


Ilustración 31: Evolución del valor de la función de pérdida del modelo de detección y clasificación de enfermedades pulmonares.

Este es el modelo que más tardó en aprender, llegando hasta las 348 épocas para obtener su mejor versión, pero que, a diferencia del modelo para la detección y clasificación de tumores cerebrales, no tuvo fluctuaciones tan violentas y agresivas, lo que implica un proceso de aprendizaje lento pero seguro. Este modelo aprendió a esta velocidad posiblemente a la cantidad de imágenes brindadas sumado a que, de todas las imágenes, en este set de datos era en el que más similitudes se podían encontrar entre ellas.

Siguiendo la metodología planteada y continuando la línea de trabajo utilizada previamente, se utilizaron por completo todas las imágenes recolectadas para la detección de enfermedades pulmonares y enviada a la red para que este brindara un diagnóstico para cada una, dejando el siguiente reporte de clasificación y matriz de confusión correspondiente:

Model loaded and sent to CUDA				
100%				
	precision	recall	f1-score	support
normal	0.98	0.51	0.67	9222
pneumonia	0.80	0.98	0.88	5856
tuberculosis	0.37	0.98	0.54	1894
accuracy			0.72	16972
macro avg	0.72	0.82	0.70	16972
weighted avg	0.85	0.72	0.73	16972

Ilustración 32: Reporte de clasificación para el modelo de detección y clasificación de enfermedades pulmonares

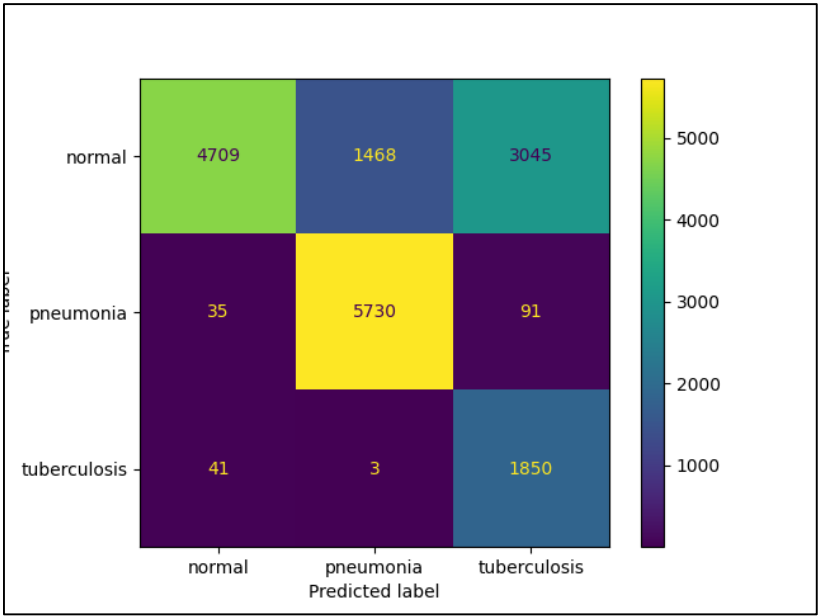


Ilustración 33: Matriz de confusión para el modelo de detección y clasificación de enfermedades pulmonares

Este modelo es el que más falsos negativos diagnosticó. El principal motivo del porque ocurre esto es debido a la calidad de las imágenes médicas usadas para el diagnóstico de enfermedades pulmonares y la anatomía tan distinta entre individuos. Profundizando en lo anterior, en lo que respecta a la calidad de los rayos x, estos son en su gran mayoría similares teniendo muchas manchas oscuras dentro de la imagen y esto nos lleva directamente al segundo punto, pues para un ojo experimentado de un neumólogo, es sencillo identificar que manchas dentro de la imagen

representan realmente una afección al pulmón provocada por estas enfermedades y cuáles no ya sea por un motivo ajeno a enfermedades pasadas como lo puede ser asma o bien ya la biología y genética del individuo. Otro motivo del porque se obtuvieron estos resultados es el desbalance de los datos. Caso contrario a los datasets de Alzheimer y tumores cerebrales, el set de datos de enfermedades pulmonares tenía más imágenes pertenecientes al pulmones sanos y neumonía que de tuberculosis y esto afecta negativamente el resultado final.

En este caso lo positivo a resaltar sobre los errores de la red, es que la misma no lanzó una cantidad exagerada de imágenes de pulmones enfermos como pulmones sanos, sino todo lo contrario, clasificó varias imágenes de pulmones sanos como enfermos lo cual llevaría al médico a realizar una revisión de segunda mano y comenzar acciones preventivas leves y muy poco invasivas sobre el paciente para descartar totalmente la afección y dejar al paciente en observación.

Esto nos permite afirmar que, si bien la red no logra cumplir completamente la problemática en su totalidad, sigue siendo una muy buena solución pues en temas de salubridad, se suele tener en cuenta que es mejor tomar acciones preventivas sobre un falso positivo que dejar a su suerte a un falso negativo que pueda extender alguna enfermedad o en el peor de los casos, empeorar su estado y fallecer.

5.1.4 CLASIFICACIÓN PREVIA DE IMÁGENES MÉDICAS

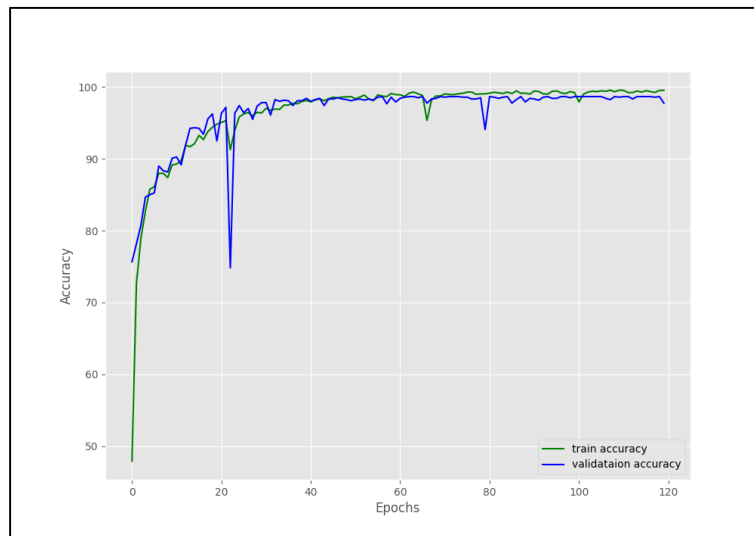


Ilustración 34: Evolución de la exactitud del modelo de clasificación previa de imágenes.

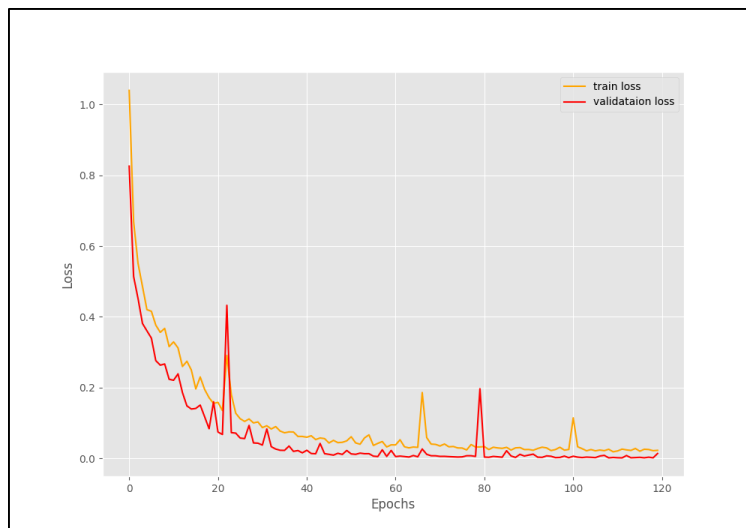


Ilustración 35: Evolución del valor de la función de pérdida del modelo de clasificación previa de imágenes.

En este modelo los resultados obtenidos fueron completamente satisfactorios, pues es en el que más cerca se logró de llegar a 0 con el valor devuelto por la función de pérdida. Existieron muy pocas fluctuaciones en el proceso de aprendizaje, las cuales eran esperadas pues el proceso de aprendizaje de una red neuronal convolucional desde cero, que no involucre aprendizaje heredado, suele tener estos altibajos ocasionales pues la red altera los pesos de cada neurona, llegando a existir combinaciones que provoquen lo mencionado anteriormente.

Acto seguido y continuando la línea de trabajo planteada, se siguió con el proceso de evaluación el cual brindó el siguiente reporte de clasificación y matriz de confusión respectiva:

Model loaded and sent to CUDA

100%|

	precision	recall	f1-score	support
alzheimer-diagnosis	1.00	0.99	1.00	40384
brain-tumor-clasification	0.98	0.99	0.99	10184
tuberculosis-pneumonia-diagnosis	0.99	1.00	0.99	16972
accuracy			0.99	67540
macro avg	0.99	0.99	0.99	67540
weighted avg	0.99	0.99	0.99	67540

Ilustración 36: Reporte de clasificación del modelo de clasificación previa de imágenes.

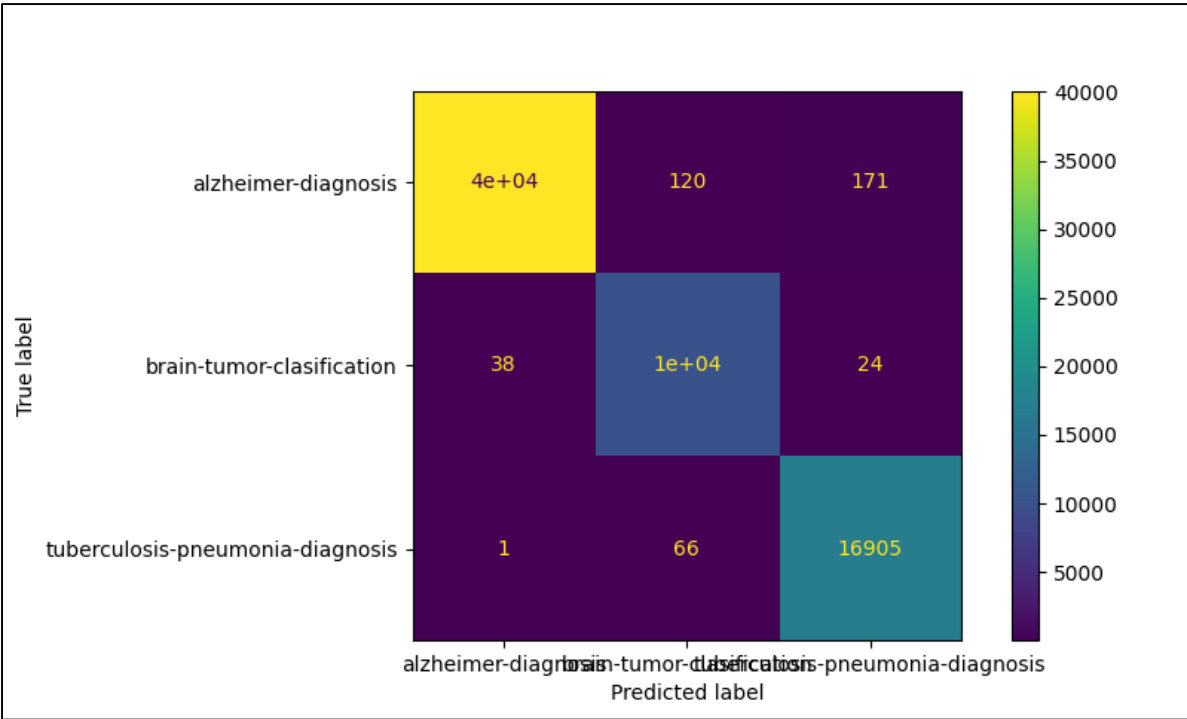


Ilustración 37: Matriz de confusión del modelo de clasificación previa de imágenes.

Este modelo logró obtener en todas sus métricas valores de 0.99 (99%) o 1.00 (100%) lo que indica un modelo casi perfecto, esto fue debido a que la red no debía aprender por completo cada característica de las imágenes; solo debía obtener los rasgos principales de las mismas y brindar la predicción del para que iba a ser utilizada.

5.2 INTERFAZ GRÁFICA DE USUARIO

A continuación, se mostrarán los resultados obtenidos del desarrollo de una interfaz gráfica de usuario para el uso fácil y amigable del aplicativo, enfocándose en el uso diario del usuario. Cabe mencionar que los diagnósticos brindados por la red oscilan en tiempos entre 1 y 3 segundos dependiendo del tamaño, calidad y tipo de imagen cargada, siendo registrado un tiempo de 3.4 segundo como el diagnóstico más lento.



Ilustración 38: Pantalla principal del programa.

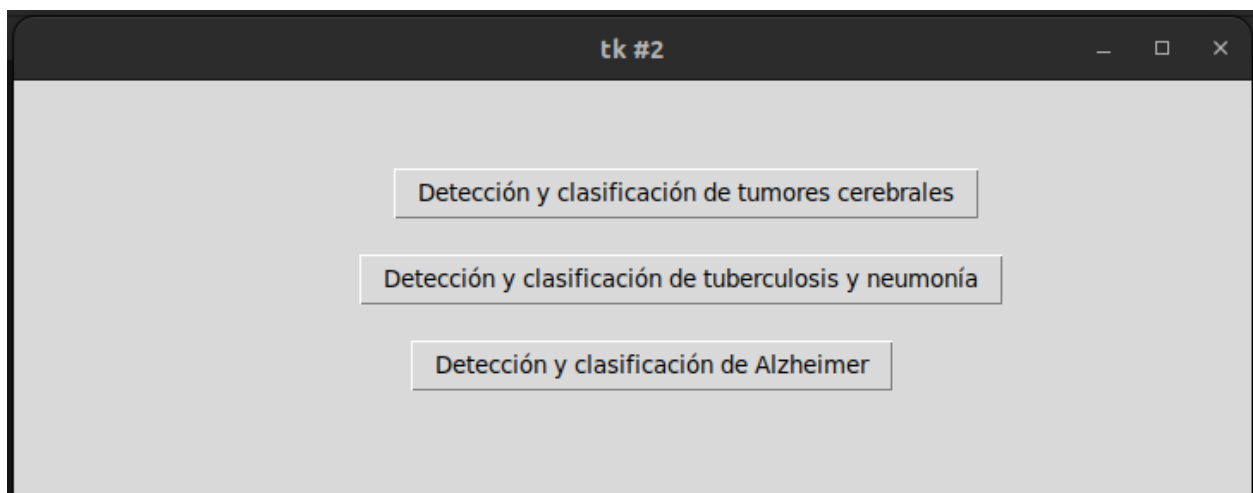


Ilustración 39: Pantalla de selección de un modelo de diagnóstico específico.

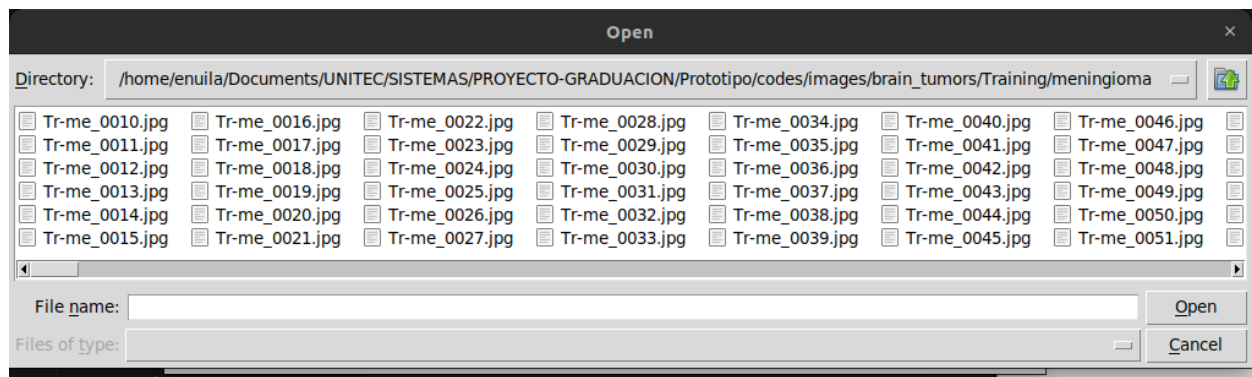


Ilustración 40: Pantalla de selección de imagen a cargar.

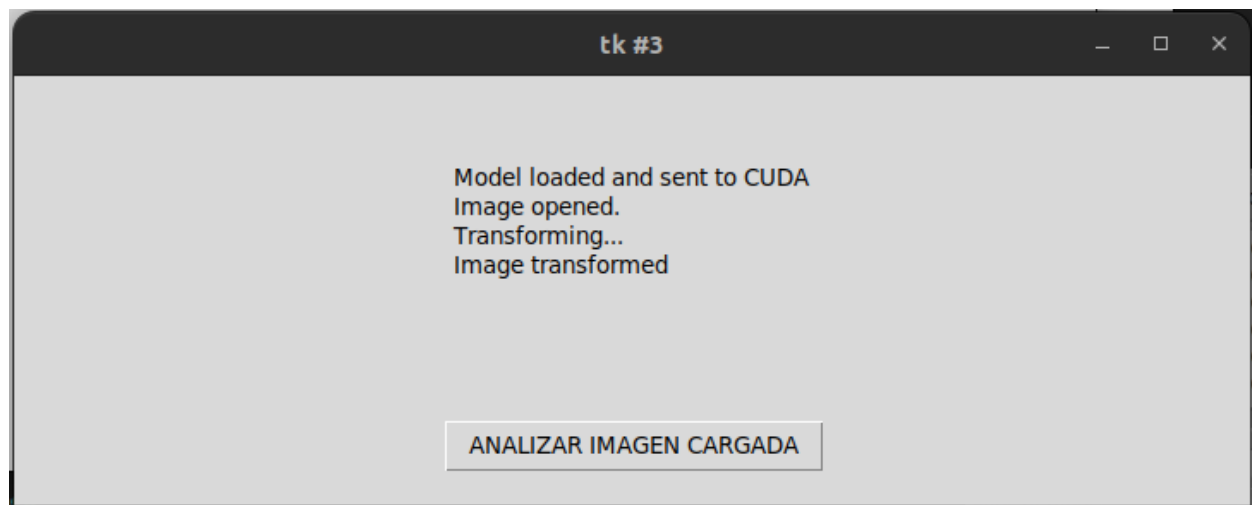


Ilustración 41: Retroalimentación dirigida al usuario de la imagen y el modelo cargado.

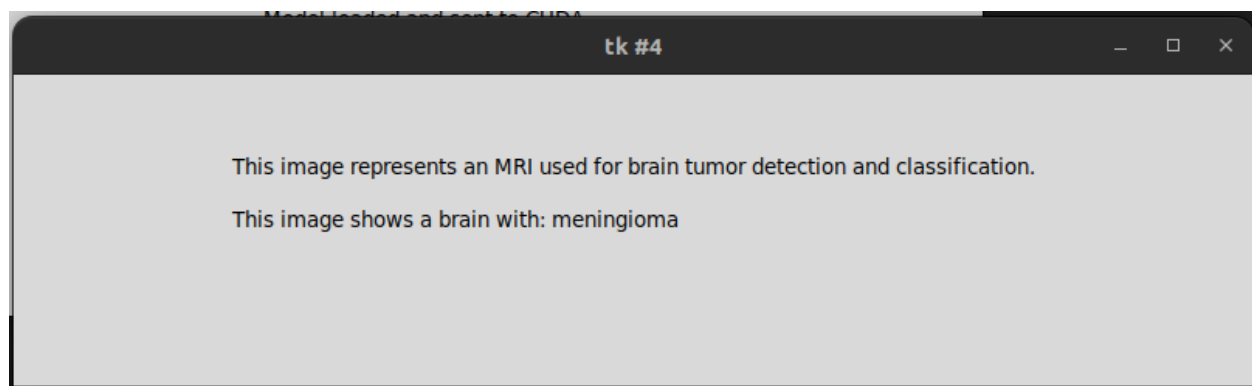


Ilustración 42: Ejemplo de diagnóstico final de imagen cargada.

VI CONCLUSIONES

- Se construyó un set de datos útil para el proceso de entrenamiento y validación de los 4 tipos de redes neuronales necesarias en este proyecto, el cual consta de un total de 67,500 imágenes en total.
- Se programó un pre procesador para las imágenes usando el módulo torch.transforms para la normalización y estandarización de todas las imágenes del set de datos construido.
- Se realizaron dos diseños efectivos de arquitectura de redes neuronales para implementar en la programación de las mismas, contando uno con cinco capas de convolución y tres lineales, diseño usado para los modelos de diagnósticos y el segundo diseño contando con tres capas convolucionales y dos lineales, siendo usado para la clasificación previa de imágenes.
- Se desarrollaron tres redes neuronales convolucionales funcionales que detectan y clasifican los distintos diagnósticos posibles de las enfermedades planteadas originalmente y una red neuronal convolucional funcional que clasifica estas imágenes previamente dependiendo del tipo de diagnóstico que debe realizar.
- Se diseñó y programó una interfaz gráfica de usuario amigable y fácil de usar en la vida cotidiana de usuarios del aplicativo.

VII RECOMENDACIONES

- Para una construcción de un set de datos útil para entrenamientos de este tipo, es necesario tener un mínimo de 65,000 imágenes de muy buena calidad separadas en una proporción idealmente de 80/20 (80 – entrenamiento, 20 – validación). Es posible usar estrategias de data augmentation, pero estas deben de ser aplicada con cautela, pues con un dato que se configure mal, estas estrategias pueden simular cantidades de hasta cinco veces más lo requerido.
- Un buen valor para el learning rate podría estar entre 0.015 y 0.025, siendo 0.02 un buen valor sobre el cual comenzar para luego realizar ajustes sobre este mismo dependiendo del rendimiento y aprendizaje de la red.
- Es recomendable utilizar una tarjeta gráfica dedicada para el proceso de entrenamiento y validación, El proceso del diagnóstico de una sola imagen es posible realizarlo desde el propio CPU del computador pues no se necesitan tantas operaciones matemáticas simultáneas como en el proceso de entrenamiento y/o validación de una red.

VIII BIBLIOGRAFÍA

1. Acosta, D., López Contreras, R., Garita, N. R., Mora, J., Jimenez, X., & Chinchilla, D. (2013). *Primer Consenso Centroamericano Multidisciplinario de Enfermedad y Alzheimer y Demencias Asociadas*. Granada, Nicaragua: Febrero.
2. *Análisis Descriptivo, Predictivo y Prescriptivo de datos - IArtificial.net*. (21 de febrero de 2019). Obtenido de IArtificial.net: <https://www.iartificial.net/analisis-predictivo-y-prescriptivo-con-machine-learning/>
3. Artola Moreno, Á. (2019). Clasificación de imágenes usando redes neuronales convolucionales en Python. Sevilla, España.
4. Banco Central de Honduras. (2021). *Honduras en Cifras 2018-2020*. censo, Tegucigalpa, Honduras.
5. Bermejo, M. C., Clavera, I., Michel de la Rosa, F. J., & Marín, B. (2007). Epidemiología de la tuberculosis. *Anales del Sistema Sanitario de Navarra*, 7-19.
6. Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.
7. Brownlee, J. (21 de Octubre de 2019). *A Gentle Introduction to Cross-Entropy for Machine Learning*. Obtenido de Machine Learning Mastery: <https://machinelearningmastery.com/cross-entropy-for-machine-learning/>
8. Cáceres, M., & Marín, M. (2019). Efectividad de la terapia ocupacional en el deterioro cognitivo del adulto mayor con Alzheimer. Lima, Perú: Universidad Norbert-Wiener.
9. Cazana Cachís, S. (2017). El concepto de inteligencia artificial según Allen Newell. *Educare Et Comunicare*.
10. Center for Disease Control And Prevention. (30 de septiembre de 2022). *Pneumonia*. Obtenido de CDC 24/7; Saving Lives, Protecting People: <https://www.cdc.gov/pneumonia/index.html>

11. Center for Disease Control And Prevention. (22 de marzo de 2023). *Tuberculosis (TB)*. Obtenido de CDC 24/7; Saving Life, Protecting People: <https://www.cdc.gov/tb/default.htm>
12. *Colegio Médico de Honduras*. (6 de febrero de 2018). Obtenido de Censo 2017: <https://www.colegiomedico.hn>
13. Comparación del Desempeño de Funciones de Activación en Redes Feedforward para aproximar Funciones de Datos con y sin Ruido. (2007). *Avances en Sistemas e Informática*, 81-88.
14. Cullity, B. D. (1956). *Elements Of X Ray Diffraction*. Addison-Wesley Publishing Company, Inc.
15. Dávila, K. (Noviembre de 2022). Computer Vision. *Clase Magistral Sistemas Inteligentes*. Tegucigalpa, Francisco Morazán, Honduras.
16. Dávila, K. (2022). Redes Neuronales y Algoritmos de Aprendizaje de Máquina. *Clase magistral Sistemas Inteligentes*. Tegucigalpa.
17. Donoso, A. (2003). La Enfermedad de Alzheimer. *Revista Chilena de neuro-psiquiatría*, 13-22.
18. Fausett, L. (1994). *Fundamentals Of Neural Networks*. Upper Saddle River, New Jersey: Prentice-Hall, Inc.,.
19. García Serrano, A. (2016). *Inteligencia Artificial. Fundamentos, prácticas y aplicaciones*. CDMX: RC Libros.
20. Gómez-Vega, J. C., Ocampo Navia, M. I., & Lee, O. F. (2019). *Epidemiología y caracterización general de los tumores cerebrales primarios en el adulto*.
21. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Cambridge, Massachusetts, USA: The MIT Press.
22. Greenspan, H., van Ginneken, B., & Summers, R. (2016). *Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique*. IEEE.

23. Hara, K., & Nakayama, K. (1995). Comparison of activation functions in multilayer neural network for pattern classification. *Kanazawa University Repository for Academic Resources*, 133-138.
24. Hodsom, T. O., Over, T. M., & Foks, S. S. (2021). Mean Squared Error Deconstructed. *Journal of Advances in Modeling Earth Systems JAMES*. doi:2021MS002681
25. Li, B., & Lu, P. (1 de Junio de 2023). *Regresión de Poisson*. Obtenido de Microsoft: <https://learn.microsoft.com/es-es/azure/machine-learning/component-reference/poisson-regression?view=azureml-api-2>
26. López Marín, L., Díaz Otero, F., Vallecillo Maza, A. J., Esquível Solís, H., & Gutierrez Pabello, J. Á. (2006). *Tuberculosis Humana y bovina en Latinoamérica: De estudios sobre virulencia hacia herramientas para su control*. CDMX: Revista Latinoamericana de Microbiología.
27. López Takeyas, B. (2007). Introducción a la Inteligencia Artificial. Nuevo Laredo, Tamaulipas, México.
28. Mahesh, B. (2018). Machine Learning Algorithms - A Review. *International Journal of Science and Research*, 381-386. doi:10.21275/ART20203995
29. Mayo Clinic. (4 de Mayo de 2022). *Meningioma*. Obtenido de Mayo Clinic: <https://www.mayoclinic.org/diseases-conditions/meningioma/symptoms-causes/syc-20355643>
30. Mayo Clinic. (2 de febrero de 2023). *Alzheimer's Disease*. Obtenido de Mayo Clinic: <https://www.mayoclinic.org/diseases-conditions/alzheimers-disease/symptoms-causes/syc-20350447>
31. Mayo Clinic. (21 de Marzo de 2023). *Gliomas*. Obtenido de Mayo Clinic: <https://www.mayoclinic.org/es/diseases-conditions/glioma/symptoms-causes/syc-20350251>
32. Mayo Clinic. (21 de abril de 2023). *Mayo Clinic*. Obtenido de Tumor Cerebral: <https://www.mayoclinic.org/es/diseases-conditions/brain-tumor/symptoms-causes/syc->

20350084#:~:text=Los%20tumores%20cerebrales%20pueden%20comenzar,conocen%20como%20tumores%20cerebrales%20metast%C3%A1sicos.

33. Mayo Clinic. (8 de Febrero de 2023). *Tumores Pituitarios*. Obtenido de Mayo Clinic: <https://www.mayoclinic.org/es/diseases-conditions/pituitary-tumors/symptoms-causes/syc-20350548>
34. MedlinePlus. (7 de Mayo de 2022). *Resonancia Magnética*. Obtenido de MedlinePlus: <https://medlineplus.gov/spanish/ency/article/003335.htm#:~:text=Es%20un%20examen%20imagenol%C3%B3gico%20que,o%20imprimir%20en%20una%20pel%C3%ADcula>.
35. Michalski, R. S. (1983). 4 - A THEORY AND METHODOLOGY OF INDUCTIVE LEARNING. En R. S. Michalski, *Machine Learning* (págs. 83-134). Morgan Kaufmann. doi:10.1016/B978-0-08-051054-5.50008-X
36. National Heart, Lung, and Blood Institute. (24 de Marzo de 2022). *What is Pneumonia?* Obtenido de NHLBI: <https://www.nhlbi.nih.gov/health/pneumonia#:~:text=Pneumonia%20is%20an%20infection%20that,or%20fungi%20may%20cause%20pneumonia>.
37. National Institute of Biomedical Imaging and Bioengineering. (Junio de 2022). *Rayos X*. Obtenido de National Institute of Biomedical Imaging and Bioengineering, Engineering the Future of Health: <https://www.nibib.nih.gov/espanol/temas-cientificos/rayos-x>
38. Nilsson, N. J. (2001). *Inteligencia Artificial: Una nueva síntesis*. (Primera ed., Vol. 1). (C. F. Madrid, Ed., R. Marín Morales, J. T. Palma Méndez, & E. Paniagua Aris, Trans.) Madrid, España: McGraw Hill. Recuperado el Agosto de 2023
39. Ponce, J. C., Torres, A., Quezada, F. S., Silva, A., Martínez, E. U., Casali, A., . . . Pedreño, O. (2014). *Inteligencia Artificial*. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIn).
40. Russel, R. (2018). *Redes Neuronales: Guia Sencilla de Redes Neuronales Artificiales*. CreateSpace Independent Publishing Platform.
41. Russel, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach* (Tercera ed.). New Jersey, USA: Pearson Education, Inc.

42. Sánchez Gatica, J. G. (2002). *CARACTERISTICAS EPIDEMIOLOGICAS, CLINICAS Y TOMOGRAFICAS DE TUMORES CEREBRALES*.
43. Sandoval, L. J. (2018). ALGORITMOS DE APRENDIZAJE AUTOMÁTICO PARA ANÁLISIS Y PREDICCIÓN DE DATOS. *ITCA-FEPADE*, 36-40.
44. Sell, E. (2007). Resonancia Magnética Funcional. *Medicina*, 661-664.
45. Sharma, V., Rai, S., & Dev, A. (2012). A Comprehensive Study of Artificial Neural Networks . *International Journal of Advanced Research in Computer Science and Software Engineering* , 278-284.
46. Stockman, G., & Shapiro, L. G. (2001). *Computer Vision*. Prentice Hall PTR.
47. Suzuki, K. (2011). *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. Tokyo, Japon: InTech.
48. Togami, M., Masuyama, Y., Komatsu, T., & Nakagome, Y. (2020). Unsupervised Training for Deep Speech Source Separation with Kullback-Leibler Divergence Based Probabilistic Loss Function. *IEEE Xplore*, 56-60.
49. Valencia, I. F., Valdovinos Rosas, R. M., Marcial Romero, J. R., & Eleuterio, R. A. (2022). *Artificial Intelligence and Board Games: From Turkish to AlphaZero*. ReCIBE.
50. Varma, R. (9 de Enero de 2018). Picking Loss Functions - A comparison between MSE, Cross Entropy, and Hinge Loss. San Jose, California, USA. Obtenido de <https://rohanvarma.me/Loss-Functions/>
51. World Health Organization. (15 de Marzo de 2023). *Dementia*. Obtenido de World Health Organization: <https://www.who.int/news-room/fact-sheets/detail/dementia#:~:text=Alzheimer%20disease%20is%20the%20most,dependenc y%20among%20older%20people%20globally>.
52. World Health Organization. (21 de abril de 2023). *Tuberculosis*. Obtenido de World Health Organization: <https://www.who.int/news-room/fact-sheets/detail/tuberculosis>

53. Zaragoza-Velasco, K., & Fernández-Tapia, S. (2013). Ligamentos y tendones del tobillo: Anatomía y afecciones más frecuentes analizadas mediante resonancia magnética. *Anales de Radiología México*, 81-94.