



**MODELO DE RED NEURONAL CONVOLUCIONAL PARA EL DIAGNÓSTICO
DE NEUMONÍA EN IMÁGENES RADIOLÓGICAS**

PRESENTADO POR:

**NAAR PEREZ AGUSTIN
BARRETO MARTINEZ FABIAN**

**UNIVERSIDAD DEL SINÚ ELÍAS BECHARÁ ZAINÚM SECCIONAL
CARTAGENA**

ESCUELA DE INGENIERÍA DE SISTEMAS

CARTAGENA-COLOMBIA

MAYO 2019



UNIVERSIDAD DEL SINÚ
Elías Bechara Zainúm
Seccional Cartagena

**MODELO DE RED NEURONAL CONVOLUCIONAL PARA EL DIAGNÓSTICO
DE NEUMONÍA EN IMÁGENES RADIOLÓGICAS**

**Trabajo de grado presentado como requisito para optar el título de
INGENIERO DE SISTEMAS**

Asesor:

EUGENIA ARRIETA RODRÍGUEZ

**UNIVERSIDAD DEL SINÚ ELÍAS BECHARÁ ZAINÚM SECCIONAL
CARTAGENA**

ESCUELA DE INGENIERÍA DE SISTEMAS

CARTAGENA-COLOMBIA

MAYO 2019

NOTA DE ACEPTACIÓN DE TRABAJO DE GRADO

NOTA: _____

NOMBRE COMPLETO CALIFICADOR

NOMBRE COMPLETO CALIFICADOR

NOMBRE COMPLETO CALIFICADOR

DIRECTOR DEL PROYECTO

ASESOR METODOLOGICO

DIRECTOR DEL PROGRAMA

Lugar, Fecha, _____

**EL DIRECTOR DE INVESTIGACIONES DE LA UNIVERSIDAD DEL SINU
“ELIAS BECHARA ZAINUM” SECCIONAL CARTAGENA**

HACE CONSTAR QUE:

En Cartagena, a los 13 días del mes de Junio del 2019, en la Oficina de la Dirección de Investigaciones de la Universidad, se aprobó por el jurado y se realizó la sustentación del Trabajo de Grado titulado **“MODELO DE RED NEURONAL CONVOLUCIONAL PARA EL DIAGNÓSTICO DE NEUMONÍA** “que se desarrolló bajo la dirección del Ingeniero(a) **EUGENIA ARRIETA RODRÍGUEZ** y presentado por los egresados **AGUSTIN NAAR PEREZ** y **FABIAN BARRETO MARTINEZ**.

Los jurados designados fueron los ingenieros **LUIS FERNANDO MURILLO FERNANDEZ** y **MARIA CLAUDIA BONFANTE**.

Teniendo en cuenta la aprobación emitida, se encuentra que los egresados han cumplido con los requisitos de presentación y sustentación del trabajo de investigación, exigidos por el programa de **INGENIERÍA DE SISTEMAS**, Resolución 0178 de 15 de marzo de 2010.

Se expide esta constancia a los 13 días del mes de junio de 2019

DIRECCIÓN DE INVESTIGACIONES
Universidad del Sinú

COORDINADOR DE INVESTIGACIONES
Escuela de Ingeniería de Sistemas

Agradecimientos

Agradezco a Dios por brindarme esta oportunidad y la fortaleza para lograr esta meta. A mi familia por su apoyo incondicional.

A mi asesora de tesis Eugenia Arrieta Rodríguez por su constancia y aporte de conocimientos de acuerdo a su amplia experiencia investigativa. A Fernando López Por su disposición y colaboración como experto en aprendizaje automático en el Área de la salud. Agradecimiento especial a Yerliz Alzate Naar por su apoyo incondicional y valiosos aportes de conocimiento sobre la neumonía.

Agustín Naar.

Agradecimientos

El presente trabajo de grado va dedicado a Dios, quien como guía estuvo presente en el caminar de mi vida, bendiciéndome y dándome fuerzas para continuar con mis metas trazadas sin desfallecer. A mi hija Antonella que ha sido motivo de superación, a mis profesores que con apoyo incondicional, y confianza permitieron que logre culminar mi carrera profesional, a Fernando López y Eugenia Arrieta gracias por su apoyo y paciencia.

Fabián Barreto.

Dedicatoria

El presente trabajo investigativo lo dedicamos principalmente a Dios, por ser el inspirador y darnos fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A nuestros padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes hemos logrado llegar hasta aquí y convertirnos en lo que somos. Ha sido el orgullo y el privilegio de ser sus hijos, son los mejores padres.

A nuestros hermanas (os) por estar siempre presentes, acompañándonos y por el apoyo moral, que nos brindaron a lo largo de esta etapa de nuestras vidas.

A todas las personas que nos han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que nos abrieron las puertas y compartieron sus conocimientos.

Fabián Barreto Martínez

Agustín Naar Pérez

TABLA DE CONTENIDO

RESUMEN	9
INTRODUCCIÓN	10
I DISEÑO METODOLÓGICO	13
1.1 Planteamiento del problema	13
1.2 Descripción del problema	13
1.3 Justificación	15
1.4 Formulación del problema	16
2. Alcance	16
3. Objetivos	17
3.1 Objetivo general:	17
3.2 Objetivos específicos:	17
4. Estado del Arte	18
4.1 CheXNet – UNIVERSIDAD DE STANFORD	18
4.2 Avances de la inteligencia artificial (IA) en el diagnóstico de posibles casos de cáncer de próstata – UNIVERSIDAD DE MURCIA	19
3.3 Inception v3 – GOOGLE	20
4.4 Inteligencia artificial y técnica de aprendizaje automático para detectar rápidamente enfermedades oculares y neumonía.	22
5. Marco de Referencia	24
5.1 Marco Teórico	24
5.1.1 Neumonía	24
5.1.2 Machine Learning	25
5.1.3 Deep Learning	25
5.1.4 Redes Neuronales Artificiales	26
5.1.5 Convolución	27
5.1.6 Capa de activación ReLu.	27
5.1.7 Capa de reducción o pooling	27
5.1.8 Algoritmo Backpropagation	28
5.1.9 Gradiente Descendente	28
5.1.10 Redes Neuronales Convolucionales	29
6. Marco Conceptual	30
6.1 Inteligencia Artificial	30

6.2	Microsoft Azure	30
6.3	Máquina virtual de Azure Data Science (DSVM)	31
6.4	Anaconda Navigator	31
6.5	Jupyter Notebook	31
6.6	Python	32
6.7	Tensorflow	32
6.8	Imágenes DICOM	33
6.9	Dataset	33
6.10	Otras referencias	34
7.	Marco Legal	34
7.1	Leyes aplicables	35
7.2	Aspectos Éticos	35
8.	Metodología	35
8.1	Muestra y Población	40
8.2	Línea de Investigación	40
8.3	Tipo de Investigación	40
II	CONSTRUCCIÓN DEL CONJUNTO DE DATOS	42
2.1	Adquisición y comprensión de datos	43
2.2	Extraer datos de la imagen (Extraer variables independientes)	43
2.3	Algoritmo para la extracción de variables independientes desde un notebook	45
2.4	Extracción de variables dependientes	51
2.5	Algoritmo para la extracción de variables dependientes desde un notebook	52
III	MODELADO DE LA RED NEURONAL CONVOLUCIONAL	55
3.1	Cargue de los datos	55
3.2	Normalización de los datos	55
3.3	Segmentación de los Datos	57
3.4	Construcción de CNN (Capas de Convolución)	59
3.4	Validación del Modelo	76
3.5	Pruebas del modelo	76
IV	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	78
4.1	Resultados	78
4.2	Conclusiones	79
4.2	Recomendaciones	80

TABLA ILUSTRACIONES

ILUSTRACIÓN 1 ÁRBOL PROBLEMA. FUENTE PROPIA.	13
ILUSTRACIÓN 2 METODOLOGÍA UTILIZADA. FUENTE (HTTPS://DOCS.MICROSOFT.COM/EN-US/AZURE/MACHINE-LEARNING/TEAM-DATA-SCIENCE-PROCESS/OVERVIEW)	36
ILUSTRACIÓN 3 METODOLOGÍA APLICADA. FUENTE PROPIA.	39
ILUSTRACIÓN 4 RESUMEN DEL ALGORITMO DE MACHINE LEARNING. FUENTE PROPIA.	42
ILUSTRACIÓN 5 REPRESENTACIÓN ILUSTRADA DEL ALGORITMO DE PREDICCIÓN DE NEUMONÍA USANDO CNN. FUENTE PROPIA.	43
ILUSTRACIÓN 6 MATRIZ QUE ALMACENA LAS IMÁGENES EN COLUMNAS. FUENTE PROPIA.	44
ILUSTRACIÓN 7 VISUALIZACIÓN CON MATPLOTLIB DE UNA MATRIZ DE PÍXELES OBTENIDOS DE UNA IMAGEN. FUENTE PROPIA	45
ILUSTRACIÓN 8 COLUMNAS DEL ARCHIVO LABELS_UNICOS.CSV. FUENTE PROPIA.	45
ILUSTRACIÓN 9 LIBRERÍAS PARA "CREATE_DATASET ()". FUENTE PROPIA.	46
ILUSTRACIÓN 10 VARIABLE TIPO NUMPY PARA DATOS FINALES DEL PRE PROCESAMIENTO. FUENTE PROPIA.	47
ILUSTRACIÓN 11 CICLO PARA LA LECTURA DE LAS IMÁGENES. FUENTE PROPIA.	47
ILUSTRACIÓN 12 "RESIZE" DE LAS IMÁGENES. FUENTE PROPIA.	47
ILUSTRACIÓN 13 DIMENSIÓN DE UNA IMAGEN DE ENTRENAMIENTO (256x256). FUENTE PROPIA.	47
ILUSTRACIÓN 14 MOLDEO DE MATRIZ A COLUMNAS. FUENTE PROPIA.	48
ILUSTRACIÓN 15 IMAGEN EN FORMA DE COLUMNA. FUENTE PROPIA.	48
ILUSTRACIÓN 16 FUNCIÓN PARA AGREGAR IMÁGENES TRANSFORMADAS A UN ARREGLO. FUENTE PROPIA.	49
ILUSTRACIÓN 17 CONSTRUCCIÓN DE ARREGLO DE IMÁGENES APLANADAS. FUENTE PROPIA.	49
ILUSTRACIÓN 18 EJEMPLO MATRIZ DE IMÁGENES APLANADAS	50
ILUSTRACIÓN 19 INSTRUCCIÓN PARA CREAR LA MATRIZ DADA POR FILAS DE IMÁGENES. FUENTE PROPIA.	50
ILUSTRACIÓN 20 TRANSPOSICIÓN DE IMÁGENES DADA POR FILAS. FUENTE PROPIA.	50
ILUSTRACIÓN 21 MATRIZ DE IMÁGENES DADAS POR COLUMNAS. FUENTE PROPIA.	50
ILUSTRACIÓN 22 MATRIZ DE DATOS. FUENTE PROPIA.	51
ILUSTRACIÓN 23 PERSISTENCIA DE LOS DATOS. FUENTE PROPIA.	51
ILUSTRACIÓN 24 LIBRERÍAS PARA CREACIÓN DE DATA SET DE LABELS. FUENTE PROPIA.	52
ILUSTRACIÓN 25 LECTURA QUE ARCHIVO LABELS_UNICOS.CSV. FUENTE PROPIA.	53
ILUSTRACIÓN 26 EXTRACCIÓN DE LABEL ASOCIADO A UNA IMAGEN. FUENTE PROPIA.	53
ILUSTRACIÓN 27 PERSISTENCIA DE LABELS. FUENTE PROPIA.	53
ILUSTRACIÓN 28 EJEMPLO DATOS SIN NORMALIZAR. FUENTE PROPIA.	56
ILUSTRACIÓN 29 EJEMPLO LABELS EN ONE HOT. FUENTE PROPIA	56
ILUSTRACIÓN 30 EJEMPLO DATOS NORMALIZADOS. FUENTE PROPIA	56
ILUSTRACIÓN 31 REPRESENTACIÓN DEL SPLIT DE DATOS. FUENTE PROPIA.	57
ILUSTRACIÓN 32 EJEMPLO DE DATOS AFECTADOS CON UNDERFITTING Y OVERFITTING. FUENTE (HTTP://INTROTODEEPLARNING.COM/)	57
ILUSTRACIÓN 33 SPLIT DE CONJUNTO DE DATOS. FUENTE PROPIA.	58
ILUSTRACIÓN 34 DIAGRAMA DE LA ESTRUCTURA DE LAS CAPAS DEL MODELO. FUENTE PROPIA	59
ILUSTRACIÓN 35 GRAFICO DE ESTRUCTURA DE LAS CAPAS DEL MODELO. FUENTE PROPIA	60
ILUSTRACIÓN 36 EJEMPLO DE CADA TIPO DE CAPA. FUENTE (HTTP://INTROTODEEPLARNING.COM/).	61
ILUSTRACIÓN 37 TAMAÑO DE LA "VENTANA DESLIZANTE". FUENTE PROPIA	62
ILUSTRACIÓN 38 EJEMPLO DE UN MAXPOOLING CON UN FILTRO DE 2X2. FUENTE (HTTP://INTROTODEEPLARNING.COM)	63
ILUSTRACIÓN 39 GRAFICA DE LA FUNCIÓN RELU. FUENTE PROPIA	64
ILUSTRACIÓN 40 EJEMPLO DE DROPOUT. FUENTE (HTTP://INTROTODEEPLARNING.COM).	65
ILUSTRACIÓN 41 VISUALIZACIÓN DE CÓMO TRABAJA LEARNING RATE.	66

ILUSTRACIÓN 42 FÓRMULA MATEMÁTICA DE TF.NN.SOFTMAX_CROSS_ENTROPY_WITH_LOGITS_V2().	66
ILUSTRACIÓN 43 MINIBATCHSIZE OBTENIDOS ALEATORIAMENTE. FUENTE PROPIA	69
ILUSTRACIÓN 44 RESULTADO LA DISMINUCIÓN DEL ERROR POR CADA ÉPOCA. FUENTE PROPIA	70
ILUSTRACIÓN 45 REPRESENTACIÓN DE UNA MATRIZ DE CONFUSIÓN. FUENTE (HTTPS://RPUBS.COM/CHZELADA/275494)	71
ILUSTRACIÓN 46 DIAGRAMA DE FLUJO PARA ENCONTRAR EL MEJOR AJUSTE DE PARÁMETROS. FUENTE PROPIA.	72
ILUSTRACIÓN 47 REPRESENTACIÓN DE PRECISIÓN Y EXACTITUD. FUENTE (HTTP://ANNYSANZ421.BLOGSPOT.COM/2017/09/DIFERENCIA-ENTRE-PRECISION-Y-RESOLUCION.HTML?M=1)	73
ILUSTRACIÓN 48 TABLA DIFERENTES PARÁMETROS APLICADOS PARA ENTRENAMIENTO Y TEST. FUENTE PROPIA	74
ILUSTRACIÓN 49 MATRIZ DE CONFUSIÓN CON LOS DATOS DE TEST. FUENTE PROPIA.	75
ILUSTRACIÓN 50 REPORTE DE CLASIFICACIÓN CON LOS DATOS DE TEST. FUENTE PROPIA	75
ILUSTRACIÓN 51 TABLA DIFERENTES PARÁMETROS APLICADOS PARA EVALUACIÓN. FUENTE PROPIA.	76
ILUSTRACIÓN 52 MATRIZ DE CONFUSIÓN CON DATOS DE EVALUACIÓN. FUENTE PROPIA.	77
ILUSTRACIÓN 53 TABLA DE RESULTADOS. FUENTE PROPIA.	78
ILUSTRACIÓN 54 CARACTERÍSTICAS DE EQUIPO COMPUTACIONAL REQUERIDO. FUENTE PROPIA.	79

RESUMEN

La presente tesis realiza la construcción de un algoritmo de Machine Learning (aprendizaje automático) para la identificación de neumonía en imágenes radiológicas, en el cual consiste en enseñarle a las máquinas a realizar las tareas con una mayor eficiencia que la del ser humano, esto permitirá optimizar los procesos de diagnóstico de esta enfermedad.

En el primer punto, explica que es la neumonía, a que hace referencia las Redes Neuronales Convolucionales, explica un poco sobre la Inteligencia Artificial (AI) y el Machine Learning, a su vez plantea la problemática, se establece el objetivo general y los objetivos específicos, así como la justificación de la tesis, los alcances, y el estado del arte de la misma. En el segundo punto se encuentra el marco referencial, donde muestra que teorías sustentan la investigación, las leyes que la rigen y los conceptos básicos para una mayor comprensión de lo escrito. En el tercer punto se explica la metodología utilizada, la línea de investigación escogida con el fin de dar desarrollo y avances tecnológicos a la optimización de procesos y generación de nuevos conocimientos, así mismo en que sector de la población va dirigido, el tipo de investigación. En el cuarto punto se enuncian las estrategias implementadas para desarrollar los objetivos generales y específicos de la tesis, por último, en el quinto punto se presentan las conclusiones a las que se llegó con el estudio.

INTRODUCCIÓN

La neumonía es una infección aguda del parénquima pulmonar lo cual puede ocasionar desde una inflamación hasta acumulación de líquido en los pulmones, su etiología puede ser por la infección de virus, bacterias u hongos.

Afecta a pacientes no hospitalizados y hospitalizados, se caracteriza por una serie de manifestaciones clínicas como es la aparición de fiebre y/o síntomas respiratorios (tos, expectoración) entre otros, junto con la presencia de alteraciones en la radiografía de tórax, representa más del 15% de todas las muertes de niños menores de 5 años a nivel mundial [1].

En Estados Unidos representa más de 500.000 visitas a los departamentos de emergencia y más de 50.000 muertes dato del 2015, lo que mantiene la dolencia en la lista de las 10 principales causas de muertes en el mundo [2].

En Colombia, según Estadísticas Vitales (DANE), en el año 2016 se registraron 7.570 muertes, cuya causa de defunción (CIE-10) es Neumonía, organismo no especificado y 24 por Neumonía Bacteriana, no clasificada en otra parte [3].

Aunque existen distintas medidas para prevenir, diagnosticar y tratar, la mitad de las muertes por neumonía específicamente infantil en el mundo, tienen lugar en el continente africano, especialmente en Nigeria (210.000 muertes), la República Democrática de Congo (132.000) y Etiopía (114.000). En Asia, la India acumula 410.000 muertes, seguida de Pakistán (92.000) y Afganistán (89.000). Así, por cada niño que pierde la vida a causa de la neumonía en un país desarrollado, más de 2.000 mueren en un país en desarrollo. Según Hospital Clínico Universitario de Santiago de Compostela (España) en la revista Neumo Expertos en Prevención, se estima que cada año, 150 millones de niños desarrollan la enfermedad y 11 millones de niños son hospitalizados a causa de la neumonía, y casi todos ellos viven en países en desarrollo, como Colombia [4].

Como parte de soluciones a este tipo de problemas de escala mundial la tecnología ha logrado un significativo aporte para combatir enfermedades donde la Inteligencia Artificial (AI, del inglés Artificial Intelligence) es protagonista por cuanto plantea la combinación de algoritmos con el propósito de crear máquinas que sean de ayuda y presenten las mismas capacidades del ser humano.

Ahora, es bien conocido en el sector de la Salud que para el diagnóstico de la Neumonía se usan Imágenes Radiológicas (CXR), Sin embargo, actualmente los proyectos enfocados al estudio y/o sistemas de predicción y procesamiento de datos pueden ayudar en el diagnóstico de la Neumonía, así, la tecnología avanzada, propone el uso de las Redes Neuronales que pertenecen a la disciplina de la Inteligencia Artificial (Redes Neuronales Artificiales (RNA), inspiradas en la biología, esto significa que están formadas por elementos que se comportan de manera análoga a las neuronas (en las funciones más elementales) y están organizadas de una forma similar a la del cerebro.

El nombre que reciben el tipo de Redes Neuronales que se encargan de procesar imágenes es Redes Neuronales Convolucionales (Convolutional Neural Network "CNN") las cuales son necesarias para generar un modelo de predicción para ayudar a dar soluciones a un problema en este caso del área de la salud, específicamente NEUMONÍA.

Las Redes Neuronales Convolucionales es implementado en este proyecto de investigación para obtener características significativas sobre un conjunto de imágenes CXR de pacientes con y sin neumonía, que a su vez se usaron para entrenar un modelo matemático con la capacidad de generalizar predicciones a partir de las distintas imágenes de neumonía. Este tipo de entrenamiento en el campo de inteligencia artificial es reconocido como aprendizaje supervisado, donde se dispone al algoritmo de entrenamiento un conjunto de datos y sus correspondientes etiquetas (Labels) que denotan, a qué corresponde cierto dato en particular. Se trata de obtener mayor precisión en el diagnóstico y en los resultados tal que permita una oportuna intervención e invasión en el cuerpo, mejorar en la

calidad de la atención, brindar mayor comodidad a los pacientes y brindar mayor acceso a los servicios de la tecnología en la medicina.

Está claro que la Inteligencia Artificial no va a reemplazar a los médicos, pero si es de gran ayuda a la hora de tomar decisiones en el diagnóstico de la neumonía También ayuda a reforzar los conocimientos de médicos generales que no tiene suficiente experiencia como los especialistas en radiología a la hora de interpretar una radiografía de tórax.

A pesar de que existen enormes desafíos, los beneficios a la hora de implementar avances tecnológicos en el sector médico son monumentales. El desafío para el futuro en este sentido, es lograr la incorporación total de la tecnología en las clínicas, hospitales y laboratorios que hoy son ajenos a ella por la falta de recursos económicos y con la que se salvarían un número importante de vidas diarias, al igual que se mejoraría la atención y el bienestar de todos los ciudadanos.

I DISEÑO METODOLÓGICO

1.1 Planteamiento del problema

En la ilustración 1, se presenta un árbol sobre el problema a tratar en este trabajo de investigación sobre el diagnóstico de la Neumonía en imágenes radiológicas.

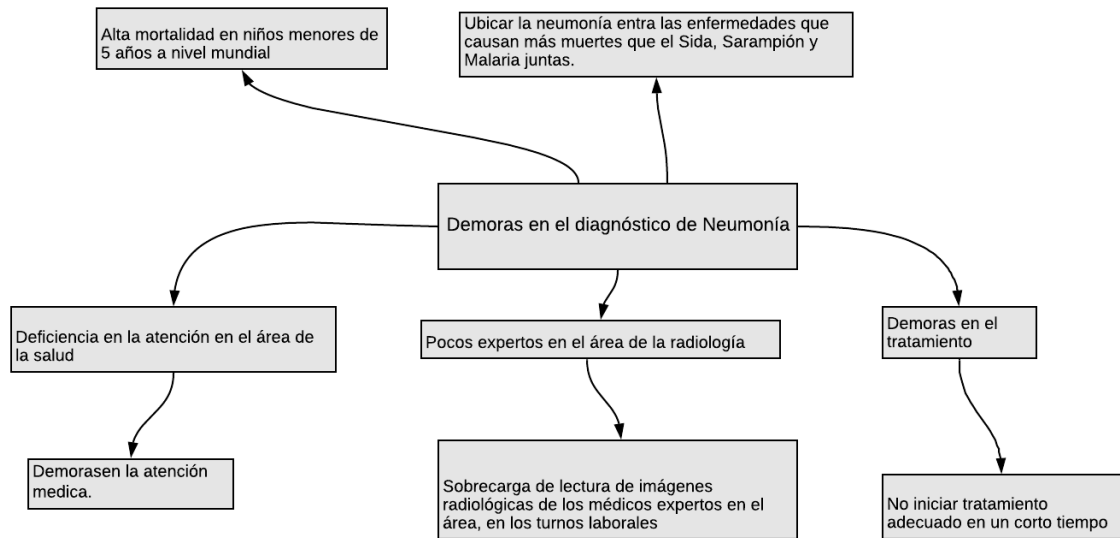


Ilustración 1 Árbol problema. Fuente propia.

1.2 Descripción del problema

La Neumonía es la enfermedad que causa el mayor número de muertes en los menores de cinco años de edad; anualmente mueren más de 2 millones de niños por esta enfermedad en el mundo, siendo responsable de más muertes que el SIDA, la Malaria y Sarampión juntas [5].

Si bien es común, diagnosticar con precisión la Neumonía es una tarea difícil, requiere la revisión de una radiografía de tórax (CXR) por especialistas altamente capacitados y la confirmación a través de la historia clínica, los signos vitales y los exámenes de laboratorio.

La Neumonía generalmente se manifiesta como un área o áreas de opacidad aumentada en CXR, sin embargo, el diagnóstico de Neumonía en CXR es complicado debido a una serie de otras afecciones en los pulmones, como sobrecarga de líquidos (edema pulmonar), sangrado o pérdida de volumen (atelectasia o colapso). Fuera de los pulmones, el líquido en el espacio pleural (derrame pleural) también aparece como un aumento de la opacidad en el CXR cuando está disponible. La comparación de los CXR del paciente tomado desde diferentes puntos temporales y la correlación con los síntomas clínicos y la historia clínica son útiles para establecer el diagnóstico.

En Colombia para el año 2016, según datos de Así Vamos en Salud, se presentó una tasa de mortalidad por Infecciones Respiratorias Agudas (IRA) en menores de 5 años de 13,84 por 100.000 menores de 5 años. Desde 1998 se ha dado un descenso sostenido en el comportamiento de este indicador. A nivel departamental, durante el 2015 los departamentos de Vaupés, Amazonas, Guaviare, Chocó, La Guajira, Cesar, Meta, Guainía y Vichada, presentaron tasas por encima de 20 muertes por 100.000 menores de 5 años. En contraste Antioquia, Huila, Norte de Santander, Arauca, Boyacá y Santander presentaron tasas por debajo de 10 muertes por 100.000 menores de 5 años.

La OMS y la UNICEF señalan la situación mundial con respecto a la neumonía y resaltan la necesidad de controlarla para alcanzar el Objetivo de Desarrollo del Milenio (ODM) sobre la mortalidad infantil.

Ahora bien, conociendo esta problemática, es necesario implementar un algoritmo, apoyado en la Inteligencia Artificial, específicamente en la utilización de Redes Neuronales que permitan optimizar estos procesos de diagnóstico, para lo cual las Redes Neuronales Convolucionales CNN, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones y por esto pueden ayudar con mayor precisión en problemas de esta área, las imágenes utilizadas para la implementación de este Modelo de Red Neuronal Convolutiva Para el Diagnóstico de Neumonía, fueron tomadas de la

plataforma Kaggle, de la National Institutes of Health de Estados Unidos (Instituto Nacional de la Salud de América del Norte), las cuales son de uso libre para el desarrollo de investigaciones en esta área.

La población beneficiada con la implementación a futuro de este proyecto serán los médicos y expertos en el área de la radiología, pero no será solo esa población la parte beneficiada sino que las entidades prestadoras de salud, pues mejoraran la atención de los pacientes, lo cual bajará los índices de mortalidad por Neumonía, puesto que la atención será prestada de manera oportuna y esto constituye un beneficio para los pacientes que padecieren esta enfermedad.

1.3 Justificación

Este trabajo de investigación está enmarcado en la línea de investigación de inteligencia artificial del grupo DEARTICA, de la Universidad del Sinú Seccional Cartagena, en la cual se estudian las aplicaciones de las técnicas de Machine Learning, enfocadas en el área de la salud.

Por otra parte, la salud es uno de los sectores más deficientes de Colombia que en opiniones de muchos analistas “está en crisis” [6] y en ese contexto, la tecnología puede aportar ciertos beneficios que mejorarían la calidad del servicio de las Instituciones Prestadoras de Salud.

Es nuestro interés principal el diagnóstico oportuno de la Neumonía, debido a que esta representa más del 15% de todas las muertes de niños menores de 5 años a nivel internacional, en 2015, 920,000 niños menores de 5 años murieron a causa de esta enfermedad, Así mismo, la radiografía de tórax, es un instrumento que facilita el estudio por imagen, más comúnmente utilizado para el diagnóstico de Neumonía.

Entre las causas de la baja calidad de las imágenes se encuentran la posición y la profundidad de inhalación que debe mantener el paciente en el momento de la toma de la imagen, estas pueden alterar la apariencia de la imagen radiológica, lo que

complica aún más la interpretación. Sumado a esto, los médicos se enfrentan a la lectura de grandes volúmenes de imágenes en cada turno, lo que afecta el diagnóstico efectivo [7].

Partiendo de esta necesidad, se visibiliza la aplicación de las técnicas de Inteligencia Artificial (IA) que son cada vez más comunes en el área de la salud y se convierte en una herramienta más sofisticada que apoyan al diagnóstico clínico de manera más rápida y a un menor costo [8].

Es así como el desarrollo de un algoritmo, apoyado en la Inteligencia Artificial, específicamente en la utilización de Redes Neuronales que permitan optimizar los procesos de diagnóstico de la Neumonía, para lo cual las Redes Neuronales Convolucionales CNN, puede ayudar con mayor precisión por medio del reconocimiento de imágenes.

Este proyecto, pretende servir de base para trabajos futuros a mediano plazo enfocados a la investigación e implementación de las tecnologías en este caso Inteligencia Artificial, para estudiar y ayudar en el diagnóstico de otras enfermedades a través de imágenes, el uso de estas técnicas de Machine Learning, hacen parte de la cuarta revolución industrial de la tecnología.

1.4 Formulación del problema

¿Cómo construir un modelo de Red Neuronal Convolutiva para el diagnóstico de Neumonía en imágenes radiológicas?

2. Alcance

El proyecto de grado está limitado al uso de la base de datos cargada en la plataforma de Kaggle, aportada por el Instituto Nacional de la Salud de EEUU, que contiene imágenes radiológicas de uso libre, el cual se le aplicará el algoritmo para diagnóstico de Neumonía, esperando un modelo matemático de predicción, con una tasa de verdaderos positivos superior al 70%.

Como resultado se entregará los archivos finales de modelado los cuales se podrán usar para realizar predicciones con otro tipo de imágenes de radiografía de tórax, y determinará el resultado en un 74% de exactitud.

El modelo no tendrá interfaz gráfica, ya que la metodología aplicada da como trabajo final un modelo matemático que puede ser desplegable para cualquier plataforma.

En el modelo se usarán otras técnicas y tecnologías como son el (Machine Learning o aprendizaje automático, el cual consiste en “enseñarle a las maquinas “ a realizar tareas con una eficiencia mayor que la del ser humano), las Redes Neuronales Convolucionales y como herramientas se usaran librerías de uso libre, lo cual no requerirá inversión en efectivo. Adicionalmente, el algoritmo se probará con los recursos computacionales disponibles de forma gratuita por las plataformas de Azure de Microsoft.

3. Objetivos

3.1 Objetivo general:

Desarrollar un modelo de Machine Learning usando Redes Neuronales Convolucionales (CNN) para identificar neumonía en imágenes radiológicas con formato DICOM.

3.2 Objetivos específicos:

- Construir el conjunto de datos que contenga un número representativo de imágenes radiológicas de neumonía que permitirán entrenar y validar la Red Neuronal Convolutiva.
- Modelar la Red Neuronal Convolutiva, haciendo entrenamiento con el 70% del conjunto de datos previamente construido, que permita identificar de forma correcta entre la clase positiva (con neumonía) y clase negativa (sin neumonía).

- Validar el modelo con el 30% del conjunto de datos que no fueron usados para el entrenamiento de la red neuronal, para lograr obtener un modelo generalizado que clasifique de forma correcta con datos desconocidos.

4. Estado del Arte

Los autores del proyecto realizaron una revisión de los diferentes trabajos que en su esencia se relacionan directamente con esta investigación tomando como base el uso de las Redes Neuronales para diagnósticos bajo imágenes radiológicas, encontrándose varios trabajos los cuales se citan a continuación:

4.1 CheXNet – UNIVERSIDAD DE STANFORD

Un equipo de investigadores de la universidad de Stanford desarrolló un algoritmo de IA capaz de diagnosticar la Neumonía mejor que los radiólogos, llamado CheXNet, que es un algoritmo de aprendizaje profundo capaz de evaluar las radiografías de tórax de los pacientes en busca de signos de enfermedad, se trata de una red neuronal convolucional de 121 capas, el algoritmo es capaz de diagnosticar 14 tipos diferentes de enfermedad y ha demostrado ser mejor que los radiólogos expertos en el diagnóstico de la neumonía, tanto en sensibilidad como en especificidad, el conjunto de datos utilizado en este algoritmo fue aportado por el Centro Clínico de los Institutos Nacionales de Salud (NIH), que contiene 112.120 radiografías de tórax de vista frontal etiquetadas con hasta 14 posibles patologías. Junto con los datos, el NIH publicó un algoritmo capaz de diagnosticar muchas de esas 14 patologías con cierto éxito, con el fin de alentar a otros a avanzar en ese trabajo.

En cuanto vieron estos materiales, los miembros del grupo de aprendizaje automático de Stanford, dirigido por el profesor adjunto de ciencias de la computación Andrew Ng, decidieron dirigir su investigación hacia esa línea.

Según los investigadores, el motivo por el que eligieron centrarse en la neumonía es porque, según los centros para el control y la prevención de enfermedades en los EE.UU., esta enfermedad lleva a un millón de estadounidenses al hospital cada año y es especialmente difícil de detectar en las radiografías.

En colaboración con Matthew Lungren, profesor asistente de Radiología, los investigadores hicieron que cuatro radiólogos de Stanford, trabajando de manera independiente, señalaran posibles signos de neumonía en 420 de las radiografías [9].

4.2 Avances de la inteligencia artificial (IA) en el diagnóstico de posibles casos de cáncer de próstata – UNIVERSIDAD DE MURCIA

Otro trabajo importante en el campo de la salud, trata sobre la Inteligencia Artificial en el proceso de diagnóstico del cáncer de próstata a través del uso de técnicas de Deep Learning. Este trabajo del Departamento de Ingeniería y Tecnología de Computadores y de Anatomía Patológica de la Universidad de Murcia (UMU), logró aplicar la Inteligencia Artificial (IA) al diagnóstico de posibles casos de cáncer de próstata, uno de los más extendidos en la población masculina.

En el campo de la informática, la inteligencia computacional permanece en auge constante. La aplicación de Redes Neuronales en los ordenadores, inspiradas en el funcionamiento del cerebro, está siendo aplicada para ser capaces de reconocer y aprender determinados patrones. Grandes empresas como Google las integran en sus servicios para el reconocimiento de voz, la traducción de voz, el reconocimiento de texto y, desde sus inicios, han sido aplicadas a la identificación de imágenes. Su campo de actuación ha seguido ampliándose en los últimos años, extendiéndose al mundo de la salud, donde se estudia su implantación como herramienta de diagnóstico por imagen.

El diagnóstico actual en Medicina se ha basado en la observación de imágenes analógicas a través de microscopio. Así, los expertos analizan a base de técnicas

microscópicas si en el paciente sufre cáncer. Hasta ahora, la única forma de llegar a un diagnóstico certero era a través de la biopsia, es decir, del análisis de un pequeño fragmento de tejido dañado, las muestras debían ser analizadas por métodos visuales estándares, con los parámetros que se pueden emplear para reconocer personas o distinguir una cara de otra. “Al aplicar el nuevo sistema queremos ganar precisión, tiempo y efectividad mediante métodos más cuantitativos, porque cualitativamente nos podemos equivocar, podemos creer que dos personas son idénticas, pero si incorporamos métodos técnicos, y aquí entra la informática, el reconocimiento será más rápido y mejor”, confirma el patólogo.

Para comenzar esta investigación se analizaron aproximadamente 500 biopsias, a partir de la digitalización de estas imágenes analógicas, se procesaron mediante técnicas de (Deep Learning) o aprendizaje profundo, que reconoció las características significativas que determinan la enfermedad.

A través de esta técnica, no se le proporciona al ordenador qué particularidades se distinguen, sino que a partir de un conjunto de entradas representadas en un gran número de muestras, la propia máquina alcanza un proceso de abstracción, al igual que los humanos, para determinar el diagnóstico de la enfermedad.

Actualmente en el campo médico se han alcanzado resultados por encima del 80%, y en otros tipos de imágenes no médicas se ha superado la capacidad de los humanos, con un ratio del 95%. Según Poblet [10], aumentar la precisión en el diagnóstico es fundamental. A partir de esta innovación se podrá incluso buscar tratamientos personalizados, adaptados a cada persona según el tamaño o tipo de tumor. En este caso, a través de esta colaboración inicial, se está analizando el cáncer de próstata, con la idea de mejorar nuestra capacidad diagnóstica”.

4.3. Inception v3 – GOOGLE

Otra investigación muy interesante es un Algoritmo de Google, que detecta el cáncer de pulmón en un 99%, el algoritmo llamado INCEPTION v3, proporciona un alto rendimiento con un costo computacional relativamente bajo ya que es de código

abierto. En este estudio, el algoritmo se entrenó con las imágenes procedentes de The Cancer Genome Atlas (El programa Atlas del Genoma del Cáncer) del Instituto Nacional de Cáncer, para clasificarlas con precisión y automáticamente en LUAD, LUSC o tejido pulmonar normal. También capacitaron a la red para predecir los diez genes más comúnmente mutados en LUAD.

Esta herramienta de aprendizaje profundo (en inglés, Deep Learning) distinguió las células cancerosas de las sanas en un 99% de las ocasiones. También fue capaz de diferenciar entre los dos tipos de tumor en casi 9 de cada 10 casos. Probando el algoritmo con muestras independientes de pacientes con cáncer, la precisión disminuyó pero, aun así, diagnosticó de forma correcta entre el 83% y el 97% de los casos.

Por otra parte, la inspección visual de las placas de histopatología es uno de los principales métodos utilizados por los patólogos para evaluar la etapa, el tipo y el subtipo de los tumores pulmonares. Estos hallazgos sugieren que los modelos de aprendizaje profundo pueden ayudar a los patólogos a detectar el subtipo de cáncer o las mutaciones genéticas. Los investigadores señalan que este algoritmo se puede aplicar a cualquier tipo de cáncer.

Hasta esta investigación ya se había probado este algoritmo para detectar cáncer de piel en el 2017, con resultados similares, concluyendo que con un pequeño conjunto de datos el modelo INCEPTION v3 puede alcanzar una precisión comparable a la de un profesional capacitado. La inteligencia artificial es cada vez más usada en la detección y análisis de cáncer en sus distintas clases y etapas [11].

Es importante resaltar que el uso de IA y Redes Neuronales Convolucionales que estudian imágenes clínicas se han ido enfocando a estudiar e interactuar con otro tipo de enfermedades.

4.4 Inteligencia artificial y técnica de aprendizaje automático para detectar rápidamente enfermedades oculares y neumonía.

En esta otra investigación se evidencia cómo se implementa las CNN para el diagnóstico de enfermedades oculares. Es el caso de una nueva herramienta basada en inteligencia artificial y técnicas de aprendizaje automático para detectar rápidamente enfermedades oculares y neumonía, según un estudio publicado en la revista Cell [12].

"La Inteligencia Artificial tiene un gran potencial para revolucionar el diagnóstico y la gestión de las enfermedades haciendo análisis y clasificaciones que involucran una gran cantidad de datos que son difíciles para los expertos humanos", señaló el autor principal Kang Zhang, profesor de oftalmología en Shiley Eye Institute en San Diego (California, EE.UU.).

Con esta tecnología los científicos pueden reconocer con mayor claridad las estructuras anatómicas discretas del ojo.

En este nuevo estudio, Zhang y sus colegas utilizaron una red neuronal basada en inteligencia artificial para revisar más de 200.000 exploraciones oculares realizadas con tomografía de coherencia óptica.

Posteriormente, los investigadores emplearon una técnica llamada aprendizaje por transferencia, en la cual el conocimiento adquirido para resolver un problema es almacenado por un ordenador y aplicado a problemas diferentes pero relacionados.

Por ejemplo, una red neuronal de Inteligencia Artificial optimizada para reconocer las estructuras anatómicas discretas del ojo, como la retina, la córnea o el nervio óptico, puede identificarlas y evaluarlas de manera más rápida y eficiente cuando se examinan las imágenes de un ojo completo.

Esto permite que el sistema aprenda de manera efectiva con un conjunto de datos mucho más pequeño que los métodos tradicionales, Los investigadores agregaron

luego pruebas de oclusión en las que la computadora identifica las áreas en cada imagen que son de mayor interés y la base de sus conclusiones.

Con las pruebas de oclusión, el ordenador puede decir dónde se ve en una imagen para llegar a un diagnóstico, para que podamos descubrir por qué el sistema obtuvo el resultado, esto hace que el sistema sea más transparente y que garantice la confianza en el diagnóstico.

El estudio se centró la degeneración macular y el edema macular diabético, dos causas comunes de ceguera irreversible que son tratables si se detectan temprano. Los diagnósticos derivados de la máquina se compararon con los diagnósticos de cinco oftalmólogos que revisaron las mismas exploraciones.

Además de realizar un diagnóstico médico, la plataforma también generó una referencia y recomendación de tratamiento no realizada en estudios previos.

Con una capacitación simple, la máquina se comportó de manera similar a un oftalmólogo bien entrenado, y podría generar una decisión sobre si el paciente debería o no derivarse para recibir tratamiento en 30 segundos, con más del 95 % de precisión. Tal velocidad y precisión representarían un profundo avance en el diagnóstico y tratamiento médico, según Zhang, la atención médica actual a menudo "es larga", ya que los pacientes son derivados de médicos generales a especialistas, consumiendo tiempo y recursos y retrasando el tratamiento efectivo.

Los científicos no limitaron su estudio a las enfermedades oculares y también probaron su herramienta para diagnosticar la neumonía infantil, una de las principales causas de muerte en todo el mundo en niños menores de 5 años.

5. Marco de Referencia

5.1 Marco Teórico

5.1.1 Neumonía

Según la publicación No. 351 de la reconocida revista Infectio de la Asociación Colombiana de Infectología, La neumonía es el proceso inflamatorio del tejido parenquimatoso pulmonar desencadenado por diversas especies de bacterias, virus, hongos o parásitos donde se ve afectada la porción distal del tracto respiratorio, bronquiolos y alvéolos; la reacción inflamatoria puede afectar también el intersticio alveolar y puede generar infiltrado celular inflamatorio y exudación en el espacio alveolar (consolidación del espacio aéreo), cuya expresión final es la alteración del intercambio gaseoso, la liberación de citosinas y mediadores inflamatorios, que se traducen en un complejo de síntomas y signos de compromiso del tracto respiratorio inferior, respuesta inflamatoria sistémica y evidencia de dicho compromiso en la radiografía del tórax [13].

Existen varios tipos de neumonía en los que se destacan la Neumonía Bacteriana, provocada por diferentes bacterias, *Streptococcus Pneumoniae* es la bacteria que con mayor frecuencia provoca la Neumonía bacteriana.

Neumonía Viral que es causada por diferentes virus entre los que se incluyen: El virus sincitial respiratorio (más frecuente en los niños menores de 5 años), El virus parainfluenza, Virus de la influenza (gripe), Adenovirus.

Neumonía por micoplasma presenta síntomas y signos físicos un tanto diferentes a los de los otros tipos de neumonía. Suelen causar una Neumonía leve y generalizada que afecta a personas de todas las edades, pero se presentan más casos en niños mayores de 5 años.

5.1.2 Machine Learning

Es un campo de las ciencias de la computación que, de acuerdo a Arthur Samuel en 1959, le da a las computadoras la habilidad de aprender sin ser explícitamente programadas.

Machine Learning es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos. La máquina que realmente aprende es un algoritmo que revisa los datos y es capaz de predecir comportamientos futuros. Automáticamente, también en este contexto, implica que estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana.

El propósito del (Machine Learning) es que las personas y las máquinas trabajen de la mano, a estas ser capaces de aprender como un humano lo haría. Precisamente esto es lo que hace los algoritmos, permiten que las máquinas ejecuten tareas, tanto generales como específicas.

Si bien al principio sus funciones eran básicas y se limitaban a filtrar correos, hoy en día puede hacer cosas tan complejas como predicciones de tráfico en intersecciones muy transitadas, detectar cáncer, mapear sitios para generar proyectos de construcción en tiempo real, e incluso, definir la compatibilidad entre dos personas [14].

5.1.3 Deep Learning

Aprendizaje Profundo (Deep Learning) es un conjunto de algoritmos de clase aprendizaje automático (Machine Learning) que intenta modelar abstracciones de alto nivel en datos usando arquitecturas compuestas de transformaciones no lineales múltiples.

El aprendizaje profundo es parte de un conjunto más amplio de métodos de aprendizaje automático basados en asimilar representaciones de datos. Una observación (por ejemplo, una imagen) puede ser representada en muchas formas (por ejemplo, un vector de píxeles), pero algunas representaciones hacen más fácil aprender tareas de interés (por ejemplo, "¿es esta imagen una cara humana?") sobre la base de ejemplos, y la investigación en esta área intenta definir qué representaciones son mejores y cómo crear modelos para reconocer estas representaciones [15].

5.1.4 Redes Neuronales Artificiales

Las Redes Neuronales Artificiales son modelos matemáticos que buscan replicar el comportamiento de las neuronas en la naturaleza, organizando su estructura de manera que emulen al cerebro.

Una Red Neuronal es un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas.

Las unidades de procesamiento se organizan en capas. Hay tres partes normalmente en una red neuronal: una capa de entrada, con unidades que representan los campos de entrada; una o varias capas ocultas; y una capa de salida, con una unidad o unidades que representa el campo o los campos de destino. Las unidades se conectan con fuerzas de conexión variables (o ponderaciones). Los datos de entrada se presentan en la primera capa, y los valores se propagan desde cada neurona hasta cada neurona de la capa siguiente, al final, se envía un resultado desde la capa de salida [16].

5.1.5 Convolución

La Convolución es una operación matemática, en la que una función se "aplica" de alguna manera a otra función. El resultado se puede entender como una "mezcla" de las dos funciones. La Convolución se representa por un asterisco (*), las convoluciones son buenas para detectar estructuras sencillas de una imagen, y después para juntar esas sencillas funciones para construir funciones aún más complejas. En una red convolucional, este proceso ocurre sobre una serie de muchas capas, en la que cada una de ellas realiza una Convolución sobre el resultado de la capa anterior.

5.1.6 Capa de activación ReLU.

La unidad lineal rectificada (ReLU) permite un entrenamiento más rápido y efectivo al asignar los valores negativos a cero y mantener los valores positivos. Esto a veces se conoce como activación, ya que solo las características activadas prosiguen su camino hacia la siguiente capa.

5.1.7 Capa de reducción o pooling

La capa de reducción o pooling se coloca generalmente después de la capa convolucional. Su utilidad principal radica en la reducción de las dimensiones espaciales (ancho x alto) del volumen de entrada para la siguiente capa convolucional. No afecta a la dimensión de profundidad del volumen. La operación realizada por esta capa también se llama reducción de muestreo, ya que la reducción de tamaño conduce también a la pérdida de información. Sin embargo, una pérdida de este tipo puede ser beneficioso para la red por dos razones:

La disminución en el tamaño conduce a una menor sobrecarga de cálculo para las próximas capas de la red, también trabaja para reducir el sobreajuste.

La operación que se suele utilizar en esta capa es max-pooling, que divide a la imagen de entrada en un conjunto de rectángulos y, respecto de cada subregión, se va quedando con el máximo valor.

5.1.8 Algoritmo Backpropagation

El algoritmo Backpropagation es el método de entrenamiento más utilizado en redes con conexión hacia delante. Es un método de aprendizaje supervisado de gradiente descendente, en el que se distinguen claramente dos fases: primero se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir la salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de las capas intermedias.

Backpropagation es un método utilizado en Redes Neuronales artificiales para calcular un gradiente que se necesita en el cálculo de los pesos que se utilizarán en la red. La propagación hacia atrás es una abreviatura de "propagación de errores hacia atrás", ya que un error se calcula en la salida y se distribuye hacia atrás a través de las capas de la red. Se usa comúnmente para entrenar redes neuronales profundas, la propagación hacia atrás es una generalización de la regla Delta a redes de avance de múltiples capas, que se hace posible al usar la regla de la cadena para calcular iterativamente los gradientes de cada capa.

5.1.9 Gradiente Descendente

Lo que busca el gradiente descendente es encontrar un mínimo de la función, ya sea local o no.

Para saber cuándo un valor se acerca más que otro es tomar un valor y restarle la derivada de la función en ese valor multiplicado por un Alpha, que es un número de cuando se irá desplazando.

Es importante considerar que Alpha debe ser un valor adecuado para nuestra función, si se toma un valor muy pequeño nuestra gradiente descendente podría ir muy lento, pero si se toma uno muy grande, podría pasar el mínimo.

Optimizador Gradient Descent es uno de los algoritmos más populares para realizar la optimización y el más común para optimizar Redes Neuronales.

5.1.10 Redes Neuronales Convolucionales

Una Red Neuronal Convolutiva (CNN o ConvNet) es uno de los algoritmos más populares para Deep Learning (aprendizaje profundo), que es un tipo de aprendizaje automático (Machine Learning) en el que un modelo aprende a realizar tareas de clasificación directamente a partir de imágenes, vídeos, textos o sonidos.

Las CNNs son especialmente útiles para localizar patrones en imágenes con el objetivo de reconocer objetos, caras y escenas. Aprenden directamente a partir de los datos de imágenes, utilizando patrones para clasificar las imágenes y eliminar la necesidad de una extracción manual de características.

Las aplicaciones que involucran reconocimiento de objetos y visión artificial (como los vehículos de conducción autónoma y las aplicaciones de reconocimiento facial) se basan fundamentalmente en las CNNs. En función de su aplicación, puede crear una CNN desde cero, o bien utilizar un modelo previamente entrenado con su conjunto de datos [17].

6. Marco Conceptual

6.1 Inteligencia Artificial

La Inteligencia artificial es el campo científico de la informática que se centra en la creación de programas y mecanismos que pueden mostrar comportamientos considerados inteligentes. En otras palabras, la IA es el concepto según el cual “las máquinas piensan como seres humanos”.

Normalmente, un sistema de IA es capaz de analizar datos en grandes cantidades (big data), identificar patrones y tendencias y, por lo tanto, formular predicciones de forma automática, con rapidez y precisión. Para nosotros, lo importante es que la IA permite que nuestras experiencias cotidianas sean más inteligentes. ¿Cómo? Al integrar análisis predictivos y otras técnicas de IA en aplicaciones que utilizamos diariamente.

Durante décadas se ha hablado del concepto de la “Inteligencia Artificial” (IA) y el impacto que tendría en nuestras vidas. El cine y la literatura han fantaseado con las posibilidades y los dilemas que se plantean desde HAL 9000 hasta Skynet, sin embargo, la realidad estaba siempre muy lejos de la ficción.

En los últimos años se han dado una serie de condiciones y cambios tecnológicos que han propiciado el resurgimiento de la IA y su aplicación en nuevas áreas, nos referimos a Big Data, Cloud, IoT, Mobility, una verdadera nueva revolución industrial. Las empresas se están posicionando en este nuevo escenario y, sin duda, Google es uno de los referentes en Inteligencia Artificial desde sus orígenes [18].

6.2 Microsoft Azure

Microsoft Azure es un conjunto en constante expansión de servicios en la nube para ayudar a su organización a satisfacer sus necesidades comerciales. Le otorga la libertad de crear, administrar e implementar aplicaciones en una red mundial enorme con sus herramientas y marcos favoritos.

6.3 Máquina virtual de Azure Data Science (DSVM)

La Máquina Virtual de Data Science (DSVM) es una imagen de máquina virtual personalizada en la nube Azure de Microsoft, creada específicamente para hacer ciencia de datos. Tiene muchas herramientas de ciencia de datos populares y otras preinstaladas y preconfiguradas para comenzar a crear aplicaciones inteligentes para análisis avanzado [19].

6.4 Anaconda Navigator

Anaconda es una Suite de código abierto que abarca una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la Ciencia de datos con Python. En líneas generales Anaconda es una distribución de Python que funciona como un gestor de entorno, un gestor de paquetes y que posee una colección de más de 720 paquetes de código abierto.

Anaconda Distribution se agrupa en 4 sectores o soluciones tecnológicas, Anaconda Navigator, Anaconda Project, Las librerías de Ciencia de datos y Conda [20].

6.5 Jupyter Notebook

Jupyter Notebook es una aplicación web de código abierto que le permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y mucho más [21].

6.6 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

6.7 Tensorflow

TensorFlow es el sistema de aprendizaje automático de segunda generación de Google Brain, liberado como software de código abierto a finales del año 2015. Mientras la implementación de referencia se ejecuta en dispositivos aislados, TensorFlow puede correr en múltiple CPUs y GPUs. Esta biblioteca también está disponible en Linux de 64 bits, macOS, y plataformas móviles que incluyen Android e iOS.

Los cómputos de TensorFlow están expresados como stateful dataflow graphs. El nombre TensorFlow deriva de las operaciones que tales Redes Neuronales realizan sobre arrays multidimensionales de datos. Estos arrays multidimensionales son referidos como “tensores”. En junio de 2016, Jeff Dean de Google declaró que 1.500 repositorios en GitHub mencionaron TensorFlow, de los cuales solo 5 pertenecían a Google.

TensorFlow es una librería open source utilizada para computación numérica. Fue creada y es soportada por Google bajo la licencia de Apache 2.0. La API está disponible para el lenguaje de programación Python, aunque hay acceso a la API C

++ subyacente (La mayoría de los ejemplos en los tutoriales los encontraras en Python). A diferencia de otras bibliotecas numéricas diseñadas para Deep Learning como Theano, TensorFlow se diseñó para su uso tanto en investigación y desarrollo como en sistemas de producción, por ejemplo, RankBrain, Google Search o el proyecto DeepDream. Se puede ejecutar en sistemas de CPU individuales, GPU, dispositivos móviles o sistemas distribuidos a gran escala con cientos de máquinas.

6.8 Imágenes DICOM

El estándar en Imagenología Digital y Comunicaciones en Medicina (DICOM) describe los medios de formato e intercambio de imágenes médicas y la información relacionada para facilitar la conectividad de dispositivos y sistemas médicos. El estándar DICOM respaldado por la Asociación de Fabricantes Eléctricos (NEMA) es el resultado de esfuerzos en conjunto de usuarios y fabricantes de imagenología médica y tecnología de la información sobre la salud, El estándar DICOM es el mecanismo de codificación, almacenamiento y transmisión de imágenes aceptado universalmente por la comunidad médica. La cabecera de este formato, extremadamente rica, permite almacenar información sobre el paciente, las condiciones en las que se tomó la imagen, y el formato interno de esta. La estructura de un fichero DICOM es compleja por el amplio conjunto de campos que almacenan información sobre el paciente, las condiciones en las que se tomó la imagen y el formato interno de ésta que puede ser múltiple dependiendo del formato de grabación de la imagen [22].

6.9 Dataset

Los Dataset en Machine Learning son conjuntos de datos históricos que sirven de base para entrenar un algoritmo con el objetivo de que una máquina pueda tomar decisiones.

6.10 Otras referencias

- H5py: La librería como interfaz conectada al formato de rescate HDF5 de Pythonic, necesario para guardar modelos de entrenamientos en discos. Instalado con pip `install h5py`.
- Matplotlib.pyplot as plt: Matplotlib es una biblioteca de trazado 2D de Python que produce cifras de calidad de publicación en una variedad de formatos de papel y entornos interactivos en todas las plataformas. Matplotlib se puede usar en los scripts de Python, las carcasas de Python e IPython , el cuaderno Jupyter , los servidores de aplicaciones web y cuatro kits de herramientas de interfaz gráfica de usuario.
- Numpy: La librería “numpy” de Python es muy útil para gestionar matrices y array en general.
- Pydicom as dicom: pydicom es un paquete de python puro para trabajar con archivos DICOM. Fue hecho para inspeccionar y modificar los datos DICOM de una manera fácil. Las modificaciones se pueden escribir de nuevo en un nuevo archivo.
- Pandas: Pandas es una biblioteca de código abierto con licencia BSD que proporciona estructuras de datos de alto rendimiento y fáciles de usar y herramientas de análisis de datos para el lenguaje de programación Python.
- Skimage.transform import resize: Esta librería implementa la function resize la cual permitió redimensionar las imagines
- Cnn_utils: Es una librería privada se utilizó para convertir los labels binarios en formato one hot formato de codificación compatible con tensorflow.

7. Marco Legal

A continuación, se identifica el marco normativo que regula particularmente a la Inteligencia Artificial en el manejo de datos:

7.1 Leyes aplicables

En Colombia no existe como tal una legislación o marco legal que rige la Inteligencia Artificial, además que haga énfasis en el uso y manejo de información de las personas para la ejecución de proyectos de Inteligencia Artificial.

Ley estatutaria 1581 “Ley de Habeas Data” esta ley trata sobre el manejo adecuado de la información recolectada, tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma.

7.2 Aspectos Éticos

En este trabajo de investigación se garantiza la integridad de los datos de los pacientes, son datos anónimos, no involucra directamente la intervención de pacientes físicamente, no existe riesgo biológico ni ambiental con el desarrollo del proyecto.

8. Metodología

La metodología que la investigación utilizará es Team Data Science Process (TDSP, la cual se muestra en la ilustración 2. Esta metodología de ciencia de datos es ágil e iterativa para ofrecer soluciones de análisis predictivo y aplicaciones inteligentes de manera eficiente. TDSP ayuda a mejorar la colaboración y el aprendizaje en equipo. Contiene una destilación de las mejores prácticas y estructuras de Microsoft y otros en la industria que facilitan la implementación exitosa de las iniciativas de ciencia de datos. El objetivo es ayudar a aprovechar al máximo los beneficios de su programa de análisis [23].

Para el desarrollo de este proyecto el Proceso de Ciencia de Datos del Equipo (TDSP) proporciona un ciclo de vida para estructurar el desarrollo de sus proyectos de ciencia de datos. El ciclo de vida describe los pasos, de principio a fin, que los proyectos suelen seguir cuando se ejecutan.

Este ciclo de vida ha sido diseñado para proyectos de ciencia de datos que se envían como parte de aplicaciones inteligentes. Estas aplicaciones implementan el aprendizaje automático o los modelos de inteligencia artificial para el análisis predictivo. Los proyectos de ciencia de datos exploratorios o los proyectos de análisis ad hoc también pueden beneficiarse de este proceso. Pero en tales casos, algunos de los pasos descritos pueden no ser necesarios.

El ciclo de vida describe las etapas principales que normalmente ejecutan los proyectos, a menudo de manera iterativa:

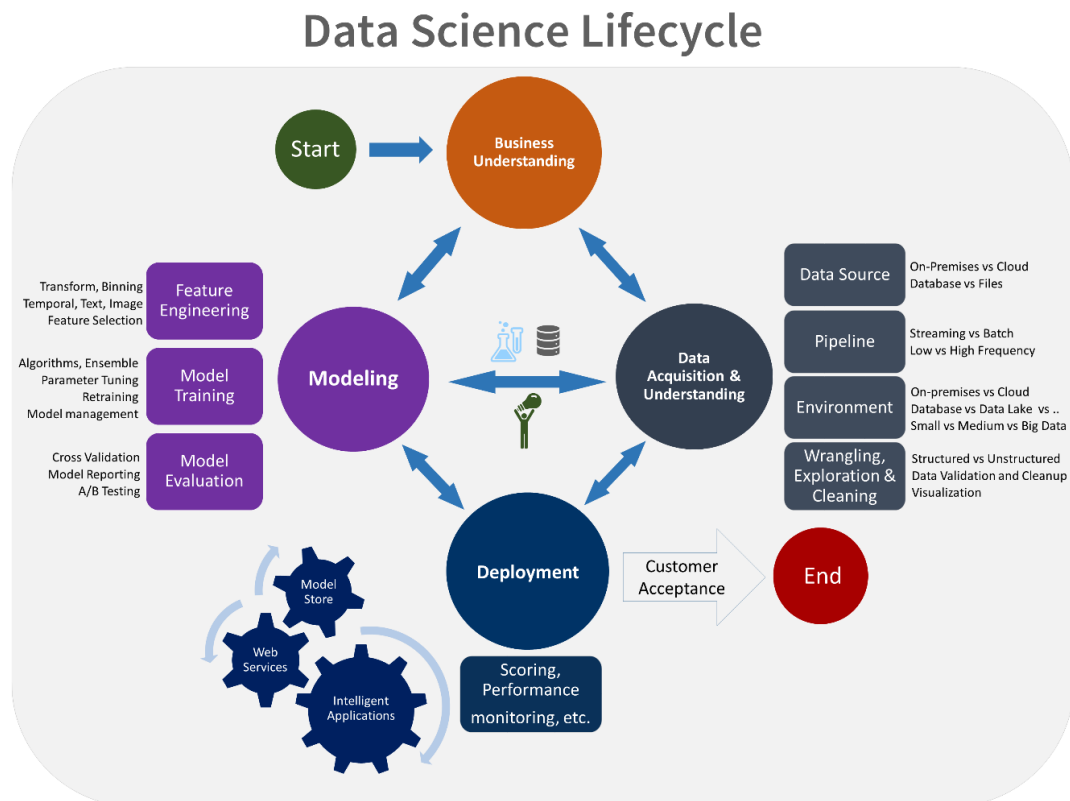


Ilustración 2 Metodología Utilizada. Fuente (<https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview>)

Entendimiento de negocios

Hay dos tareas principales abordadas al definir los objetivos:

- Trabaje con su cliente y otras partes interesadas para comprender e identificar los problemas comerciales.
- Formule preguntas que definan los objetivos comerciales que las técnicas de ciencia de datos pueden abordar.
- Identifique las fuentes de datos: encuentre los datos relevantes que le ayuden a responder las preguntas que definen los objetivos del proyecto.

Adquisición y comprensión de datos

Hay tres tareas principales abordadas en esta etapa:

- Procese los datos en el entorno analítico objetivo.
- Explore los datos para determinar si la calidad de los datos es adecuada para responder la pregunta.
- Configure un flujo de datos para puntuar datos nuevos o actualizados periódicamente.

Modelado

Hay tres tareas principales abordadas en esta etapa:

- Ingeniería de características: cree características de datos a partir de los datos sin procesar para facilitar el entrenamiento del modelo.
- Capacitación del modelo: encuentre el modelo que responde a la pregunta con mayor precisión comparando sus métricas de éxito.
- Determine si su modelo es adecuado para la producción.

Despliegue

La principal tarea abordada en esta etapa:

Operacionalice el modelo: implemente el modelo y la canalización en un entorno de producción o similar a la producción para el consumo de aplicaciones.

Aceptación del cliente

Hay dos tareas principales abordadas en esta etapa:

- Validación del sistema: confirme que el modelo y la tubería implementados satisfacen las necesidades del cliente.
- Transferencia del proyecto: entregue el proyecto a la entidad que ejecutará el sistema en producción.
- El cliente debe validar que el sistema satisface sus necesidades comerciales y que responde a las preguntas con una precisión aceptable para implementar el sistema en producción para que lo use la aplicación de su cliente.

Para el cumplimiento del objetivo (Construir el conjunto de datos que contenga un número representativo de imágenes radiológicas de neumonía que permitirán entrenar y validar la red neuronal convolucional), previamente se hizo un entendimiento de negocios no directamente con un cliente si no en la investigación del problema que se quiso resolver.

Para el pleno cumplimiento del objetivo relacionado en el anterior punto se hace una adquisición de datos y comprensión de los mismos de manera que se pudo entender la magnitud de estos datos y la capacidad de cómputo mínima para trabajar el manejo de los datos.

Para el cumplimiento del segundo objetivo de este proyecto (Entrenar la red neuronal convolucional con el 70% del conjunto de datos previamente construido, que permita identificar de forma correcta entre la clase positiva (con neumonía) y clase negativa (sin neumonía)), se procedió a cumplir con el punto del ciclo de vida que es el modelado, en el cual también se incluye el tercer objetivo que trata de

validar el modelo con el 30% del conjunto de datos que no fueron usados para el entrenamiento de la red neuronal y con otro conjunto de datos reservados los cuales se les llamo como datos de evaluación, para analizar los diferentes resultados y definir el mejor ajuste de entrenamiento, logrando obtener un modelo que pueda generalizar y clasifique de forma correcta con datos desconocidos.

Cabe aclarar que en el ciclo de vida de TDSP solo se aplicó hasta la fase de modelado ya que las demás fases no son de vital importancia para ejecución de los objetivos, como se puede observar en la ilustración 3.

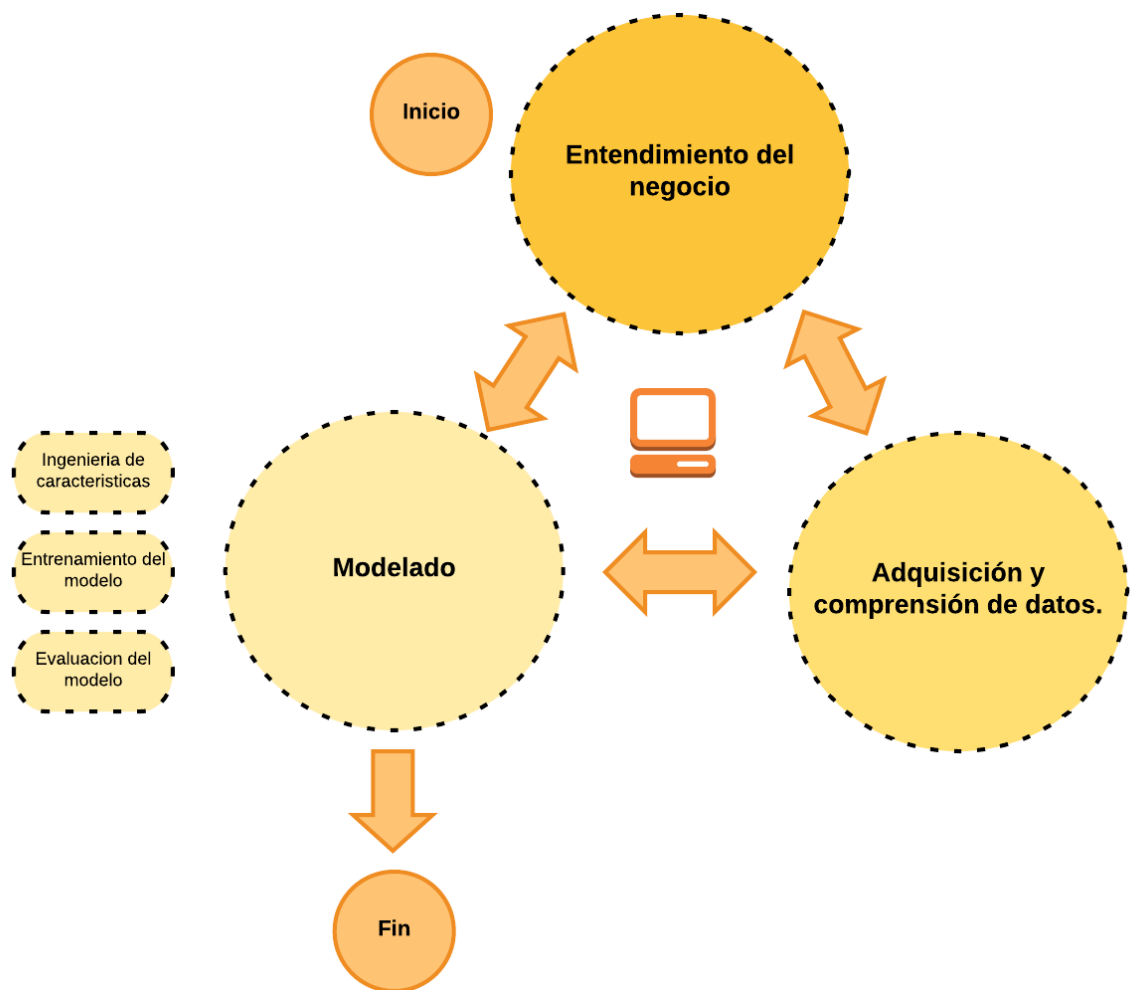


Ilustración 3 Metodología aplicada. Fuente propia.

8.1 Muestra y Población

Se tomó una muestra significativa de imágenes radiológicas de 10.000 imágenes entre pacientes con neumonía y pacientes sanos para el entrenamiento de nuestra red neuronal convolucional estos datos fueron tomados de National Institutes of Health de Estados Unidos (Instituto Nacional de la Salud de América del Norte) [24].

Dado que la privacidad del paciente es primordial, el conjunto de datos se examinó rigurosamente para eliminar toda la información de identificación personal antes de su publicación.

El criterio de selección fue dado por el algoritmo, las primeras imágenes del conjunto de imágenes total, donde estaban mezcladas las que correspondían a neumonía y cuales no correspondía a neumonía, La muestra tomada para el proyecto son 630 imágenes de entrenamiento, 270 imágenes para test y 270 imágenes para evaluación final.

8.2 Línea de Investigación

Este proyecto aportará a la línea de investigación de Inteligencia Artificial del grupo de investigación DEARTICA. Debido a que este proyecto involucra técnicas de Machine Learning y Deep Learning que son unas de las temáticas de gran impacto a nivel de investigaciones en el campo de la ingeniería y ciencias de la computación.

8.3 Tipo de Investigación

Este trabajo corresponde a una investigación aplicada, de cohorte retrospectiva donde los sujetos se estudian después de haberse producido la enfermedad. Los datos se obtendrán de las imágenes radiológicas y el diagnóstico de las mismas realizadas por un profesional de la salud; donde se aplican los conocimientos y las técnicas de inteligencia artificial para aportar en la solución de un problema de la

vida real, como es el apoyo diagnóstico en el área de radiología, específicamente aplicado a neumonía.

En este tipo de investigación el énfasis del estudio está en la aplicación efectiva de las técnicas de inteligencia artificial para obtener resultados positivos en términos de sensibilidad y especificidad.

II CONSTRUCCIÓN DEL CONJUNTO DE DATOS

En este capítulo se describe la construcción del conjunto de datos, en el cual se detalla paso a paso como está construido el conjunto de datos. Se identificaron una serie de etapas que determinan la ejecución de una tarea tanto para procesamiento de los datos como los resultados.

En el siguiente diagrama de flujo se presenta las tareas para la adquisición de los datos, el modelamiento de la red neural convolucional y finaliza con las evaluaciones del modelo construido. Ver ilustración 4.

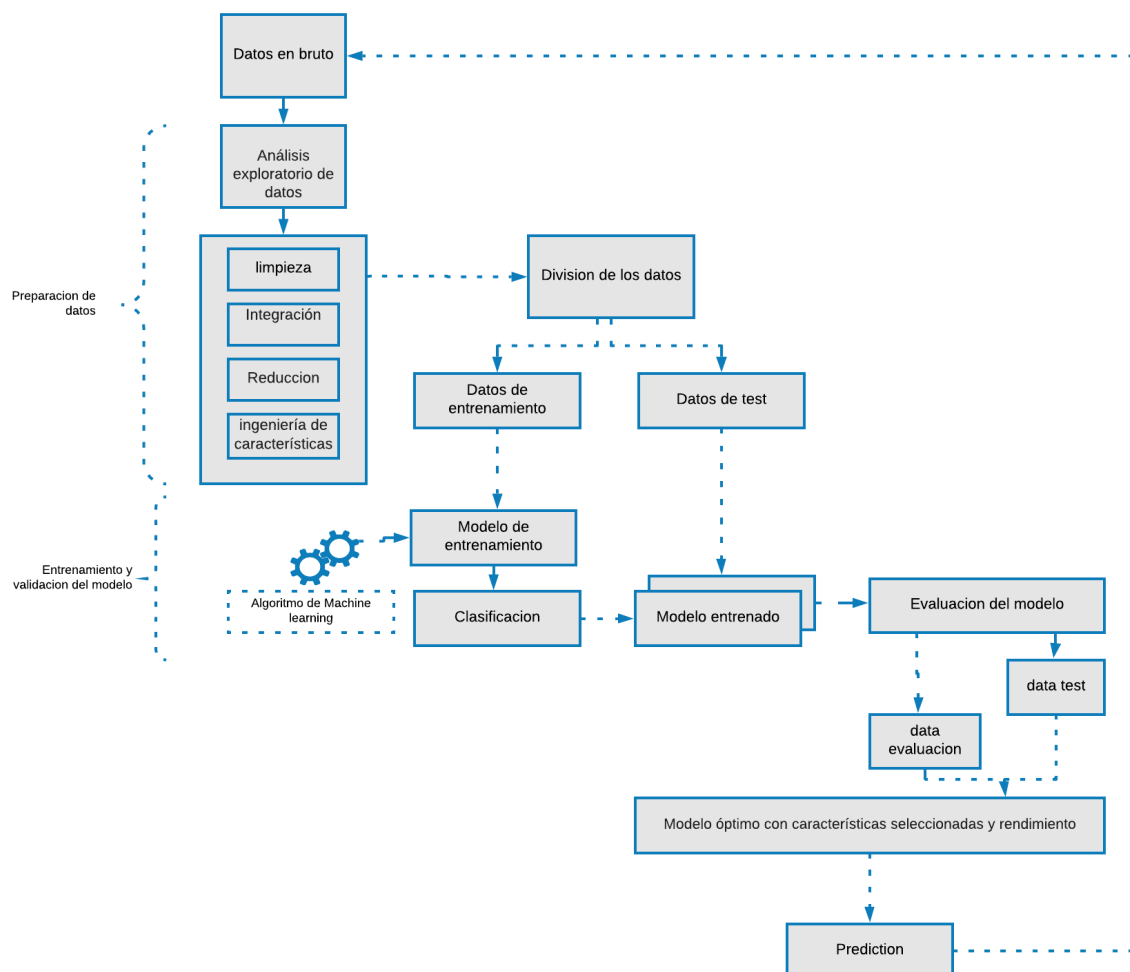


Ilustración 4 Resumen del algoritmo de Machine Learning. Fuente propia.

2.1 Adquisición y comprensión de datos

Para el cumplimiento de la primera fase de la metodología relacionada con el entendimiento del negocio, fue necesario para llevar a cabo la identificación de la fuente de los datos, el cual fue la base de este proyecto, sin tener directamente un cliente definido ya que esta investigación no lo ameritó.

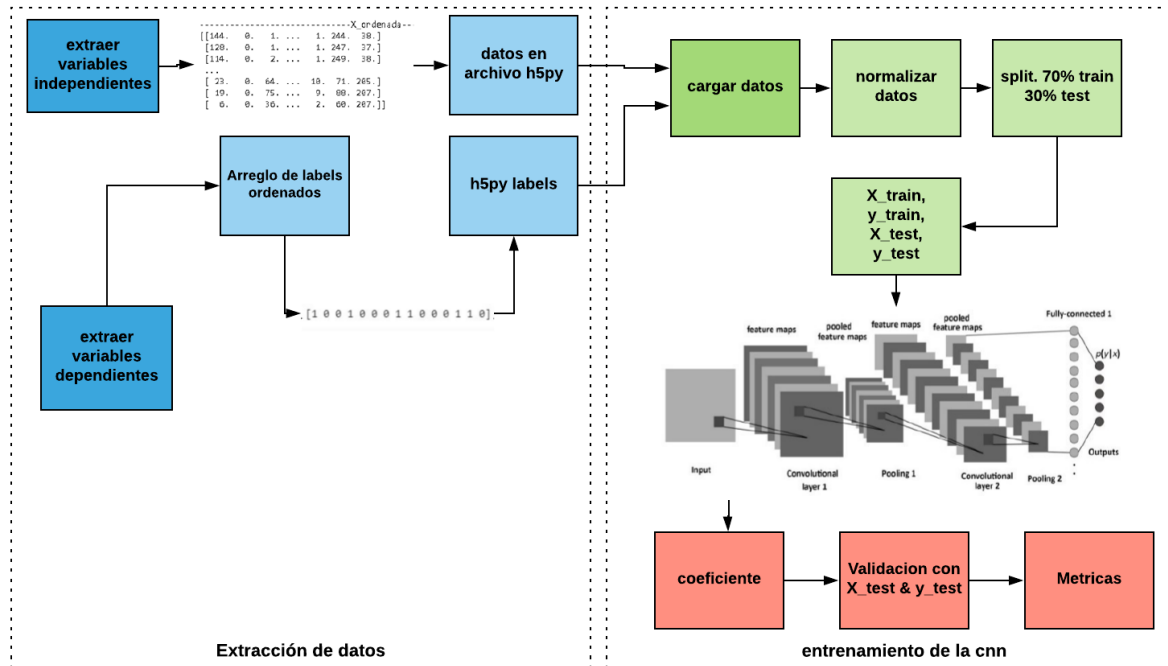


Ilustración 5 Representación ilustrada del algoritmo de predicción de Neumonía usando CNN. Fuente Propia.

En la Ilustración 5, se presenta el diseño del algoritmo para detección de neumonía, este plantea el desarrollo segmentado en dos fases, extracción de datos y entrenamiento. La primera fase consiste en extraer las características de los datos, esta a su vez implica una serie de pasos que se detallan a continuación.

2.2 Extraer datos de la imagen (Extraer variables independientes)

Se toman como dato de entrada las imágenes DICOM (Digital Imaging and Communication in Medicine), este formato de datos DICOM contiene la información en conjuntos de datos, eso significa que un archivo de una imagen de rayos X del

tórax lo que realmente agrupa es la identificación del paciente dentro del archivo, de tal modo que la imagen nunca puede separarse de esta información.

Un objeto de datos DICOM está construido a partir de una serie de atributos, que corresponden elementos como el nombre, ID, etc. También un atributo especial que contiene los datos de píxeles de la imagen, esta información proporciona al algoritmo los datos necesarios para la creación conjunto de datos de entrenamiento y pruebas. El nombre original de las imágenes DICOM fue previamente cambiado de forma manual a un orden numérico ascendente, iniciando desde el 1 hasta 10.000, para proporcionar una lectura ordenada y secuencial de cada imagen en el proceso de extracción de datos de la imagen DICOM.

Para la construcción del conjunto de datos se utilizó el algoritmo “create_dataset”, encargado de obtener los datos de interés y crear la persistencia de estos mismos. De esta manera se pudo conseguir un resultado dado en datos organizados, teniendo en cuenta que los datos de interés de cada imagen DICOM es una matriz de pixeles/8-bits con valores desde 0 hasta 255, la cual representa una imagen de tamaño 1024 ancho y 1024 de alto como se puede observar en la ilustración 7 que en cada iteración se fueron transformando las imágenes a un tamaño de 256 alto y 256 ancho, estas imágenes se han leído desde el disco local y almacenado en memoria por iteraciones para darle una forma y orden en (tamaño de la imagen dado un arreglo x número de imágenes) hasta finalmente tenerlas serializadas en un archivo de tipo h5py como se puede observar en la ilustración 6. Podrá comprender a detalles más específico en el anexo “algoritmo para la extracción de variables dependientes e independientes”.

```
-----X_ordenada-----  
[[144.  0.  1. ...  1. 244. 38.]  
 [128.  0.  1. ...  1. 247. 37.]  
 [114.  0.  2. ...  1. 249. 38.]  
 ...  
 [ 23.  0. 64. ... 10.  71. 205.]  
 [ 19.  0. 75. ...  9.  88. 207.]  
 [  6.  0. 36. ...  2.  60. 207.]]
```

Ilustración 6 Matriz que almacena las imágenes en columnas. Fuente propia.

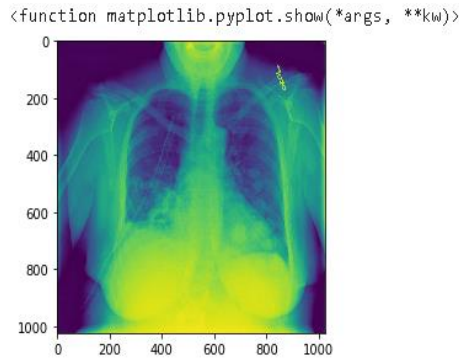


Ilustración 7 Visualización con matplotlib de una matriz de pixeles obtenidos de una imagen. Fuente propia

2.3 Algoritmo para la extracción de variables independientes desde un notebook

Para la construcción del dataset (conjunto de datos) se dio inicio a partir de la adquisición de un conjunto de imágenes (10000 imágenes radiológicas) que corresponden a neumonía y no neumonía, con un archivo “labels_unicos.csv” que contiene un identificador “patientid” que permitió asociar cual imagen corresponde a neumonía y no neumonía por medio clasificación binaria dentro de una columna llamada “Target”.

En la ilustración 8 se puede observar las columnas que están en el archivo “labels_unicos.csv”

1	patientid,x,y,width,height,Target		
2	0004cfab-14fd-4e49-80ba-63a80b6bddd6,,,,,0		
3	00313ee0-9eaa-42f4-b0ab-c148ed3241cd,,,,,0		
4	00322d4d-1c29-4943-afc9-b6754be640eb,,,,,0		
5	003d8fa0-6bf1-40ed-b54c-ac657f8495c5,,,,,0		
6	00436515-870c-4b36-a041-de91049b9ab4,264.0,152.0,213.0,379.0,1		
7	00569f44-917d-4c86-a842-81832af98c30,,,,,0		
8	006cec2e-6ce2-4549-bffa-eadfcd1e9970,,,,,0		

Ilustración 8 Columnas del archivo labels_unicos.csv. Fuente propia.

Las imágenes son de formato DICOM las cuales contienen una variedad de características, pero las que son de interés para obtener de cada imagen dicom son (patientId y pixelarray), donde cada (pixelarray) corresponde al mapa de pixeles que visualiza como tal la imagen radiológica de una dimensión origen de (1024x1024), teniendo en cuenta que debemos obtener estas características de cada imagen DICOM, se ha creado una función “ create_dataset ()” la cual tiene importada las librerías de uso libre que se pueden observar en la Ilustración 9 y un entorno de desarrollo que contaba con una capacidad de hardware en 56gb de RAM y un disco duro de 150 gb por el servicio de máquinas virtuales de Azure Microsoft.

```
import numpy as np
import matplotlib.pyplot as plt
import pydicom as dicom
import pandas as pd
import h5py
from skimage.transform import rescale, resize, downscale_local_mean
```

Ilustración 9 Librerías para “create_dataset ()”. Fuente propia.

Numpy fue usada para manipulación de los datos dado en matrices y vectores.

Matplotlib.pyplot fue usada para la visualización de los datos y gráficas.

Pydicom fue usada para la lectura y manipulación de los archivos “DICOM”.

Panda fue utilizada para la lectura y manipulación de archivos “csv”

H5py fue usada para crear la persistencia de los datos finales.

Resize de **skimage.transform** fue usada para manipular la dimensión (ancho, alto) de las imágenes.

Tensorflow 1.8 para la construcción de la Red Neuronal Convolutiva.

Python como lenguaje de programación para la construcción del algoritmo, dentro del ambiente de Anaconda.

Como primer paso del algoritmo se definió una variable de tipo “numpy” en la cual se almacenó la matriz final, como se observa en la Ilustración 10.

```
X=np.array([])
```

Ilustración 10 Variable tipo numpy para datos finales del pre procesamiento. Fuente propia.

En la Ilustración 11, se puede observar un ciclo iterativo que inicia desde la imagen 1 hasta la 10000, el algoritmo lee cada imagen que está alojada en la unidad (c:/) con la librería “pydicom” y extrae el arreglo de pixeles de cada imagen.

```
for m in range(1,10001):  
    imagen=(dicom.dcmread("C:\stage_2_train_images/"+str(m)+".dcm"))  
    imagen_dicom=np.array(imagen)# FICHERO DICOM*****  
    pixel_arr=np.array(imagen.pixel_array)
```

Ilustración 11 ciclo para la lectura de las imágenes. Fuente propia.

```
print("-----arreglo de la imagen "+str(m)+"-----")  
my_image = resize(pixel_arr, (pixel_arr.shape[0] / 4, pixel_arr.shape[1] / 4),anti_aliasing=True)
```

Ilustración 12 “Resize” de las imágenes. Fuente propia.

En la Ilustración 12, se puede observar la línea de código que dio uso de la librería (skimagen.transform), permitió realizar un “resize” por cada imagen a una dimensión de 256x256. Esta nueva dimensión permitió trabajar muchas más imágenes con el poder de computo que se disponía.

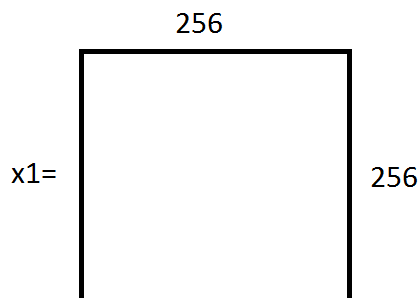


Ilustración 13 Dimensión de una imagen de entrenamiento (256x256). Fuente propia.

La Ilustración 13, refleja el alto y ancho de una imagen de entrenamiento.

En la Ilustración 14, se puede observar cómo a la matriz “my_image” de dimensiones 256x256 se le dio una nueva forma dada en 65536 filas x 1 columna (65536x1).

```
print("-----mas-----")
#print(my_image)
Xn=my_image.reshape((my_image.shape[0]*my_image.shape[1]),1)# CREAR LAS COLUMNAS Xn
```

Ilustración 14 Moldeo de matriz a columnas. Fuente propia.

En la Ilustración 15, se observa la dimensión de una imagen dentro el data set (conjunto de imágenes) que corresponde a un arreglo de una 1 columna y 65536 filas.

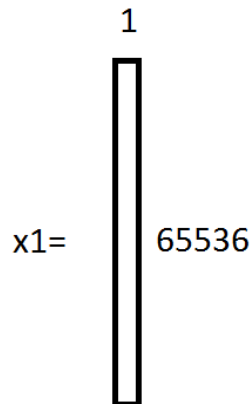


Ilustración 15 Imagen en forma de columna. Fuente propia.

Luego de moldear una imagen en la forma anterior se dio continuidad con unificar estos datos en un arreglo.

La Ilustración 16, muestra la función “np.append” que se usó para agregar cada arreglo de X1 a un nuevo arreglo X, que dio como resultado un arreglo que se puede observar en la ilustración 17.

```
X=np.append(X,Xn)
```

Ilustración 16 Función para agregar imágenes transformadas a un arreglo. Fuente propia.

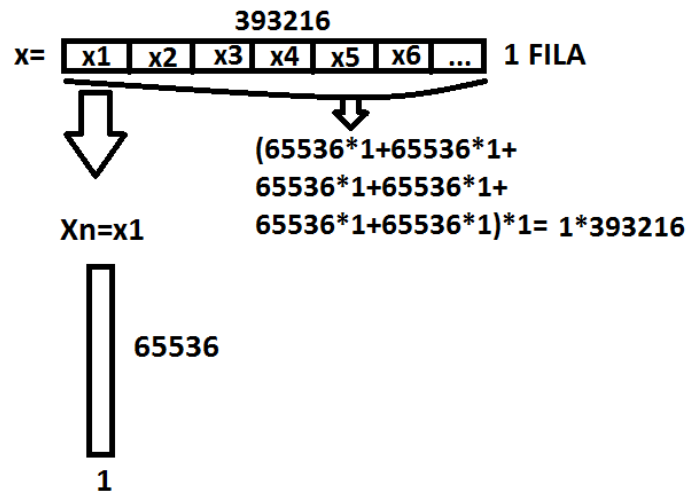


Ilustración 17 Construcción de arreglo de imágenes aplanadas. Fuente propia.

Luego de terminar el ciclo de lectura de imágenes como resultado se obtuvo el arreglo X el cual dio la forma que se puede observar en la ilustración 18, por medio de la instrucción de código como se ve en la ilustración 19, que divide el arreglo anterior (ilustración 17) entre el número de imágenes leídas.


```

[[144.  0.  1. ...  1. 244. 38.]
 [128.  0.  1. ...  1. 247. 37.]
 [114.  0.  2. ...  1. 249. 38.]
 ...
 [ 23.  0. 64. ... 10.  71. 205.]
 [ 19.  0. 75. ...  9.  88. 207.]
 [  6.  0. 36. ...  2.  60. 207.]]

```

Ilustración 22 Matriz de datos. Fuente propia.

Las instrucciones de código que permitieron crear la persistencia del data set (conjunto de datos) se puede observar en la Ilustración 23, esta tarea se realizó con el objetivo de poder tener los datos de forma uniforme y accesible en un solo archivo.

```

#crear el archivo hdf5
f=h5py.File("train_.hdf5", "w")
# crear el dataset
dset=f.create_dataset('train_', data=inputx)

dataset1 = h5py.File('train_.hdf5', "r")
#leer los elementos del dataset
D = np.array(dataset1['train_'][:])
print(D.shape)
plt.imshow(D[:,0].reshape(256,256))
plt.show

print("-----leabels-----")

```

Ilustración 23 Persistencia de los datos. Fuente propia.

2.4 Extracción de variables dependientes

Al mismo tiempo de que se realizó la extracción de los datos de la imagen, correspondiente a las variables independientes; también se realizó el proceso extracción de las variables dependientes del archivo "labels_unicos.csv", el cual contiene todos los diagnósticos correspondientes a cada imagen DICOM en forma binaria. Cada dato del archivo a su vez se asocia a cada imagen por medio de un identificador (*Patientid*). Estos valores indican si la imagen tiene diagnóstico de neumonía o no, esto permite construir el algoritmo de aprendizaje supervisado automático, especialmente al momento de realizar el entrenamiento y evaluación de la CNN (Convolutional Neural Network).

De igual forma el algoritmo para obtener las variables dependientes del archivo “labels_unicos.csv” cumple con la función de serializar sus datos como archivos h5py, pero teniendo en cuenta que este obtiene un resultado expresado como binario (1 y 0).

Fueron reorganizadas las etiquetas obtenidas, de tal forma que estas tuvieran la misma posición u orden de las imágenes. Teniendo en cuenta que las imágenes se organizaron inicialmente de forma ascendente por nombre.

Como resultado de la creación del conjunto de datos se obtuvieron dos subconjunto de datos de tipo h5py de forma organizada, que fueron útiles para implementar en el entrenamiento de la CNN con Tensorflow donde se llamó a los dos subconjuntos de datos con la función load_dataset(), para seguir procesándolos en memoria.

La persistencia de estos dos subconjuntos de datos está disponible en la ruta que se indicó en su creación (c:/) con una extensión Hdf5 los cuales luego se realizó con copia de seguridad en Onedrive para su uso desde cualquier otro ordenador.

2.5 Algoritmo para la extracción de variables dependientes desde un notebook

En otra celda del notebook (cuaderno de trabajo) se realizó la extracción de las variables dependientes a partir de archivo “labels_unicos.csv”.

Librerías utilizadas para la creación del data set (conjunto de datos) de los labels (etiquetas) fueron las que se observan en la ilustración 24.

```
import numpy as np
import pandas as pd
import pydicom as dicom
import h5py
```

Ilustración 24 Librerías para creación de data set de labels. Fuente propia.

La ilustración 25, muestra el uso de la función “read_csv ()” de la librería pandas, para realizar la lectura del archivo “labels_unicos.csv” que contiene los labels (etiquetas) y guardarlo en memoria con la variable labels.


```
labels = pd.read_csv('C:/labels_unicos.csv')
```

Ilustración 25 Lectura que archivo labels_unicos.csv. Fuente propia.

Con un ciclo iterativo partiendo desde la imagen 1 hasta la 10000 se realizó la lectura de cada imagen “dicom” para que de esta forma poder obtener los labels de cada imagen correspondiente por medio de su identificador (patientid) y como resultado guardar de “Target” la respuesta en binarios de cada imagen como se observa en la ilustración 26.

```
print("-----labels cruzados-----")
for d in range(1,10001):

    imagen=(dicom.dcmread("C:\stage_2_train_images/"+str(d)+".dcm"))
    pac=np.array(int(labels.Target[imagen.PatientID==labels.patientId]))

    print("-----id"+str(d)+"-----")
    print(imagen.PatientID)

    print(pac)
    j=np.append(Y,pac)
```

Ilustración 26 Extracción de label asociado a una imagen. Fuente propia.

Para concluir con la construcción de labels serializados con “h5py” se obtuvo un arreglo de dimensiones 1 columna x 10000 filas como se observa en la Ilustración 27.

```
f=h5py.File("data_set_test_Y.hdf5", "w")
# crear el dataset
dset=f.create_dataset('labels_unicos_test', data=j)

dataset1 = h5py.File('data_set_test_Y.hdf5', "r")
#leer los elementos del dataset
D = np.array(dataset1['labels_unicos_test'][:])
print(D.shape)
print(D[5])
```

Ilustración 27 Persistencia de labels. Fuente propia.

Como resultados de este capítulo se obtuvieron dos subconjuntos de datos serializados con la librería h5py los cuales corresponden al conjunto datos obtenidos de las imágenes y otro corresponde a las etiquetas (labels) de cada imagen, para

posteriormente comenzar el modelamiento de la red neuronal convolucional a partir de estos datos serializados.

III MODELADO DE LA RED NEURONAL CONVOLUCIONAL

En este capítulo se explica el proceso de entrenamiento al que fue necesario exponer los datos moldeado, y además se explica cómo se distribuyen y procesan los datos para el algoritmo de Machine Learning diseñado.

3.1 Cargue de los datos

Con los subconjunto de datos obtenidos para la CNN (Convolutional Neural Network) se permitió seguir con la fase de entrenamiento en la cual se implementó la herramienta de uso libre (Tensorflow) para la construcción y evaluación de la CNN (Convolutional Neural Network), donde se requirió una función que fue creada con el nombre `load_dataset()` con el fin de cargar en memoria todos los datos con una dimensión (tamaño de imagen dado un [] x número de imágenes), y las variables dependientes con una dimensión (cantidad de variables dependientes x una sola columna), tanto las variables dependientes y la matriz de datos están serializados en formato h5py y en la mencionada función `load_dataset()` fueron convertidos a arreglos de tipo (Numpy) para permitir su manipulación en memoria y terminar de preparar los datos que (Tensorflow) debe recibir.

Cabe aclarar que solo se pudieron cargar en memoria 900 imágenes ya que el poder computacional con el que se trabajó no era lo suficiente para un mayor trabajo.

3.2 Normalización de los datos

Fue necesario tomar la matriz de datos y realizar una normalización aplicando una división de la matriz de tipo (Numpy) que contiene los datos de entrenamiento entre 255 (datos de entrenamiento/255 el valor máximo de un pixel), esta normalización permitió que las operaciones matriciales que se realizaron en el entrenamiento sean mucho más rápidas y eficientes tanto para la multiplicación que se realiza con los pesos que se inicializan y para la actualización de pesos con respecto a nivel de error que se debe disminuir. Al mismo tiempo a los labels se les aplico una

conversión de formato que recibe (Tensorflow) llamado (One Hot Encoding) la cual consiste en este caso de un matriz (cantidad labels en filas x columnas), en la cual cada columna representa la categoría de la imagen (neumonía y no-neumonía) y cada fila pertenece al resultado de una imagen en específico como se puede observar en la ilustración 29, su valor en binario fue permitiendo determinar a qué categoría pertenece dicha imagen en una posición.

En la ilustración 28, se puede observar una matriz de datos sin normalizar a diferencia de una matriz de datos normalizados en la ilustración 30.

```
[[[ 0  0  0 ... 102 102 101]
 [ 0  0  0 ... 101 101 100]
 [ 0  0  0 ... 101 101 100]
 ...
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]]
```

Ilustración 28 Ejemplo datos sin normalizar. Fuente propia.

```
[[1 0]
 [1 0]
 [0 1]
 [1 0]
 [0 1]
 [0 1]
 [0 1]
 [0 1]
 [0 1]
 [1 0]
 [0 1]]
```

Ilustración 29 Ejemplo labels en one hot. Fuente propia

```
[[0.      0.      0.      ... 0.4      0.4      0.39607843]
 [0.      0.      0.      ... 0.39607843 0.39607843 0.39215686]
 [0.      0.      0.      ... 0.39607843 0.39607843 0.39215686]
 ...
 [0.      0.      0.      ... 0.      0.      0.      ]
 [0.      0.      0.      ... 0.      0.      0.      ]
 [0.      0.      0.      ... 0.      0.      0.      ]
 ]]
```

Ilustración 30 Ejemplo datos normalizados. Fuente propia

3.3 Segmentación de los Datos

En la ilustración 31, se representa la técnica (Split), que opera con una división del total de datos que se tienen cargados en memoria en una distribución de 70% para datos de entrenamiento y un 30% para test, el objetivo de tener estos datos divididos es pertinente al corroborar con los datos del test, si el modelo generalizaba con datos conocidos.

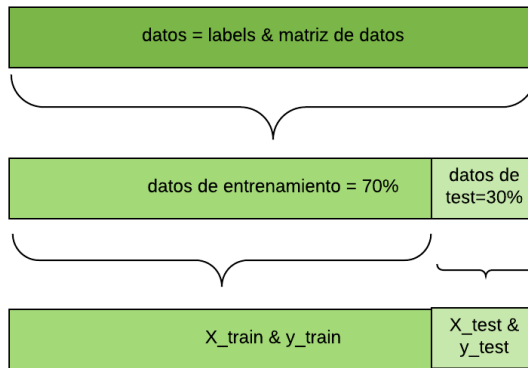


Ilustración 31 Representación del Split de datos. Fuente propia.

Encontrar un balance en el aprendizaje del modelo CNN en el que no estemos incurriendo en (Underfitting) y tampoco en (Overfitting) como muestra la ilustración 32, a veces esto puede resultar una tarea un poco difícil.

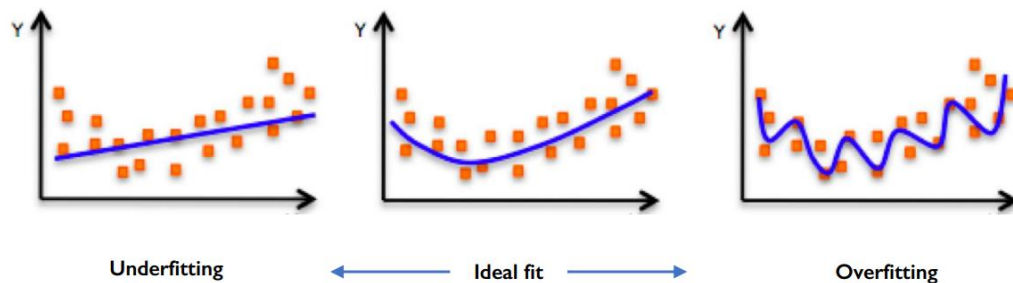


Ilustración 32 Ejemplo de datos afectados con underfitting y overfitting. Fuente (<http://introtodeeplearning.com/>)

Overfitting: Lo que ocurriría es que nuestra máquina sólo se ajustará a aprender los casos particulares que le enseñamos y será incapaz de reconocer nuevos datos de entrada.

Underfitting: Si nuestros datos de entrenamiento fueran muy pocos nuestra máquina no será capaz de generalizar el conocimiento y estará incurriendo en underfitting.

```
SHAPE DE X_train ES: (630, 65536)
SHAPE DE X_test ES: (270, 65536)
SHAPE DE y_train ES: (630, 2)
SHAPE DE y_test ES: (270, 2)
```

Ilustración 33 Split de conjunto de datos. Fuente propia.

En la ilustración 33, se puede observar el Split de los 900 datos “X” (variables independientes), & “y” (como variables dependientes) que fueron cargados en memoria de los cuales se destinaron un 30% para test y un 70% para entrenamiento.

3.4 Construcción de CNN (Capas de Convolución)

La ilustración 34, muestra la secuencia de la CNN dada por las funciones que ayudaron a la construcción de la arquitectura de red neuronal convolucional.

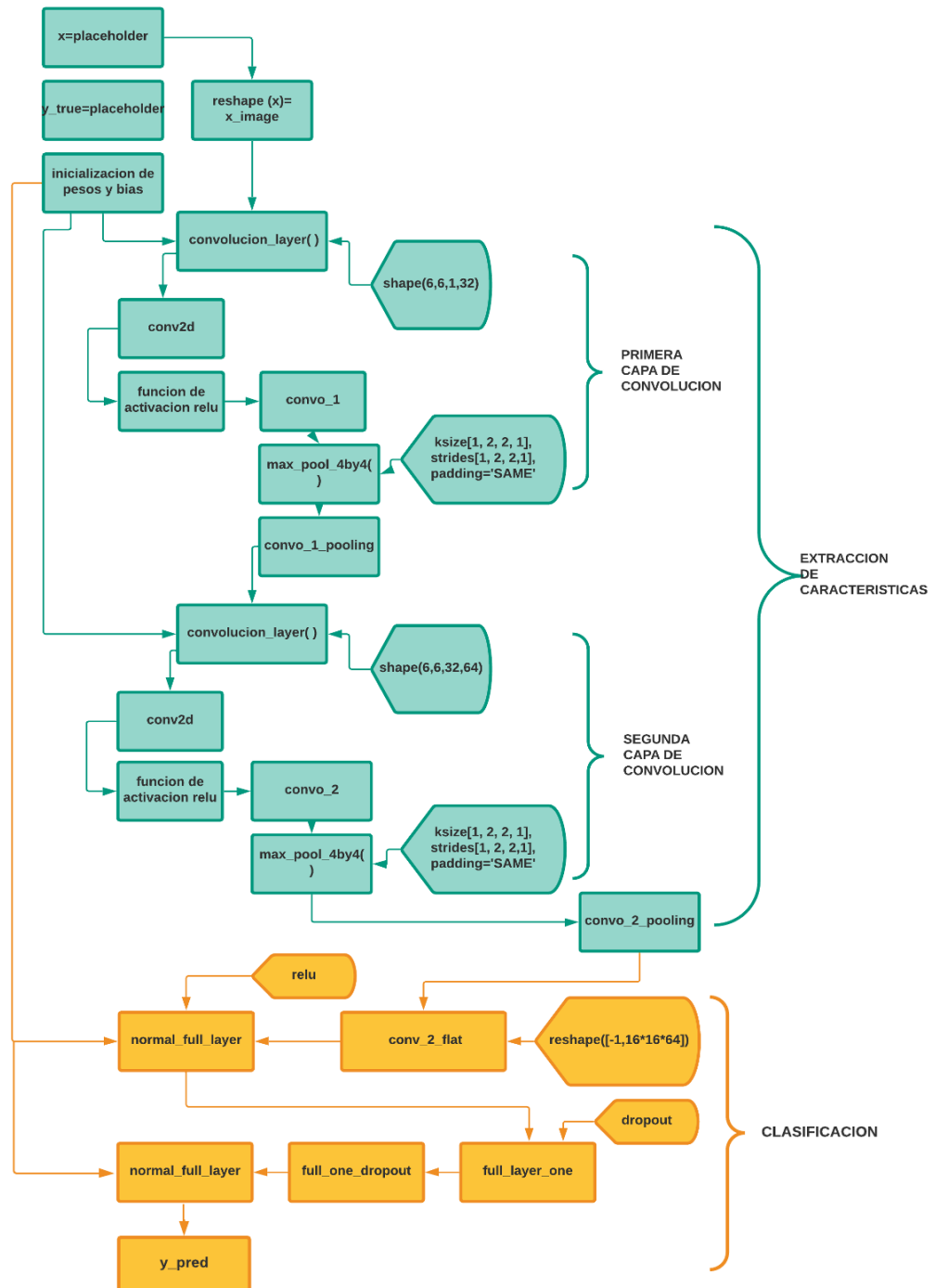


Ilustración 34 Diagrama de la estructura de las capas del modelo. Fuente propia

La arquitectura de la red neuronal convolucional costo de dos subconjuntos de procesos en el proceso de entrenamiento los cuales son extracción de características y clasificación como muestra la ilustración 35.

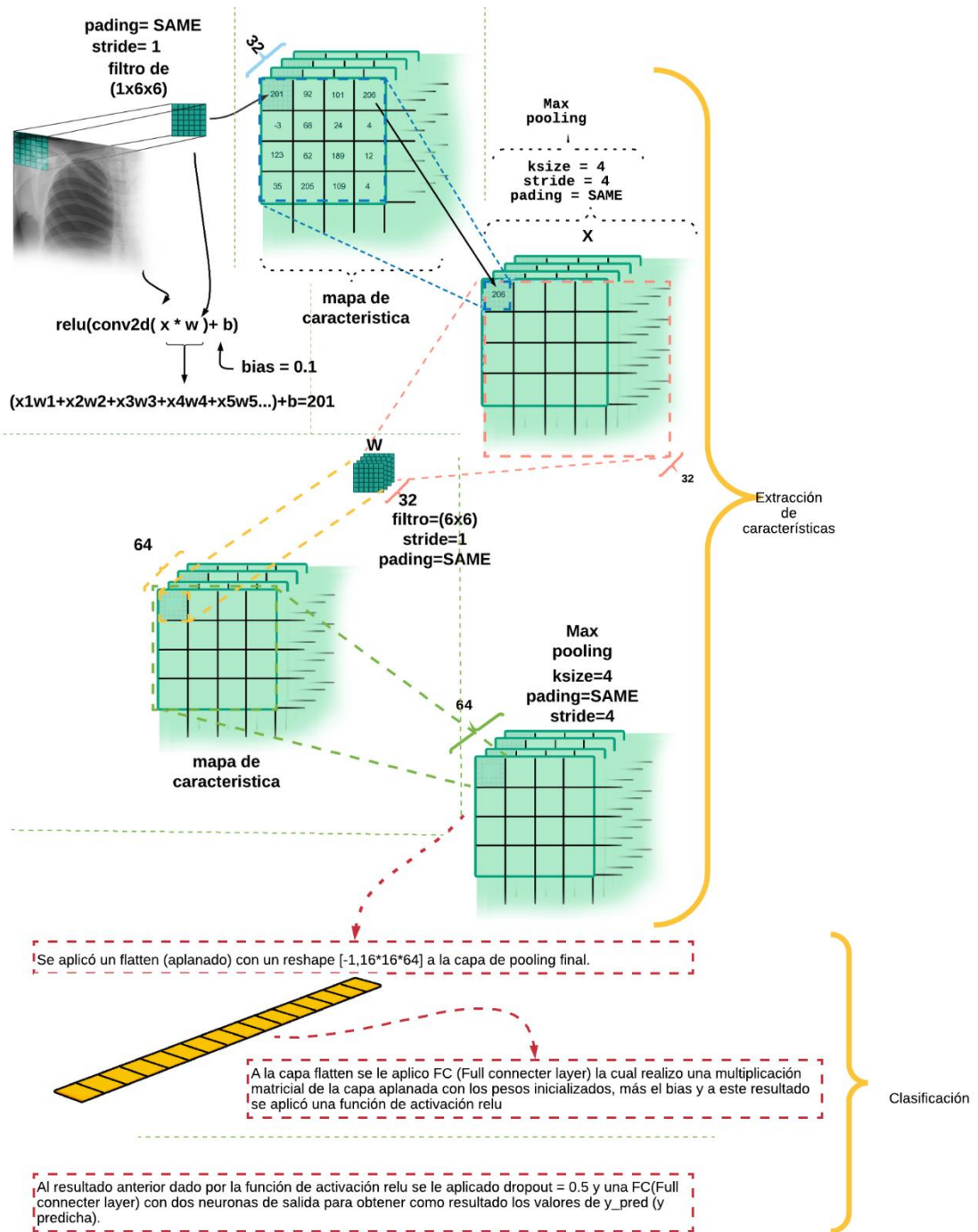


Ilustración 35 Grafico de estructura de las capas del modelo. Fuente propia

Para la creación de arquitectura de la CNN se implementaron las siguientes funciones que permitieron ajustar y determinar los parámetros de cada capa. En la ilustración 36, se puede observar el orden con el que es normalmente es aplicando cada tipo de capa convolucional.

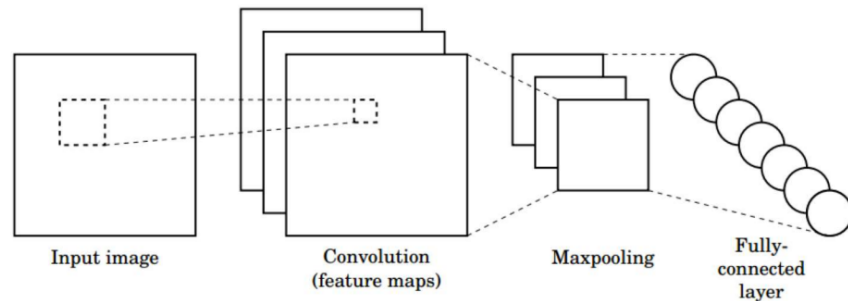


Ilustración 36 Ejemplo de cada tipo de capa. Fuente (<http://introtodeeplearning.com/>).

Int_weights (shape)

Esta función permitió la inicialización de los pesos de la red neuronal convolucional a partir del argumento shape [6,6,1,32], el cual se le entrega a la función “Int_weights (shape)” y retorna valores aleatorios en una distribución normal con una desviación estándar de 0.1

Donde el $6 \times 6 \times 1$ inicial representa el tamaño “ventana deslizante” como se observa en la ilustración 37, que corrió dentro de la imagen, en este caso un área de 6×6 con una dimensión de 1 porque estas imágenes están en una escala de grises (monocromáticas) a diferencia de una imagen (RGB) que corresponde a una dimensión de 3. El 32 que se colocó de último determina el número de mapas de características resultantes como se observa en la ilustración 35.

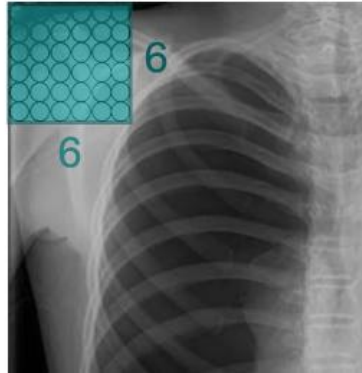


Ilustración 37 Tamaño de la "ventana deslizante". Fuente propia

Init_Bias (bias)

Esta función permitió la inicialización del bias con una constante igual a 0.1 que recibe la forma y número de filtros en cada convolución.

conv2d(x, W)

Esta función permitió el proceso de convolución que fue llamado en cada capa convolucional, para que cada convolución pudiera extraer características por medio de los argumentos que recibe (datos de entrada & pesos), con un stride = 1 que determina que tanto va a ir deslizando el filtro en adquisición de características dada para cada entrada x.

max_pool_4by4(x)

Esta función recibió como entrada una capa de convolución a la cual se le aplicó una técnica llamada (Max Pooling) que sencillamente lo que hizo es obtener el valor máximo dentro de un filtro de 4 x 4 (ksize), que va pasando de 4 en 4 posiciones (strides) dentro de la capa convolucional que recibió, con un relleno (Padding), igual a "same" (que indica que se añadan tantas filas y columnas de ceros como sea necesario para que la salida tenga la misma dimensión que la entrada) que permitió

establecer un relleno de borde para que el tamaño de salida sea el proporcional e independientemente al número de strides (que nos indica el número de pasos en que se mueve la ventana de los filtros) que se determinará.

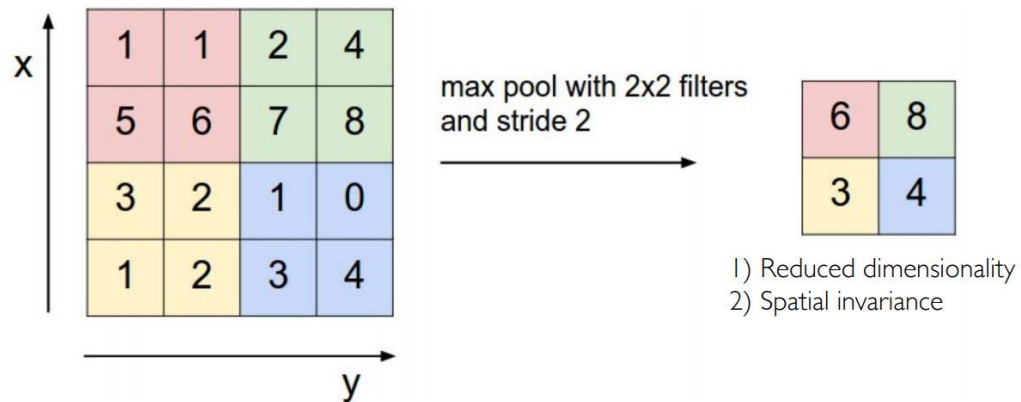


Ilustración 38 Ejemplo de un Maxpooling con un filtro de 2x2. Fuente (<http://introtodeeplearning.com>)

En la ilustración 38, se puede identificar cómo funciona la técnica de (Maxpooling) con un filtro de 2x2 pasando en una imagen de entrada con una dimensión de 4x4.

ReLU – Rectified Lineal Unit

La función relu convierte los valores negativos a cero y dejando los positivos tal y como entran, así como se muestra en la Ilustración 39, se define como $R(z) = \max(0, z)$.

Características de la función ReLU:

- No está acotada.
- Se pueden morir demasiadas neuronas.
- Se comporta bien con imágenes.
- Buen desempeño en Redes Convolucionales.

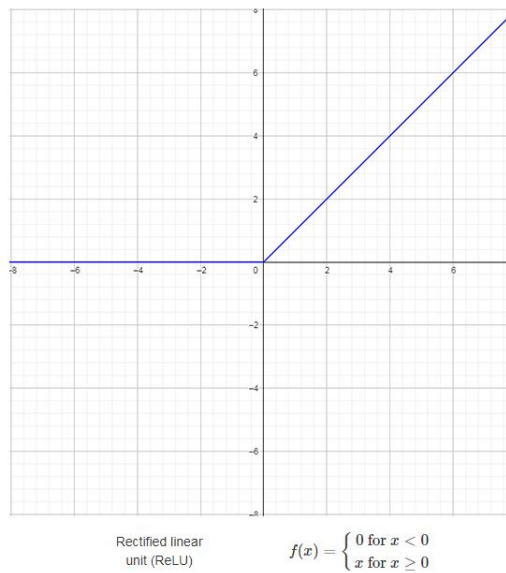


Ilustración 39 Grafica de la función Relu. Fuente propia

Convolutional_layer (input_x, shape)

Esta función permitió la creación de las capas convolucionales por medio de los argumentos que recibió como datos de entrada (input_x) y tamaño del filtro (shape), con el que se hizo las convoluciones en dada capa convolucional y a su vez tomo estos argumentos para llamar la función que inicializo los “pesos y bias”, para a su vez tener como resultado el retorno de la convolucion con una función de activación “relu”.

Convo_2_flat

Para poder implementar la técnica Flattening (aplanado) la cual consiste en darle una forma aplanada a una matriz fue necesario realizar un reshape (Remodelar) con la forma [-1,16*16*64] para iniciar la fase de clasificación, y como resultado aplanado (flatening) la matriz que tenía una dimensión con alto y ancho ahora será una imagen alargada (arreglo alargado).

normal_full_layer (input_layer, size)

Esta función determinada como la capa completamente conectada está dentro de la fase de clasificación y solo recibió datos de entrada y número de mapas de características de los datos entrantes, para que así se pudiera realizar la multiplicación matricial de datos de entrada con los pesos inicializados más el bias.

tf.nn.dropout ()

Por medio de esta función se pudo aplicar Dropout (Técnica de regularización patentada por Google para reducir el sobreajuste de las Redes Neuronales mediante la prevención de complejas adaptaciones en los datos de entrenamiento. Es una forma muy eficiente de realizar promedios de modelos con redes neuronales.), el cual permitió apagar un 50% de las neuronas para hacer que a medida que el modelo fuera llegando a la última capa de clasificación no se viera afectada de sobreajustes y en el entrenamiento la Red Neuronal fuera forzada a encontrar nuevas características. En la ilustración 40, se puede observar cómo trabaja Dropout en una red neuronal

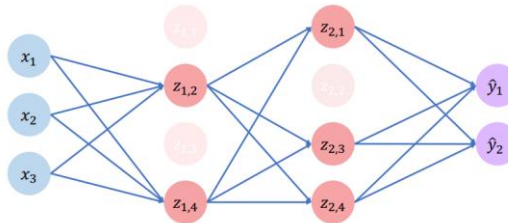


Ilustración 40 Ejemplo de dropout. Fuente (<http://introtodeeplearning.com>).

Luego de haber creado la arquitectura con las funciones de ayuda para la (CNN) se definió la tasa de aprendizaje o también llamado (Learning Rate) con valor igual a 0.0001, permitió establecer que tan rápido pudo ser el aprendizaje y la actualización de los pesos por medio de la función de optimización. En la ilustración 41, se puede ver cómo trabaja un Learning Rate(Velocidad de aprendizaje) con un valor alto y bajo para encontrar un mínimo local.

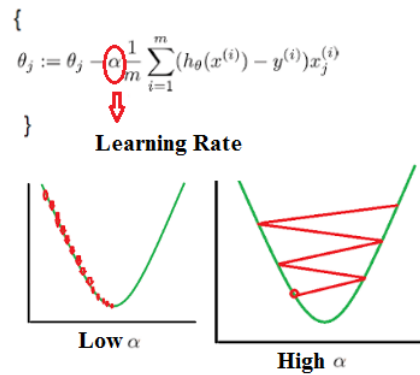


Ilustración 41 Visualización de cómo trabaja learning rate.

Fuente (<https://stackoverflow.com/questions/42332058/gradient-descent-thetas-not-converging>)

Para el cálculo de pérdidas se usó la entropía cruzada por medio de la función **tf.nn.softmax_cross_entropy_with_logits_v2()** que recibe los datos “y_pred & y_true” como se observa en la ilustración 42, para así poder medir el error de probabilidad en esta tarea de clasificación discreta.

Además se utilizó la operación “reduce_mean()” al cálculo de las pérdidas (error) para conseguir la media de los elementos.

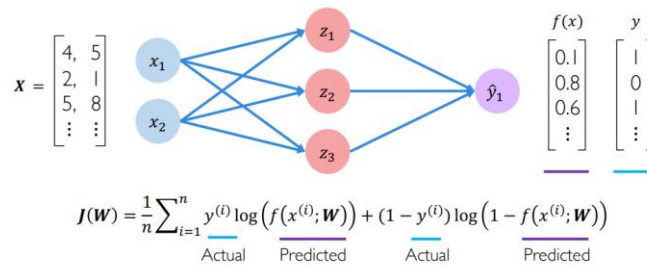


Ilustración 42 Fórmula matemática de tf.nn.softmax_cross_entropy_with_logits_v2().

Fuente (<http://introtodeeplearning.com>)

El optimizador para minimizar el error previamente calculado fue Adam Optimizer por medio de la función “tf.train.AdamOptimizer()”, que recibió como argumento learning rate y el resultado de función de costo, con el objetivo de disminuir el error a un porcentaje muy bajo

Optimizador (Adam)

Para minimizar el error previamente calculado por medio de la función “tf.train.AdamOptimizer()”, fue Adam Optimizer, que recibió como argumento (Learning Rate) y el resultado de función de costo, con el objetivo de disminuir el error a un porcentaje muy bajo

La estimación del momento adaptativo (Adam) es otro método que calcula las tasas de aprendizaje adaptativo para cada parámetro. Además de almacenar un promedio de decaimiento exponencial de gradientes cuadrados pasados, Adam también mantiene un promedio de decaimiento exponencial de gradientes pasados $M(t)$, similar al impulso:

$M(t)$ y $V(t)$ son valores del primer momento que es la Media y el segundo momento que es la varianza no centrada de los gradientes, respectivamente.

Ecuación 1 Las fórmulas para el primer momento (media) y el segundo momento (la varianza) de los degradados.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

Ecuación 2 Las fórmulas para el primer momento (media) y el segundo momento (la varianza) de los degradados.

$$v_t = \frac{v_t}{1 - \beta_2^t}$$

Entonces la fórmula final para la actualización de parámetros es:

Ecuación 3 Fórmula final para la actualización de parámetros.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

Los valores para β_1 son 0.9, 0.999 para β_2 y $(10 \times \exp(-8))$ para ϵ .

Adam funciona bien en la práctica y se compara favorablemente con otros algoritmos de método de aprendizaje adaptativo, ya que converge muy rápido y la velocidad de aprendizaje del modelo es silenciosa, rápida y eficiente, y también rectifica todos los problemas que se enfrentan en otras técnicas de optimización, como la desaparición de la tasa de aprendizaje, convergencia lenta o alta variación en las actualizaciones de parámetros que conducen a la función de pérdida fluctuante [25].

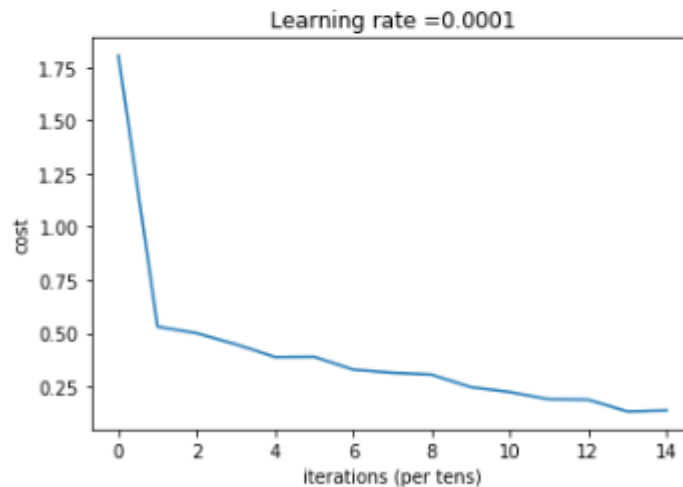
Una característica que tiene Tensorflow es la manera en como permite diseñar toda la estructura de la CNN en la cual primero se estableció un grafo de las capas convolucionales y las distintas variables de Tensorflow que son inicializadas en la función “`tf.global_variables_initializer()`”.

Luego de haber diseñado el grafo de las capas con la función de pérdida y el optimizador se definió la función “`random_mini_batches_v2()`” que permitió segmentar los grupos de imágenes aleatorios que se proporcionaron a la CNN por cada época (iteración), en la que realiza entrenamiento, actualización de los pesos. Con el objetivo de tener un seguimiento de cómo va aprendiendo el modelo con respecto a la disminución del error e inclusive estableció mucha más estabilidad en cuando al rendimiento de la memoria RAM con los procesos de cálculo de los datos cargados.

Métricas

Luego de haber entrenado el modelo se pudo observar una disminución del error por cada época como se puede ver en la ilustración 44, llegando a un valor de 0.138335 lo cual representa un buen resultado por parte del error.

```
Cost after epoch 0: 1.804402
Cost after epoch 10: 0.531185
Cost after epoch 20: 0.500114
Cost after epoch 30: 0.448518
Cost after epoch 40: 0.388601
Cost after epoch 50: 0.390130
Cost after epoch 60: 0.329111
Cost after epoch 70: 0.314362
Cost after epoch 80: 0.306148
Cost after epoch 90: 0.246856
Cost after epoch 100: 0.224177
Cost after epoch 110: 0.189994
Cost after epoch 120: 0.187817
Cost after epoch 130: 0.133510
Cost after epoch 140: 0.138335
```



```
Train Accuracy: 0.96349204
Test Accuracy: 0.7740741
```

Ilustración 44 Resultado la disminución del error por cada época. Fuente propia

También se pudo evaluar la exactitud del modelo en su proceso de test con un resultado superior a 70% y una exactitud en su entrenamiento igual 96%, los cuales son buenos resultados

Una manera de representar los resultados obtenidos fue mediante una matriz de confusión la cual permitió obtener los números de predicciones acertadas y cuales no fueron acertadas en cuanto al test como se puede observar en la ilustración 45.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Ilustración 45 Representación de una matriz de confusión. Fuente (<https://rpubs.com/chzelada/275494>)

VP es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.

VN es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.

FN es la cantidad de positivos que fueron clasificados incorrectamente como negativos.

FP es la cantidad de negativos que fueron clasificados incorrectamente como positivos.

En la ilustración 46, se puede observar el ciclo apropiado para poder analizar los resultados dados por la aplicación ajustes a los parámetros tanto para test como para evaluación.

El primer ciclo consta de un entrenamiento con el conjunto de datos de entrenamiento que luego dio paso a la evaluación del modelo ya entrenado para esperar los resultados dados en un reporte de clasificación y una matriz de

confusión, que permitieron el análisis en función de los datos de test para encontrar el ajuste de los parámetros que mejor se comportara con este conjunto de datos.

Cuando se encontró un ajuste apropiado para el modelo según las evaluaciones de test se prosiguió a con otro tipo de datos, que ya estaban reservados para seguir evaluando el modelo y determinar qué tan capaz fue el modelo para generalizar con un conjunto de datos distinto.

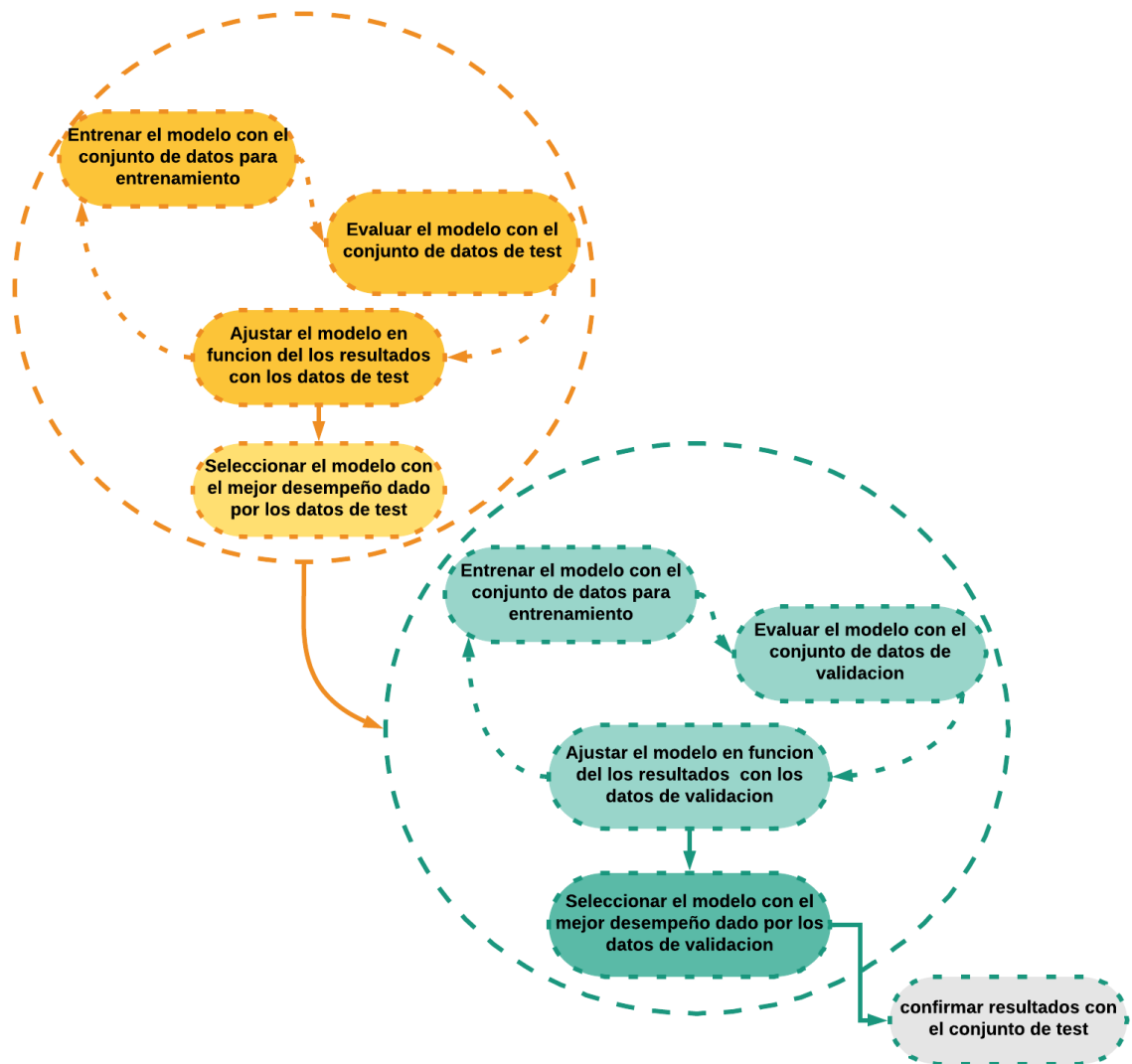
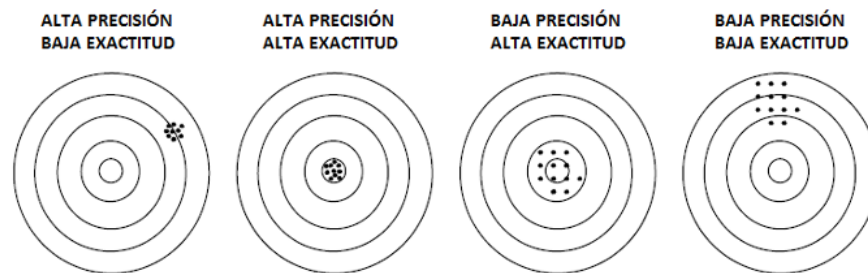


Ilustración 46 Diagrama de flujo para encontrar el mejor ajuste de parámetros. Fuente propia.

Como se muestra en la ilustración 47, la Precisión o “Accuracy” (AC) se refiere a la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una magnitud. Cuanto menor es la dispersión mayor la precisión. Se representa por la proporción entre el número de predicciones correctas (tanto positivas como negativas)

La Exactitud, se refiere a lo cerca que está el resultado de una medición del valor verdadero. En términos estadísticos, la exactitud está relacionada con el sesgo de una estimación. También se conoce como Verdadero Positivo (o “True positive rate”). Se representa por la proporción entre los positivos reales predichos por el algoritmo y todos los casos positivos.

La Sensibilidad se refiere a la respuesta que el instrumento de medición tenga para medir una variable y que tan rápida sea este para estabilizar su medida [26].



*Ilustración 47 Representación de precisión y exactitud. Fuente
(<http://annysanz421.blogspot.com/2017/09/diferencia-entre-precision-y-resolucion.html?m=1>)*

F1-score en el análisis estadístico de la clasificación binaria, la puntuación F 1 (también la puntuación F o la medida F) es una medida de la precisión de una prueba. Considera tanto la precisión p como el recordatorio r de la prueba para calcular la puntuación: p es el número de resultados positivos correctos dividido por el número de todos los resultados positivos devueltos por el clasificador, y r es el número de resultados positivos correctos dividido por el número de todas las muestras relevantes (todas las muestras que deberían haberse identificado como positivas). La puntuación F 1 es la media armónica. De la precisión y recuperación,

donde una puntuación F 1 alcanza su mejor valor en 1 (precisión perfecta y recuperación) y peor en 0.

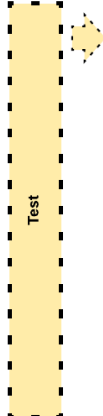
La ilustración 48, denota el registro de distintos parámetros (learning rate, batch_size, épocas y pasos) que más se acercaron a un resultado significativo, con el reporte de métricas que se dieron por cada evaluación con los datos de test.

Learning rate (velocidad de aprendizaje)

Batch size (grupo de imágenes por cada época)

Época (iteración en la que se realiza entrenamiento y minimización de error)

Pasos (pasos que se dan en cada época)



parámetros de ajuste				datos (30%) & (70%)		Métricas							
learning rate	batch_size	Épocas	pasos	cantidad de datos de test	cantidad de datos de entrenamiento	Precisión	exactitud	sensibilidad	F1	TP	TN	FP	FN
0.0001	32	180	15	270	630	0.77	0.75	0.91	0.83	164	48	42	16
0.0001	32	100	10	270	630	0.84	0.77	0.81	0.82	146	63	27	34
0.0002	32	100	10	270	630	0.79	0.76	0.86	0.83	157	50	40	23
0.0001	32	80	10	270	630	0.80	0.77	0.86	0.83	156	52	38	24
0.0001	32	100	10	270	630	0.82	0.77	0.83	0.82	150	58	32	30
0.0001	40	180	10	270	630	0.8	0.72	0.8	0.8	144	54	36	36
0.0001	32	150	10	270	630	0.78	0.74	0.86	0.82	157	40	40	33

Ilustración 48 Tabla diferentes parámetros aplicados para entrenamiento y test. Fuente propia

En una evaluación dada con los datos de test se contó con un total de 270 imágenes las cuales determinaron los resultados en una matriz de confusión que están representado en la ilustración 49, y 270 imágenes normalizadas para realizar otra prueba de evaluación frente los resultados de la prueba de test.

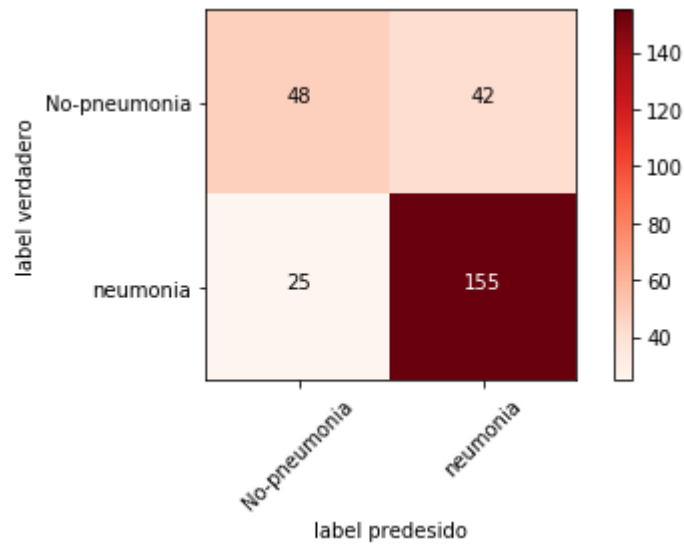


Ilustración 49 Matriz de confusión con los datos de test. Fuente propia.

Luego de haber obtenido todos los resultados dados por los campos de la matriz de confusión con el mayor balance de resultados en la evaluación de test, se pudo definir un reporte de clasificación como se muestra en la ilustración 50, en términos de precisión, exactitud, sensibilidad y f1, en comparación con los demás resultados se puede establecer que estos son los más altos con estos parámetros dados.

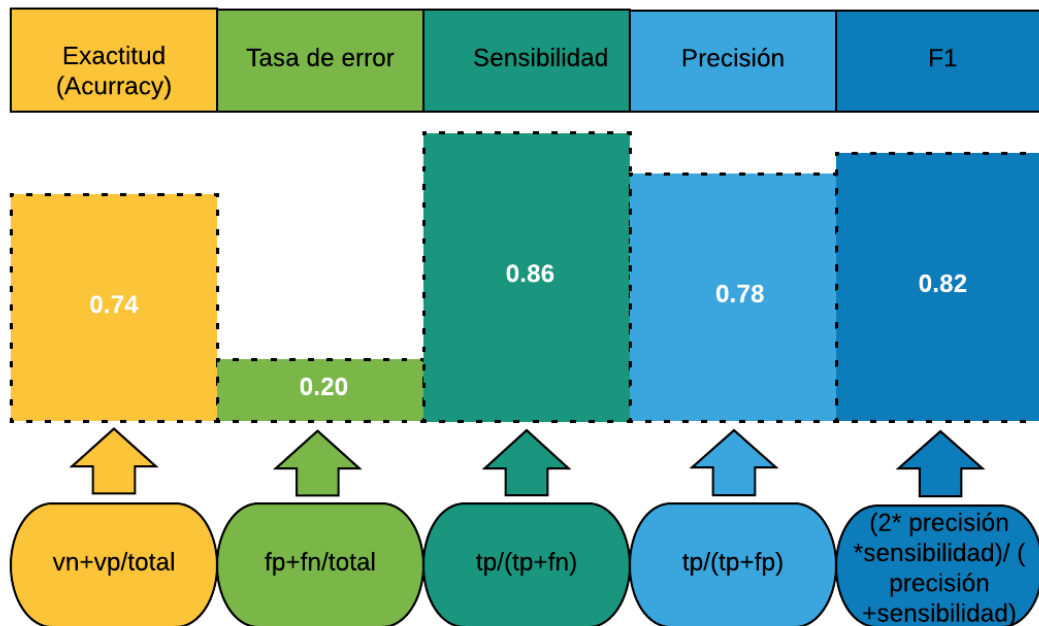


Ilustración 50 Reporte de clasificación con los datos de test. Fuente propia

3.5 Validación del Modelo

Para la validación del modelo se reservó un conjunto de datos con el objetivo de probar el modelo con ellos y determinar un análisis según sus resultados dados por la métrica frente las métricas dadas con el test.

3.6 Pruebas del modelo

Del conjunto de datos serializados con h5py se reservó un conjunto de datos igual a 270 imágenes de 256 alto con 256 ancho y un solo canal de color con su respectivo labels (etiqueta) asignado, el objetivo de preservar este grupo de imágenes es para realizar evaluaciones dadas con ajustes de parámetros de modelo para analizar e inferir en base a los resultados que fueron dados en el test y tomar como forma de medir la capacidad del modelo para generalizar.

A estos datos también se les realizo el mismo procedimiento (cargar datos y normalización) como se les realizo a los datos de test

En la ilustración 51, se puede observar los parámetros que fueron definidos para la evaluación con los datos reservados para evaluación.

learning rate	batch_size	Épocas	pasos	cantidad de datos de evaluación	cantidad de datos de entrenamiento	Precisión	exactitud	sensibilidad	F1	TP	TN	FP	FN
0.0001	32	50	10	270	630	0.77	0.74	0.87	0.81	166	31	49	24
0.0001	32	80	10	270	630	0.79	0.74	82	0.80	156	39	41	34
0.0001	32	100	10	270	630	0.80	0.72	0.81	0.80	154	43	37	36
0.0002	32	100	10	270	630	0.83	0.71	0.78	0.81	150	51	29	40
0.0002	40	100	10	270	630	0.78	0.73	0.80	0.79	153	38	42	37
0.0002	32	150	10	270	630	0.79	0.72	0.85	0.82	162	39	41	28
0.0001	32	180	10	270	630	0.75	0.70	0.91	0.82	155	39	41	35
0.0001	32	150	10	270	630	0.79	0.74	0.82	0.81	157	40	40	33

Ilustración 51 Tabla diferentes parámetros aplicados para evaluación. Fuente propia.

3.6 Matriz de confusión de los datos de evaluación

Los resultados reflejados en la ilustración 52, expresan el comportamiento dado en un matriz de confusión con los datos de evaluación y con los mismos parámetros que fueron aplicados en el test los cuales se acercaron al mejor resultado posible con este conjunto de imágenes, Los mejores parámetros de ajuste que dan como resultado esta matriz de confusión es:

Learning rate = 0.0001

Bacht size = 32

Pasos = 12

Epocas = 150

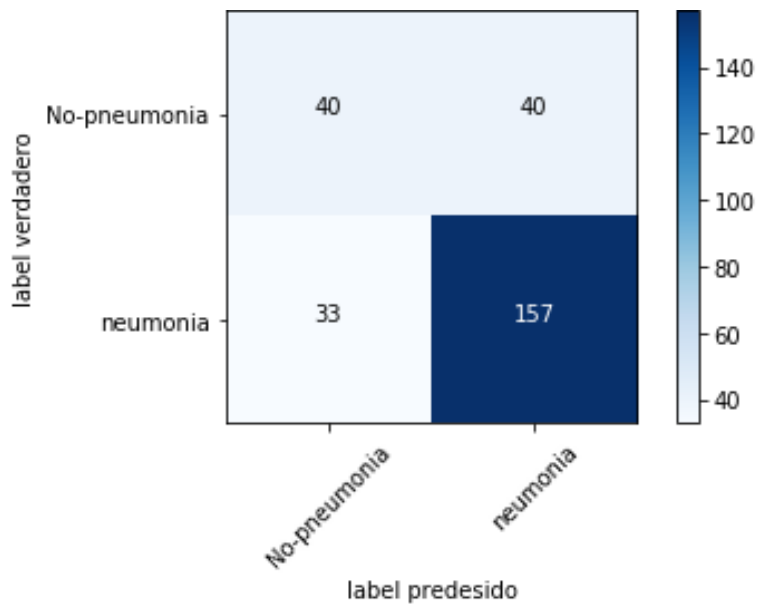


Ilustración 52 Matriz de confusión con datos de evaluación. Fuente propia.

IV RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se exponen los resultados obtenidos después de haber realizado todas las etapas de construcción del Modelo de Red Neuronal para el diagnóstico de Neumonía, se encuentran las observaciones finales y el resultado del mejor ajuste del modelo dado, para el conjunto de datos, con el que fue entrenado y que recomendaciones son pertinentes para llegar a un mejor resultado por parte de las métricas.

4.1 Resultados

Los resultados finales para presentar son un modelo de Redes Neuronales Convolucionales para la detección de neumonía con las características y ajustes de parámetros de entrenamiento que se pueden observar en la ilustración 53, además un gráfico del modelo con sus ajustes por medio de tensorboard.

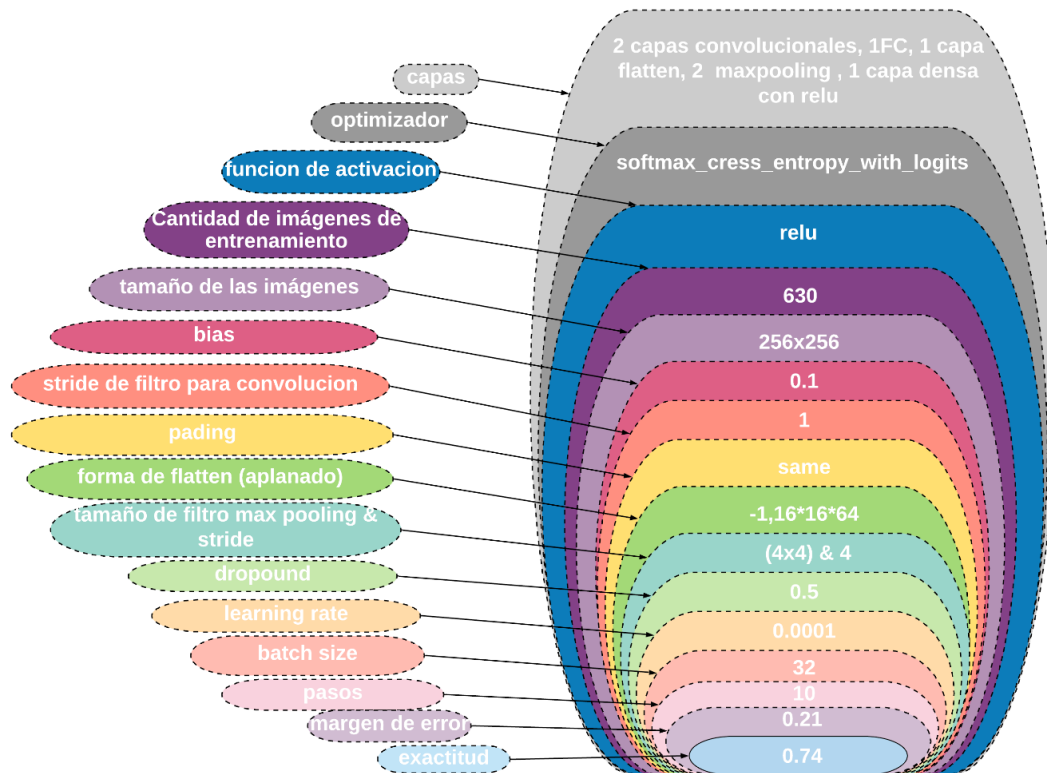


Ilustración 53 Tabla de resultados. Fuente propia.

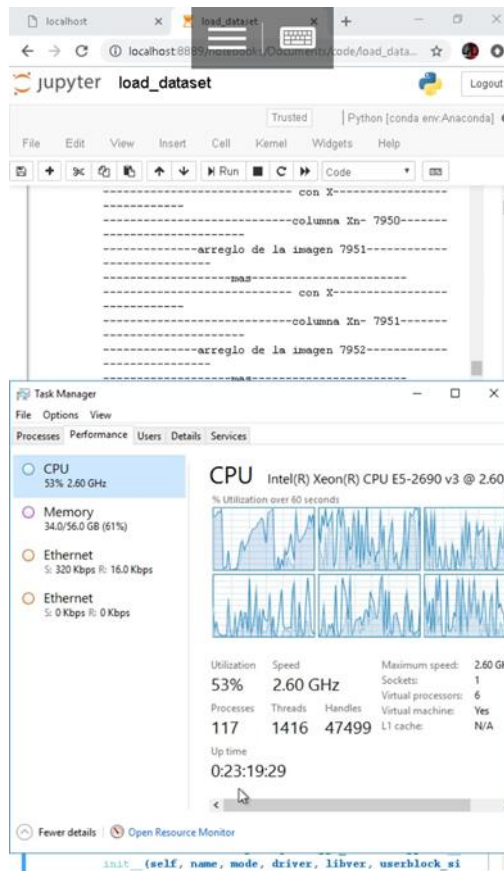


Ilustración 54 Características de equipo computacional requerido. Fuente propia.

La ilustración 54, muestra el poder de cómputo con el que estuvo trabajando la máquina.

4.2 Conclusiones

Al finalizar el trabajo de investigación, se puede concluir que a pesar de que existen diferentes medidas para identificar, precaver y tratar la neumonía, no ha sido suficiente para detener o disminuir las muertes infantiles entre los menores de 5 años, por ello se crea la posibilidad de desarrollar un modelo de machine learning usando Redes Neuronales Convolucionales (CNN) para identificar neumonía en imágenes radiológicas con formato DICOM.

Se logró obtener como resultado la construcción de un algoritmo de Machine Learning en base a una métrica de evaluación representada en términos de

porcentaje con una exactitud superior al 73% y con un porcentaje de error inferior al 21%; para llegar al cumplimiento de este objetivo se cumplieron los tres objetivos específicos de la siguiente manera:

- Se construyó un conjunto de datos para lo cual se utilizó el algoritmo “create_dataset”, encargado de obtener los datos de interés y crear la persistencia de estos mismos.
- Se entrenó la red neuronal con el conjunto de datos de entrenamiento que luego dio paso a la evaluación del modelo ya entrenado para esperar los resultados dados en un reporte de clasificación y una matriz de confusión.
- La etapa de evaluación fue un trabajo en conjunto con la etapa de test ya que luego de observar los resultados del test fue posible analizar resultados para determinar el mejor comportamiento de los parámetros en entrenamiento
- Finalmente, se describen los resultados de las pruebas de interfaz y funcionales del sistema realizadas en conjunto con el usuario principal del sistema, lo cual permitió validar los requerimientos identificados en la fase de análisis.

4.2 Recomendaciones

En el transcurso de la investigación y desarrollo del Modelo de Red Neuronal Convolutiva para el diagnóstico de Neumonía, se fueron dando inconvenientes técnicos, los cuales se llegaron a minimizar y así no afectar a la ejecución de los objetivos, dentro de los inconvenientes está integrado una deficiencia de equipo computacional (hardware) la cual dio como observación que es recomendable adquirir equipo de cómputo con capacidad de memoria RAM superior a 56gb, 500gb de almacenamiento y cualquier procesador de una DSVM (Data Science Virtual

Machine) de azure con capacidad para trabajar en paralelo, dada recomendación con el objetivo que llevar a cabo este proyecto a un escalón más alto.

Si este proyecto llega a escalar a nivel de hardware es recomendable analizar el balance de los datos para tomar la decisión si es necesario implementar un balanceo de datos con la técnica “smote” de la librería “imblearn.over_sampling”.

Bibliografía

- [1] O. M. D. L. S. OMS. [En línea]. Available: https://www.who.int/mediacentre/news/releases/2014/child_mortality_estimates/es/. [Último acceso: 15 MARZO 2019].
- [2] «www.kaggle.com,» [En línea]. Available: <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge>. [Último acceso: 09 11 2018].
- [3] DANE, «DIEZ PRIMERAS CAUSAS DE DEFUNCIÓN, SEGÚN DEPARTAMENTO DE RESIDENCIA,» Santa Fe de Bogotá, 2016.
- [4] H. C. U. d. S. d. Compostela, «NEUMO EXPERTOS EN PREVENCIÓN,» Hospital Clínico Universitario de Santiago de Compostela, [En línea]. Available: <https://neumoexpertos.org/2017/05/17/pronostico-y-mortalidad-de-la-neumonia/>.
- [5] OMS, «<https://www.who.int/es/news-room/fact-sheets/detail/pneumonia>,» 07 11 2016. [En línea]. Available: <https://www.who.int/es/news-room/fact-sheets/detail/pneumonia>. [Último acceso: 5 11 2018].
- [6] E. TIEMPO, «EL TIEMPO - BLOGS,» ELTIEMPO.COM, [En línea]. Available: <http://blogs.eltiempo.com/motor-de-busqueda/2017/03/05/la-crisis-del-sistema-de-salud/>.
- [7] S. R. d. A. d. Norte, «kaggle.com,» Sociedad Radiologica de America del Norte, 2018. [En línea]. Available: <https://www.kaggle.com/c/rsna-pneumonia-detection-challenge>. [Último acceso: 15 enero 2019].
- [8] clustersalud.americaeconomia.com, «americaeconomia.com,» clustersalud.americaeconomia.com, 2018. [En línea]. Available: <https://clustersalud.americaeconomia.com/insumos-y-servicios-hospitalarios/5-problemas-del-sector-salud-que-la-inteligencia-artificial-puede-resolver>. [Último acceso: 13 12 2018].
- [9] «observatorio-ia,» observatorio-ia, [En línea]. Available: <http://observatorio-ia.com/chexnet-ia-de-stanford-diagnostica-neumonia>. [Último acceso: 14 09 2018].
- [10] C. d. D. y. D. d. I. Informática, «coddii.org,» La opinion de Murcia, [En línea]. Available: <https://coddii.org/inteligencia-artificial-en-el-proceso-de-diagnostico-del-cancer-de-prostata>. [Último acceso: 20 Septiembre 2018].

- [11 nmas1.org/, «nmas1.org/,» [En línea]. Available:
] <https://nmas1.org/news/2018/09/18/algorithm-Google-detecta-cancer>. [Último
acceso: 10 octubre 2018].
- [12 elcomercio.pe/, «elcomercio.pe/,» [En línea]. Available:
] [https://elcomercio.pe/tecnologia/ciencias/salud-crean-herramienta-utiliza-
inteligencia-artificial-detectar-enfermedades-oculares-neumonia-noticia-499508](https://elcomercio.pe/tecnologia/ciencias/salud-crean-herramienta-utiliza-inteligencia-artificial-detectar-enfermedades-oculares-neumonia-noticia-499508).
[Último acceso: 18 ENERO 2019].
- [13 A. C. d. Infectología, «Recomendaciones para el diagnóstico, tratamiento y prevención de
] la neumonía adquirida en la comunidad en adultos inmunocompetentes,» *Infectio*, vol.
351, p. 38, 2013.
- [14 buzzword.com.m, «buzzword.com.m,» [En línea]. Available:
] <https://buzzword.com.mx/noticia.aspx?idNoticia=6&categoria=1>. [Último acceso: 04
Nnoviembre 2018].
- [15 unipython.com, «unipython.com,» [En línea]. Available: [https://unipython.com/deep-
\] learning-aprendizaje-profundo/](https://unipython.com/deep-learning-aprendizaje-profundo/). [Último acceso: 28 Noviembre 2018].
- [16 t-systemsblog, «t-systemsblog,» [En línea]. Available: [https://www.t-
\] systemsblog.es/redes-neuronales/](https://www.t-systemsblog.es/redes-neuronales/). [Último acceso: 28 Noviembre 2018].
- [17 a.mathworks, «la.mathworks.com,» [En línea]. Available:
] [https://la.mathworks.com/solutions/deep-learning/convolutional-neural-
network.html](https://la.mathworks.com/solutions/deep-learning/convolutional-neural-network.html). [Último acceso: 18 octubre 2018].
- [18 salesforce.com.mx, «salesforce,» [En línea]. Available:
] <https://www.salesforce.com/mx/blog/2017/6/Que-es-la-inteligencia-artificial.html>.
[Último acceso: 18 Noviembre 2018].
- [19 M. Azure, «docs.microsoft.com,» [En línea]. Available: [https://docs.microsoft.com/en-
\] us/azure/machine-learning/data-science-virtual-machine/overview](https://docs.microsoft.com/en-us/azure/machine-learning/data-science-virtual-machine/overview). [Último acceso: 28
Noviembre 2018].
- [20 desdelinux, «desdelinux.net,» [En línea]. Available: [https://blog.desdelinux.net/ciencia-
\] de-datos-con-python/](https://blog.desdelinux.net/ciencia-de-datos-con-python/). [Último acceso: 28 Noviembre 2018].
- [21 jupyter.org, «<https://jupyter.org/>,» [En línea]. Available: <https://jupyter.org/>. [Último
] acceso: 28 Noviembre 2018].
- [22 revistas.ucm.es, «<http://revistas.ucm.es/>,» [En línea]. Available:
] <http://revistas.ucm.es/index.php/RGID/article/viewFile/54719/49922>. [Último
acceso: 28 Noviembre 2018].

- [23 M. Azure. [En línea]. Available: <https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview>. [Último acceso: 15 ABRIL 2019].
- [24 «nih.gov,» [En línea]. Available: <https://www.nih.gov/>.
]
- [25 «engmrk.com,» adam-optimization-algorithm, [En línea]. Available:
] <https://engmrk.com/adam-optimization-algorithm/>.
- [26 «data-speaks.luca-d3.com,» data-speaks.luca-d3.com, [En línea]. Available: <https://data-speaks.luca-d3.com/2018/01/ML-a-tu-alcance-matriz-confusion.html>.
]