



Benemérita Universidad Autónoma de Puebla

---

Facultad de Ciencias Físico Matemáticas

---

Medición de concentraciones de hierro en tejido hepático  
con resonancia magnética

Tesis presentada al

**Posgrado en Física Aplicada**

como requisito parcial para la obtención del grado de

**MAESTRO EN CIENCIAS**

por

Gerardo Uriel Perez Rojas

Asesorado por

Dr. Benito de Celis Alonso

Dr. Eduardo Moreno Barbosa

Puebla Pue.  
1 de julio de 2024





Benemérita Universidad Autónoma de Puebla

---

Facultad de Ciencias Físico Matemáticas

---

Medición de concentraciones de hierro en tejido hepático  
con resonancia magnética

Tesis presentada al

**Posgrado en Física Aplicada**

como requisito parcial para la obtención del grado de

**MAESTRO EN CIENCIAS**

por

Gerardo Uriel Perez Rojas

Asesorado por

Dr. Benito de Celis Alonso

Dr. Eduardo Moreno Barbosa

Puebla Pue.  
1 de julio de 2024



**Título:** Medición de concentraciones de hierro en tejido hepático con resonancia magnética

**Estudiante:** GERARDO URIEL PEREZ ROJAS

COMITÉ

---

Dr. Javier Miguel Hernández López  
Presidente

---

Dr. José Gerardo Suárez García  
Secretario

---

Dr. Jorge Velazquez Castro  
Vocal

---

Dr. Benito de Celis Alonso  
Asesor

---

Dr. Eduardo Moreno Barbosa  
Coasesor



A mis padres; Gerardo y Elvira



# Agradecimientos

Agradezco al Dr. Benito de Celis Alonso y al Dr. Eduardo Moreno Barbosa por permitirme ser su estudiante de maestría, han sido 2 años arduos y llenos de trabajo que sin su tutela no podría haberlo logrado.

Al Dr. José Gerardo Suárez por apoyarme en la transición entre la ya alejada licenciatura y la etapa de la maestría ya que son etapas completamente distintas, me enseñó a ser autodidacta, entregar propuestas de soluciones a problemas presentados en el desarrollo del proyecto y que uno tiene que prepararse para cualquier escenario posible en el futuro.

Al Dr. Moisés Arredondo, me enseñó lo que una clase creada con cautela y bajo un plan de estudios puede dar como frutos, su trabajo como docente es admirable a mi parecer, lo reconozco por haberme puesto a prueba y haberme enseñado lo que significa estar en un posgrado.

Al Dr. Jorge Velazquez Castro por resolver mis dudas y darme muchos consejos con respecto a mi trabajo de tesis, lo conozco desde la licenciatura y siempre he admirado su forma de ser y su manera de explicar las cosas.

Al Dr. Hector Novales, por abrirme las puertas de su oficina cada vez que necesitaba escuchar un consejo, ir por un café o al menos platicar de algo no relacionado a lo laboral.

A la abogada, notaria y actuaria Abigail Ortiz por escucharme hasta altas horas de la noche con lo que realizaba en el posgrado, por aconsejarme, cuidarme, curarme, tenerme paciencia y ser ese soporte emocional que no sabía que necesitaría pero ahí estaba.

A mis padres, por respetar mis decisiones por muy arriesgadas que sean, cuidarme y apoyarme en el proceso sin importar los resultados pero con la garantía de que creen en mí. En el vasto infinito del universo, las probabilidades dieron como resultado unos papás que no podría pedirlos de una forma diferente, los quiero mucho.

Finalmente a todos mis amigos, Carolina, Paola, Miguel, Gustavo y todos los demás a los que no podría mencionar pues me llevaría al menos 5 hojas; todos ustedes son una parte vital de mi vida, han dejado huella en mí y han trascendido en lo que ahora soy. Además con mención especial quiero agradecer al ahora M.C. Eduardo Rodríguez, por haber estado conmigo en esta aventura llamada maestría, enfrentar retos, compartir emociones y poder apoyarnos en nuestra vida profesional. El doctorado nos espera con nuevas aventuras.

Agradezco también al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) por el apoyo económico otorgado para la realización de mis estudios de maestría.



# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Marco teórico</b>	<b>9</b>
2.1. Resonancia Magnética (RM) y su Aplicación en Medicina	9
2.1.1. Principios básicos de la resonancia magnética	9
2.1.2. Elementos que componen la RM	12
2.1.3. Conceptos de detección y métodos de adquisición de señales.	15
2.1.4. Espacio “k” y secuencia eco de gradiente	24
2.2. Tejido hepático y su Relación con el Hierro	29
2.2.1. Anatomía y fisiología del hígado	29
2.2.2. Funciones del hierro en el organismo	30
2.2.3. Pérdida y acumulación de hierro en el tejido hepático: causas y consecuencias	31
2.3. Diagnóstico de Sobrecarga de Hierro en el Hígado	32
2.3.1. Métodos tradicionales de medición de hierro hepático	32
2.3.2. Limitaciones y desafíos en la medición de hierro hepático	33
2.4. Resonancia Magnética en la Medición de Hierro Hepático	34
2.4.1. Secuencias de RM utilizadas para medir hierro hepático (relajometría)	35
2.5. Inteligencia Artificial (IA), Machine Learning(ML) y Aprendizaje Profundo (Deep Learning)	36
2.5.1. Tipos de Machine Learning	37
2.5.2. Proceso de Machine Learning: Conceptos y consideraciones	38
2.5.3. Regresión lineal en ML	47
2.5.4. Redes Neuronales Artificiales (RNA)	49
2.5.5. El descenso de gradiente en Deep Learning	53
2.5.6. Proceso de entrenamiento para una Red Neuronal Artificial	56
2.5.7. Funciones de activación	59
2.5.8. Regularización	61
2.5.9. Optimizadores	64
2.5.10. Redes Neuronales Convolucionales (RNC)	66
2.6. Uso de RNA para diagnóstico clínico, casos de estudio con problemas que involucran hierro	70
2.6.1. La Integración de Redes Neuronales en la medición de hierro por RM	71
2.7. Métodos estadísticos para interpretación de resultados	72
2.7.1. Pruebas de distribución	72
2.7.2. Pruebas de correlación y test de prueba	72
2.7.3. Gráfico de Bland-Altman	74

<b>3. Métodos</b>	<b>75</b>
3.1. Base de datos y preprocesamiento . . . . .	75
3.1.1. Corrección de inhomogeneidades . . . . .	75
3.1.2. Normalización . . . . .	78
3.2. Diseño e implementación de red neuronal convolucional . . . . .	78
3.2.1. Arquitectura de RNC . . . . .	79
<b>4. Resultados</b>	<b>83</b>
4.1. Resultados de los métodos estadísticos . . . . .	83
<b>5. Discusión</b>	<b>97</b>
5.1. Normalización y corrección de inhomogeneidades . . . . .	97
5.2. Arquitecturas de RNC . . . . .	97
5.3. Métricas obtenidas de Bland-Altman . . . . .	98
<b>6. Conclusiones</b>	<b>101</b>
<b>A. Ecuaciones de Bloch</b>	<b>103</b>
<b>B. Transformada de Fourier</b>	<b>105</b>
<b>C. Regresión lineal</b>	<b>107</b>
C.1. Regresión múltiple . . . . .	108
<b>D. Código de RNC</b>	<b>111</b>
<b>Bibliografía</b>	<b>121</b>

# Índice de figuras

2.1. Efectos de gradientes. (a) Campo magnético $\mathbf{B}_0$ haciendo que los núcleos hagan movimientos de precesión a la misma frecuencia. (b) Campo magnético $\mathbf{B}_0$ con un gradiente añadido a lo largo del eje x, la frecuencia de precesión hace dependencia de la posición ahora. 2.1 . . . . .	12
2.2. Efectos de gradientes en la señal de RM. Se tendrá una señal que proviene de diversas posiciones con cambios de fase dependientes de la posición. [5] . . . . .	14
2.3. Bucles de hilos que experimentan cambios de flujo con una dependencia temporal: (a) Una bobina cuadrada estacionaria con un campo aplicado oscilando en un eje fijo. (b) Una bobina circular girando en un campo estático. (C) Una bobina estacionaria sumergida en un campo giratorio de magnitud fija (ejemplo relevante para mediciones de RM) [6] . . . . .	16
2.4. Desplazamientos en frecuencia de las señales procedentes de una muestra uniforme debidos a la demodulación y el filtrado, con un desplazamiento $\delta\omega$ . Las líneas gruesas representan la señal en las frecuencias indicadas. [6] . . . . .	19
2.5. (a) Señal FDI con espines precesando a la frecuencia de Larmor. (b) Condición demodulada. (C) Señal FDI demodulada con demodulación en una frecuencia diferente a la de Larmor. (d) Señal FID demodulada total de varios espines, cada uno con una frecuencia de Larmor variada, mostrando un decaimiento con oscilaciones lentas y un decaimiento más rápido que en $T_2$ . [6] . . . . .	21
2.6. Pulso de RF de $\frac{\pi}{2}$ aplicado al eje x en $t = 0$ y un pulso de RF de $\pi$ a lo largo del eje y en $t = \tau$ para invertir la fase acumulada por los espines. El eco muestra un máximo local positivo, en lugar de un mínimo local negativo después de invertir fases. Los subíndices indican el eje a lo largo del cual se aplica el pulso de RF. [6] . . . . .	23
2.7. Conjunto de espines simulados durante el método de eco-spin. Un pulso $\frac{\pi}{2}$ rota los espines alrededor del eje x, hacia el plano transversal generando un movimiento de precesión. Al acumular una fase extra, se invierte una vez mandado otro pulso de amplitud $\pi$ , al seguir acumulando fase y volver al eje “y” positivo al mismo ritmo, se generará un eco. [6] . . . . .	23
2.8. Gráficas de señal y tiempo para una RM de una dimensión, el campo magnético colocado a la izquierda del cilindro hace referencia al campo que aparece en el intervalo de tiempo $(t_3, t_4)$ , los espines isocronos están en un plano igual, pese a las diferencias en sus coordenadas en z, la figura también muestra los gradientes positivo y negativo. [6] . . . . .	27
2.9. Espacio k para secuencia eco de gradiente de forma básica en (a). mientras que en (b) se muestra la secuencia eco-spin donde los gradientes son positivos. [6] . . . . .	28
2.10. El espacio k para una RM de prueba en la formación de imágenes FID. El tiempo $t''$ se define como $t'' = t - t_1$ y comienza desde que los gradientes aparecen. El tiempo total de la adquisición de señales se toma como $T_s$ . Solo se considera la mitad del espacio k y no hay progresión a través del espacio k hasta el tiempo $t_1$ . [6] . . . . .	28

2.11. Anatomía hepática. Tomado de <i>Anatomía hepática</i> , por Mèdica Capacitaciòn, 2017, <a href="https://diplomadomedico.com/anatomia-hepatica/">https://diplomadomedico.com/anatomia-hepatica/</a> . . . . .	29
2.12. Segmentaciòn del hígado. Tomado de <i>Segmentaciòn hepática. Evaluaciòn ecogràfica</i> , por Mèdica Capacitaciòn, 2016, <a href="https://diplomadomedico.com/segmentacion-hepatica-evaluacion-ecografica-2/">https://diplomadomedico.com/segmentacion-hepatica-evaluacion-ecografica-2/</a> . . . . .	30
2.13. Absorciòn de hierro por enterocitos en el duodeno. Tomado de <i>Metabolismo del hierro: Absorciòn, transporte, reciclado y almacenamiento</i> , por CardioTeca, 2023, <a href="https://www.cardioteca.com/metabolismo-del-hierro.html">https://www.cardioteca.com/metabolismo-del-hierro.html</a> : :text=El%20metabolismo%20del%20hierro%20incluye,la%20degradaci%C3%B3n%20de%20los%20eritrocitos . . . . .	31
2.14. Esquema de una biopsia hepática percutánea. [11] . . . . .	33
2.15. Sobrecarga de hierro en (a) el hígado de un especimen de rata con 1.8mg de hierro por gramo de masa, se presenta deposiciòn en los hepatocitos como granulos negros. (b) Imagenes hígado en técnica de RM de 4 especimenes de control (C) y 2 con sobrecarga de hierro (I1,I2). Los contenidos de hierro en I1 es de 3.9mg y en I2 de 4.0mg, se observa una diferencia de intensidades significativa entre los especimenes de control y los que tienen excesos de hierro. [13] . . . . .	34
2.16. Esquema de campos necesarios para el desarrollo de IA. . . . .	36
2.17. Esquema de algoritmo para aprendizaje supervisado. . . . .	37
2.18. Esquema de algoritmo para aprendizaje no supervisado. . . . .	38
2.19. Esquema de algoritmo para aprendizaje semi-supervisado. . . . .	38
2.20. Esquema de algoritmo para aprendizaje por refuerzo. . . . .	39
2.21. Proceso de Machine Learning. . . . .	39
2.22. Conjunto de datos en crudo. . . . .	41
2.23. Subajuste, ajuste "óptimo" y sobreajuste . . . . .	43
2.24. Diagrama de Diana que muestra los errores a considerar en un modelo con base en un modelo "perfecto" . . . . .	43
2.25. Gráfico Complejidad de modelo vs. Error. El crecimiento del error en el conjunto de prueba aparece cuando un modelo es robusto . . . . .	44
2.26. Regresiòn lineal simple . . . . .	48
2.27. Regresiòn lineal multiple . . . . .	48
2.28. Neurona motora. . . . .	50
2.29. Modelo matemático simplificado de una neurona. Este modelo es también conocido como una red neuronal artificial de una sola capa . . . . .	51
2.30. Grafo DAG de una Red Neuronal Artificial (RNA), con sus capas ocultas completamente conectadas. . . . .	52
2.31. Proceso de entrenamiento de una RNA. . . . .	56
2.32. Funciòn sigmoide. . . . .	60
2.33. Funciòn tangente hiperbólica. . . . .	60
2.34. Funciòn ReLU y su derivada. . . . .	62
2.35. Dropout en un modelo de RNA. A la izquierda, una RNA estándar con 2 capas ocultas. A la derecha, un ejemplo de una red que aplicó dropout al modelo de la izquierda, algunas conexiones cruzadas han sido eliminadas. [24] . . . . .	62
2.36. Comparaciòn de las operaciones basicas entre una RNA estándar y una con Dropout. [24] . . . . .	63
2.37. Matriz de entrada (izquierda) y kernel (derecha)[29] . . . . .	68
2.38. Matriz de entrada con el kernel en convoluciòn (izquierda) y la matriz convolucionada (derecha) en convoluciòn tipo valid[29] . . . . .	68
2.39. Ejemplo de la operaciòn <i>Average Pooling</i> [31] . . . . .	69
2.40. Arquitectura de RNA para el diagnóstico de anemia. [32] . . . . .	70

2.41. Variación de observaciones mediante comparación de segmentación de los 2 expertos y el modelo de DL. En (a) un diagrama de cajas que presenta las métricas de segmentación y en (b) diagramas de diana que muestran las diferencias en segmentación de acuerdo al modelo de la <i>American Health Association</i> (paneles izquierdos) y el índice de correlación entre los expertos y el modelo de DL (paneles derechos)[34] .	71
3.1. RM de un paciente con secuencia eco-gradiente en tiempo de relajación $T_1$ , en corte transversal. . . . .	76
3.2. RM de un paciente sin corrección de inhomogeneidades . Visualización desde el software ITK-Snap. [37] . . . . .	77
3.3. Resultado de la aplicación del algoritmo N4 para corrección de inhomogeneidades.	78
3.4. Arquitectura de RNC inicial, se muestran las imágenes de entrada, capas de convolución, pooling y red neuronal completamente conectada. . . . .	80
3.5. Arquitectura de RNC inicial en forma de diagrama de flujo, para esta arquitectura se tienen en la capa de la RNA (capa completamente conectada) 3852288 características.	81
4.1. Prueba de Smirnov-Kolmogorov. . . . .	84
4.2. Gráfico de dispersión y Bland-Altman para una capa de convolución y 10 neuronas en la capa completamente conectada. . . . .	85
4.3. Gráfico de dispersión y Bland-Altman para una capa de convolución y 15 neuronas en la capa completamente conectada. . . . .	86
4.4. Gráfico de dispersión y Bland-Altman para una capa de convolución y 20 neuronas en la capa completamente conectada. . . . .	86
4.5. Gráfico de dispersión y Bland-Altman para una capa de convolución y 25 neuronas en la capa completamente conectada. . . . .	87
4.6. Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 10 neuronas en la capa completamente conectada. . . . .	87
4.7. Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 15 neuronas en la capa completamente conectada. . . . .	88
4.8. Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 20 neuronas en la capa completamente conectada. . . . .	88
4.9. Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 25 neuronas en la capa completamente conectada. . . . .	89
4.10. Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 10 neuronas en la capa completamente conectada. . . . .	89
4.11. Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 15 neuronas en la capa completamente conectada. . . . .	90
4.12. Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 20 neuronas en la capa completamente conectada. . . . .	90
4.13. Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 25 neuronas en la capa completamente conectada. . . . .	91
4.14. Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 10 neuronas en la capa completamente conectada. . . . .	91
4.15. Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 15 neuronas en la capa completamente conectada. . . . .	92
4.16. Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 20 neuronas en la capa completamente conectada. . . . .	92
4.17. Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 25 neuronas en la capa completamente conectada. . . . .	93
4.18. Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 10 neuronas en la capa completamente conectada. . . . .	93

4.19. Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 15 neuronas en la capa completamente conectada. . . . .	94
4.20. Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 20 neuronas en la capa completamente conectada. . . . .	94
4.21. Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 20 neuronas en la capa completamente conectada. . . . .	95
4.22. Diagrama de cajas, comparación de etiquetas reales y predicciones. . . . .	95

# Índice de tablas

4.1. Tabla de configuraciones de RNC, coeficiente de Pearson, valor p, significancia y d de Cohen . . . . .	83
4.2. Tabla de métricas obtenidas de los gráficos de Bland-Altman para cada arquitectura de red neuronal . . . . .	85



# Capítulo 1

## Introducción

### Resumen

En el campo de la medicina, la cuantificación precisa del contenido de hierro en el hígado desempeña un papel crucial en el diagnóstico y seguimiento de diversas enfermedades hepáticas. El exceso o la deficiencia del hierro pueden ser indicadores de patologías como la hemocromatosis hereditaria, la anemia ferropénica y la cirrosis, entre otras. Por lo tanto, contar con métodos no invasivos y precisos para evaluar el contenido de hierro en el hígado se ha convertido en un desafío importante en la práctica clínica.

Actualmente, los métodos tradicionales de cuantificación de hierro hepático, como la biopsia hepática invasiva, presentan limitaciones significativas. Estos procedimientos son dolorosos, costosos y están asociados con riesgos potenciales para los pacientes. Como resultado, ha surgido la necesidad de desarrollar enfoques no invasivos que ofrezcan resultados precisos y confiables.

La resonancia magnética (RM) ha emergido como una técnica para la evaluación no invasiva del contenido de hierro en el hígado. La RM proporciona imágenes detalladas del órgano, permitiendo una evaluación precisa de las concentraciones de hierro en diferentes regiones hepáticas. Sin embargo, el análisis manual de las resonancias magnéticas para cuantificar el hierro es una tarea compleja y sujeta a errores inter e intraobservador, lo que limita su aplicabilidad clínica.

Por otro lado, las redes neuronales convolucionales (CNN, por sus siglas en inglés) son una clase de algoritmos de aprendizaje profundo que han demostrado un rendimiento sobresaliente en el procesamiento de análisis de imágenes. Estas redes tienen la capacidad de extraer características relevantes de las imágenes de resonancia magnética y aprender patrones complejos, lo que las convierte en una herramienta prometedora para mejorar la cuantificación precisa del hierro hepático.

El objetivo de este trabajo de tesis es desarrollar y evaluar un modelo de CNN para la cuantificación precisa del contenido de hierro en el hígado a partir de resonancia magnética. Se propone utilizar un enfoque basado en el uso de CNN para automatizar y mejorar la precisión de la cuantificación del hierro, superando las limitaciones de los métodos tradicionales.

La metodología a seguir en esta investigación incluirá la recopilación de un conjunto de datos de resonancia magnética hepática con información de referencia del contenido de hierro, el diseño e implementación de un modelo de CNN específicamente adaptado para el análisis de resonancia magnética hepáticas, la validación exhaustiva del modelo propuesto haciendo uso de estadísticos y métodos ofrecidos por los algoritmos de Machine Learning para la comparación entre los resultados obtenidos mediante un método estandarizado y el método propuesto.

El desarrollo de un enfoque automatizado y preciso para la cuantificación de hierro en hígado a partir de resonancia magnética podría tener un impacto significativo en el diagnóstico temprano y el seguimiento de enfermedades hepáticas relacionadas con el hierro, mejorando así la atención clínica y la calidad de vida de los pacientes.

**Palabras clave:** Resonancia magnética, Redes Neuronales Convolucionales (CNN), Hierro hepático, hemocromatosis

## Abstract

In the field of medicine, the precise quantification of liver iron content plays a crucial role in diagnosing and monitoring various liver diseases. Excess or deficiency of iron can be indicators of conditions such as hereditary hemochromatosis, iron deficiency anemia, and cirrhosis, among others. Therefore, having non-invasive and accurate methods to assess liver iron content has become an important challenge in clinical practice.

Currently, traditional methods for liver iron quantification, such as invasive liver biopsy, have significant limitations. These procedures are painful, costly, and associated with potential risks for patients. As a result, there is a need to develop non-invasive approaches that offer precise and reliable results.

Magnetic Resonance Imaging (MRI) has emerged as a technique for non-invasive assessment of liver iron content. MRI provides detailed images of the organ, allowing for precise evaluation of iron concentrations in different hepatic regions. However, the manual analysis of MRI scans to quantify iron is a complex task, subject to inter- and intra-observer errors, which limits its clinical applicability.

On the other hand, Convolutional Neural Networks (CNNs) are a class of deep learning algorithms that have demonstrated outstanding performance in image analysis processing. These networks have the ability to extract relevant features from MRI images and learn complex patterns, making them a promising tool for improving the precise quantification of liver iron.

The objective of this thesis is to develop and evaluate a CNN model for the precise quantification of liver iron content from MRI. The proposal is to use a CNN-based approach to automate and improve the accuracy of iron quantification, overcoming the limitations of traditional methods. The methodology for this research will include the collection of a dataset of hepatic MRI scans with reference information on iron content, the design and implementation of a CNN model specifically adapted for hepatic MRI analysis, and the exhaustive validation of the proposed model using statistical methods and machine learning algorithms to compare the results obtained with a standardized method and the proposed method.

The development of an automated and precise approach for liver iron quantification from MRI could have a significant impact on the early diagnosis and monitoring of liver diseases related to iron, thereby improving clinical care and the quality of life for patients.

**Keywords:** Magnetic Resonance Imaging, Convolutional Neural Networks (CNNs), Liver Iron, Hemochromatosis



# Hipótesis y objetivos

## Hipótesis

Las red neuronal convolucional(CNN) propuesta hace predicciones correctas de los valores de hierro medidos en el hígado. Esta red muestra resultados similares a los obtenidos mediante aplicaciones comerciales.

## Objetivo general

Cuantificar el hierro en tejido hepático haciendo uso de redes neuronales convolucionales aplicadas a imágenes de resonancia magnética en secuencia  $T_2^*$ .

## Objetivos específicos

- Realizar la búsqueda de las RN más utilizadas para problemas similares.
- Diseñar y justificar las CNN que se utilizará.
- Comparar las distintas correlaciones de cada CNN entre la predicción y los valores medidos mediante un método comercial, los cuales fueron usados como estándar de referencia.
- Alcanzar un valor de correlación mayor a 0.75 y que sea una correlación significativa.



# Resumen de capítulos

El trabajo de tesis está estructurado de la siguiente manera: En el capítulo 2 se presenta el marco teórico donde se describe de manera general el metabolismo del hierro, como es procesado en el cuerpo humano y bajo que condiciones existen los “excesos” o “deficiencias” de éste. En el capítulo 3 se profundiza sobre el diseño y estructura de las CNN a utilizar, así como los pre-procesamientos aplicados al conjunto de datos estudiados en este problema. En el capítulo 4 se presentan los resultados obtenidos de las CNN, las predicciones obtenidas y las variaciones entre arquitecturas, así como los recursos estadísticos empleados para determinar la eficacia de nuestra propuesta de CNN. El capítulo 5 discute los resultados de manera resumida, lo hallado y la interpretación de los mismos. Finalmente, se destina un apartado para presentar las conclusiones y trabajo futuro.



## Capítulo 2

# Marco teórico

### 2.1. Resonancia Magnética (RM) y su Aplicación en Medicina

La técnica de diagnóstico por resonancia magnética nuclear (NMR por sus siglas en inglés) o simplemente resonancia magnética (RM) revolucionó la forma de adquisición de imágenes médicas de diagnóstico de forma no invasiva. ¿Cuáles son los pasos que sigue esta técnica de adquisición de imágenes para realizar una toma de estas?

#### 2.1.1. Principios básicos de la resonancia magnética

Las Resonancias Magnéticas tienen como componentes principales a las antenas transmisoras y emisoras de radiofrecuencia, un imán superconductor y un conjunto de bobinas de gradiente de campo magnético. Para uso médico, de forma típica, se utiliza un magneto superconductor de 3 Teslas que interactúan con los protones del cuerpo humano provocándoles una precesión (rotación) la cual se ejecuta en una frecuencia proporcional a la fuerza del campo magnético, también conocida como “frecuencia de resonancia”. El hecho de tener el fenómeno de interacción del campo magnético con los protones hace que los cambios del campo magnético hagan que la frecuencia de resonancia dependa de la ubicación espacial de cada protón en el cuerpo humano, haciendo posible la creación de imágenes por esta diferencia en las frecuencias de cada protón.

Tomando en consideración lo anteriormente mencionado, la imagen formada de una RM es formada por las señales provenientes de los protones que se encuentran en los tejidos del cuerpo como el agua o la grasa del paciente.

A escala atómica, un protón es una partícula cargada la cual gira alrededor de un eje interno de rotación con un valor de momento angular ( $P$ ), también tiene momento magnético ( $\mu$ ), por lo que se puede considerar como una pequeña barra magnética que tiene un polo norte y un polo sur. [1] Asimismo, la magnitud del momento angular está cuantizado por un valor fijo. Utilizando lo anterior se tiene que la magnitud del momento magnético de un protón es proporcional a la magnitud del momento angular de la forma:

$$|\vec{\mu}| = \gamma |\vec{P}| \quad (2.1)$$

Siendo  $\gamma$  la constante de radio giromagnético, la cual tiene por valor  $267,54 \frac{MHz}{Tesla}$  por protones de hidrógeno (H). Como consecuencia, se tiene al momento magnético como una cuantización con un valor fijo.

Excluyendo la influencia del campo magnético de la RM, la magnitud del momento magnético de cada protón en el cuerpo humano permanece constante, aunque su dirección es completamente

aleatoria. En consecuencia, la magnetización total, que representa la suma de todos los momentos magnéticos individuales del cuerpo, se anula (haciendo caso omiso de los efectos pequeños del campo magnético terrestre).

De acuerdo con la mecánica cuántica, la componente del momento magnético en dirección al campo inducido ( $B_0$ ) puede tener solamente dos valores discretos, el cual resulta en los momentos magnéticos alienados a un ángulo de  $54.7^\circ$  con respecto a la dirección de  $B_0$  ya sea de forma paralela (misma dirección) o de forma anti-paralela (dirección contraria). Cabe destacar que las formas “paralela” y “anti-paralela” hacen referencia únicamente a la componente z de  $\mu$  y que en realidad  $\mu$  esta alineada a un ángulo con respecto a  $B_0$ .

El número de protones de manera paralela o anti-paralela dependen del valor del campo  $B_0$ , pues, los protones paralelos estarán en el estado de energía “favorable” en comparación con la configuración anti-paralela. La diferencia de energía entre estas dos configuraciones estará dada por:

$$\Delta E = \frac{\gamma h B_0}{2\pi} \quad (2.2)$$

Con “h” la constante de Plank.

Es necesario calcular el número relativo de protones, haciendo uso de la ecuación 2.2 y la ecuación de Boltzmann, se tiene la razón:

$$\frac{N_{anti-paralelos}}{N_{paralelos}} = \exp - \left[ \frac{\Delta E}{kT} \right] \quad (2.3)$$

Haciendo expansión en serie de Taylor hasta primer orden debido a que el valor del argumento de la exponencial es muy pequeño, tenemos:

$$\frac{N_{anti-paralelos}}{N_{paralelos}} = 1 - \left[ \frac{\Delta E}{kT} \right] \quad (2.4)$$

Notemos que la señal de RM depende de la diferencia entre ambos tipos de protones, haciendo que la ecuación 2.4 se vuelva:

$$N_{paralelos} - N_{anti-paralelos} = N \left[ \frac{\gamma h B_0}{4\pi kT} \right] \quad (2.5)$$

Con “N” el total de protones. Teniendo en consideración el número de protones alineados, la diferencia energética que tienen se puede realizar también el calculo de la precesión que estos tienen al momento de ser inducidos por un campo magnético externo  $B_0$ . Debido a una ausencia de perturbaciones externas, más protones tenderan a alinearse paralelamente con el campo magnético, lo que resulta de una magnetización neta en la dirección del campo magnetico; como el campo magnético se coloca en dirección al eje z, la magnetización neta será  $M_z$ , en virtud de la transitividad, se concluye que la diferencia entre protones paralelos y antiparalelos es la magnetización neta del eje z, la cual se utilizará y abordará más adelante con las ecuaciones de Bloch y los tiempos de relajación.

De manera clásica, cuando un campo  $B_0$  aparece, este intenta que los momentos magnéticos de los protones se alineen consigo mismo haciendo que se cree un torque que es la resultante del producto cruz de 2 campos magnéticos:

$$\vec{T} = \vec{\mu} \times \vec{B}_0 = |\mu| |B_0| \sin(\theta) \hat{e}_N \quad (2.6)$$

Con  $\hat{e}_N$  un vector unitario normal a los vectores  $\vec{\mu}$  y  $\vec{B}_0$ . Se observa que la dirección de  $\vec{T}$  es tangencial a la dirección de  $\vec{\mu}$  y eso genera un movimiento de precesión a los protones alrededor del eje del campo magnético, además, se mantiene un ángulo constante de  $54.7^\circ$  entre  $\vec{\mu}$  y  $\vec{B}_0$ . El movimiento de precesión de un protón también se puede calcular con el hecho de saber que el torque es la tasa de cambio del momento angular del proton con respecto a una variable temporal, es decir:

$$\vec{T} = \frac{d\vec{P}}{dt} = \vec{\mu} \times \vec{B}_0 \quad (2.7)$$

La magnitud de la componente del momento angular el cual tiene precesión en el plano perpendicular al campo  $\vec{B}_0$  es  $|\vec{P}|\sin(\theta)$ . Dado un instante de tiempo  $dt$ , el movimiento de precesión de  $\vec{\mu}$  es realizado a través de un ángulo diferencial  $d\phi$  generando así un cambio en  $d\vec{P}$  del momento angular. Se tiene la relación siguiente:

$$\sin(d\phi) = \frac{d\vec{P}}{|\vec{P}|\sin(\theta)} = \frac{\vec{C}dt}{|\vec{P}|\sin(\theta)} \quad (2.8)$$

Si el ángulo  $\phi$  es muy pequeño, se puede realizar la aproximación senoidal a únicamente el valor del argumento por lo que, la frecuencia de precesión angular o  $\omega$  se volverá:

$$\omega = \frac{d\phi}{dt} = \frac{\vec{T}}{|\vec{P}|\sin(\theta)} = \frac{\vec{\mu} \times \vec{B}_0}{|\vec{P}|\sin(\theta)} = \frac{\gamma|\vec{P}||\vec{B}_0|\sin(\theta)}{|\vec{P}|\sin(\theta)} = \gamma B_0 \quad (2.9)$$

Por lo tanto, un protón bajo los efectos de un campo magnético es realizar un movimiento de precesión a una frecuencia directamente proporcional a la fuerza que tenga el campo magnético, esta frecuencia es conocida como la frecuencia de Larmor la cual se ve como:

$$\omega_0 = \gamma B_0 \quad (2.10)$$

Además de la frecuencia de Larmor, utilizando la ecuación 2.2 y la relación De Broglie que indica una relación de la forma  $\Delta E = hf$  se tiene que:

$$f = \frac{\gamma B_0}{2\pi} \vee \omega = \gamma B_0 \quad (2.11)$$

Se observa que la ecuación de frecuencia de precesión 2.10 es similar con el campo de radiofrecuencia.

Ahora, es necesario mostrar varios elementos que sirvan para entender el funcionamiento de las RM.

### Magnetización transversal

De forma similar al análisis realizado para el movimiento de precesión de un protón, un pulso de radiofrecuencia (RF) aplicado a  $90^\circ$  de la dirección del campo  $B_0$  tendrá una componente magnética del pulso de RF llamada campo  $B_1$  produce un torque el cual hace que la magnetización neta rote en el plano x-y. La rotación generada en la magnetización neta viene en conjunto con un ángulo de inclinación, el cual es proporcional a la fuerza del campo aplicado de RF y su tiempo de aplicación llamado  $\tau_{B_1}$ , el ángulo se expresa de la forma:

$$\alpha = \gamma B_1 \tau_{B_1} \quad (2.12)$$

Un ángulo de inclinación de  $90^\circ$ , también conocido como pulso de  $90^\circ$ , resulta en el valor máximo de la componente magnética en el eje y, es decir  $M_y$ , a su vez, un ángulo de  $180^\circ$  no produce la magnetización de  $M_y$ , pero si rota la magnetización neta  $M_0$  del eje positivo en z a su parte negativa.

### 2.1.2. Elementos que componen la RM

Los elementos anteriormente descritos no son suficientes, pues, para generar la RM es necesario hablar de codificaciones entre espacio y frecuencia, gradientes, y demás elementos que la conforman.

Los protones de hidrogeno tienen spin el cual al ser expuestos ante un campo magnético  $\mathbf{B}_0$  se pueden excitar, esta excitación puede ocurrir mediante un campo que tendrá variabilidad con el tiempo, el campo de radiofrecuencia  $\mathbf{B}_1$  cuando este último es aplicado de manera perpendicular al campo  $\mathbf{B}_0$ . La RF que tiene una duración de apenas unos milisegundos afectará a los núcleos, provocando que tenga niveles altos de energía, haciendo que tengan que liberarla para volver a su estado óptimo de energía (o estado de relajación). La energía que absorben los núcleos generará una diferencia de potencial que afectará a una bobina de alambre (para un “decaimiento de inducción libre” o FID en inglés). Como los pulsos de RF tienen intervalos de aparición, el sistema comenzará a relajarse hasta que llega a un equilibrio térmico. En consecuencia, los núcleos comenzarán a resonar, pero, en secuencias distintas al momento de exponerlos de nuevo al campo magnético. Emitiendo varios pulsos RF se obtendrán promedios de decaimientos FID, los cuales, se tomarán como un promedio para observar la relación entre señal y ruido (coeficiente SNR) el cual mejora realizando este promedio de los datos. [3]

#### Gradientes

Un gradiente hace referencia en RM a las variaciones lineales espaciales de acuerdo con la intensidad del campo estático en el eje z, en otras palabras, a lo largo del campo  $\mathbf{B}_0$ . Al ser espacial, se tendrán 3 gradientes:  $G_x$ ,  $G_y$  y  $G_z$ , a cada uno de estos gradientes se les sumará o restará (según sea el caso) la magnitud del campo estático a lo largo de su eje y dirección correspondiente.[5] La figura 2.1 muestra en (a) cómo es el campo magnético  $\mathbf{B}_0$  con un gradiente en el eje x, mientras que en (b) se muestran las líneas de campo de un campo total.

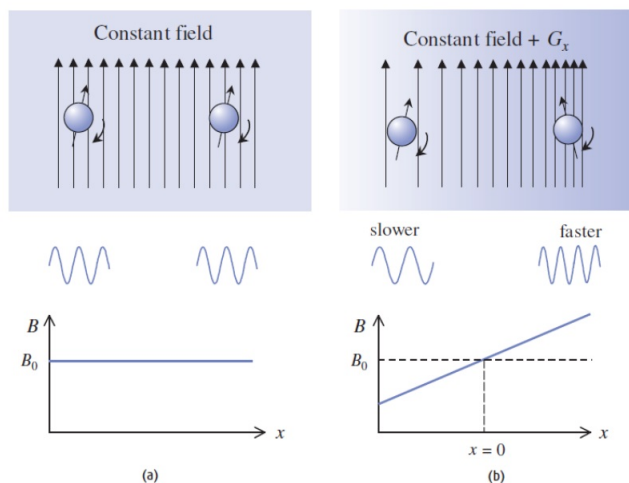


Figura 2.1: Efectos de gradientes. (a) Campo magnético  $\mathbf{B}_0$  haciendo que los núcleos hagan movimientos de precesión a la misma frecuencia. (b) Campo magnético  $\mathbf{B}_0$  con un gradiente añadido a lo largo del eje x, la frecuencia de precesión hace dependencia de la posición ahora. 2.1

De acuerdo con los gráficos de la figura 2.1 todos los espines en (a) están bajo el efecto del mismo campo magnético, al añadir un gradiente en (b), el campo que se produce por añadir el gradiente se suma al campo inicial  $\mathbf{B}_0$ . En el punto  $X = 0$ , el campo total sigue siendo el inicial, por lo que

los espines hacen resonancia con la frecuencia de Larmor, pero, al recorrer en dirección positiva o negativa del eje x, el campo total cambia su magnitud, aumentando o disminuyendo según sea la posición de los protones. Esto implica que de acuerdo a la posición de los protones, la resonancia será mayor o menor. Una precesión rápida muestra una frecuencia alta en la señal de RM, por lo que las mediciones de frecuencia se pueden utilizar para hacer una distinción entre diversas señales de RM en diversas posiciones del espacio. Los gradientes se aplican en un breve periodo de tiempo en forma de pulsos, aplicados en cualquier dirección y orientación según sea lo necesario. El conjunto de gradientes en las 3 dimensiones, hace que la RM pueda volverse tridimensional.[5]

El efecto de los gradientes se puede ver matemáticamente definiendolos de forma ortogonal a  $\mathbf{B}_z$  como:

$$\begin{aligned} G_x &= \frac{\partial B_z}{\partial x} \\ G_y &= \frac{\partial B_z}{\partial y} \\ G_z &= \frac{\partial B_z}{\partial z} \end{aligned} \quad (2.13)$$

Al aplicar un gradiente, el campo total en la dirección z que experimentan los espines depende de la posición en el espacio, tomando como ejemplo la dirección en x con un gradiente aplicado:

$$B(x) = B_0 + xG_x \quad (2.14)$$

Cuando se aplica el gradiente, la frecuencia de Larmor ahora dependerá de la componente total en z del campo magnético, es decir, se vuelve dependiente del espacio, tomando la ecuación 2.11, esta se transforma de la siguiente manera:

$$\omega = \gamma(B_0 + xG_x) \quad (2.15)$$

En el cuerpo, al tener una distribución uniforme de agua, los efectos que tendrá al aplicar un gradiente durante un tiempo serán las variaciones de la señal de RM, dependiente de la posición. Los espines que hacen el movimiento de precesión lo harán con una acción de gradiente que hará que al preceder más rápido entre ellos se tendrá un desfase o una separación, mientras que los que precesan más lentamente tendrán un desfase similar pero en una dirección contraria. Este efecto combinado es conocido como *fanning out* o “abanico” por el desfase. La figura 2.2 muestra el desfase de estos protones. La velocidad de desfase depende de la amplitud del gradiente. Se tiene un ángulo total de desfase que depende del producto de la intensidad del gradiente por su duración, también conocido como “momento de gradiente”. Aplicar ahora un gradiente en sentido o dirección contraria, las señales rápidas que se tenían previamente ahora serán más lentas y viceversa. Los espines tendrán un desfase tal que se llegará a un punto en el que los momentos de gradientes serán iguales y las componentes de la señal de RM apuntarán en la misma dirección. Cuando lo anterior ocurre se tiene una señal de RM medible, la cual se conoce como **eco de gradiente**. Cada impulso de gradiente se conoce como *lobulo* y se define como un desfase si se pierde parte de la señal o como refase si corrige un desfase previo.

Cabe destacar que la acción del gradiente se puede ver también como un desfase de los componentes de magnetización transversal en el plano xy. Su cambio de fase en un tiempo “t” y una posición “x” se verá como:

$$\phi(x, t) = e^{(i\gamma*xG_x*t)} \quad (2.16)$$

Cuando el gradiente ya no está activo, las modificaciones que se realizaron en la fase aun permanecen hasta que la magnetización transversal decae a cero o se aplica otro gradiente. Un eco de gradiente es el resultado de aplicar un gradiente de igual magnitud en su momento de gradiente pero de sentido opuesto.

$$\int_0^{TE} G^-(t)dt = - \int_0^{TE} G^+(t)dt \quad (2.17)$$

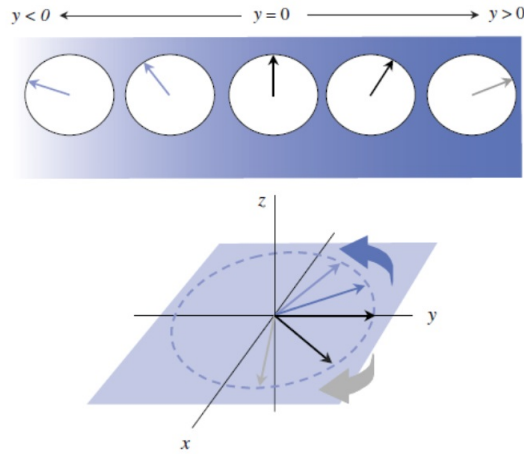


Figura 2.2: Efectos de gradientes en la señal de RM. Se tendrá una señal que proviene de diversas posiciones con cambios de fase dependientes de la posición. [5]

La ecuación 2.17 muestra al tiempo TE ocurriendo donde los signos más y menos se refieren a la forma de onda del gradiente. La pérdida de señal debida a la falta de homogeneidad del campo  $\mathbf{B}_0$  principal no es restablecido. La señal de RM decae de forma exponencial con una constante de tiempo que se llamará  $T2^*$

### Tiempos de relajación T1 y T2

Un equilibrio del estado de magnetización es debido a las componentes  $M_z$ ,  $M_x$  y  $M_y$  igualadas a cero. Aplicando una RF se crea un estado de no equilibrio añadiendo energía al sistema. Después de terminar el pulso de RF, el sistema debe relajarse y regresar a un equilibrio. Este fenómeno de relajación o tiempo de relajación aparece en cada componente llamados: relajación tipo T1 que afecta a la magnetización en el eje z, la relajación tipo T2 que afecta a la magnetización en los ejes x e y. También, al tiempo de relajación T1 se le conoce como espín-red y al tiempo T2 se le conoce como espín-espín.

Los tiempos de relajación se pueden expresar matemáticamente como una serie de ecuaciones diferenciales de primer orden conocidas como “ecuaciones de Bloch” cuyas soluciones aparecen después de un pulso de RF, un ángulo de punta  $\alpha$ , el valor de  $M_z$  y un tiempo t el cual se expresa matemáticamente como:

$$M_z(t) = M_0 \cos(\alpha) + (M_0 - M_0 \cos(\alpha))(1 - e^{-\frac{t}{T_1}}) \quad (2.18)$$

De acuerdo con la ecuación 2.18, el aplicar un pulso de  $90^\circ$  produce una diferencia de la unidad y la parte exponencial, es decir:

$$M_z(t) = (M_0)(1 - e^{-\frac{t}{T_1}}) \quad (2.19)$$

Debido a la existencia de diferentes valores para  $T_1$  originados por los diferentes tejidos existentes en el cuerpo humano, así como la diferencia entre valores promedio y los valores detectados en el momento de una toma de muestra; se generarán contrastes en la imagen.

La ecuación de Bloch para las componentes x e y después de emitir un pulso de RF en un ángulo  $\alpha$  a lo largo de nuestro eje x, se tiene un pulso de la forma:

$$M_y(t) = M_0 \sin(\alpha) e^{-\frac{t}{T_2}} \quad (2.20)$$

De las ecuaciones de Bloch se observa que las secuencias en  $T_1$  y  $T_2$  no están correlacionadas ya que un valor más grande del primer tiempo de relajación no implica una relación proporcional directa o inversa.

Una limitante para la construcción de imágenes por RM es la inhomogeneidad que existe en los diferentes tipos de tejidos, pues, cada tejido tiene diferentes susceptibilidades magnéticas. Un implante a base de metal como el titanio o el hierro existente en la hemoglobina alteran el campo magnético efectivo  $B_0$ . Las variaciones locales entre tejidos con diferentes propiedades deben ser consideradas, pues, el desfaseamiento en el movimiento de precesión de los protones será más rápido que el tiempo de relajación  $T_2$ , por ende, es necesario considerar estos efectos en un tiempo de relajación llamado  $T_2^*$ . La suma de los recíprocos de  $T_2$  y  $T_2^*$  originan el tiempo de relajación  $T_2^*$  modelado por la ecuación:

$$\frac{1}{T_2^*} = \frac{1}{T_2^{\dagger}} + \frac{1}{T_2} \quad (2.21)$$

Los valores para  $T_2^*$  consideran variaciones mínimas del orden de milisegundos en elementos metálicos o con propiedades magnéticas (como el aire) y también se consideran tejidos homogéneos con variaciones de cientos de milisegundos como el cerebro

### 2.1.3. Conceptos de detección y métodos de adquisición de señales.

Previamente se ha hecho mención sobre los cambios de la magnetización debida a gradientes o la aplicación de un pulso de RF haciendo cambios de direcciones a los campos magnéticos y la precesión de los protones de hidrógeno. Al tener una magnetización transversal se puede considerar la detección de la precesión de los protones al rededor de  $B_0$ , sus líneas de campo magnético (las de los protones) hacen un barrido junto con el movimiento de precesión que constituye una clave para medir la magnetización. Una fuerza electromotriz (fem) se crea en cualquier bobina a través de la cual el flujo magnético del espín del protón se barre como una consecuencia de la ley de Faraday, al depender la corriente del tiempo se puede transportar información de tal forma que se convierte en una imagen de RM.

#### Inducción de Faraday

La inducción de Faraday es un fenómeno fundamental de la RM que describe la generación de una corriente eléctrica en un conductor cuando este es expuesto a un campo magnético variable en el tiempo. Se tiene que la ley de inducción de Faraday se ve matemáticamente como:

$$emf = -\frac{d\Phi}{dt} \quad (2.22)$$

Siendo  $\Phi$  el flujo que pasa a través de una bobina.

$$\Phi = \int_{\text{Área de la bobina}} \vec{B} \cdot d\vec{S} \quad (2.23)$$

Se puede considerar que el flujo es proporcional al número de líneas de campo que penetran en el área efectiva que presentan las espiras que componen una bobina. Las corrientes inducidas en una espira conductora por un flujo cambiante producen un campo que se opone a los cambios inducidos por el campo externo. Aplicar la ecuación 2.22 a la RM, resulta útil el estudio de ejemplos elementales como una espira de alambre. Considerando una bobina fija en un campo magnético de eje fijo que depende del tiempo, al considerar un campo magnético armónico (con un comportamiento sinusoidal con el paso del tiempo) con una frecuencia angular  $\omega$  que formará un ángulo  $\theta$  con la normal del plano de la bobina que se muestra en la figura 2.3(a), el campo se expresará como:

$$\vec{B}(t) = B(\sin\theta\hat{y} + \cos\theta\hat{z})\sin\omega t \quad (2.24)$$

En la ecuación 2.24 se tiene un campo constante  $B$  que es mayor a 0. Se toma el campo de 2.24 como uniforme sobre el espacio de la bobina en cualquier instante de tiempo. Al escoger la bobina en el plano x-y, se tendrá que el diferencial de área será:

$$d\vec{S} = dx dy \hat{z}$$

Tomando en consideración las ecuaciones 2.22 y 2.23 se tiene que la fem generada en la bobina por la variación del campo magnético con respecto del tiempo es:

$$fem = -\frac{d}{dt} \int_{-\frac{L}{2}}^{\frac{L}{2}} dx \int_{-\frac{L}{2}}^{\frac{L}{2}} dy \hat{z} \cdot \vec{B}(t) = -L^2 B \omega \cos \theta \cos \omega t \quad (2.25)$$

Considerando el resultado de 2.25 si se sitúa en un tiempo igual a 0 se tendrá una fem negativa

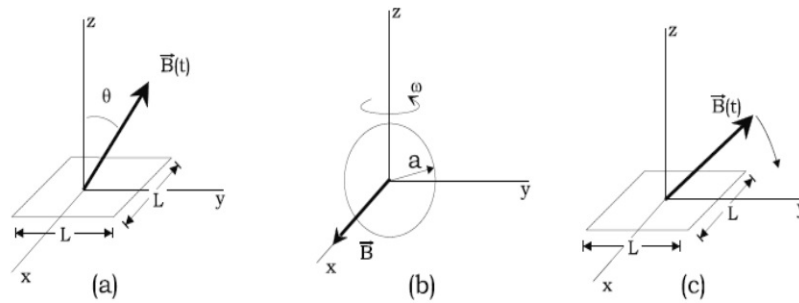


Figura 2.3: Bucles de hilos que experimentan cambios de flujo con una dependencia temporal: (a) Una bobina cuadrada estacionaria con un campo aplicado oscilando en un eje fijo. (b) Una bobina circular girando en un campo estático. (c) Una bobina estacionaria sumergida en un campo giratorio de magnitud fija (ejemplo relevante para mediciones de RM) [6]

con una corriente en sentido de las manecillas del reloj en la espiral. La aparición del factor  $\omega$  en 2.25 que resulta de la derivada temporal, es un ejemplo claro de que la fem aumenta con la frecuencia. La situación en la que una bobina estacionaria con un campo rotando es un ejemplo aun mas estrecho a la RM. En un experimento de RM, el campo asociado a una magnetización en precesión barre bobinas receptoras. [6]

La ecuación 2.23 se puede transformar para ser útil en la RM, ya que se invierten papeles de la fuente de magnetización y la bobina de detección. El cambio de sus papeles en una ecuación para la señal es un ejemplo del principio de reciprocidad.

Se tiene un campo magnetico asociado con la magnetización de una muestra que surge de la densidad de corriente efectiva:

$$\vec{J}_M(\vec{r}, t) = \vec{\nabla} \times \vec{M}(\vec{r}, t) \quad (2.26)$$

Una densidad de corriente  $\vec{J}$  implica  $|\vec{J}|$  una carga por unidad de tiempo por unidad de área en la dirección de la corriente  $\hat{J}$ . El producto cruz de la ecuación 2.26 calcula la "circulación" neta de la

magnetización, es decir, la magnetización puede aproximarse como una distribución de densidad de varios bucles pequeños de corriente. Las corrientes netas asociadas a los bucles corresponderán a una densidad de corriente efectiva. Por lo que, el potencial vectorial en la posición  $\vec{r}$  proveniente de una fuente de corriente se muestra matemáticamente como:

$$\vec{A}(\vec{r}) = \frac{\mu_0}{4\pi} \int d^3r' \frac{\vec{J}(\vec{r}')}{|\vec{r} - \vec{r}'|} \quad (2.27)$$

Se ignoran los efectos debido al retardo temporal entre la fuente y la medición del campo. El campo magnético se calcula a partir de:

$$\vec{B} = \vec{\nabla} X \vec{A} \quad (2.28)$$

Utilizando 2.28 y el teorema de Stokes, se puede reescribir la ecuación 2.23 como:

$$\Phi = \int_{Area} \vec{B} \cdot d\vec{S} = \int_{Area} (\vec{\nabla} X \vec{A}) \cdot d\vec{S} = \oint d\vec{l} \cdot \vec{A} \quad (2.29)$$

Con estas ecuaciones previas, se detalla y demuestra que el flujo de la magnetización  $M$  a través de una bobina debido a una fuente de magnetización se puede relacionar con un flujo debido a la bobina que atraviesa la magnetización, se hace uso de 2.26, 2.27 y 2.29 así como una integración por partes y la identidad vectorial  $\vec{A} \cdot (\vec{B} X \vec{C}) = -(\vec{A} X \vec{C}) \cdot \vec{B}$ , obteniendo:

$$\begin{aligned} \Phi &= \oint d\vec{l} \cdot \left[ \frac{\mu_0}{4\pi} \int d^3r' \frac{\vec{\nabla}' X \vec{M}(\vec{r}')}{|\vec{r} - \vec{r}'|} \right] = \frac{\mu_0}{4\pi} \int d^3r' \oint d\vec{l} \cdot \left[ (-\vec{\nabla}' \frac{1}{|\vec{r} - \vec{r}'|}) X \vec{M}(\vec{r}') \right] \\ &= \frac{\mu_0}{4\pi} \int d^3r' \vec{M}(\vec{r}') \cdot \left[ \vec{\nabla} X \left( \oint \frac{d\vec{l}}{|\vec{r} - \vec{r}'|} \right) \right] \end{aligned} \quad (2.30)$$

Usando la ecuación de 2.27 para lazo de corriente evaluada en la posición  $\vec{r}'$ , muestra que la integral de línea sobre la trayectoria de la corriente en 2.1.3 será en realidad un campo magnético por unidad de corriente que produciría la bobina en el punto  $\vec{r}$

$$\vec{B}^{receive}(\vec{r}') = \vec{B}(\vec{r}')/I = \vec{\nabla}' X \left( \frac{\mu_0}{4\pi} \oint \frac{d\vec{l}}{|\vec{r} - \vec{r}'|} \right) \quad (2.31)$$

Finalmente, el flujo se puede mostrar como:

$$\Phi_M(t) = \int_{muestra} d^3r \vec{B}^{receive}(\vec{r}) \cdot \vec{M}(\vec{r}, t) \quad (2.32)$$

La dependencia temporal de la magnetización se vuelve explícita. El hecho de que el flujo en 2.31 dependa del campo magnético producido por la bobina de detección en todos los puntos donde la magnetización es distinta de cero se conoce como **principio de reciprocidad**. La expresión original como integración superficial sobre el área de la bobina de detección se ha sustituido por la integración volumétrica sobre la región de magnetización distinta de cero, es decir, el flujo a través de la bobina de detección debido a la magnetización puede encontrarse en su lugar calculando el flujo que emanaría de la bobina de detección, por unidad de corriente, a través de la magnetización. [6]

### Señales de la magnetización en precesión

Se asume que la muestra (un proton de agua) esta inmerso en un campo magnético, estático y uniforme en dirección z y se ha excitado mediante un pulso de RF haciendo que en un tiempo “t” surjan componentes transversales de magnetización  $M_x, M_y$  y una componente longitudinal  $M_z$ . La señal que se medirá por un sistema electronico será proporcional a la energía electromotriz la cual se puede escribir por componentes de la forma:

$$Signal \propto -\frac{d}{dt} \int d^3r [\mathcal{B}_x^{receive}(\vec{r}) \vec{M}_x(\vec{r}, t) + \mathcal{B}_y^{receive}(\vec{r}) \vec{M}_y(\vec{r}, t) + \mathcal{B}_z^{receive}(\vec{r}) \vec{M}_z(\vec{r}, t)] \quad (2.33)$$

Cuando se colocan las soluciones conocidas para  $\vec{M}(\vec{r}, t)$  en la integral, la evaluación de la derivada temporal muestra que la manetización longitudinal se puede despreciar, incluso si la componente en z es distinta de 0 para el campo de la bobina receptora.

Las soluciones de campo estático se mantienen para cada  $\vec{r}$ . La solución longitudinal será:

$$M_z(\vec{r}, t) = e^{-\frac{t}{T_1(\vec{r})}} M_z(\vec{r}, 0) + (1 - e^{-\frac{t}{T_1(\vec{r})}}) M_0 \quad (2.34)$$

Es útil el emplear la representación compleja para las componentes transversales y no solo por la parte temporal, más bien para una serie de fases.

$$\begin{aligned} M_+(\vec{r}, t) &= e^{-\frac{t}{T_2(\vec{r})}} e^{-i\omega_0 t} M_+(\vec{r}, 0) \\ &= e^{-\frac{t}{T_2(\vec{r})}} e^{-i\omega_0 t + i\phi_0} M_\perp(\vec{r}, 0) \end{aligned} \quad (2.35)$$

La fase  $\phi_0$  y la magnitud de  $M_\perp(\vec{r}, 0)$  son determinados por el pulso de RF como condición inicial. Se tiene, entonces, una componente imaginaria y real.

$$M_x = Re(M_+) \quad M_y = Im(M_+)$$

Las ecuaciones 2.35 y 2.1.3 se pueden sustituir en 2.33 para operar directamente con 2.34 y 2.35. En campos estáticos al nivel de los protones de agua, la frecuencia de Larmor es al menos 4 órdenes de magnitud mayor que los valores típicos de  $\frac{1}{T_1}$  y  $\frac{1}{T_2}$  por lo que las exponenciales que guardan como argumento a estos 2 tiempos de relajación se pueden despreciar en comparación con la derivada que tiene en consideración a la frecuencia. La señal al sustituir por las exponenciales con el numero imaginario “i”, se vuelve una integral de senos y cosenos de angulos dobles, el término principal en el cálculo de la derivada temporal indicará que las oscilaciones rápidas de Larmor de la magnetización transversal inducen la “señal dominante en la bobina receptora”. También se muestrs por que la magnetización transversal  $M_+$  o su magnitud vista como  $M_\perp = \sqrt{M_x^2 + M_y^2}$  se refiere de forma heuristica como la “señal”. [6]

La expresión de la señal puede ser simplificada, considerando que se puede describir en terminos del campo campo magnetico transversal y el angulo con una parametrización:

$$\mathcal{B}_x^{receive} \equiv \mathcal{B}_\perp \cos\theta_B, \quad \mathcal{B}_y^{receive} \equiv \mathcal{B}_\perp \sin\theta_B \quad (2.36)$$

Con la posible dependencia de la posición explicita y el uso de la identidad trigonométrica para el ángulo doble  $\sin(a + b) = \sin a \cos b + \cos a \sin b$

$$signal \propto \omega \int d^3r e^{-\frac{t}{T_2(\vec{r})}} M_\perp(\vec{r}, 0) \mathcal{B}_\perp(\vec{r}) \sin(\omega t + \theta_B(\vec{r}) - \phi_0(\vec{r})) \quad (2.37)$$

La expresión 2.37 puede ser modificada para tener una generalización. Modificando  $T_2$  por  $T_2^*$  se hace al considerar inhomogeneidades como el hierro en las globinas que afectan al campo externo.

Es posible que, para pequeñas muestras homogéneas, se puedan despreciar las dependencias espaciales, considerando que todas las cantidades se encuentren en la integral 2.37 son independientes de  $\vec{r}$ . Si el volumen de la muestra es  $V_s$ , la señal se puede considerar:

$$signal \propto \omega V_s e^{-\frac{t}{T_2}} M_{\perp} \mathcal{B}_{\perp} \sin(\omega t + \theta_B - \phi_0) \tag{2.38}$$

Para que la frecuencia de precesión sea constante en todas partes, el campo estático  $B_0$  también debe ser uniforme en todo el espacio. Las fases constantes de la ecuación 2.38 son especialmente importantes en la comparación de señales de diferentes fuentes donde es necesario ver si se cancelan estas señales. [6]

Las oscilaciones rápidas a la frecuencia de Larmor en las expresiones de la señal se eliminan, en la práctica, mediante un paso electrónico llamado “demodulación”, que equivale a ver la señal desde un marco referencias inercial. La demodulación hace referencia a la multiplicación de la señal por una sinusoidal o cosinusoidal con una frecuencia igual o cercana a la de Larmor creando una RF de transmisión o “irradiación”  $\omega_{rf}$ , pero, se suele escoger la frecuencia  $\omega_0$ . La formula independiente del espacio 2.38 se utiliza para ilustrar una señal demodulada tomando en consideración la señal de referencia, que genera por separado para la demodulación para tener una frecuencia  $\Omega = \omega + \delta\omega$  donde el término  $\delta\omega$  se denomina “frecuencia de desplazamiento” de la frecuencia de Larmor. Se considera tanto la multiplicación de seno como la de coseno, respectivamente, al almacenamiento de datos en “dos canales”, es decir, una matriz de numeros reales y una matriz de numeros imaginarios. Los canales imaginarios y reales se expresan de la forma:

$$CRF = \frac{1}{2} \cos(\delta\omega * t - \zeta) = \frac{1}{2} Re(e^{i\delta\omega*t-i\zeta}) \tag{2.39}$$

$$CIF = \frac{1}{2} \sin(\delta\omega * t - \zeta) = \frac{1}{2} Im(e^{i\delta\omega*t-i\zeta}) \tag{2.40}$$

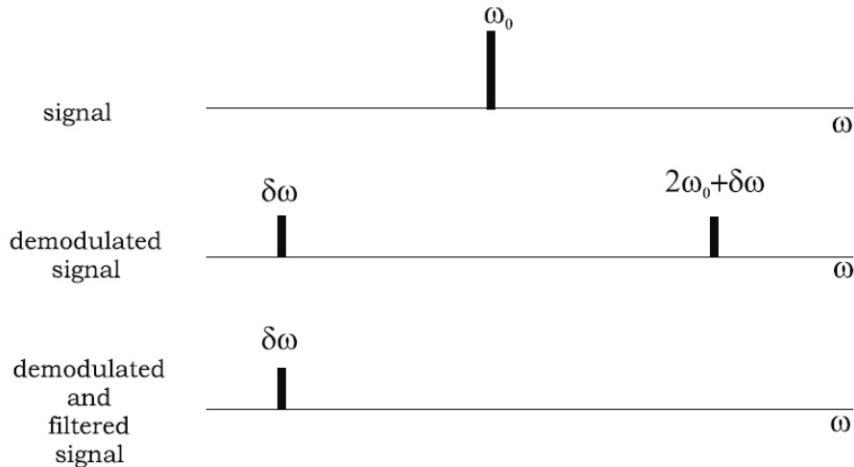


Figura 2.4: Desplazamientos en frecuencia de las señales procedentes de una muestra uniforme debidos a la demodulación y el filtrado, con un desplazamiento  $\delta\omega$ . Las líneas gruesas representan la señal en las frecuencias indicadas. [6]

### Decaimiento por inducción libre

Al considerar un pulso  $\frac{\pi}{2}$  aplicado uniformemente a los espines de protones en un campo magnético estático asociado, con cualquier átomo de hidrogeno presente (en una vista macroscópica). El pulso rota la magnetización longitudinal (exceso de espines) hacia el plano transversal tras lo cual los espines inclinados precesan libre y colectivamente. El campo magnético variable en el tiempo derivado de la suma de todos los campos de espín de protón en precesión induciría una pequeña fem en cualquier bobina de RF orientada adecuadamente para detectar los cambios de flujo correspondientes. Las consideraciones anteriores son las que originan el decaimiento por inducción libre (FID por sus siglas en inglés *Free Induction Decay*). Un FID se realiza de forma rutinaria en las máquinas de RM para sintonizar las bobinas de RF y optimizar la respuesta del sistema. Puede utilizarse para localizar el pico de resonancia del agua y determinar la amplitud y duración de RF necesarias para producir la señal máxima.

La base para la expresión matemática de la señal FID se realiza generalizando la ecuación 2.37, de forma compleja de una señal demodulada debida a un spin de RF en el tiempo igual a 0.

$$s(t) \propto \omega \int d^3r e^{-\frac{t}{T_2(\vec{r})}} \mathcal{B}_\perp(\vec{r}) M_\perp(\vec{r}, 0) e^{i((\Omega - \omega(\vec{r}))t + \phi_0(\vec{r}) - \theta_B(\vec{r}))} \quad (2.41)$$

Teniendo una frecuencia de referencia de demodulación y ángulo de campo  $\theta B$ . Existe en general, una dependencia en  $T_2$  y la magnetización inicial (incluso con un pulso  $\frac{\pi}{2}$  perfecto, el valor de equilibrio  $M_0$  depende de la densidad de espín local. También se ha incluido la posibilidad de que la frecuencia de precesión pueda cambiar con la posición,  $\omega = \omega(\vec{r})$ . Suponiendo que la frecuencia de precesión es constante ( $\omega$ ), en todo el espacio, y que la fase de magnetización inicial, la dirección del campo receptor y la constante de decaimiento son también independientes del espacio. Los factores asociados pueden tomarse fuera de la integral 2.41 de la forma:

$$s(t) \propto \omega e^{-\frac{t}{T_2}} e^{i((\Omega - \omega)t + \phi_0 - \theta_B)} \int d^3r \mathcal{B}_\perp(\vec{r}) M_\perp(\vec{r}, 0) \quad (2.42)$$

Se considera la no uniformidad de la bobina receptora. La exponencial compleja da paso a una función seno o coseno dependiendo de la detección de señal real o imaginaria del canal. La ecuación 2.42 se puede utilizar para comprender la dependencia temporal de la señal. Al no aplicar demodulación (siendo  $\Omega = 0$ ), las oscilaciones rápidas a una frecuencia  $\omega_1$  se amortiguan por el factor de relajación, como se muestra en la figura 2.5(a). Una demodulación en resonancia, es decir,  $\Omega = \omega$  elimina la componente imaginaria, quedando solo la envolvente en  $T_2$  de la figura 2.5(b). Finalmente, una demodulación fuera de resonancia como lo indica la figura 2.5(c) deja una componente de baja frecuencia, siendo que  $\Omega - \omega = \delta\omega$ . La superposición de señales demoduladas (en un valor central a  $\Omega$ ) de varios protones, cada una con una frecuencia de Larmor diferente, conduce a una suma de términos de acuerdo con la ecuación 2.42 dando lugar a una señal que decae más rápido que una única señal demodulada que sólo tiene amortiguamiento en el tiempo  $T_2$ . El desfase entre espines causa una señal reducida parecida a un efecto  $T_2$ .

La contribución de la fase de la ecuación 2.41 debido a la magnetización (también conocida como spin isocromático) en  $\vec{r}$  es:

$$\phi(\vec{r}, t) = -\omega(\vec{r})t + \phi_0(\vec{r}) = -\gamma B_z(\vec{r})t + \phi_0(\vec{r}) \quad (2.43)$$

Considerando la condición inicial en un tiempo igualado a 0, se observa que la fase  $\phi$  acumula una cantidad adicional, en el tiempo t, debido a la precesión sobre el componente z del campo. Asumido el hecho de que la magnetización transversal es creada por un pulso de RF, el ángulo  $\phi_0$ , la fase de la magnetización inicial  $M_+(\vec{r}, 0)$  es determinada por la dirección del pulso de RF. Las fases iniciales, como la que esta expuesta en la ecuación 2.43 a menudo se redefinirán. Dado a que se puede elegir un origen a conveniencia, las coordenadas polares del plano x-y se definen de tal forma que el ángulo  $\phi = 0$  a lo largo del eje X. Asimismo, es muy útil elegir el origen a lo largo de la magnetización inicial. Elegir el ángulo 0 se debería realizar a lo largo del eje y para originar una

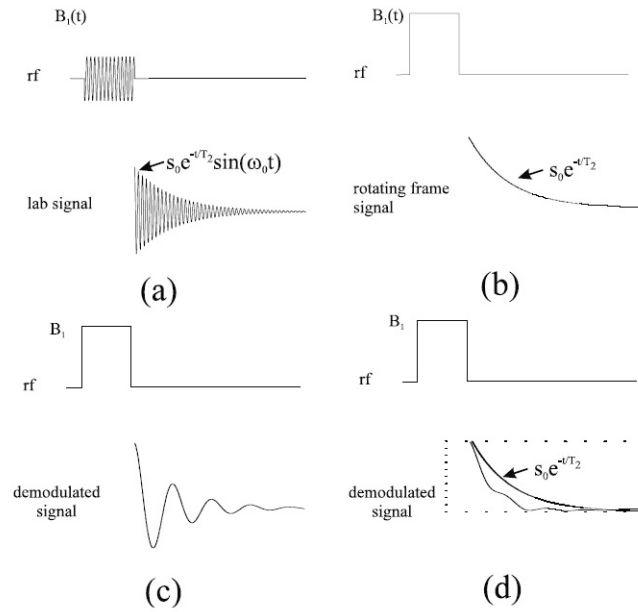


Figura 2.5: (a) Señal FDI con espines precesando a la frecuencia de Larmor. (b) Condición demodulada. (c) Señal FDI demodulada con demodulación en una frecuencia diferente a la de Larmor. (d) Señal FID demodulada total de varios espines, cada uno con una frecuencia de Larmor variada, mostrando un decaimiento con oscilaciones lentas y un decaimiento más rápido que en  $T_2$ . [6]

dirección de pulso de  $\frac{\pi}{2}$  en dirección x. En el tiempo 0, las direcciones “x” e “y” coincidirán con sus sentidos contrarios  $x'$  e  $y'$ , manteniendo la rotación en sentido antihorario para tener un ángulo  $\phi$  creciente.

### Decaimiento para $T_2^*$

Considerando los tiempos de relajación, se presentan inhomogeneidades del campo magnético causadas por desfases en la magnetización ante otro tipo de moléculas distintas a las del agua, haciendo que se haga una supresión adicional de la señal modulada por la ecuación 2.41, considerando ahora a la magnetización transversal para el tiempo de relajación  $T_2^*$  se tiene:

$$M_{\perp}(\vec{r}, t) = M_{\perp}(\vec{r}, 0)e^{-\frac{t}{T_2^*}} \quad (2.44)$$

De forma cualitativa, se tiene la misma señal que la presentada en la figura 2.5(b), haciendo que el nuevo factor envolvente decae mucho más rápido en el tiempo.

De acuerdo con la ecuación 2.21, se presenta una combinación de efectos inducidos por el campo externo. Es importante mencionar que el tiempo que complementa a  $T_2^*$ , es decir  $T_2^{\dagger}$  depende tanto de la máquina de resonancia magnética, como de la muestra de tejido de la cual se realiza la RM. Una diferencia que distingue al tiempo  $T_2$  es que la pérdida de la señal de  $T_2^{\dagger}$  es reversible, es decir, se puede recuperar.

Las FID usan la dependencia de  $T_2^{\dagger}$  para determinar si la homogeneidad del imán principal es adecuada. Se realiza una prueba colocando una muestra uniforme en el imán y se observa la FID. Si el decaimiento es como el mostrado en la figura 2.5(d), se concluye que el imán no tiene la suficiente homogeneidad y se tienen que hacer correcciones en el campo magnético principal para recuperar la señal.

Pese a que para el tiempo de relajación  $T_2^*$  se pueda obtener mediante ajuste de datos de forma

exponencial de la ecuación 2.44, hacer una distinción entre  $T_2^\dagger$  y  $T_2$  se vuelve una tarea complicada. Por lo que es necesario saber como se estima (mide) el valor de  $T_2^\dagger$ .

El origen del tiempo  $T_2^\dagger$  en su decaimiento parte del efecto colectivo de desfase isócrono de los espines. Matemáticamente, es un límite de desfase cuando se presenta un tiempo que tiende a infinito.

$$\sum_{muestra} e^{i\phi(\vec{r},t)} \rightarrow_{desfase} 0$$

La suma de la ecuación anterior hace referencia a una forma discreta de la señal demodulada de la ecuación 2.41 en la que se omiten factores de campo y magnetización restantes. Considerando la aplicación de un argumento de fase para estimar el tiempo de decaimiento en  $T_2^\dagger$  se tendrá una ventaja adicional de introducir variaciones de campo como campos lineales externos, efectos de susceptibilidad interna, entre otros. Considerando el uso de la ecuación 2.43 para un spin isócrono en una posición  $\vec{r}$  donde hay una variación espacial independiente del tiempo  $\Delta B(\vec{r})$  en la componente z del campo magnético total. Despreciando la fase inicial, la fase acumulada será:

$$\phi(\vec{r}, t) = -\gamma(B_0 + \Delta B(\vec{r}))t \tag{2.45}$$

Es importante destacar que las inhomogeneidades de los campos magnéticos externos ya estan considerados en el factor exponencial  $e^{-\frac{t}{T_2^\dagger}}$ , como en la ecuación 2.44. La fase  $\phi$  de la magnetización compleja  $M_+$  se redefine cuando el factor del tiempo de relajación  $T_2^\dagger$  esta presente, esto con el fin de no incluir los efectos de las inhomogeneidades que ya se toman en cuenta en el tiempo de relajación complementario. Lo mismo ocurre con el tiempo de relajación  $T_2$  cuando esta amortiguado, que se incluyen en el factor exponencial anteriormente descrito. Asimismo, para poder recuperar la señal de  $T_2^\dagger$  se tiene que realizar el método eco-spin.

### Método Eco-Spin

Este método (o secuencia) esta basado en aplicar 2 pulsos de RF, primero se aplica un pulso  $\frac{\pi}{2}$  seguido de un pulso de “reenfoque”  $\pi$ . La recuperación de la señal del tiempo de relajación  $T_2^\dagger$  se puede observar en la figura 2.6.

Para exponer esta secuencia se considera que la magnetización de un protón se inclina por el primer pulso hacia el plano transversal. Considerando que esto ocurre en un tiempo  $t = 0$ , de manera que los espines (en su mayoría) apuntan a lo largo del eje y, recordando que el campo de un pulso de este tipo se encuentra a lo largo del eje x, donde se supuso que se aplica el pulso en el marco de referencia rotado. Los espines, tendrán diferentes posiciones que se comenzarán a desfazar, unos con respecto a otros, a medida que se experimentaran diferentes intensidades del campo magnético que, estrictamente hablando, no serán exactamente iguales al campo magnético inicial  $B_0$ . La fase que se ha acumulado en un spin en una posición  $\vec{r}$  en el marco de rotación se encontrará restando del término de la frecuencia de Larmor en la ecuación 2.45. Por lo que la fase en este marco referencial relativo al eje “y” será:

$$\phi(\vec{r}, t) = -\gamma\Delta B(\vec{r})t \text{ para } 0 < t < \tau \tag{2.46}$$

Con  $\phi_0 = 0$  de la ecuación 2.43

Después de este efecto, de forma instantánea se aplica otro pulso de RF con el doble de amplitud. Este pulso es a lo largo del eje y, girando los espines sobre este mismo eje con un ángulo  $\pi$  en un tiempo  $\tau$ . Los espines que previamente acumularon una fase positiva extra, ahora después del pulso de  $\pi$ , tendrán el negativo de esa fase, lo mismo ocurre con los espines de fase negativa, se vuelven positivos.

$$\phi(\vec{r}, \tau^+) = -\phi(\vec{r}, \tau^-) = \gamma\Delta B(\vec{r})\tau$$

Finalmente, los spines continuan, después de un tiempo  $\tau$ , acumulando fase, de acuerdo con la ecuación 2.46.

$$\phi(\vec{r}, t) = \phi(\vec{r}, \tau^+) - \gamma\Delta B(\vec{r})(t - \tau)$$

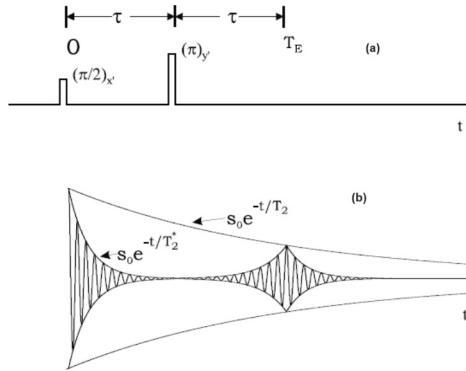


Figura 2.6: Pulso de RF de  $\frac{\pi}{2}$  aplicado al eje x en  $t = 0$  y un pulso de RF de  $\pi$  a lo largo del eje y en  $t = \tau$  para invertir la fase acumulada por los espines. El eco muestra un máximo local positivo, en lugar de un mínimo local negativo después de invertir fases. Los subíndices indican el eje a lo largo del cual se aplica el pulso de RF. [6]

$$\begin{aligned}
 & -\gamma \Delta B(\vec{r})(t - 2\tau) \\
 & = -\gamma \Delta B(\vec{r})(t - T_E), \text{ con } \tau < t
 \end{aligned}
 \tag{2.47}$$

La velocidad a la que se acumula la fase de cada spin no cambiará, por lo que todos los espines volverán a  $\phi = 0$  al mismo tiempo, el tiempo de eco el cual será  $T_E \equiv 2\tau$ . La ecuación 2.47 demuestra que la fase acumulada de todos los espines que experimentan una variación de campo independiente del tiempo volverán a 0 en el tiempo de eco, es decir,  $t = T_E$  sin importar el valor del campo magnético “B” y la posición  $\vec{r}$ . Por lo tanto, todos los espines se alinean al mismo tiempo, este realineamiento es el denominado “eco de spin”. La combinación de los pulsos de RF a diferentes amplitudes producen un “eco positivo” a lo largo del eje y, es decir, el eco hace un reenfoque a todos los espines a lo largo de la dirección en y de forma positiva. La figura 2.7 simplifica el comportamiento individual de los espines durante este efecto. [6]

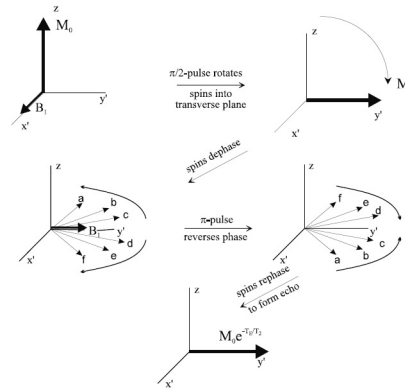


Figura 2.7: Conjunto de espines simulados durante el método de eco-spin. Un pulso  $\frac{\pi}{2}$  rota los espines alrededor del eje x, hacia el plano transversal generando un movimiento de precesión. Al acumular una fase extra, se invierte una vez mandado otro pulso de amplitud  $\pi$ , al seguir acumulando fase y volver al eje “y” positivo al mismo ritmo, se generará un eco. [6]

### 2.1.4. Espacio “k” y secuencia eco de gradiente

Cuando se realiza el diagnóstico por imagen de RM, se está realizando una determinación referente a la magnitud de espines distribuidos en el cuerpo, es decir, se toma en consideración la distribución de densidades de los espines, en vez de tomar propiamente a la magnetización de estos. Considerando una señal en función del tiempo que será una señal compleja modulada, es decir, la unión de las ecuaciones 2.39 y 2.40, que ahora se definirá como  $s(t)$ , para modificar las proporcionalidades de la señal en una ecuación se consideran las aportaciones realizadas por el sistema electrónico de detección. Además, se toma en cuenta que las bobinas emisoras y receptoras de RF tienen uniformidad, de modo que la fase  $\phi_0$ , la fase de dirección del campo  $\theta_B$  y la amplitud del campo de recepción  $B_\perp$  son invariantes ante cualquier posición. Dado a que se toman en consideración varias veces las posiciones a las que se encuentra el sistema, las fases constantes se pueden despreciar o igualar a 0 o ser absorbidas en el término  $\Lambda$ . La señal es generalizada para incluir la frecuencia de precesión dependiente de una posición y el tiempo  $\omega(\vec{r}, t)$ , además se desprecian los efectos de los tiempos de relajación, por lo que se obtiene:

$$s(t) = \omega_0 \Lambda B_\perp \int d^3r M_\perp(\vec{r}, 0) e^{i(\Omega t + \phi(\vec{r}, t))} \quad (2.48)$$

con  $\Omega$  la referencia o frecuencia de demodulación sin desplazamiento, es decir,  $\Omega = \omega_0$ , la  $\omega_0$  representará la frecuencia de Larmor. El ángulo  $\phi$  es la fase acumulada, siguiendo la convención de signos positivos en sentido contrario a las manecillas del reloj para ángulos, se tiene:

$$\phi(\vec{r}, t) = - \int_0^t dt' \omega(\vec{r}, t')$$

donde  $\omega = \omega_0$  solo si hay un campo magnético estático y uniforme. Añadir un campo de gradientes será la razón por la cual la frecuencia de precesión sea una función espacio-temporal. Si se quita el campo de gradientes, el campo estático será estrictamente homogéneo y constante.

Si se considera un pulso de  $\frac{\pi}{2}$  aplicado sobre la muestra y se desprecian los efectos que los tiempos de relajación ocasionan, la magnetización transversal será una magnetización en estado de equilibrio por lo que se puede expresar en términos de la densidad de spin del protón. Resultará una expresión que involucrará el campo estático  $B_0$  y la temperatura  $T$ .

$$M_\perp(\vec{r}, 0) = M_0(\vec{r}) = \frac{1}{4} \rho_0(\vec{r}) \frac{\gamma^2 \hbar^2}{kT} B_0 \quad (2.49)$$

Haciendo una combinación de las ecuaciones 2.48 y 2.49 se obtiene:

$$s(t) = \int d^3r \rho(\vec{r}) e^{i(\Omega t + \phi(\vec{r}, t))} \quad (2.50)$$

Considerando ahora en la nueva ecuación la densidad efectiva de espines, la cual, se muestra de forma explícita como:

$$\rho(\vec{r}) \equiv \omega_0 \Lambda B_\perp M_0(\vec{r}) = \frac{1}{4} \omega_0 \Lambda B_\perp \rho_0(\vec{r}) \frac{\gamma^2 \hbar^2}{kT} B_0 \quad (2.51)$$

Si la magnitud del campo receptivo  $B_\perp$  no es homogénea, su dependencia espacial puede ser considerada también en la ecuación 2.51. Los factores como los tiempos de relajación también se pueden considerar. Considerando que la ecuación 2.50 dependerá de una sola dimensión, la señal se puede reescribir como:

$$s(t) = \int dz \rho(z) e^{i(\Omega t + \phi(z, t))} \quad (2.52)$$

con su respectiva densidad de spin efectiva para el caso de una dimensión.

$$\rho(z) = \int \int dx dy \rho(\vec{r})$$

Las regiones de interés serán aquellas zonas que tengan una densidad de spin distinta de 0, esto determina los límites de integración. Asimismo, mediante una proyección de las dimensiones  $x$  e  $y$  se puede obtener la "densidad lineal"  $\rho(z)$ , se vuelve entonces, un problema en una sola dimensión. Despreciar los efectos de relajación provoca errores si se realiza una única adquisición, si el tiempo de muestreo total (llamado  $T_s$ ) es mucho menor que el tiempo de relajación  $T_2^*$ , es decir, en la ecuación 2.52 si se tiene  $0 \leq t \leq T_s$ .

Si se realizan adquisiciones múltiples hacen aparición los tiempos de relajación  $T_1$  y  $T_2$  al utilizar el método de eco-spin. La ecuación 2.52 aún no perderá su validez siempre y cuando se tenga un tiempo de repetición muy grande, es decir,  $TR \gg T_1$ , mientras que el tiempo de eco, tiene que ser muy pequeño, en otras palabras,  $TE \ll T_2$ . Con las consideraciones anteriores, las exponenciales que albergan a estos tiempos de relajación se anulan o se vuelven una unidad. Como los efectos de relajación en la vida real no se pueden despreciar, la densidad de espines se verá como  $\rho(z, T_1, T_2)$  y, de forma general para un espacio tridimensional será  $\rho(\vec{r}, T_1, T_2)$

### Transformadas de Fourier y ecuaciones para una dimensión

Considerando un gradiente en la dirección  $z$ , la dependencia temporal del gradiente muestra que se pueden tener modificaciones del mismo, por lo que se puede definir una frecuencia angular de los espines, con una contribución a esta de parte del gradiente.

$$\omega(z, t) \equiv \omega_0 + \omega_G(z, t) \quad (2.53)$$

Tomando al campo magnético en  $z$  producido por el campo inicial y su respectivo gradiente, es decir,  $B_z(z, t) = B_0 + zG(t)$  se tiene una variante de la frecuencia de Larmor que dependerá del gradiente y de la coordenada espacial en  $z$ , es decir:

$$\omega_G(z, t) = \gamma z G(t) \quad (2.54)$$

Se presenta una relación entre la posición de los espines a lo largo de la dirección  $z$  y sus velocidades de precesión, esta relación se conoce como "codificación de frecuencia" a lo largo de la dirección. El objetivo de una RM en pacientes es poder "extraer" información sobre la distribución de hidrógeno de la zona de interés, observando espines mediante su señal codificada en una frecuencia. La fase que se acumula, en un tiempo  $t$ , por causa del gradiente será:

$$\begin{aligned} \phi_G(z, t) &= - \int_0^t dt' \omega_G(z, t') \\ &= -\gamma z \int_0^t dt' G(t') \end{aligned} \quad (2.55)$$

Se asume que el gradiente se aplica después del primer pulso de RF, es decir en  $t = 0$ .

Se realiza la suposición en la que la señal de la ecuación 2.52 tiene una frecuencia demodulada, es decir,  $\Omega = \omega_0$  y una frecuencia de precesión tal y como lo describe la ecuación 2.53 es considerada, por lo que, con estas consideraciones se obtiene:

$$s(t) = \int dz \rho(Z) e^{i\phi_G(z, t)} \quad (2.56)$$

Aquí, la fase después de la demodulación es determinada por el campo de gradiente. La ecuación 2.56 se puede utilizar para mediciones con campos de gradiente que tienen una dependencia en  $z$  de forma arbitraria, se le conoce como "ecuación de imagen 1D" por ser para una sola dimensión, pero, puede ser generalizada haciendo el cambio de  $\phi_G$  por simplemente  $\phi$  ya que se consideran las variaciones del gradiente y de las RF. A su vez, la dependencia espacial en " $z$ " de la ecuación 2.55 de forma explícita se define en la ecuación siguiente:

$$s(k) = \int dz \rho(Z) e^{-i2k\pi z} \quad (2.57)$$

Haciendo que la dependencia temporal se encuentre de forma implícita en un *espacio de frecuencias*  $k = k(t)$  de tal forma que:

$$k(t) = \tilde{\gamma} \int_0^t dt' G(t') \tag{2.58}$$

Se presenta, entonces, que el hecho de aplicar gradientes hace que la señal  $s(k)$  sea la transformada de Fourier de la densidad de espín de muestra. La densidad del espín esta codificada en el espacio de Fourier a lo largo del eje  $z$  por el gradiente. Una de las propiedades importantes de la transformada de Fourier es su inversa que tiene una definición propia, dada una señal  $s(k)$  para todo valor en "k", la densidad de espines efectiva del tejido se puede hallar utilizando una transformada inversa de la señal:

$$\rho(z) = \int dk s(k) e^{i2\pi kz} \tag{2.59}$$

La señal  $s(k)$  y la imagen que se obtiene de  $\rho(z)$  son transformadas de Fourier que estan pareadas.

**Eco de gradiente**

Para este tipo de secuencias, se puede tomar de manera ilustrativa la figura 2.8, colocando un gradiente negativo constante de tal forma que  $G_z = -G$  si el gradiente "G" es mayor que 0 y se coloca en un intervalo de tiempo definido por  $(t_1, t_2)$ . De la ecuación 2.55, la acumulación de fase debida a los gradientes para un spin en  $z$ , y un tiempo  $t$ , durante la aplicación de un primer gradiente, se modelará matemáticamente como:

$$\phi_G(z, t) = +\gamma G_z(t - t_1) \text{ para } t_1 < t < t_2 \tag{2.60}$$

La precesión en sentido contrario a las manecillas del reloj es correcta para un gradiente negativo y una coordenada en  $z$  positiva. Cuando se tiene que  $G_z = G$  durante un intervalo ahora denominado  $(t_3, t_4)$  el comportamiento de la fase de manera relativa al eje  $z$  y para cualquier momento dentro del intervalo anteriormente descrito será:

$$\phi_G(z, t) = +\gamma G_z(t_2 - t_1) - \gamma G_z(t - t_3) \tag{2.61}$$

Como no hay cambios de fase durante el tiempo en que el gradiente aplicado es 0, entonces se tiene un eco de gradiente. La fase de la ecuación 2.61 vuelve a ser 0 y por la figura 2.8 se muestra que es en:

$$t = t_3 + t_2 - t_1 \equiv T_E$$

para cualquier posición en  $z$ . El eco hace referencia al segundo gradiente aplicado en el que el área del gradiente cancela al primer gradiente lo que supone una condición para mencionar la secuencia *eco de gradiente*.

$$\int G(t) dt = 0$$

En otras palabras, la secuencia eco de gradiente aparece en el momento que el gradiente de la integral es 0. De acuerdo con la figura 2.8, se puede tomar a elección el segundo intervalo de tiempo para el siguiente eco, si se quiere colocar en el centro se puede definir una nueva coordenada temporal con su origen en el eco, es decir  $t' \equiv t - t_3 - (t_2 - t_1) = t - T_E$ , por lo que la fase del gradiente se puede escribir como:

$$\phi_G(z, t) = -Gzt' \text{ para } -\frac{t_4 - t_3}{2} < t' < \frac{t_4 - t_3}{2}$$

En el intervalo mostrado anteriormente, los datos son adquiridos con esta secuencia. Además, la señal a lo largo del eje z, se puede reescribir tomando en consideración el intervalo anterior, de modo que quedaría de la forma:

$$s(t') = \int dz \rho(z) e^{-i\gamma G_z t'}$$

$$= \int dz \rho(z) e^{-i2\pi k(t')z} \text{ para } -\frac{t_4 - t_3}{2} < t' < \frac{t_4 - t_3}{2} \quad (2.62)$$

Con  $k = \gamma G t'$  y ñ desplazamiento temporal dirigido hacia  $t'$ , se hace un análisis de la señal para los datos referidos en el centro del segundo gradiente, este gradiente también se conoce como "gradiente de desfase". Los desplazamientos temporales contribuyen para el análisis de una RM. En el espacio k, se tiene un rango de valores tanto negativos como positivos para mediciones realizadas alrededor del eco, el eco del gradiente negativo se utilizó para crear otro eco de gradiente en medio del refase positivo en donde  $k = 0$ . Así, se puede hacer un diagrama del espacio K, como se muestra en la figura 2.9, la señal se verá matemáticamente como:

$$s(k) = \int dz \rho(z) e^{-i2\pi k z} \text{ para } -k_{max} < k < k_{max} \quad (2.63)$$

Siendo el valor de  $k_{max} = \gamma G \frac{t_4 - t_3}{2}$

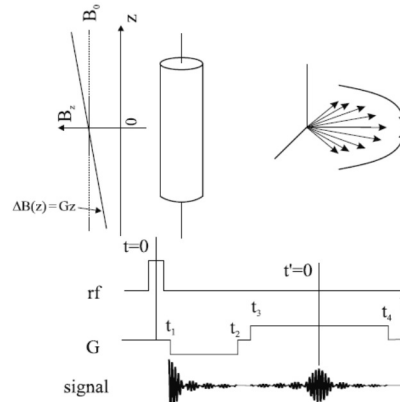


Figura 2.8: Graficas de señal y tiempo para una RM de una dimensión, el campo magnético colocado a la izquierda del cilindro hace referencia al campo que aparece en el intervalo de tiempo  $(t_3, t_4)$ , los espines isocronos estan en un plano igual, pese a las diferencias en sus coordenadas en z, la figura tambien muestra los gradientes positivo y negativo. [6]

Se puede hacer también una reconstrucción de imágenes mediante los valores positivos del espacio k, usando imágenes FID con reconstrucción parcial de Fourier. Para el caso actual, la imagen se realiza con un unico gradiente positivo. En la figura 2.10 se muestra un diagrama del espacio K para la formación de imagenes FID. Los límites del espacio de muestra serán  $0 < k < k_{max}$  para un gradiente constante positivo y un tiempo de muestreo  $T_s$

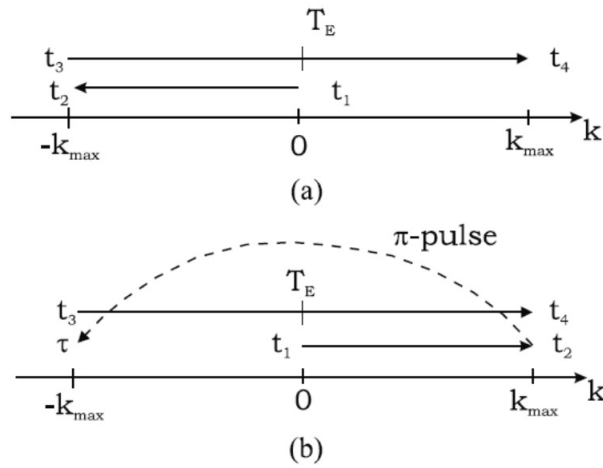


Figura 2.9: Espacio k para secuencia eco de gradiente de forma básica en (a). mientras que en (b) se muestra la secuencia eco-spin donde los gradientes son positivos. [6]

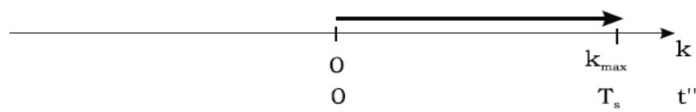


Figura 2.10: El espacio k para una RM de prueba en la formación de imágenes FID. El tiempo  $t''$  se define como  $t'' = t - t_1$  y comienza desde que los gradientes aparecen. El tiempo total de la adquisición de señales se toma como  $T_s$ . Solo se considera la mitad del espacio k y no hay progrsión a través del espacio k hasta el tiempo  $t_1$ . [6]

## 2.2. Tejido hepático y su Relación con el Hierro

En una persona sana, el hierro se encuentra depositado en el hígado, bazo y en el sistema musculoesquelético en forma de ferritina. Esta a su vez, se encuentra en casi todas las células. La ferritina sérica es un indicador que muestra una cantidad de hierro que se localiza en los depósitos de hierro.

El hierro almacenado es una forma no tóxica e inactiva que será utilizada de manera inmediata para poder satisfacer las necesidades de las células.

### 2.2.1. Anatomía y fisiología del hígado

El hígado, en condiciones saludables, tiene características diferenciadas de otros órganos como su color marrón, tener una superficie lisa, pesar 1.4 kg para mujeres y 1.8 kg en hombres lo cual corresponde a un 2% del peso total de un ser humano adulto. Recibe sangre de la vena porta en un 80% del total y el otro 20% de la arteria hepática que corresponde a sangre oxigenada; la vena porta proviene del bazo y el intestino. La vena porta es una unión entre la vena mesentérica superior y la vena esplénica. (Ver figura 2.11).

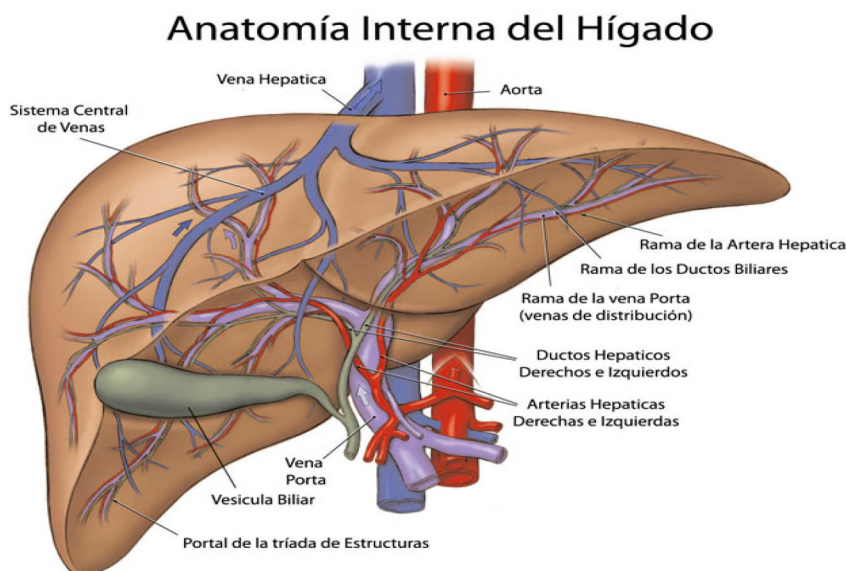


Figura 2.11: Anatomía hepática. Tomado de *Anatomía hepática*, por Médica Capacitaciòn, 2017, <https://diplomadomedico.com/anatomia-hepatica/>

De acuerdo con el Dr. Claude Couinaud, los sistemas de aporte sanguíneo hepático, las venas hepáticas y la vena porta no tienen una coincidencia entre ellas, pero, se tiene una interdigitación dejando áreas de irrigación y drenaje biliar independiente entre ambos sistemas de venas, estas áreas son 8 segmentos hepáticos que funcionan de forma independiente. Cada uno de estos segmentos cuenta con una rama de arteria hepática, vena porta y posee un conducto biliar. La segmentación de este órgano dadas las condiciones previamente expuestas se tiene la figura 2.12:

El segmento I, también llamado lóbulo caudado está ubicado en la parte posterior del hígado. Tanto los segmentos II y III se ubican, respectivamente, en la parte posterior y anterior del lóbulo izquierdo; en conjunto, los segmentos se conocen como: lateral izquierdo o lóbulo izquierdo topográfico. El segmento IV recibe por nombre "Segmento medial del lóbulo izquierdo" y está dividido en 2 partes pues el segmento IVA corresponde a los segmentos superiores (en conjunto con los seg-

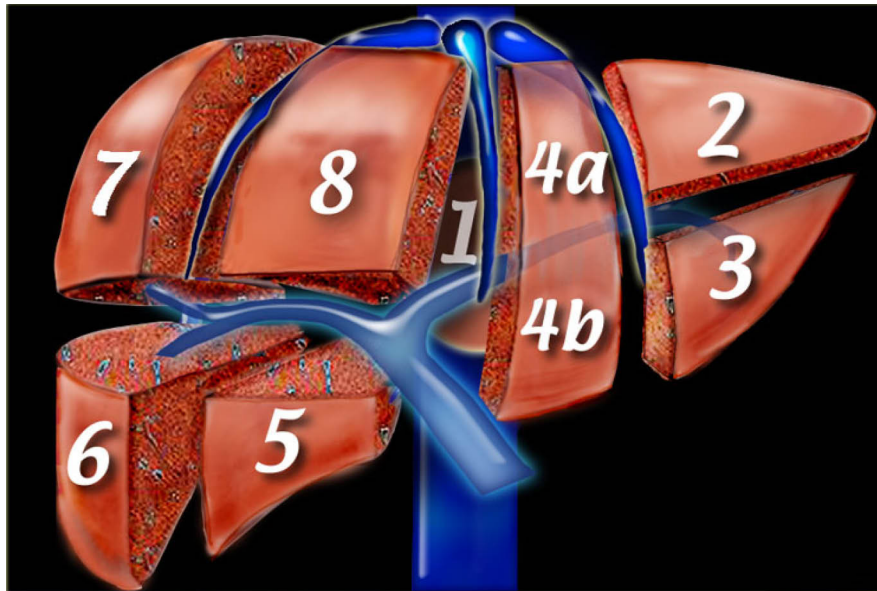


Figura 2.12: Segmentación del hígado. Tomado de *Segmentación hepática. Evaluación ecográfica*, por Mèdica Capacitaciòn, 2016, <https://diplomadomedico.com/segmentacion-hepatica-evaluacion-ecografica-2/>

mentos II, VII y VIII) mientras que el segmento IVB corresponde a los segmentos inferiores (junto con el segmento V); el segmento IV está ubicado entre las venas hepáticas mediana e izquierda. Los segmentos V y VIII forman la parte anterior, mientras que la parte posterior es formada por los segmentos VI y VII del lóbulo hepático funcional derecho.

El hígado tiene el trabajo importante de servir como filtro para la sangre que proviene del sistema digestivo y la sangre dentro del organismo. A través de la vena porta, la sangre derivada de los intestinos llega; esta misma es filtrada en láminas de células hepáticas y termina drenándose hacia las venas hepáticas que desembocan en la vena cava inferior. En el tránsito por las láminas hepáticas se presentan cambios clínicos tales como la formación de bilis en la parte trasera de cada lámina, esta bilis pasa al intestino a través del colédoco.

En cada lóbulo del hígado se producen brechas entre células endoteliales y el plasma en contacto con el estrecho de las células. En la arteria hepática la sangre fluye hacia los sinusoides. Asimismo, las venas centrales forman a las venas hepáticas para drenar hacia la vena cava inferior. El flujo de la sangre a través del lóbulo hepático desde la vénula portal hasta la vena hepática central es de 8.4s en promedio.

### 2.2.2. Funciones del hierro en el organismo

El hierro que se consume suele aparecer en su forma hemo la cual tiene como composición una molécula de protoporfirina IX y un ion de  $Fe^{2+}$ , en su forma no hemo se tiene la forma férrica  $Fe^{+3}$  o ferrosa  $Fe^{+2}$ , estas 2 formas de hierro (férrico y ferroso) se pueden encontrar en alimentos de origen animal, cereales, verduras, legumbres y frutas. La forma ferrosa es la que se absorbe en el cuerpo mientras que en la alimentación se encuentra en su mayoría la forma férrica. Las actividades gástricas hacen que el hierro se disuelva, permitiendo que se formen complejos solubles con el ácido ascórbico y otras sustancias que reducen al  $Fe^{+3}$  en  $Fe^{+2}$ .

Casi toda la absorción del hierro es realizada por el duodeno (ver figura 2.13). Para la transportación del hierro de tipo  $Fe^{+2}$  hacia los enterocitos se realiza mediante el transportador de metal divalente 1 o DMT1 por sus siglas en inglés. Una parte de este es almacenado en una molécula llamada

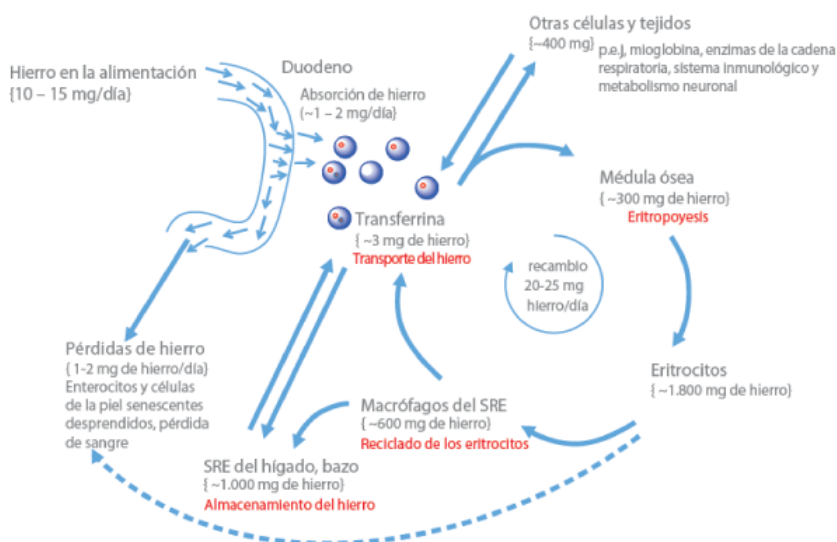


Figura 2.13: Absorción de hierro por enterocitos en el duodeno. Tomado de *Metabolismo del hierro: Absorción, transporte, reciclado y almacenamiento*, por CardioTeca, 2023, <https://www.cardioteca.com/metabolismo-del-hierro.html>: :text=El %20metabolismo %20del %20hierro %20incluye,la %20degradaci %C3 %B3n %20de %20los %20eritrocitos

ferritina y la parte restante se mueve se mueve fuera de los enterocitos por la ferroportina 1 la cual es un transportador de la membrana basolateral. La proteína hefaestina forma un vínculo con la ferroportina 1 para su transporte; si bien la hefaestina no es un transportador si hace que la ferroportina 1 se mueva con mayor facilidad en el transporte baso lateral. Una vez adentrándose en el plasma, el  $Fe^{+2}$  se convierte en  $Fe^{+3}$  y se une con la proteína transferrina, la cual, se encarga de transportar el hierro. Las condiciones normales de los valores de transferrina es de una saturación de 35% mientras que los valores para la concentración plasmática del hierro para hombres es de  $\frac{130\mu m}{100ml}$  o  $\frac{23\mu mol}{L}$ , mientras que en mujeres los valores son de  $\frac{110\mu m}{100ml}$  o  $\frac{19\mu mol}{L}$ .

El 70% del hierro del organismo se encuentra en la hemoglobina 3% en mioglobina y el restante en la ferritina que esta presente en órganos como el cerebro, bazo o hígado, este ultimo, al absorber y distribuir hierro en todo el cuerpo humano para la formación de hemoglobina lo vuelve uno de los depositos más grandes de hierro en el cuerpo por lo que el control del mismo y su conteo puede ser adquirido desde este órgano.

### 2.2.3. Pérdida y acumulación de hierro en el tejido hepático: causas y consecuencias

En un organismo adulto, las pérdidas de hierro son mínimas; estas a su vez no son reguladas y las reservas de hierro en el cuerpo son controladas por la rapidez con la que se absorbe desde el intestino, como se mencionó anteriormente, el duodeno es el principal encargado de realizar esta absorción). Las personas del género masculino pierden en promedio  $\frac{0,6mgFe}{da}$  a través de la excreción. En una persona del género femenino, los valores de perdida de hierro son del doble en comparación del masculino ya que la pérdida adicional de hierro durante la menstruación es un factor a considerar.

El consumo de hierro diario es, en promedio de 20 mg, sin embargo, la cantidad absorbida y funcional solo representa el 6% de la ingerida, esto es debido a diversos factores que afectan la

absorción como el ácido fólico, el cual, reacciona con el hierro y se vuelve insoluble en el intestino, por lo que ya no se puede absorber.

Entre las consecuencias por los trastornos de absorción del hierro existe la anemia ferropénica la cual se origina por una deficiencia de hierro. Por otra parte, el exceso o sobrecarga de hierro produce hemosiderosis la cual es una acumulación en tejidos de hemosiderina. Entre las consecuencias, una de ellas es la hemocromatosis cuyas características como síndrome son: cirrosis hepática, lesión pancreática, atrofia gonadal y carcinoma hepático.

La hemocromatosis puede ser hereditaria o adquirida. Para la parte hereditaria, suele ser motivo de su aparición la mutación del gen HFE principalmente en población caucásica. El gen está localizado en el brazo corto del cromosoma 6, ligado con el antígeno leucocitario humano-A (HLA-A). La hemocromatosis por la mutación HFE se caracteriza por absorber cantidades en exceso de hierro ya que el gen normalmente inhibe la expresión de los transportadores duodenales que participan en la absorción del hierro. Entre las causas de la aparición de hemocromatosis, se tiene la destrucción crónica de eritrocitos, tener transfusiones repetidas en un periodo corto de tiempo, al no responder al tratamiento de anemia, el exceso de hierro superará los niveles de regulación de hierro y tendrá aparición la hemocromatosis. [7]

## 2.3. Diagnóstico de Sobrecarga de Hierro en el Hígado

La sobrecarga de hierro en tejido hepático (Hígado) es un objeto de estudio que, mediante diversas técnicas, se ha logrado medir para posteriormente diagnosticar enfermedades como la hemocromatosis hereditaria o la talasemia.

### 2.3.1. Métodos tradicionales de medición de hierro hepático

#### Muestras de sangre.

Entre los métodos más comunes para la medición de hierro existen las pruebas de hierro, las cuales son muestras de sangre tomadas al paciente para poder determinar los niveles de hierro en el cuerpo. Entre las pruebas que se realizan existen:

- Prueba de hierro en suero: Medición de hierro en sangre
- Prueba de transferrina: Medición de la proteína que transporta el hierro hacia todo el cuerpo
- Capacidad total de unión al hierro (TIBC): Medición de la unión del hierro con la transferrina para tener un transporte y deposición correctas según sean las necesidades del organismo.
- Prueba de ferritina: Esta prueba cuantifica el hierro total almacenado en el cuerpo. [9]

Para una prueba de ferritina, los valores normales en el cuerpo humano oscilan entre los 33 a 193  $\frac{\mu\text{g}}{\text{dL}}$  como límites clínicos. Un valor menor a 33  $\frac{\mu\text{g}}{\text{dL}}$  podría desencadenar como consecuencia anemia, por otra parte, un valor mayor a 193  $\frac{\mu\text{g}}{\text{dL}}$  puede ser diagnóstico de hemocromatosis, talasemia o cirrosis no alcohólica.

## Biopsia

Si se tiene sospecha de posible sobrecarga de hierro se puede solicitar una biopsia, específicamente, una biopsia hepática percutánea. Las biopsias de este tipo son un procedimiento para tomar muestras de células hepáticas para realizar análisis de laboratorio.[10] Con estas pruebas se puede determinar que tejido o tejidos están siendo afectados en el hígado. Usualmente, el uso de esta técnica es para medir el daño realizado en el hígado, mediante un proceso llamado "estadificación". La figura 2.14 muestra un esquema básico de este tipo de biopsia.

Asimismo, existe la biopsia hepática laparoscópica, la cual, se utiliza para obtener muestras de

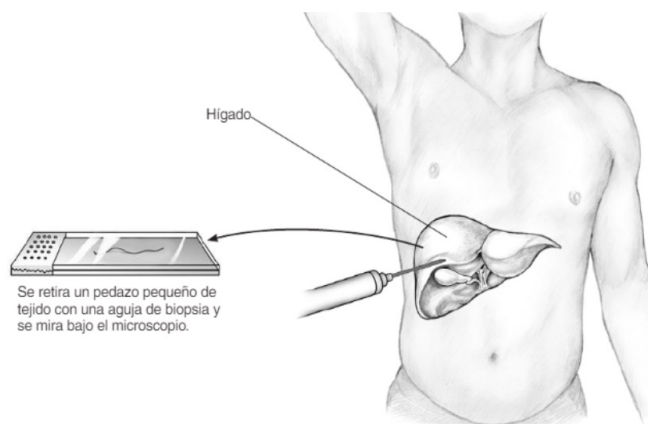


Figura 2.14: Esquema de una biopsia hepática percutánea. [11]

una o varias zonas específicas del tejido hepático cuando hay una sospecha de cáncer o infección. El período de recuperación de una biopsia hepática es de 1 a 2 días. En este periodo, los pacientes no pueden realizar actividad física intensa o ejercicio.

El riesgo de una hemorragia es de 1 en 12000, pero, si ocurre se tiene que realizar un procedimiento conocido como embolización; este procedimiento es asistido por angiografía y puede acompañarse con transfusiones de sangre.

Otro riesgo poco probable es la punción de otros órganos internos, infección y propagación de células cancerosas (en caso de tener) mediante el proceso de metástasis. [11]

### 2.3.2. Limitaciones y desafíos en la medición de hierro hepático

Entre las limitaciones importantes radica el tiempo que toma realizar estas pruebas, pese a que sean solo días para la obtención de resultados en la prueba de biopsia y horas en los análisis de sangre, los resultados tienen que ser interpretados y, de ser necesario, realizar más estudios para realizar un diagnóstico preciso, ya que, un nivel de hierro dentro de los límites clínicos no necesariamente son un indicador de descarte para la anemia, pues, existen diversas variaciones de esta, la cual toma en consideración a las globinas.

Otro punto limitante radica en el uso de procedimientos quirúrgicos para la realización de la biopsia, el tiempo de recuperación y obtención de resultados se podría mejorar mediante algunas otras técnicas, una de ellas es el uso de las resonancias magnéticas para la cuantificación de hierro en hígado como método no invasivo.

## 2.4. Resonancia Magnética en la Medición de Hierro Hepático

El uso de resonancias magnéticas para medición de hierro en el hígado se ha desarrollado desde hace más de 30 años cuando Stark y colaboradores [13], realizaron estudios de resonancia magnética en ratas de laboratorio simulando problemas y enfermedades humanas, creando una sobrecarga de hierro mediante ingesta excesiva de tocino con 2.5% de carbonilo purificado el cual es un tipo de hierro altamente puro utilizado para prevenir y tratar deficiencias de hierro. Este método fue utilizado durante 4 meses en las ratas para producir depósitos hepatocelulares en los roedores. Los resultados de estas condiciones fue una sobrecarga de hierro férrico en los roedores que al ser examinados mediante RM se detectó el exceso de hierro reduciendo el contraste en la RM, obteniendo correlaciones entre la cantidad de hierro en el roedor y su tiempo de relajación  $T_1$  de 0.98, La examinación histológica reflejó que las distribuciones de hierro en el hígado tenían similitudes con los pacientes que eran diagnosticados con hemocromatosis. Los valores de control para el tejido de agua no cambió pero si se veían diferencias de intensidad significativa entre las ratas que estaban sanas y las ratas que tenían una sobrecarga de hierro inducida. [13] La figura 2.15 muestra como se ve el hierro en el hígado de una rata, esta a su vez no presentó inflamación, degeneración hepatocelular o necrosis. Por otra parte, en la misma figura se muestran los hígados de los especímenes de control (marcados con la letra "C") y los especímenes con sobrecarga de hierro inducida (marcados como "I1" y "I2"), los que tenían la sobrecarga de hierro presentaron intensidades muy bajas en la imagen, mientras que los sanos mostraban intensidades similares entre sí, haciendo una distinción muy marcada entre ratas sanas y no sanas. Por otro lado, Wood y colaboradores [12] propusie-

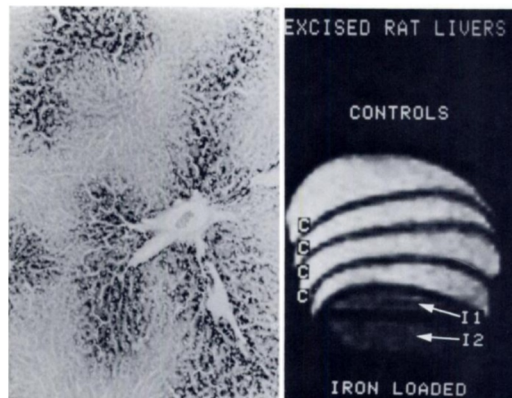


Figura 2.15: Sobrecarga de hierro en (a) el hígado de un espécimen de rata con 1.8mg de hierro por gramo de masa, se presenta deposición en los hepatocitos como granulos negros. (b) Imágenes hígado en técnica de RM de 4 especímenes de control (C) y 2 con sobrecarga de hierro (I1,I2). Los contenidos de hierro en I1 es de 3.9mg y en I2 de 4.0mg, se observa una diferencia de intensidades significativa entre los especímenes de control y los que tienen excesos de hierro. [13]

ron configuraciones gradiente de eco  $T_2$  para cuantificar hierro mediante un ajuste lineal. Entre los pacientes, se encontraban personas sanas y con hemocromatosis hereditaria que se realizaron estudios de biopsia, por lo que el diagnóstico de sobrecarga de hierro tendría en consideración a la RM contra la biopsia.

Las RM fueron tomadas en un escáner de 1.5T con secuencia de gradiente de eco, con un tiempo eco espaciado en 0.25 milisegundos, con un intervalo de tiempo de 0.8 a 4.8 ms. Asimismo, utilizaron un tiempo de repetición de 0.25ms, un campo de visión de 48x24 y un grosor de corte de 15 mm. Se consideró como suposición un ajuste lineal que tuviese una variable, los tiempos de eco, para

poder ajustar una ecuación exponencial de tipo:

$$S(TE) = Ae^{(-TE^*R2^*)} + C \quad (2.64)$$

Así, lograron ajustar una cuantificación de hierro de la forma:

$$[Fe]_{R2-SP} = (29,75 - \sqrt{900,7 - 2,283R2})^{1,424} \quad (2.65)$$

La ecuación 2.65 se interpreta como una relación entre el valor de hierro medido en una biopsia y el arrojado por la secuencia  $T_2$ . Mediante una prueba de correlación R se obtiene un valor de 0.98 y se llega a la conclusión de que el valor del hierro se puede obtener mediante una biopsia o una secuencia de  $T_2$  e incluso  $T_2^*$ . Se muestra reproductibilidad para este trabajo con un sesgo del 5%, asimismo, se mostró una técnica no invasiva para la medición de hierro en hígado.

Un trabajo similar fue el realizado por Pierre y colaboradores [18] en donde realizan un estudio en personas y maniquies (objetos para caracterizar y calibrar secuencias) para poder cuantificar el hierro en hígado mediante cortes de imágenes y la detección de talasemia. Los métodos realizados varían del realizado por Wood et al. [12] pues, se utiliza la secuencia spin-echo en un escáner de 1-5T en técnicas multicorte. Tiempo de repetición de 2500 ms, TE de 6,9,12,15 y 18 ms y cortes de 5 mm. Un tamaño de matriz de 256mm. La técnica utilizada fue la T2 y  $T2^*$  para pasar a  $T_2$  y  $T_2^*$ . Entre los resultados presentados se menciona que los coeficientes de correlación para la medición de hierro en hemocromatosis hereditaria no eran relevantes. Sin embargo, en el diagnóstico de talasemia se encontró una correlación del 95% en la cual, esto sirve como una propuesta para poder cuantificar hierro en hígado. Una propuesta que usa ajustes lineales también.

### 2.4.1. Secuencias de RM utilizadas para medir hierro hepático (relajometría)

Los métodos para evaluación de concentraciones de hierro en hígado se consideran mejores si se miden en tiempos de relajación T2 [17]. De acuerdo con Alustiza y colaboradores [16] el valor en T2 es definido como el tiempo requerido para que la magnetización tenga un alcance de al menos 37% de su valor original. La medición en T2 permiten trazar una curva de decaimiento exponencial. Al realizar las secuencias de eco de gradiente se tendrá una medición en el tiempo de relajación que será denominado como  $T2^*$ . De acuerdo con los autores, la medición mediante relajometría mostró una correlación mayor al 95% poniendo como técnica de oro a la biopsia, este valor representaba una correlación muy buena para cuantificar hierro, la desventaja observada radicaba en que las secuencias espín-eco utilizadas tomaban varios minutos y eran costosas, dando como trabajo relevante el de Wood et al. [12] pues, sus tiempos cortos y excelente correlación permitían reproducir los estudios en relajometría para cuantificación de hierro y los ajustes lineales eran reproductibles en otros escaneres. Cabe destacar, que el parámetro más importante en relajometría es el tiempo de eco TE, pero, no implica que tenga que ser mayor, implica que debe ser menor a 1ms para poder identificar sobrecargas de hierro altas, como el hierro es un metal que afectará al campo magnético inducido, el tiempo de adquisición debe ser menor al tiempo que le tomará al hierro para comenzar a aparecer como artefacto en la imagen, es decir, que el hierro tenga el tiempo suficiente para hacer variar el campo magnético inducido y a sus vecinos cercanos.

Una limitación matemática importante es no tener un estándar para el modelado de las cuantificaciones de hierro, ya que algunos autores usan modelos de ajuste lineal mientras que otros utilizan modelos monoexponenciales con factores de correlación y truncamiento haciendo que en la práctica algunos resultados sean irreproducibles. Asimismo, los autores mencionan que es posible hacer mediciones de hierro con el radio de intensidad de señal, pero este método se ve afectado cuando existe una saturación en los niveles de hierro (mayor a 250  $\mu\text{mol}$ ) [16]

## 2.5. Inteligencia Artificial (IA), Machine Learning(ML) y Aprendizaje Profundo (Deep Learning)

La palabra "artificial" en Inteligencia Artificial (o por sus siglas en inglés AI) significa que es diseñada y creada por el ser humano. La Inteligencia Artificial (ahora en adelante llamada AI) es una disciplina que estudia y desarrolla teorías, técnicas y sistemas usados para realizar simulación de la inteligencia humana. El termino fue acuñado por John McCarthy en 1956, que lo definió como: "La ciencia e ingeniería que vuelve inteligentes a las máquinas, especialmente inteligentes mediante programas computacionales". Las premisas de la tecnología de AI es volver capaces a las máquinas para aprender a partir de datos recolectados, y tomar "decisiones" como lo harían los humanos. Es un campo interdisciplinario que toma en consideración más campos.

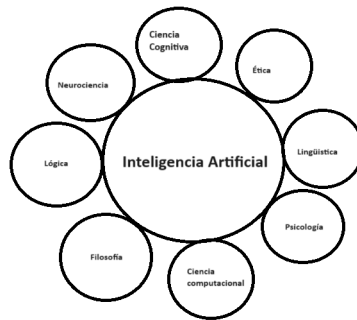


Figura 2.16: Esquema de campos necesarios para el desarrollo de IA.

Por otro lado, el Machine Learning (ML) o aprendizaje máquina se puede entender de diversas formas. Tom Mitchell define al ML como: "Se dice que un programa informático aprende de la experiencia (E) con respecto a una clase de tareas (T) en una medida de rendimiento (P) si su rendimiento en tareas (T), medido por P, mejora con la experiencia (E)." [20]

La definición anterior, aunque simple y abstracta, fue la pionera para poder comprender el aprendizaje máquina. Asimismo la connotación y extensión de la definición del ML cambian a través del tiempo, por lo que definir de forma simple o clara no es fácil.

¿Cómo se relaciona la IA y el ML? La Inteligencia Artificial es una disciplina que que estudia y desarrolla teorías, técnicas y sistemas aplicados usados para simular y entender la inteligencia humana. El ML se refiere a la habilidad que tienen las computadoras para aprender, simular o implementar un artificio del comportamiento humano para adquirir nuevo conocimiento, habilidades y actualizar estructuras existentes de conocimientos para hacer una mejora de rendimiento.

Podemos comprender que el campo extendido es la inteligencia artificial que tiene como elementos básicos: datos, algoritmos, poder de computo y aplicación en escenarios, mientras que el ML usa las habilidades específicas del computo para la creación de nuevo conocimiento y uso de algoritmos.

Como ultimo campo, el aprendizaje profundo (Deep Learning o DL por sus siglas en inglés) es un campo de búsqueda del ML y es originado de las redes neuronales artificiales (RNA). El DL se basa principalmente en la simulación de neuronas interconectadas para interpretar datos, como imágenes, voz y texto.

El ML tiene la capacidad de resolver diversos problemas, de los cuales, existen 3 comunes:  
1. Clasificación: Para especificar una de las  $k$  categorías de entrada, el algoritmo de aprendizaje

2.5 Inteligencia Artificial (IA), Machine Learning(ML) y Aprendizaje Profundo (Deep Learning)

genera una función  $f: R^n \rightarrow (1, 2, \dots, k)$ . Por ejemplo, los algoritmos de clasificación de imágenes por características que distinguen unas de otras.

2. Regresión: El programa predice la salida para una entrada dada. Los algoritmos de aprendizaje suelen dar como salida una función  $f: R^n \rightarrow R$ .

3. Clusterización o agrupación: Basandose en similitudes, el programa agrupa una gran cantidad de datos sin etiquetas en diversas clases. Los datos de una misma clase tienen una similitud más grande que los datos de otras clases.

Tanto clasificación como regresión son los tipos de algoritmos de predicción que más se utilizan. La salida de un clasificador es un valor discreto, mientras que, para la salida de una regresión es un valor continuo.

2.5.1. Tipos de Machine Learning

El ML puede verse como una forma de resolver una tarea mediante el aprendizaje, estas tareas por aprendizaje se pueden clasificar en:

**Aprendizaje supervisado:** Se toma un conjunto conocido o “etiquetado” de muestras y se genera un modelo óptimo para generar predicciones. Después, el modelo entrenado asigna todas las entradas a las salidas y realiza un “juicio” simple sobre las salidas. De este modo, se clasifican los datos conocidos.

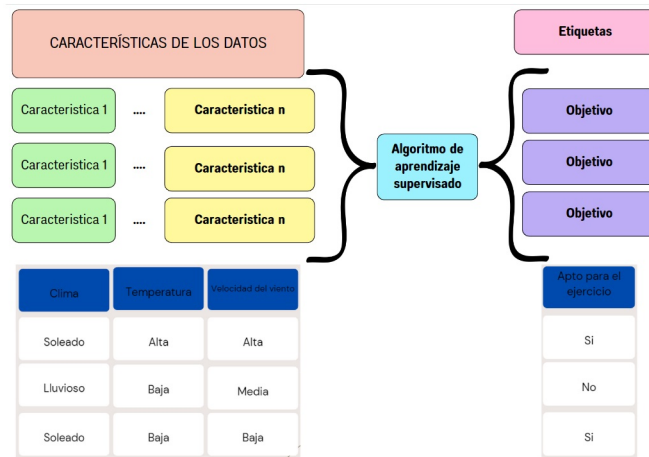


Figura 2.17: Esquema de algoritmo para aprendizaje supervisado.

**Aprendizaje no supervisado:** Se construye un modelo basado en datos de entrada no etiquetados. Un ejemplo es la clusterización que agrupa objetos basándose en similitudes. Los algoritmos de aprendizaje no supervisado modelan las muestras similares, calculan la similitud entre las muestras nuevas y las existentes y finalmente las clasifican por similitud.

**Aprendizaje semi-supervisado:** Se entrena un modelo mediante la combinación de una pequeña cantidad de datos etiquetados y una cantidad aun más grande de datos sin etiquetar.

**Aprendizaje por refuerzo:** Los sistemas de aprendizaje aprenden el comportamiento de su entorno para maximizar el valor de la función de señal de recompensa (también conocido como refuerzo). El entorno proporciona señales escalares de refuerzo para evaluar

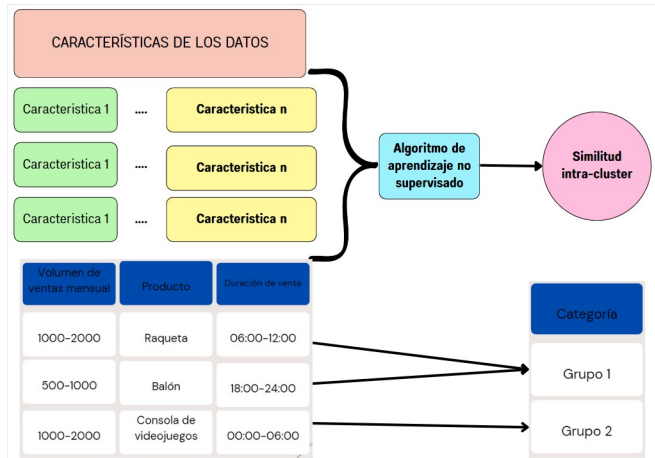


Figura 2.18: Esquema de algoritmo para aprendizaje no supervisado.

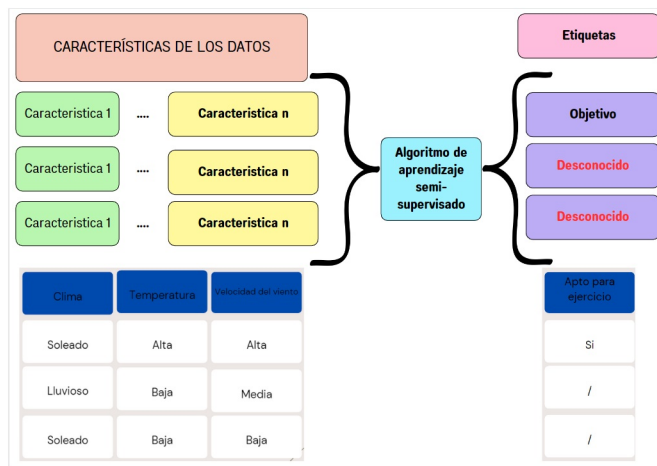


Figura 2.19: Esquema de algoritmo para aprendizaje semi-supervisado.

sus acciones.

### 2.5.2. Proceso de Machine Learning: Conceptos y consideraciones

Debido a la cantidad de algoritmos, procesos y herramientas en el ML, en esta subsección se mencionarán solamente las que se utilizarán en este trabajo.

El proceso del ML se basa en un preparado de datos, depuración, modelado, iteraciones, etc. En la figura 2.21 se puede observar un esquema bastante simplificado del Machine Learning, sin embargo, es necesario mencionar que significa cada proceso y su importancia dentro del ML.

#### Conjunto de datos

Una colección de datos usados para tareas a realizar en ML, cada parte de este conjunto de datos se conoce como “muestra”. La elección de un conjunto de datos y muestras que lo conforman no es aleatoria, pues, cada una de estas muestras tiene que reflejar particularidades

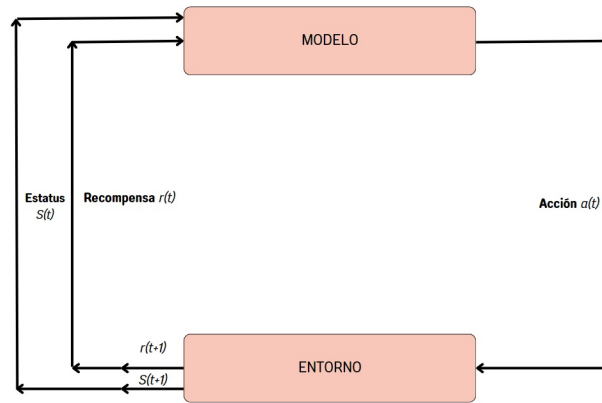


Figura 2.20: Esquema de algoritmo para aprendizaje por refuerzo.

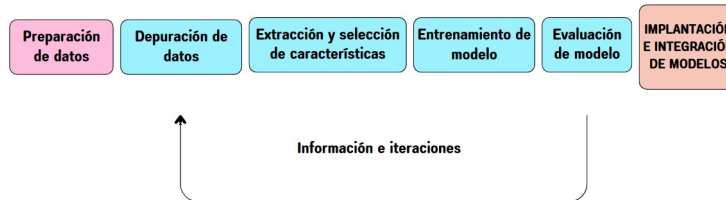


Figura 2.21: Proceso de Machine Learning.

o “características”. Antes de un entrenamiento, el conjunto de datos se tiene que separar en dos partes; un conjunto de entrenamiento y un conjunto de prueba. En ocasiones, se emplea un tercer conjunto llamado “conjunto de validación”, este conjunto servirá de refuerzo al algoritmo para saber si los resultados obtenidos se adaptan al problema planteado o no.

- **Conjunto de entrenamiento:** Este conjunto es el que se usa en el proceso de “entrenamiento”, cada muestra es llamada “muestra de entrenamiento”. Cuando se hace referencia “aprendizaje” (o entrenamiento), es necesario hacer énfasis en que es el proceso de construir un modelo a partir de los datos. El conjunto de entrenamiento suele ser el 75 % del conjunto de datos totales, pues, es necesario tener la mayor cantidad de muestras para que el modelo construido tenga un mejor aprendizaje.
- **Conjunto de prueba:** Usado para el proceso de prueba, cada muestra se conocerá como “muestra de prueba”. Probar se refiere al proceso en que el modelo entrenado es usado para predecir muestras que no fueron usadas durante el entrenamiento. El conjunto de prueba tiene

el 25 % de muestras del conjunto total. Si se usa un conjunto de validación se puede tomar parte de este 25 % para no disminuir las muestras que usará el conjunto de entrenamiento.

### Preprocesamiento de datos

Los datos son cruciales para los modelos y sus capacidades para la resolución de problemas. Los buenos modelos requieren de buenos datos.

Para el preprocesamiento de los datos se tiene la depuración de datos, la estandarización de estos o la reducción de dimensiones, cada uno con características particulares.

- **Estandarización de datos:** Estandarizar los datos reducen el ruido y mejoran la precisión del modelo.
- **Reducción de dimensiones de los datos:** Simplificar los atributos de los datos.
- **Depuración de datos:** Encontrar valores anómalos, detección de ruido y datos con sesgos importantes.

### Depuración de datos

Los modelos de ML procesan características, las cuales suelen ser representaciones numéricas de las variables de entrada que pueden ser usadas en el modelo.

En varios casos, solo los datos preprocesados pueden ser usados por los algoritmos. El procesamiento de datos se puede centrar en los siguientes puntos:

- Filtrado de datos
- Selección de datos sesgados o pérdida de características
- Tratamiento de valores erróneos o anormales
- Fusión de datos y características
- Consolidación de datos

Los “datos crudos” usualmente contienen problemas en la calidad, por ejemplo:

- **Incompletos:** Datos incompletos o valores incompletos
- **Ruido:** Datos que contienen puntos anormales, sesgos o registros incorrectos
- **Inconsistencias:** Datos que contienen registros “conflictuantes”, es decir, datos que tienen un formato incorrecto, cifras significativas distintas, errores en formato, etc.

En la imagen 2.22 se muestran datos que tienen inconsistencias, errores ortográficos o valores faltantes. Este tipo de conflictos con los demás datos genera sesgos y podría ser una parte importante para no lograr el objetivo de ML. Una vez preprocesados los datos, estos tienen que ser convertidos a una representación adecuada para los modelos de ML.

- Los datos categóricos se pueden convertir en números para clasificación
- Los datos numéricos convertidos en categóricos para disminuir el valor de las variables
- Procesado de datos en imágenes, conversión a escala de grises, conversión en espacios de colores, etc.
- Normalización y estandarización de características para asegurar que las entradas de las variables estén en un rango de valores

#	Identificador	Género	País	Ciudad	Estudiantes
1	111	H	Francia	Dublin	0
2	222	F	Alemania		15
3	333	F	Ytalia	Roma	23
4	444	A	Italia	Italia	10
5	444	F	Fransia	París	-3
6	555	M	Alemania	Berlin	15

Diagrama de anotaciones:

- Datos faltantes:** Señala a la celda vacía de la Ciudad en la fila 2.
- Cantidades incorrectas:** Señala a la celda con el valor -3 en la columna Estudiantes de la fila 5.
- Datos duplicados:** Señala a las celdas con el identificador 444 en las filas 4 y 5.
- Valor inválido:** Señala a la celda con el género 'A' en la fila 4.
- Errores ortográficos:** Señala a las celdas con 'Ytalia' y 'Fransia'.

Figura 2.22: Conjunto de datos en crudo.

- Aumento de características a partir de las existentes en el conjunto de datos

Asimismo, los conjuntos de datos que contienen diversas características, no son necesariamente útiles todas o pueden llegar a ser irrelevantes para los valores y tareas que el modelo tiene destinado realizar.

Una elección de características adecuada permite obtener los siguientes aspectos:

- Modelos simplificados para una interpretación sencilla
- Cortos tiempos de entrenamiento
- Omitir dimensiones extra
- Tener modelos generalizados y evitar sobreajustes

### Métodos de selección de características-Filtrado

Un método de filtrado es independiente del modelo durante la selección de características. Para esta forma de selección de características, se busca una correlación entre la característica y la etiqueta que le corresponde o el atributo que contribuye al objetivo de la etiqueta. La elección de qué característica tiene mejor correlación se atribuye a la estadística que puntúa que tan correlacionadas están las características con las etiquetas. Los métodos más comunes incluyen al Coeficiente de correlación de Pearson, el coeficiente  $\chi^2$  y la información mutua. Cabe recalcar que los métodos de filtrado tienden a seleccionar variables (características) redundantes ya que estos métodos no consideran una relación entre características.

### Métodos de selección de características-“Wrapper”

El método “wrapper” o de “envoltura” trata de seleccionar características como un problema de búsqueda óptimo y evalúan y comparan diferentes combinaciones, es decir, se hacen búsquedas mediante ensayo y error. El método más común es la eliminación de características de forma recursiva que se basa en realizar pruebas hasta encontrar las características óptimas para un mejor resultado.

Una limitación importante de este método es que cada vez que se escogen características de un subconjunto del conjunto de datos original, se creará un nuevo modelo, ocasionando una carga computacional intensa. Además, la elección final de un conjunto de características óptimas

generarán un modelo específico, creando que la tarea a realizar o resolver de parte del algoritmo de ML, se vuelva particular y bajo condiciones especiales.

Entonces, ¿Cómo saber si un modelo es bueno?

- **Generalización:** La precisión de las predicciones basadas en los datos actuales
- **Capacidad de explicar resultados:** Los resultados predichos deben ser sencillos de explicar e interpretar
- **Tiempo de predicción:** El tiempo necesario para realizar una predicción

### Efectividad de un modelo

Los modelos de ML tienen como objetivo adaptarse a problemas, ser entrenados y entregar resultados factibles ante muestras nuevas, no solo las utilizadas para el entrenamiento. “Robustecer” el modelo o hacer una generalización de capacidades es el objetivo para poder extender la capacidad de aprendizaje de un modelo para ser aplicado a nuevas muestras. Por otro lado, además de esperar un modelo robusto, también se espera que el modelo tenga una forma de ajustarse ante varias funciones; con suficiente capacidad para encargarse de una tarea compleja y con un volumen grande de datos, el algoritmo resultante sería el óptimo. Una capacidad insuficiente no puede encargarse de tareas complejas por lo que el modelo puede caer en una “subadaptación” o “subajuste”, es decir, una falta de aprendizaje dadas las características para resolver un problema. Por otra parte, los modelos con una capacidad grande para resolver problemas complejos pero más de la necesaria para resolver la tarea requerida recae en un “sobreajuste” lo que implica un aprendizaje en exceso, incluidos los errores, valores atípicos y “ruido” de las muestras.

En la figura 2.23 se muestra un esquema simplificado de los 3 tipos de ajustes que se tienen en el proceso de ML. Asimismo, es importante mencionar que para tener un modelo óptimo debe existir un valor de “error” pequeño. El error se puede comprender como la diferencia entre la predicción mostrada por el modelo de una muestra y el resultado real de la muestra.

Es importante mencionar que al entrenar, aparecerá el “error de entrenamiento” el cual es el error que tiene el modelo con el conjunto de entrenamiento. Asimismo, el “error generalizado” es el error que existe ante nuevas muestras (aparece cuando se usa el conjunto de validación o de prueba una vez el modelo ha realizado su entrenamiento). Teóricamente, para eliminar los errores o volverlos 0 se debe de tener un conjunto de datos infinito y un modelo de ML “perfecto”. Pero, en la práctica un modelo no es perfecto y los datos son limitados.

### Sobreajuste y errores a considerar

Las limitaciones de un modelo de aprendizaje automático se encuentran en sus algoritmos, ajustes de parámetros y métodos de regularización. Estos factores pueden llevar al modelo a tener un ajuste insuficiente o excesivo, dependiendo de cómo haya sido su proceso de aprendizaje. Para determinar si el modelo está afectado por alguno de estos problemas, es necesario analizar los errores del modelo y su significado.

En la figura 2.24 se muestra un esquema que presenta los errores a considerar para poder evaluar un modelo según su error. El centro del esquema representa un modelo que predice perfectamente y no tiene ningún tipo de sesgo o varianza. Cada punto en negro representa un modelo con diferentes capacidades, haciendo que entre más alejado este del centro se vuelva un modelo con mayor varianza y sesgos.

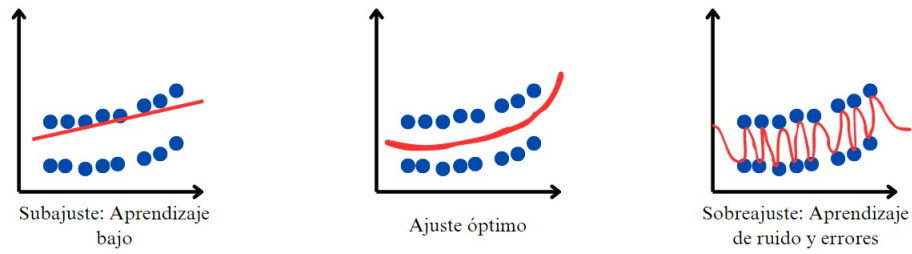


Figura 2.23: Subajuste, ajuste "óptimo" y sobreajuste

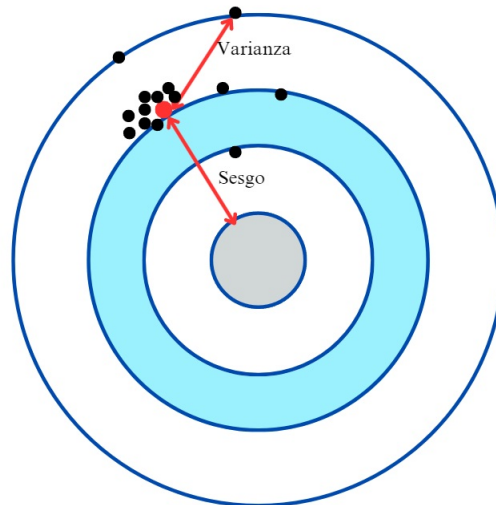


Figura 2.24: Diagrama de Diana que muestra los errores a considerar en un modelo con base en un modelo "perfecto"

Los errores se pueden agrupar en 2 conjuntos; el error debido al “sesgo” o “bias” el cual es tomado como la diferencia entre el valor predicho por el modelo y el valor real que se tiene. Por otro lado, el error debido a la varianza es una cantidad que muestra que tanta desviación tiene una predicción con respecto a la media de las predicciones realizadas por diferentes modelos. La varianza es causada por una pequeña fluctuación o cambio en el conjunto de entrenamiento. En conjunto, estos dos errores y un “error ineliminable” (EI) sirven para calcular el “error de predicción” (EP) el cual es definido como:

$$EP = Bias^2 + Varianza + EI \tag{2.66}$$

El error ineliminable es aquel que aparece cuando un modelo no es perfecto y se tiene un conjunto de datos finito.

Debido a que el EI no tiene forma real de ser eliminado pero si de ser disminuido, las variables a considerar en el EP serán la varianza y el sesgo. Tener un sesgo y una varianza pequeños generan un buen modelo. Si una de las dos variables es relativamente pequeña pero la otra es grande se tendrá un modelo inadecuado. Si se tienen ambas variables en cantidades grandes, se tendrá un modelo pésimo y este debe ser modificado.

Un modelo ideal puede capturar de forma precisa las reglas de los datos de entrenamiento y generalizar con nuevos datos. Pero, esto no se puede realizar emn un modelo de forma simultánea.

Para disminuir los errores, se puede aumentar la complejidad de los modelos; mientras más complejo sea un modelo, mas pequeño será el error de entrenamiento. No obstante, es necesario destacar que al mejorar la complejidad del modelo el error del conjunto de prueba también disminuirá, pero, se formará una curva convexa que aumentará el error entre más complejo se vuelva el modelo. La gráfica 2.25 lo ejemplifica. Esto es debido a que un modelo complejo en fase de entrenamiento es más sensible y un cambio mínimo en el conjunto de prueba generará sesgos grandes ya que son datos que no conce el modelo y su capacidad de adaptabilidad se verá reducida. Se vuelve un problema particular y se pierde la generalidad.

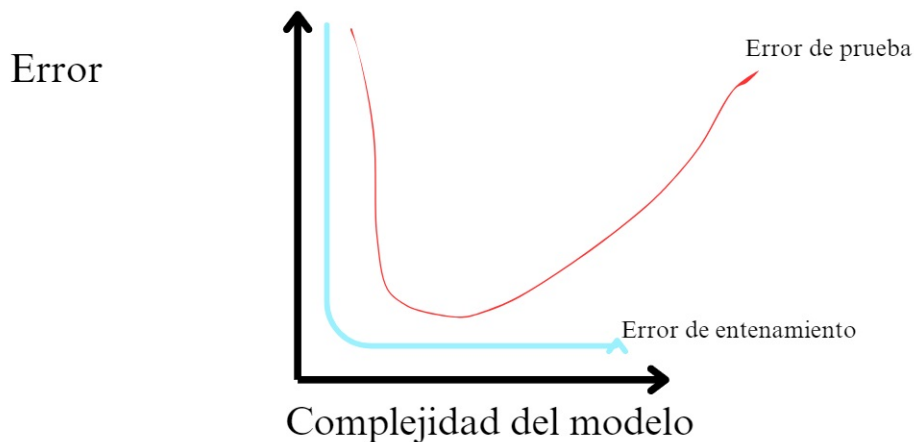


Figura 2.25: Gráfico Complejidad de modelo vs. Error. El crecimiento del error en el conjunto de prueba aparece cuando un modelo es robusto

### Evaluación de rendimiento de algoritmo de ML para regresión

El presente trabajo utilizará regresión, por lo que se mostraran las formas de evaluar el rendimiento para este caso particular, pues, existen formas de evaluar clasificación y clusterización pero esta fuera del propósito principal.

El rendimiento de un modelo de ML se basa principalmente en saber que tanta "distancia" hay entre la muestra objetivo y el punto que se ha predicho por el modelo, es decir, la hipótesis con la que trabaja el modelo, durante la fase de entrenamiento es útil para poder realizar ajustes al modelo, robustecerlo o simplificarlo según sea el caso. Entre las formas matemáticas para poder mostrar estos valores de rendimiento se tienen:

- **Error medio absoluto (EMA):** El error medio absoluto o mean absolute error (MAE en inglés), es una medida que entre más cerca de 0 se encuentre indicará que el modelo se ha ajustado adecuadamente según el conjunto de datos de entrenamiento. Su expresión matemática es dada por la ecuación 2.67.

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \quad (2.67)$$

Siendo "m" el número de muestras totales que se tienen.

$\hat{y}_i$  la predicción o salida de la función.

$y_i$  el valor real al cual se le conocen como etiquetas u objetivos.

- **Error cuadrático medio (ECM):** El error cuadrático medio o mean squared error (MSE en inglés) es una medida que considera la diferencia cuadrática promedio entre las predicciones y los valores objetivo, su expresión matemática es:

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \quad (2.68)$$

- $R^2$ : Un valor acotado de la forma [0,1], entre más cercano este a 1 indicará que el modelo se ajusta mejor a los datos del conjunto de entrenamiento. Se toma en consideración la diferencia entre las muestras y su promedio (TSS) y la diferencia entre los valores predichos y las etiquetas reales(RSS). De acuerdo con la ecuación 2.69

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{\sum_{i=1}^m (y_i - \bar{y}_i)^2} \quad (2.69)$$

Es importante destacar que los calculos de errores con MAE no penaliza errores grandes de predicción, haciendo que se puedan volver inexactas las decisiones de acuerdo a la complejidad del modelo, por lo que es necesario intentar con más formas de calcular rendimientos de algoritmo o, al menos, complementar con más herramientas disponibles para regresión como el MSE o  $R^2$ .

Como dato extra, existe una variación de MSE, llamada en inglés "Root Mean Square Error (RMSE)" la cual es una medida de la magnitud promedio de los errores al cuadrado en una predicción, esta función al ser más compleja implica un calculo más rudimentario que no necesariamente implica una mejoría en rendimiento del modelo de ML o una precisión mayor. El RMSE se expresa matemáticamente como:

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \quad (2.70)$$

Todos los valores calculados o funciones descritas en esta sección que evalúan el rendimiento del algoritmo de ML y por ende el modelo de ML también son conocidas como funciones de costo (C)

o de pérdida (loss en inglés). Lo ideal es que el costo sea el valor más pequeño de todo un espectro de valores, para poder encontrar este valor pequeño se emplean los "algoritmos de búsqueda".

### Algoritmos de búsqueda: El descenso de gradiente

Los algoritmos de búsqueda se basan en instrucciones que se otorgan a un modelo de ML para localizar un elemento en específico. En este caso, se basan en la derivada de la función de costo que pida y encuentre un mínimo local. Existen diversos tipos de algoritmos de búsqueda, pero el que se utilizará en este trabajo será el algoritmo por descenso de gradiente.

El método por descenso de gradiente utiliza la dirección negativa de la posición actual como dirección de búsqueda, la cual, se considerará como la dirección más rápida de descenso en dicha posición actual. Se utiliza la derivada en ese momento para poder hacer una actualización de pesos, estos pesos son la relevancia de la conexión existente entre la característica y las etiquetas u objetivos. Este proceso de calculo de derivadas de la función de costo, actualización de pesos y repetición del proceso es el algoritmo de busqueda de descenso de gradiente que continuará hasta converger, es decir, hasta encontrar un mínimo local.

Matemáticamente, el algoritmo de descenso de gradiente es:

$$\omega_{j+1} = \omega_j - \alpha \frac{\partial C(\omega_j)}{\partial \omega_j} \quad (2.71)$$

Siendo:

- $C(\omega_j)$  la función de costo dependiente de los pesos
- $\alpha$  es el factor de entrenamiento o "learning rate" que varia entre 0 y 1

Estos valores varian en un rango de  $j \in [1, 2, \dots, h]$  con "h" el número de iteraciones que se variaran para encontrar el mínimo local.

El algoritmo de búsqueda comienza con los valores  $\omega$  en números aleatorios y el valor de la derivada parcial inicializada en 0.

Es importante mencionar que la "convergencia" significa un cambio muy pequeño en la función de costo o alcanza un máximo en el número de iteraciones (actualizaciones de pesos).

### Parámetros e hiperparámetros

Un modelo de ML necesita ser controlado y regulado durante su entrenamiento, esto se logra mediante parámetros e hiperparámetros. Los hiperparámetros hacen posible que el modelo aprenda las configuraciones optimas de los parámetros. Lo anteriormente mencionado muestra que los hiperparámetros son valores que se ajustan de manera manual, estos son usados para controlar el entrenamiento. Por otro lado, los parámetros son automáticamente aprendidos por los modelos, es decir, son "destilados" de los datos para generar un modelo.

Es necesario destacar que los hiperparámetros son configuraciones fuera del modelo y antes del entrenamiento. Entre las características importantes de los hiperparámetros se tienen:

- Usados comunmente para la estimación de parámetros de un modelo.
- Son colocados por el usuario.
- Se colocan de forma "heurística", es decir, una manera de buscar la solución (en este caso del mejor modelo que resuelva un problema) mediante métodos no rigurosos.[21]
- Se suelen ajustar para un problema de modelado predictivo.

Existen diversas clases de hiperparámetros, sin embargo, en este trabajo se utilizarán unicamente los que se enlistan a continuación:

- Tasa de aprendizaje (learning rate  $\alpha$ )
- Tamaño de lote (batch size)
- Función de activación
- Numero de neuronas de la red neuronal artificial
- Tamaño de filtros en las capas de convolución
- Número de “pliegues” (Numero ”k” para la validación k-fold)

Se profundizará en las siguientes subsecciones sobre estos hiperparámetros.

Determinar de forma específica que hiperparámetros son mejores para un modelo suele ser uno de los principales problemas para la obtención de resultados, pues, teniendo un espectro infinitesimal en las fronteras numéricas de los hiperparámetros, pareciese que se busca el mejor número dentro de infinitos candidatos. De manera general, la búsqueda del “hiperparámetro que mejor se ajusta al modelo” se puede obtener dividiendo primero el conjunto de datos en 3: conjunto de entrenamiento, conjunto de prueba y conjunto de validación. Después, se optimizan los parámetros del modelo utilizando el conjunto de entrenamiento basado en las métricas del rendimiento del modelo. Luego, se buscan los hiperparámetros del modelo usando el conjunto de validación basándose en las métricas de rendimiento del modelo. Finalmente, se repite el proceso de uso de conjunto de entrenamiento y validación alternativamente hasta que los hiperparámetros y parámetros sean determinados y se evalúe el modelo mediante el conjunto de prueba. Cabe destacar que si el conjunto de prueba muestra buenos resultados se puede considerar como una elección “buena” de hiperparámetros y obtención de parámetros, sin embargo, esta prueba podría tener mejoras si se hace con más procesos, datos o con otro modelo por lo que no deja de ser un método heurístico.

Por otra parte, los parámetros son parte del modelo que ha aprendido mediante el entrenamiento y es la clave para que los algoritmos de ML aprendan. Los parámetros tienen como características importantes el ser requeridos para que un modelo pueda realizar predicciones. Además, los valores de los parámetros definen las funciones modeladas, es decir, para un ajuste lineal se define la función que mejor se adapta a los datos de entrenamiento utilizando los parámetros. Los parámetros son valores que no se colocan manualmente por el usuario, pues, estos son estimados y “aprendidos” de los datos y del entrenamiento. Los parámetros son considerados parate de los modelos por lo que el aplicarlos a los conjuntos de prueba y validación se utilizarán como una forma de saber si el modelo es el óptimo para la solución del problema.

Los parámetros que serán considerados únicamente en este trabajo serán los “pesos” de una red neuronal artificial (RNA)

### 2.5.3. Regresión lineal en ML

La regresión lineal en ML toma como referente el análisis de regresión de la estadística matemática para determinar de forma cuantitativa la relación entre 2 o más variables. Para ML, la regresión lineal es un tipo de aprendizaje supervisado.

Una regresión lineal simple consta en el análisis de regresión con una variable independiente y una variable dependiente en existencia, así como su relación entre ellas que puedan ser representadas mediante una línea recta. El análisis de regresión lineal múltiple toma en consideración dos o más variables independientes y se supone que la relación entre estas variables y las variables dependientes será lineal. Los modelos de regresión lineal se vuelve una línea que modela las relaciones entre variables únicamente cuando la variable independiente “x” es de una sola dimensión (ver figura 2.26). Por otra parte, deja de ser un plano lineal para volverse un hiperplano cuando la variable independiente es multidimensional (ver figura 2.27).

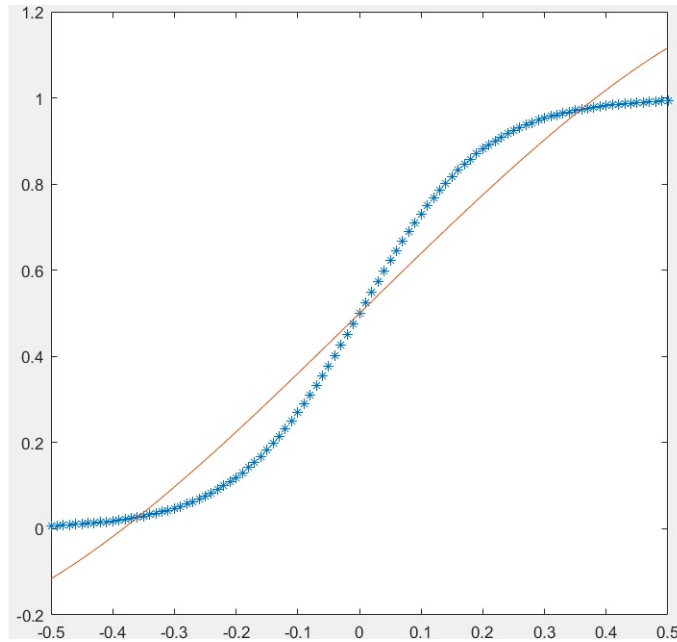


Figura 2.26: Regresión lineal simple

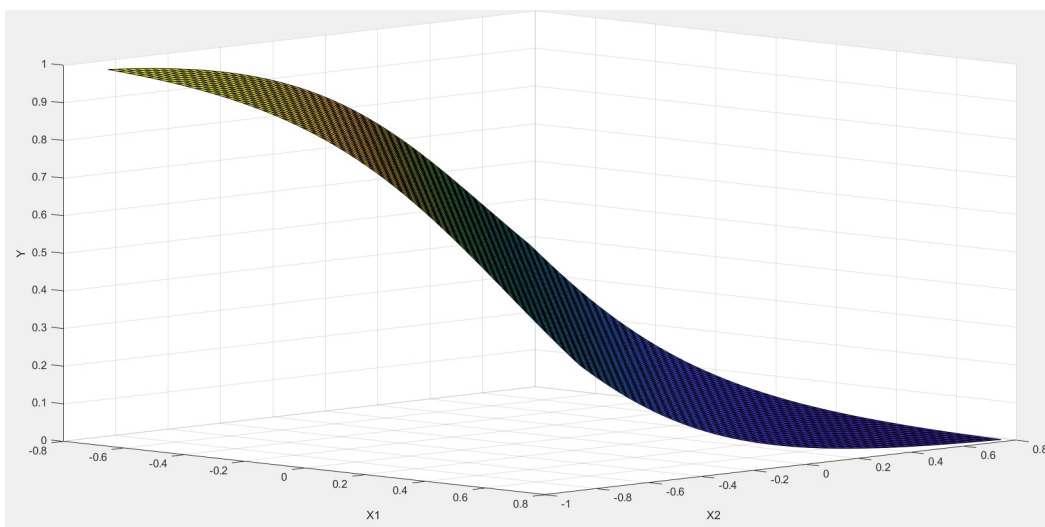


Figura 2.27: Regresión lineal multiple

Tomando como premisa, que el problema de ML se podrá ajustar a una curva de regresión lineal, se supone, que la función que modelará al algoritmo de ML será el siguiente:

$$h_{\omega}(x) = \omega^T x + b \quad (2.72)$$

Siendo  $\omega$  el parámetro llamado “peso”,  $x$  representa la muestra y “ $b$ ” representa el sesgo o error que se tendrá. Esta función también recibe el nombre de hipótesis

La salida o “predicción” se refleja matemáticamente como:

$$P = \omega^T x + b + \epsilon \quad (2.73)$$

Siendo  $\epsilon$  un error que aparece debido a que al provenir de una función de hipótesis y tener un sesgo, existirá una diferencia entre la salida real y la predicción. El error  $\epsilon$  se asume como un error con distribución normal, asimismo, la función de costo de la regresión lineal se puede obtener mediante una función de distribución normal y la estimación por máxima verosimilitud (en inglés “maximum likelihood estimation” (MLE)), es decir:

$$J(\omega) = \frac{1}{2m} \sum (h_{\omega}(x) - y)^2 \quad (2.74)$$

Se busca que el valor predicho se aproxime lo más posible al valor real, es decir, minimizar el valor de costo (Loss). Se puede utilizar el algoritmo de gradiente descendiente para calcular el parámetro de peso “ $\omega$ ” cuando la función de costo alcance el mínimo, completando la construcción del modelo. Por otra parte, existe la regresión polinómica que es una extensión de la regresión lineal debido a la complejidad del conjunto de datos que impide que se ajusten las predicciones a únicamente una curva con curvatura 0, si se utiliza la regresión lineal ocurrirá un subajuste, por lo que es necesario utilizar la regresión lineal polinómica:

$$h_{\omega}(x) = \omega_1 x + \omega_2 x^2 + \omega_3 x^3 + \dots + \omega_n x^n + b \quad (2.75)$$

“ $n$ ” representa el grado del polinomio. Considerando que las características no tengan linealidad, la relación entre sus parámetros “ $\omega$ ” sigue siendo cercana a tenerla. El uso del gradiente descendiente y la regresión lineal serán claves para el entrenamiento de ML específico, es decir, las redes neuronales artificiales (RNA).

#### 2.5.4. Redes Neuronales Artificiales (RNA)

Dentro de los algoritmos de ML se desprenden dos ramas, el aprendizaje supervisado y el aprendizaje no supervisado, asimismo, el aprendizaje supervisado se divide en regresión y clasificación. Se usará esta subrama, la regresión y se utilizará uno de sus algoritmos más utilizados, la red neuronal artificial.

El ML tradicional se basa en un análisis de problemas y localización de tareas para después hacer un filtrado, extracción y selección de características, luego, entrenar un modelo y finalmente predecir, inferir e identificar si el problema está solucionado. Pero, ¿Es posible utilizar un algoritmo que automáticamente ejecute todos los procesos previamente mencionados? Si, es posible utilizando una estructura similar a la del cerebro humano que procesa información, la filtra y entrega un resultado después de recibir estímulos.

##### La neurona y el Perceptrón de una sola capa

Una neurona del sistema nervioso central (SNC) tiene un cuerpo celular compuesto por un núcleo o soma, dendritas que se extienden desde el cuerpo celular con extensas arborizaciones, en particular del SNC con espinas dendríticas, un axón largo que se origina en el montículo del axón (ver figura 2.28) y es dividido en terminales presinápticas, las cuales, pueden clasificarse según el número de prolongaciones en unipolares, bipolares y multipolares. Muchos axones están

mielinizados, es decir, contienen mielina lo cual, mejora la velocidad de transmisión de los impulsos nerviosos. En el sistema nervioso periférico, la mielina se forma cuando las células de Schwann envuelven el axón. En el SNC, las células que forman la mielina son los oligodendrocitos, que pueden mielinizar múltiples axones vecinos.

Las neuronas reciben estímulos mediante potenciales de acción, los cuales son cambios del potencial de la membrana que tienen una propagación en las células excitables, específicamente en su superficie. Estos potenciales tienen una “subida” que es un umbral de excitación, al terminar este proceso comienza un periodo refractario que es básicamente “apagar” el potencial de acción para no desencadenar un segundo. Al durar aproximadamente 1 ms el cambio de potencial de  $-70$  mV estando en reposo hasta los  $30$  mV, se representa un diferencial de potencial grande, ya que las neuronas no miden más allá de los  $20\mu m$ .

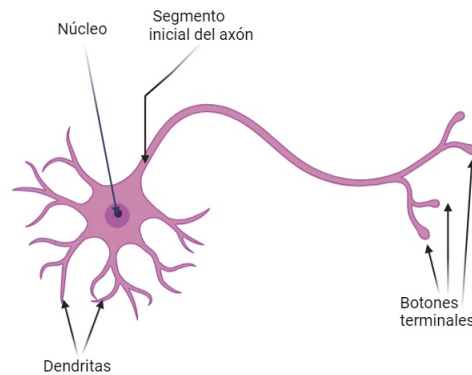


Figura 2.28: Neurona motora.

Al contener mielina, las cuales son células accesorias que envuelven a los axones, crea un aislante eléctrico haciendo que la corriente “salte” de un axón a otro, mejor conocido como “conducción saltatoria”. Al contener la mielina, se produce una conducción más rápida, en consecuencia, se reduce la cantidad de carga que se requiere para una diferencia de potencial y un potencial de acción más rápido.

Lo anterior mencionado es la forma en como opera una neurona ante un potencial de acción, que se produce a partir de un estímulo (en este caso la corriente que comienza es debida al sodio  $Na$ ), el potencial es conducido por todo el axón y termina en los botones terminales que terminan realizando una acción o sirven de estímulo para una neurona vecina.

Matemáticamente, el modelo simplificado de una neurona es llamado “perceptrón” el cual intenta emular la estructura y funcionamiento de la neurona cerebral, pues, el perceptrón utilizará las entradas y respectivas características las dendritas de la “neurona matemática”, tendrá pesos asociados “ $\omega$ ”, la “función de activación” servirá como una forma de decidir “activar a la neurona” o no, si es activada se producirá una acción similar al potencial de acción que recorrerá todo el axón y se conducirá a la salida de la neurona que será el término de esta neurona matemática la cual haría analogía la salida con los botones terminales, entregando un resultado que se puede utilizar para interpretación (una acción del cerebro) o como estímulo para un perceptrón vecino (en el cerebro serviría para una neurona vecina, la acción se conoce como “salto sináptico”).

El esquema del modelo matemático simplificado se puede observar en la figura 2.29 El modelo más

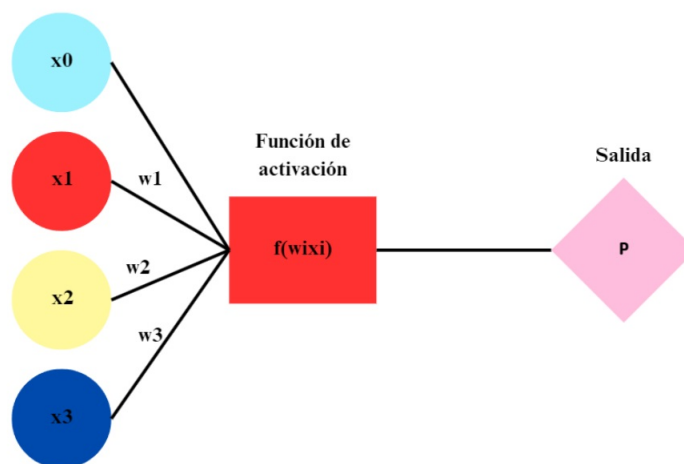


Figura 2.29: Modelo matemático simplificado de una neurona. Este modelo es también conocido como una red neuronal artificial de una sola capa

simple de una neurona artificial se puede expresar matemáticamente de la forma:

$$P = f\left(\sum_{i=0}^m x_i \omega_i + b\right) = f(\vec{\omega}^T \vec{x} + b) \quad (2.76)$$

Para simplificar las ecuaciones, es preferible optar por una variable de preactivación que suele denotarse con la letra “z”, esta variable vectorial será el argumento de la función de activación, es importante mencionar que la función Z y la función de hipótesis difieren en el termino del sesgo, la función hipótesis no lo tiene, mientras que la preactivación si la tiene y está entrará en la función de activación. Es decir:

$$z = \vec{\omega}^T \vec{x} + b \quad (2.77)$$

Originalmente el perceptrón se propuso como un algoritmo de clasificación que utilizaba la función “sigmoide” ya que era una clasificación binaria y esta función tenía valores entre 0 y 1 lo que la hacía perfecta para el problema.

$$\sigma(z) = \frac{1}{1 + \exp[-z]} \quad (2.78)$$

Como el uso del perceptrón se veía limitado para solución de problemas más complejos y diferentes a una clasificación binaria, se comenzaron a unir más perceptrones con la misma lógica de una neurona cerebral: Si una neurona crea una acción en el cuerpo, un conjunto de neuronas podrá realizar tareas más avanzadas y complejas dentro del cuerpo humano como regular un órgano, pensar o tomar decisiones. Este principio hizo que el conjunto de perceptrones se volviera una red neuronal artificial.

### Redes Neuronales Artificiales (RNA)

La limitación más grande de la red neuronal de una sola capa era su limitada capacidad para ejecución de tareas más allá de una clasificación lineal binaria, por esta razón, se introdujeron más capas “profundas” al algoritmo y el arreglo de neuronas. Una red neuronal artificial (de ahora en adelante llamada RNA) es un conjunto de neuronas que interactúan al mismo tiempo con las características o estímulos a los que se exponen, ahora se tendrán matrices de pesos asociados que tendrán una conexión entre las neuronas de la capa actual y las capas anteriores, se tienen

2.5 Inteligencia Artificial (IA), Machine Learning(ML) y Aprendizaje Profundo (Deep Learning)

diversas capas con funciones de activación entre cada capa de neuronas y cada salida de las capas anteriores sirven como entradas para las capas de neuronas siguientes, un modelo matemático simplificado de un conjunto de neuronas cerebrales y como es que estas interactúan.

Estas RNA serán completamente conectadas cuando los valores de salida de las neuronas de la capa  $l$  están conectadas a todas las neuronas (excluyendo el bias) de la capa  $l+1$ . Cada neurona, a su vez, está conectada únicamente con las neuronas de la capa previa y reciben como entrada las salidas que las neuronas previas tengan, el resultado de la salida de la capa actual será la entrada de la capa siguiente. Una estructura multicapa unidireccional es usada, por lo que no hay una retroalimentación en la red entera. Las señales enviadas son transmitidas en una sola dirección desde la capa de entrada (En la figura 2.30 es la capa en color rosa) hasta la capa de salida (capa color azul). La red puede ser representada por un grafo acíclico dirigido (DAG por sus siglas en inglés "Directed acyclic graph") el cual es la figura 2.30 Una red neuronal directa es un perceptrón

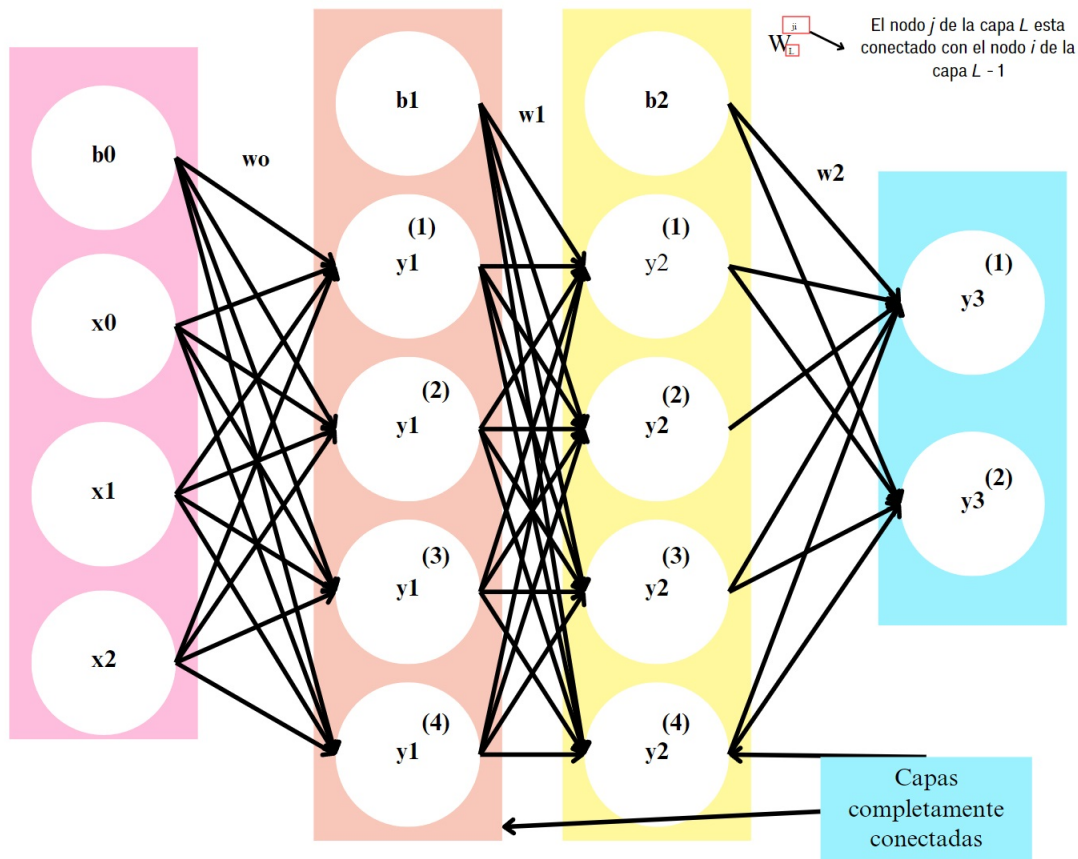


Figura 2.30: Grafo DAG de una Red Neuronal Artificial (RNA), con sus capas ocultas completamente conectadas.

multicapa, la forma más simple y comúnmente usada es un arreglo de neuronas en varias capas. Ahora, hipótesis no será únicamente una función de ajuste, ahora será la red neuronal completa (el esquema presentado en la figura 2.30 es la hipótesis que se supone, solucionará el problema que se le asigne). Los nodos de la capa de entrada no son computables pero representan únicamente los valores de un vector de entrada. Los nodos de cada capa representan neuronas que proveen una función a computar. Decir que es una red neuronal multicapa de una sola dirección indica que las neuronas dentro de una misma capa no están conectadas entre sí, únicamente se conectan las neuronas que aparecen en la capa anterior y la capa siguiente, es decir, se crea una información

intra-capas. La salida de las neuronas hasta antes de la capa final son llamadas “activaciones”. El tener varias capas profundas generan el algoritmo de red neuronal profunda (Deep Learning) y suponen que entre más capas tenga la RNA mejor será la clasificación o predicción, aunque esto es limitado hasta cierto punto, ya que forzar a la solución de un problema mediante un modelo puede generar casos de sobreajuste. Finalmente, las matrices asociadas de pesos son iniciadas mediante numeros aleatorios, ahora será una matriz de numeros aleatorios.

Más capas ocultas indican una mayor capacidad de identificación y solución de problemas de parte de la RNA, sin embargo, es necesario considerar algunas reglas de entrenamiento para aprovechar esta capacidad y no caer en sesgos.

### Funciones de costo en Deep Learning (DL)

El entrenamiento realizado mediante DL no esta exento de errores, por ello, es necesario parametrizar los errores existentes con respecto a los objetivos o etiquetas que se tienen. Como se hizo mención en secciones anteriores, una función de error o función de costo refleja el error entre el objetivo de salida y la salida real de la neurona en esa capa. En problemas de regresión, es usual utilizar la función de costo cuadratica. Para una muestra del conjunto de entrenamiento “x” la función de costo será:

$$C(\omega) = \frac{1}{2} \sum_{d \in D} (t_d - O_d)^2 \quad (2.79)$$

Siendo “d” una neurona en la capa de salida, “D” el grupo de todas las neuronas de la capa de salida,  $t_d$  indica la predicción de salida y  $o_d$  indica la salida real.

Cabe destacar que usar esta función tiene como ventaja tomar a la matriz de pesos como una variable independiente. Utilizar la suma de las raices de la diferencia entre la salida predicha y la salida real de cada muestra entrenada como factor principal para determinar el error.

Una vez que las muestras del conjunto de entrenamiento son dadas para que el algoritmo comience a entrenar, tanto entradas como salidas reales son constantes. La salida actual variará dependiendo de  $\omega$  (considerando el uso del algoritmo de gradiente descendiente), haciendo que la variable independiente de la función de error sea  $\omega$ . El coeficiente  $\frac{1}{2}$  se coloca de forma estrategica debido a que el algoritmo de gradiente descendiente necesita la derivada de la función de costo, al derivar se tendrá que el cuadrado se vuelve una multiplicación por 2 de la función de costo derivada por regla de la cadena, una vez ocurra esto, el factor  $\frac{1}{2}$  anulará al 2 multiplicando y desaparecerán ambos factores haciendo que la carga de operaciones que tendrá el ordenador para realizar sea más ligera. La función de costo puede, de forma iterativa, hacer una busqueda y actualización del parámetro  $\omega$  en la dirección negativa del gradiente para minimizar la función de costo. Pero, no es un método efectivo para resolver y dar solución a la busqueda de extremos debido a la alta dimensionalidad de la función de costo.

Entonces, se realiza un supuesto en el que la dirección negativa del gradiente es la dirección en la que se desciende de manera más rápida en la función. Por lo que el punto minimo de la ecuación 2.79 se espera que exista a lo largo de la dirección  $-C(\omega)$

#### 2.5.5. El descenso de gradiente en Deep Learning

El algoritmo de descenso de gradiente en DL para una función multivariable para una función de costo  $C(W) = f(\omega_0, \omega_1, \dots, \omega_n)$  y un arreglo matricial de pesos  $W' = [\omega'_0, \omega'_1, \dots, \omega'_n]^T$  se volverá de la forma:

$$\nabla f(\omega'_0, \omega'_1, \dots, \omega'_n) = \left[ \frac{\partial f}{\partial \omega_0}, \frac{\partial f}{\partial \omega_1}, \dots, \frac{\partial f}{\partial \omega_n} \right]^T |_{W=W^T} \quad (2.80)$$

La dirección del vector gradiente tenderá a ubicarse y dirigirse en el punto donde la función crezca más rápido. Por lo tanto, la dirección negativa del gradiente se ubicará en donde la función decrezca más rápido.

Para la actualización de pesos se utilizará un coeficiente que ajuste los pesos de acuerdo con el error que se tenga en el gradiente, denotado como  $\alpha$ .

$$W_{t+1} = W_t - \alpha \frac{dC}{dW} \quad (2.81)$$

Basandose en la tasa de aprendizaje “ $\alpha$ ” y el valor del gradiente, los parámetros actualizados van a reducir el costo de la red neuronal. Sin embargo, es necesario ajustar también las muestras para mejorar el algoritmo y no dejar que todo el trabajo sea realizado por el mismo. Las versiones alternas y más utilizadas del gradiente descendiente se enlistan a continuación.

### Descenso de gradiente por lotes (Batch Gradient Descent, BGD)

Considerando que a cada muestra se le atribuye una etiqueta real, estos pares de datos pueden manipularse en pares para mejorar el entrenamiento o disminuir el costo computacional que conlleva. El descenso de gradiente por lotes hace referencia a tomar todo el conjunto de datos de entrenamiento para poder comenzar con el entrenamiento, cada vez que se utiliza el conjunto completo para que al menos una vez haya actualizado los pesos una muestra de ese lote se le llamará época, al mismo tiempo, el número total de lotes que se necesiten para poder completar una época se conocerá como iteración. Para el BGD, época e iteración se pueden utilizar de forma indistinta. El algoritmo se describe de la siguiente forma:

- El arreglo vectorial o matricial de  $\omega$  se inicializa con números aleatorios.
- El gradiente inicial o el lugar donde se pondrán las actualizaciones de pesos tiene que estar puesto en 0, ya que en el primer momento de inicial el algoritmo no existe una derivada aun calculada, es hasta el siguiente momento de la actualización de pesos en las que ya existe una derivada de los pesos.
- Para cada par de muestra-etiqueta se asignará un peso asociado y esto determinará su contribución para llegar al modelo deseado o la solución del problema requerido.

$$\Delta\omega_i = -\alpha \frac{1}{n} \sum_x \sum_{d \in D} \frac{\partial C(t_d, o_d)}{\partial \omega_i} \quad (2.82)$$

Se tendrá que para cada  $\omega_i$   $\omega_i + = \Delta\omega_i$ .

Pese a que el algoritmo de BGD es bueno matemáticamente para actualización de pesos y de la función de costo, tiene la desventaja de tener un proceso lento en convergencia porque todas las muestras de entrenamiento necesitan ser computadas/tomadas en cuenta cada vez que los pesos son actualizados. Una propuesta para el descenso por lotes será realizarlo muestra por muestra o de forma estocástica.

### Descenso de gradiente estocástico (Stochastic Gradient Descent, SGD)

Para arreglar el defecto del algoritmo BGD, una variante desarrollada es el incremento del descenso de gradiente o volver “estocástico” al gradiente descendiente. Se conoce también como aprendizaje virtual/en línea (online learning) el cual actualiza al gradiente completo en función de una sola muestra.

$$\Delta\omega_i = -\alpha \frac{1}{n} \sum_x \sum_{d \in D} \frac{\partial C(t_d, o_d)}{\partial \omega_i} \rightarrow \Delta\omega_i = -\alpha \sum_{d \in D} \frac{\partial C(t_d, o_d)}{\partial \omega_i} \quad (2.83)$$

La forma en la que trabaja el algoritmo SGD se basa en:

- Se inicializan los pesos con valores aleatorios de valor absoluto cercano a cero.

- Se seleccionará de manera aleatoria del conjunto de entrenamiento una sola muestra con su respectiva etiqueta haciendo que cada  $\omega_i$  sea correspondida de la forma:

$$\omega_{i+} = -\alpha \sum_{d \in D} \frac{\partial C(t_d, o_d)}{\partial \omega_i} \quad (2.84)$$

Para SGD se tiene como ventaja que tiene una actualización de pesos utilizando una muestra a la vez, se considera una ventaja porque las muestras pueden contener cierta cantidad de “ruido” la cual generará que se localice un mínimo o punto extremo aproximado, la dirección del gradiente se verá orientada en ambas direcciones, pero con un punto o acotado en un valor unicamente cerca del valor extremo dificultando la convergencia y tomando en cuenta que no se estanque el algoritmo en un mínimo, esto ayuda al gradiente a no quedar en un mínimo que podría no ser general o extremo y continuar con el entrenamiento hasta localizar otro mínimo que ahora converja mucho mejor. La desventaja principal es que se necesitará más tiempo para que las muestras pasen completamente, cada muestra cuenta como una iteración y cuando pasen todas las muestras se generará una época, se recomienda no hacer solo una época ya que solo habrá pasado al menos una vez cada muestra, haciendo un modelo basado en solamente el conjunto de datos sin variar y sin volver a entrenar para aprender más.

La solución para poder mejorar el algoritmo es la combinación de los dos algoritmos previamente descritos.

### Descenso de gradiente por lotes pequeños (Mini-batch Gradient Descent, MBGD)

La mejora de los algoritmos previamente mencionados es la combinación de estos, es decir, generar un pequeño conjunto de muestras con un tamaño menor al lote completo para computar la variación de los pesos y actualizarlos.

El algoritmo es desarrollado de la forma:

- Se inicializan los pesos con valores aleatorios cercanos a cero y en valor absoluto.
- Se inicializa el gradiente en cero.
- Para las muestras que se utilizarán en el entrenamiento se sigue la siguiente igualdad matemática, asimismo, cada lote de muestras tomado se usará de manera aleatoria:

$$\Delta \omega_i = -\alpha \frac{1}{n} \sum_x \sum_{d \in D} \frac{\partial C(t_d, o_d)}{\partial \omega_i} \rightarrow \Delta \omega_i \quad (2.85)$$

- Si el lote es el ultimo utilizado para la actualización de pesos, se volverán a “barajar” las muestras para volver a tomar lotes de muestras aleatorias en la siguiente época.

El algoritmo de MBGD considerará ambas ventajas de los descensos de gradiente anteriores, es decir, se utiliza la ventaja de sobrepasar los mínimos locales que no son de interes para el problema y reducir un costo computacional, asimismo disminuye en esencia el tiempo de entrenamiento. Para saber el numero de lotes que será utilizado se tiene que dividir el numero de muestras por el tamaño del lote, el valor usual para el tamaño del lote es de 32. Es importante mencionar que el tamaño de lote es un hiperpárametro, así como el numero de épocas que se utilizarán para el entrenamiento, esto se escoge de acuerdo con las necesidades de cada problema y del programador.

### 2.5.6. Proceso de entrenamiento para una Red Neuronal Artificial

Una RNA tiene como objetivo desarrollar algoritmos mediante el descenso de gradiente (el más común en usar) para actualizar pesos y ajustarse (al menos en problemas de regresión) para entregar una serie de resultados que serán comparados con los originales, esto se hace mediante las capas de entrada, las capas ocultas, sus variables de preactivación, sus funciones de activación y las capas de salida. Finalmente se tiene que calcular el rendimiento mediante una función de costo, este proceso es conocido como “propagación hacia adelante” o “Forward propagation”, en la figura 2.31 se presenta un esquema que explica el proceso de propagación adelante y también el proceso de propagación hacia atrás o “backward propagation”.

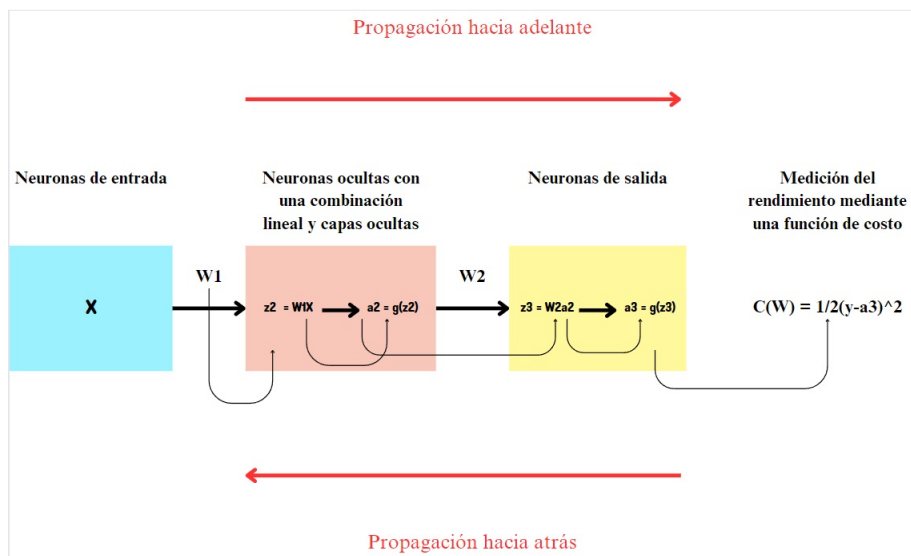


Figura 2.31: Proceso de entrenamiento de una RNA.

#### Forward propagation

Para explicar el proceso de cálculo de la propagación hacia adelante se tomará en consideración el ejemplo de la figura 2.31 la cual contiene una capa de entrada, una capa oculta y una capa de salida. Pese a ser simple se puede generalizar para ser adaptada a cualquier número de capas. Al inicio se tendrá una combinación lineal entre el vector de características  $\vec{X}$  y la fila de la matriz de pesos  $W_1$  que a su vez estará sumada, esta será la primera combinación en la variable de preactivación, es decir:

$$z_2 = \vec{\omega}_1 \vec{X} \tag{2.86}$$

En general, la variable de preactivación  $i$ -ésima se puede obtener de la forma:

$$z_i^{(l+1)} = \sum_{j=1}^n a_j^{(l)} \omega_{ji}^{(l)} + b_i^{(l)} \tag{2.87}$$

Se considera “ $l$ ” como el número de la capa actual, los subíndices de la matriz de pesos van en función del número de neuronas de la capa anterior y la capa actual, para la capa de entrada también aplica esto, considerando que  $W_1$  es del tamaño de número de características por el número de neuronas de la primera capa oculta. Asimismo, para el bias, en las capas ocultas se tiene que concatenar como en el esquema de la figura 2.30 ya que se tendrá un sesgo por cada capa. Ya que la suma

comienza en 1, el bias tendrá como correspondiente el subíndice “ $j = 0$ ”.

Una vez creada la primera variable de preactivación está pasando por una función de activación y se denotará con la letra “ $a$ ” y un superíndice que indica a que capa pertenece. Siguiendo el ejemplo, se tiene:

$$a^{(2)} = f(z^{(2)}) \quad (2.88)$$

Una vez realizada la activación, se propagará hacia adelante, la salida de la capa actual se volverá la entrada de la siguiente capa, es decir, se vuelve una nueva variable de preactivación:

$$z^{(3)} = \vec{\omega}^{(2)} a^{(2)} + b^{(2)} \quad (2.89)$$

Pasando por la capa de salida y su respectiva activación:

$$a^{(3)} = f(z^{(3)}) \quad (2.90)$$

Esto se realiza iterativamente llegando hasta la capa de salida teniendo función de activación (para clasificación) o no (para regresión).

En resumen, se puede ver el cálculo a la salida de la función de costo como:

$$C(W) = \frac{1}{2}(y - a^{(3)}) = \frac{1}{2}(y - (f(W_2 f(W_1 X))))^2 \quad (2.91)$$

Con “ $f$ ” la función de activación que se abordará en la siguiente sección.

Para la actualización de pesos es necesario tomar en consideración a la función de costo y calcular las derivadas parciales de cada parámetro en cada capa que tenga la RNA, para esto es necesario utilizar la regla de la cadena y el algoritmo de backward propagation comienza.

### Backward propagation

El algoritmo de backward propagation (BP) se basa en el cálculo de las derivadas parciales para la actualización de todos los pesos en general, es decir, calcular los errores que se tienen en toda la RNA, se denota con la letra  $\delta$  el error de la capa  $l$ -ésima para la neurona  $j$ -ésima, es decir,  $j_j^{(l)}$ , de acuerdo con la figura 2.31 se tendrán que calcular 2 errores  $\delta$ , pues, no existe error  $\delta^{(1)}$  yta que las entradas son entradas reales, se están otorgando como “veraces” para el entrenamiento por lo que estas no contienen un error de algoritmo en la RNA. Por lo tanto, se tendrán “1-1” errores  $\delta$ , como se calculan errores, se calculan las derivadas, por ende, la actualización de pesos se hace al mismo tiempo que los cálculos de errores. De forma explícita, en el ejemplo se calcula el error de la capa de salida como:

$$\delta_j^{(3)} = (a_j^{(3)} - y) f'(z^{(3)}) \quad (2.92)$$

La función de activación también tiene que ser derivada, pues, al hacer una propagación hacia atrás, la derivada afecta a todos los elementos que contengan al parámetro  $W_n$ . (Nota: El parámetro  $W$  representa un objeto matricial/vectorial según sea el caso y el símbolo  $\omega$  representa un elemento de ese arreglo matricial o vectorial)

Para el error 2, se tendrá:

$$\delta_j^{(2)} = W^{(2)} \delta^{(3)} f'(z^{(2)}) \quad (2.93)$$

La actualización de pesos de acuerdo con la igualdad:

$$W_{t+1} = W_t - \alpha \frac{dC}{dW} \quad (2.94)$$

será el uso de las derivadas con respecto a los pesos de cada una de las capas, es decir:

$$\frac{\partial C(W)}{\partial W_{ji}} = a_j^{(l)} \delta_j^{(l+1)} \quad (2.95)$$

En extenso, para calcular la matriz de pesos 1, se tiene

$$\frac{dC}{dW_1} = \frac{\partial C(W)}{\partial a_3} \frac{\partial a_3}{\partial z_3} \frac{\partial z_3}{\partial a_2} \frac{\partial a_2}{\partial z_2} \frac{\partial z_2}{\partial W_1} \quad (2.96)$$

Este problema que se generalizó con la ecuación 2.95 utiliza la regla de la cadena para enviar todos los errores existentes de la capa de salida hacia atrás de la RNA, entonces, el gradiente con respecto a los pesos de la RNA pueden ser calculados sencillamente. Cabe destacar que para cada algoritmo se deben hacer sus respectivas derivada, este problema toma como referencia a la función de costo MSE y se utilizó para 2 capas, sin embargo, las generalidades se fueron introduciendo para dar un panorama más amplio tomando en consideración un ejemplo de RNA de una sola capa oculta.

En resumen:

- El forward propagation o propagación hacia adelante hace que el conjunto de valores o características de las muestras pase por funciones de activación durante cada capa llegando hasta una capa de salida que hace uso de una función de costo para calcular el error del entrenamiento.
- El backward propagation o propagación hacia atrás hace que se calculen las derivadas, el error se propague a todas las capas que tiene la RNA y los pesos puedan ser actualizados mediante el uso de la regla de la cadena, tomando como consideración el uso de la función de costo, su derivada y que esta afecta a cada elemento que toma como parámetro a los pesos. La propagación hacia atrás al calcular errores y derivadas sigue con el algoritmo de gradiente descendiente para actualizar los pesos y continuar con el entrenamiento, es decir, volver al forward propagation.

Utilizar el algoritmo de gradiente descendiente para optimizar los modelos hace que se utilice la función de costo al final del algoritmo y se ajusten los parametros para minimizar los errores, esto es desencadenado por el hecho de tener parámetros anidados capa por capa y completamente conectados. Esto quiere decir que el calculo del gradiente se vuelve más complejo y de procesos más tardados. Para disminuir la complejidad se utiliza el algoritmo de propagación hacia atrás, la cual muestra una manera de calcular el gradiente de la función de costo haciendo uso de derivadas y la regla de la cadena. Utilizando este método, los errores de la propagación hacia adelante se propagan hacia atrás para cada neurona, los parámetros son actualizados y se sigue utilizando el algoritmo de gradiente descendiente.

Un problema importante es el desvanecimiento de gradiente, el cual aparece cuando las capas de una red crecen y los valores de las derivadas en el Back Propagation (BP) comienzan a decrecer, entonces, el gradiente comenzará a ser 0 y se estancan los pesos en un punto sin actualizarse y sin garantizar que es un mínimo.

Por otro lado, existe la contraparte llamada en inglés “exploding gradient” o un gradiente que crece sin control, aparece cuando aumentan el número de capas y las derivadas en vez de disminuir el valor actual de los pesos hace que crezcan, ocasionando que el gradiente crezca de forma desmedida y en cantidades grandes.

Como los pesos inicializados son números aleatorios menores a 1 en valor absoluto, se tiene que en el BP las derivadas de las funciones de activación se vuelvan el problema, ya que existen varias funciones de activación que debido a su naturaleza o características especiales se vuelvan un conjunto de errores que al recolectarse entre cada capa el gradiente se desvanesca o crezca de forma desmedida.

### 2.5.7. Funciones de activación

Una de las características importantes de una RNA es que las salidas de cada capa de neuronas sirven como “estímulo” o entradas de la siguiente capa, sin embargo, conservar la misma hipótesis sin tener modificaciones durante el proceso puede generar sesgos grandes, una función de costo alta o un costo computacional alto que tenga como salida un mal ajuste al problema requerido. Una solución práctica que aumenta la “complejidad” del modelo es el uso de las funciones de activación. En una RNA de Deep Learning se tiene una relación funcional entre la salida de la neurona de una capa previa con la entrada de la capa actual, esta relación recibe como nombre función de estímulo o función de activación. Sin la función de activación, las capas de la RNA solo tendrán una transformación lineal y todas las entradas serán transformaciones lineales. Los modelos lineales suelen ser insuficientes para poder ajustarse a los problemas que se le asignan, es aquí donde entra en juego la función de activación para introducir factores no lineales.[23] Entonces, se deja de tener solamente perceptrones que hacen una equivalencia con funciones lineales de la forma:

$$P = \omega_{21}(\omega_{11}x_1 + \omega_{21}x_2 + b_1) + \omega_{22}(\omega_{12}x_1 + \omega_{22}x_2 + b_2) + \omega_{23}(\omega_{13}x_1 + \omega_{23}x_2 + b_3) \quad (2.97)$$

A tener funciones de activación que añaden factores no lineales a la RNA la cual convierte todo el modelo en una hipótesis no lineal:

$$\begin{aligned} a_1 &= \omega_{11}x_1 + \omega_{21}x_2 + b_1 \\ a_2 &= \omega_{12}x_1 + \omega_{22}x_2 + b_2 \\ a_3 &= \omega_{13}x_1 + \omega_{23}x_2 + b_3 \\ P &= f(\omega_{21}(a_1) + \omega_{22}(a_2) + \omega_{23}(a_3)) \end{aligned} \quad (2.98)$$

Entre las funciones más utilizadas se enlistan las siguientes con sus propiedades:

#### Función sigmoide

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.99)$$

Esta función comúnmente utilizada en clasificación binaria tiene un rango entre (0,1), es monótona, continua, no pasa por el origen y es asimétrica. Asimismo, es fácil de derivar. Las salidas de esta función están acotadas, y la red tendrá una convergencia sencilla. Por otro lado, su derivada es:

$$f'(x) = \frac{e^{-x}}{[1 + e^{-x}]^2} = f(x)[1 - f(x)] \quad (2.100)$$

Su derivada tiene como rango el  $(0, \frac{1}{4})$ , se tiene una cercanía con el origen cuando se aleja del punto central, esto implica que una red muy profunda, tendrá que calcular más gradientes en el backward propagation y caerán en un área de saturación, haciendo que el gradiente sea muy pequeño, cercano a 0 ocasionando dificultades al entrenamiento, ocasionando el desvanecimiento del gradiente. En la figura 2.32 se observa la función sigmoide en la línea verde y la derivada en azul.

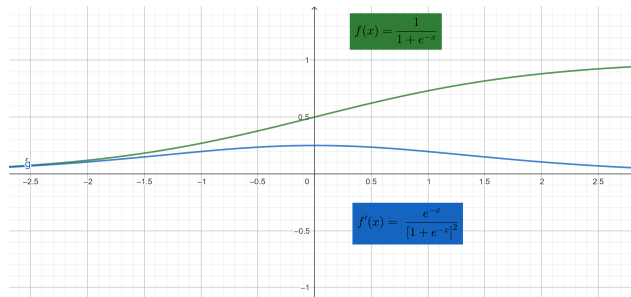


Figura 2.32: Función sigmoide.

**Función tangente hiperbólica**

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.101}$$

Esta función también utilizada para clasificación, esta en un rango de (-1,1), es monotonamente creciente, continua, pasa por el origen y es antisimétrica, al igual que la función sigmoide, se tienen las mismas deficiencias. La derivada de la función tangente hiperbólica:

$$f'(x) = \frac{1}{\cosh^2(x)} = 1 - \tanh^2(x) \tag{2.102}$$

Esta derivada tiene rango de (0,1), esta cercana al 0 en sus extremos, la precisión de las salidas estarán cercanas a 0 como con la función sigmoide. Por esta razón el algoritmo SGD puede reducir el número de iteraciones porque está cercano al gradiente descendiente normal. Es decir, se usarán iteraciones menores para alcanzar un mejor rendimiento o similar al del BGD (Batch Gradient Descend), la figura 2.33 muestra el comportamiento de la función y su derivada al rededor del origen.

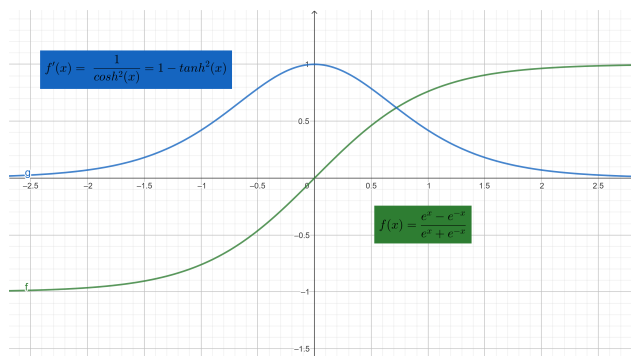


Figura 2.33: Función tangente hiperbólica.

**Función de unidad lineal rectificada (ReLU)**

$$f(x) = \begin{cases} x & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (2.103)$$

La función ReLU (por sus siglas en inglés “Rectified Linear Unit”) es una función que esta acotada entre  $(0, \infty)$ , es continua, monotonamente creciente y no pasa por el origen en un sentido matemático y estricto, pues, es una función a trozos. Su derivada es:

$$f'(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases} \quad (2.104)$$

Esta función junto con su derivada son utilizadas tanto en clasificación como en regresión, entre sus ventajas se encuentran:

- Hace una convergencia más rápida y una mejor sinergia con el algoritmo SGD.
- Comparandola con la función sigmoide y tangente hiérbolica, la función ReLU no contiene exponenciales, lo que la hace más sencilla de calcular y reduce el poder computacional.
- El desvanecimiento de gradiente se evita con esta función.
- Tiene un excelente rendimiento para preentrenamientos de aprendizaje no supervisado.
- La función puede “matar” o “apagar” neuronas para que no pasen a las siguientes capas debido a la función a trozos en la cual esta definida.

Sin embargo, esta función también cuenta con algunas desventajas, entre las cuales están:

- Al no tener un límite superior, es relativamente fácil para el entrenamiento converger sin llegar al mínimo.
- Al no ser diferenciable en  $x = 0$  se tiene que forzar a la existencia de una derivada que se puede definir a conveniencia, volviendola ambigua.
- La superficie definida en el punto 0 no es lo suficientemente suave en algunos problemas de regresión. (Debido a que hace un salto brusco en su derivada cercana a 0)

Estas desventajas no la demeritan ya que, como se hizo mención previa, se reduce la carga computacional y se reducen las operaciones que realizará el algoritmo de BP, mitiga el problema del desvanecimiento de gradiente y no se satura, haciendo que las actualizaciones de pesos en toda la red sean actualizados rápidamente. La figura 2.34 muestra su comportamiento y salto brusco en su derivada cercana a 0, en sentido estricto, no esta definida la derivada en 0, pero para este trabajo ha sido definida como 1.

**2.5.8. Regularización**

Considerando todo lo escrito en secciones anteriores podemos realizar nuestro entrenamiento y crear un modelo que solucione un problema, se mencionó que para tener seguridad de un modelo y su eficiencia fuese demostrada se tendrían que utilizar conjuntos de prueba y conjuntos de validación, pero ¿Qué ocurre si se tiene un entrenamiento excelente con una función de costo muy pequeña pero el rendimiento en el conjunto de prueba es deficiente? Las razones por las cuales ocurre esto son variadas como las dimensiones de las características que se emplean en una RNA de DL, las consideraciones que el modelo asume, los parámetros, la existencia de ruido o un escaso

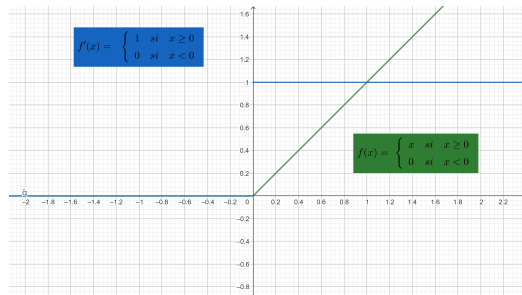


Figura 2.34: Función ReLU y su derivada.

número de datos para entrenamiento. Trayendo como consecuencia un ajuste que al menos podría ajustarse correctamente con el conjunto de entrenamiento, pero, al introducir datos nuevos que jamás pasaron por entrenamiento se volverán deficientes los resultados. En conclusión, los datos están sobreajustándose sin considerar que el modelo o la hipótesis propuesta de DL tenga la capacidad para ser generalizada y sea capaz de ajustarse a nuevos datos.

Para este trabajo se utilizará el método de regulación conocido como "dropout" (o DropConnect). En 2014, en el *Journal of Machine Learning Research 15* aparece esta técnica para prevenir el sobreajuste, el dropout es una técnica que desea prevenir el sobreajuste y poder mostrar un camino que aproximadamente combine diferentes arquitecturas de RNA de manera eficiente. [24]

"Dropout" hace referencia a un "abandono o desconexión de neuronas" (de las capas ocultas o las visibles, es decir, la capa de salida por ejemplo) de una RNA. [24]

En esencia, el retirarlas o desconectarlas no implica que se haga todo el tiempo del entrenamiento ya que en este caso se tendría una pérdida de información y rendimiento. Lo que ocurre es que de forma aleatoria se desconectan algunas neuronas, con alguna probabilidad de desconexión. Puede ser cualquier valor, pero puede ser puesto simplemente como 0.5 para mencionar que cada neurona tendrá un 50% de "pasar" a la siguiente capa o no. La siguiente figura ejemplifica de manera esquemática como funciona el dropout.

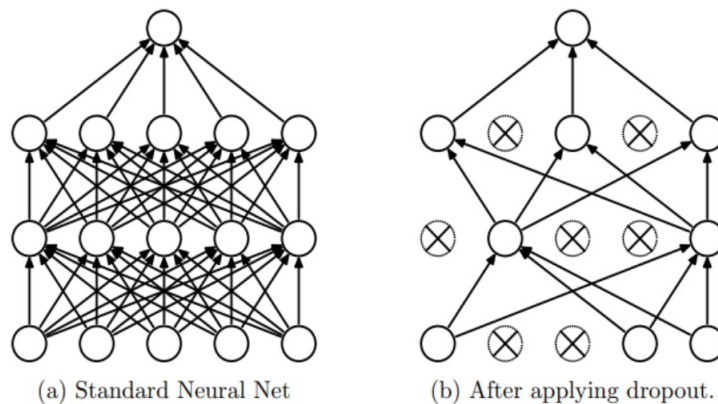


Figura 2.35: Dropout en un modelo de RNA. A la izquierda, una RNA estándar con 2 capas ocultas. A la derecha, un ejemplo de una red que aplicó dropout al modelo de la izquierda, algunas conexiones cruzadas han sido eliminadas. [24]

Es importante mencionar que una RNA con dropout es como generar más redes con menos carga de neuronas, pero que al momento de entrenar la arquitectura original, se estan entrenando más subredes pequeñas, con menos conexiones, neuronas y por ende, menos carga computacional. Al agregarle el factor de aleatoriedad, las subredes que se generan serán del orden de  $2^n$  donde “n” es el número de neuronas totales de la red original. Los pesos compartidos ahora serán de  $n^2$  lo que implica una actualización más amplia que tendrá al factor del dropout como el que decidirá si la neurona pasa o no a contribuir en ese momento del entrenamiento con la actualización de las matrices de pesos. Los sobreajustes tenderán a ser menores para el conjunto de prueba.

**Descripción matemática del Dropout**

Las matemáticas para el Dropout siguen el mismo modelo que la retropropagación o feedforward propagation, sin embargo, se anexa un término más, una distribución binomial o de Bernoulli.

$$r_j^{(l)} \sim \text{Bernoulli}(p) \tag{2.105}$$

$$\tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)}$$

Cuando se genere la nueva variable de preactivación, ahora se utilizarán las  $\tilde{\mathbf{y}}$ , es decir:

$$z_i^{(l+1)} = \mathbf{W}_i^{(l+1)} \tilde{\mathbf{y}}^{(l)} + b_i(l+1) \tag{2.106}$$

Las variables marcadas en negritas indican vectores. La RNA final, se verá como en la figura 2.36. En cualquiera de las capas de la RNA, el vector “ $\hat{\mathbf{r}}$ ” contendrá variables aleatorias en distribución de Bernoulli, cada una tendrá una probabilidad “p” de llegar a tener éxito, es decir, de un valor numérico de 1. Las distribuciones de Bernoulli multiplicaran una a una las salidas de la capa anterior elemento por elemento para crear salidas con neuronas apagadas, ahora estas salidas con neuronas apagadas serán las entradas de la capa actual, en consecuencia, las neuronas que estan apagadas no tendrán ninguna contribución en esa capa. El proceso a cada capa equivale a realizar un muestreo de varias subredes con base en una red general. El proceso de entrenamiento hace que las derivadas de la función de costo sean transferidas a través de las subredes. Cuando se usa el conjunto de prueba, ahora la red ya entrenada no utilizará Dropout, pues, esta técnica se realiza cuando se esta entrenando, no para verificar con otros conjuntos como el de prueba. [24]

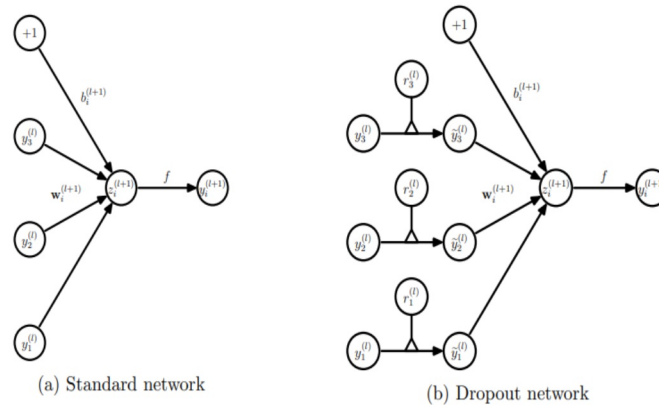


Figura 2.36: Comparación de las operaciones basicas entre una RNA estándar y una con Dropout. [24]

El Dropout por si solo hace que el problema de sobreajuste se vea mitigado en su mayoría, sin embargo, existen más técnicas para mejorar los modelos de DL. El Dropout brinda también ruido, el cual permite que el modelo de DL se pueda optimizar en diferentes partes de los pesos y así se pueda generalizar el modelo gracias a la contribución de cada subred generada por este método de regularización. Existen otros métodos como las penalizaciones L1 y L2, pero no se abordarán debido a que están fuera del alcance de este trabajo. Asimismo, existen otras formas de mejorar el modelo y arquitectura de DL, se puede aumentar o modificar la tasa de aprendizaje, o utilizar optimizadores.

### 2.5.9. Optimizadores

Existen diversas versiones del algoritmo de descenso de gradiente, dependiendo del uso que se le quiera dar tiene sus modificaciones, usualmente, estas modificaciones suelen tener consigo objetos llamados “**optimizadores**”. Entre los propósitos de incluir un algoritmo de optimización se encuentran:

- Acelerar la convergencia del descenso de gradiente utilizado.
- Prevenir saltos estrepitosos de los valores mínimos locales.
- Simplificar el colocar manualmente un hiperparámetro, concentrándose en la tasa de aprendizaje (learning rate “ $\alpha$ ”)

Los optimizadores que se describirán a continuación no son los únicos pero serán las bases para poder describir y poder utilizar el optimizador “**Adam**” en este trabajo.

#### *Momentum*

El *momentum* se basa en el principio físico del impulso; se supone como experimento mental que una pelota cae en un plano inclinado y acumula el suficiente impulso para superar un montículo y alcanzar un mínimo global en lugar de no superar el montículo y quedarse en un mínimo local. El *momentum* se basa en esta idea que creará inercia en una dirección de búsqueda y no estancarse en los mínimos locales y oscilación de gradientes con mucho ruido, tratará de buscar el mínimo global. Añadir impulso a las actualizaciones de los pesos acelerará considerablemente el proceso de optimización y mejora del algoritmo de búsqueda empleado. La cantidad de impulso será conocida como “historia” la cual tendrá un hiperparámetro “ $\eta$ ” el cual oscilará entre 0 y 1, con 0 un algoritmo de descenso de gradiente sin impulso y 1 el valor de impulso más alto que considerará más gradientes recolectados a modo de historia. El optimizador tiene la ventaja de resolver el crecimiento desmedido de gradientes para funciones no convexas con diversos mínimos locales. El *momentum* podrá con estos problemas debido a que usará gradientes ponderados de forma exponencial, evitando errores generados por iteraciones ponderadas también.[25]

Sus reglas de corrección matemáticamente se verán como:

$$\Delta\omega_{ji}^{(l)}(n) = -\alpha\delta_i^{(l+1)}x_j^{(l)}(n) + \eta\Delta\omega_{ji}^{(l)}(n-1) \tag{2.107}$$

El hiperparámetro de ajuste de impulso es  $\eta$  o coeficiente de impulso y el término  $\eta\nabla\omega_{ji}^{(l)}(n-1)$  es el término de “impulso” que recolectará en cada actualización de pesos los gradientes ponderados, es decir, la historia de los gradientes.

Una desventaja importante es que los hiperparámetros ahora serán 2,  $\eta\alpha$  que se tienen que colocar de forma manual lo que implica realizar varios experimentos para determinar un valor que se ajuste a las necesidades de la resolución del problema.

**RMSProp**

El optimizador RMSProp es un optimizador que utiliza un coeficiente de atenuación para asegurar una relación de atenuación para "r" en cada actualización de parámetros. El optimizador tiene la ventaja de no finalizar procesos tan pronto ya que esto a pesar de no tener alta carga computacional no es adecuado para la manipulación de objetivos no estables, además, los efectos de este optimizador son buenos para las RNA. Las consideraciones matemáticas para este optimizador son las siguientes:

$$g_t = \frac{\partial C(t, o)}{\partial \omega_t} \text{ Gradiente calculado} \quad (2.108)$$

$$r_t = \beta r_{t-1} + (1 - \beta) g_t^2 \text{ Acumulacion de gradiente cuadratica} \quad (2.109)$$

$$\Delta \omega_t = - \frac{\alpha}{\epsilon + \sqrt{r_t}} g_t \text{ Actualizacion calculada} \quad (2.110)$$

$$\omega_{t+1} = \omega_t + \Delta \omega_t \text{ Aplicacion de actualizacion} \quad (2.111)$$

La variable  $g_t$  indica el  $t$ -ésimo gradiente,  $r_t$  es la variable de acumulación de gradiente con valor inicial en 0.  $\beta$  es el factor de decaimiento y  $\alpha$  es la tasa de aprendizaje global la cual se coloca de forma manual. El valor de  $\epsilon$  es una cantidad infinitesimalmente pequeña, se pone para evitar las divisiones por 0. Usualmente, el valor de esta variable es de  $10^{-7}$

**Adam**

La combinación de los 2 optimizadores anteriores genera al optimizador de estimación adaptativa de momentos (*Adaptive moment estimation* o "Adam"). En 2015 se presenta el artículo *Adam: A method for stochastic optimization* en donde presentan por primera vez a este optimizador.

Los autores mencionan que "Adam es propuesto como un método para la optimización estocástica eficiente que solo requiere gradientes de primer orden"[26]

La idea de este optimizador es tener tasas de aprendizaje que se van adaptando al modelo conforme pasa el entrenamiento y son calculados los pesos. Se tendrá un valor análogo al impulso o *momentum* y se anexa un segundo momento a los gradientes. Adam combina AdaGard y RMSProp principalmente. Adam tiene como ventajas:

- Actualizar los parámetros se vuelven invariables al cambio brusco de los gradientes
- Se tendrán 2 tasas dedecaimiento que regularán las variaciones pequeñas en los parámetros actualizados
- Se ajustan automáticamente las tasas de aprendizaje conforme se realiza el entrenamiento, ahora el hiperpatámetro se ajusta y adapta al problema, se resuelve el problema de experimentar varias veces con diversas tasas de aprendizaje

El algoritmo matemáticamente se visualiza en el siguiente conjunto de ecuaciones.

$$m_0 = 0 \text{ 1}^{er} \text{ momento} \wedge v_0 = 0 \text{ 2} \text{ momento}$$

$$t = 0$$

Si aun no convergen los pesos y continua el entrenamiento, se tiene que realizar lo siguiente:

$$t \rightarrow t + 1$$

$$g_t = \nabla_w f_t(\omega_{t-1}) \text{ Gradientes obtenidos en el tiempo "t"}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \text{ Actualizacion de sesgos para estimacion de primer momento}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \text{ Actualizacion de sesgos para estimacion de primer momento}$$

$$\begin{aligned}\hat{m}_t &= \frac{m_t}{1 - \beta_1^t} \text{ Cálculo de la estimación de primer momento con correcciones de sesgo} \\ \hat{v}_t &= \frac{v_t}{1 - \beta_2^t} \text{ Cálculo de la estimación de segundo momento con correcciones de sesgo} \\ \omega_t &= \omega_{t-1} - \alpha * \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \text{ Actualización de parámetros} \quad (2.112)\end{aligned}$$

Una vez realizado este último proceso, se regresará al inicio para volver a realizar el algoritmo y continuar hasta llegar al mínimo global de ser posible.

Se debe considerar que los valores de  $\alpha$ ,  $\beta_1$  y  $\beta_2$  se tienen que colocar de forma manual, sin embargo, de acuerdo con los experimentos realizados con el optimizador, los valores por defecto recomendados serán:  $\beta_1 = 0,9$ ,  $\beta_2 = 0,999$ ,  $\epsilon = 10^{-8}$  y  $\alpha = 0,001$ . [26].

Estos valores pueden ser modificados a conveniencia, pero en la práctica, estos valores hacen que Adam pueda converger rápidamente. Después de varias veces emplear el algoritmo, se alcanzará un mínimo local. Otros parámetros no se tienen que ajustar.

El optimizador Adam tiene variaciones como AdaMax o Nadam, sin embargo, se utilizará Adam como propuesta para mejorar las arquitecturas de red empleadas en este trabajo.

Teniendo en consideración todos los pasos previos, se puede proceder a realizar arquitecturas de RNA, sin embargo, es necesario mencionar que al utilizar imágenes médicas no se emplearán como vector de entrada características, se usarán imágenes completas. por lo que se tiene que hacer una modificación a las RNA, una variación de estas; las redes neuronales convolucionales (RNC).

### 2.5.10. Redes Neuronales Convolucionales (RNC)

El uso de características para el entrenamiento de una RNA convencional tenía como delimitación principal el no poder procesar y realizar búsqueda de las características de forma automática, es decir, de un conjunto de datos buscar las características que resuelvan el problema. Asimismo, otra delimitante fue la carencia de búsquedas de forma espacial, por ejemplo, que de una imagen se puedan separar características especiales como la forma de un número o el color de alguna prenda de forma automática, sin asignar esta característica como valor de entrada. Las Redes Neuronales Convolucionales (RNC) son una variación de RNA convencional, tienen una característica especial al tener ahora pesos que abarcan un rango, ya no serán específicas de cada neurona. Las RNC vienen propuestas por Hubel y Wiesel en 1962, cuando, al estudiar las neuronas de la corteza de los gatos para poder tener visualización espacial local y poder elegir direcciones proponen que una manera sencilla de simplificar la retroalimentación neuronal es con las RNC. [27]

Una RNC tiene como estructura básica capas de convolución, capas de *pooling* y capas completamente conectadas (la estructura de una RNA convencional estará presente al final).

### Capa de convolución

Al tener una entrada de imágenes, se tendrán que realizar las extracciones de características en el proceso de aprendizaje, por ende, se necesitará de un “filtro”(o kernel) que será una matriz arreglada con numeros aleatorios principalmente que representarán los pesos contenidos en este kernel. El tamaño del kernel y el numero de filtros que se utilicen son hiperparámetros. Este filtro actuará en las imágenes mediante un proceso conocido como convolución.

La convolución se puede presentar formalmente como:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau)d\tau \quad (2.113)$$

La convolución es escrita como  $f * g$ , pero, el simbolo entre las funciones no indica una multiplicación en un sentido estricto, indica la operación de convolución. El resultado de realizar una convolución entre 2 funciones es otra función. La ecuación 2.113 muestra que: “para convolucionar un kernel con una señal de entrada, se tiene que voltear la señal, desplazar en el tiempo que se desea y acumular cada interacción con el kernel” [28] En el ámbito de las RNC, las entradas que serán imágenes que al ser operadas por la convolución mostraran como salida “mapas de características” los cuales son “imágenes” procesadas por la red que muestran las características más relevantes que considero mejor la RNC. La operación de convolución en RNA se presenta matemáticamente como:

$$\mathbf{A}_j^{(l)} = f\left(\sum_{i=1}^{M^{(l-1)}} \mathbf{A}_i^{(l-1)} * k_{ij}^{(l)} + b_j^{(l)}\right) \quad (2.114)$$

Para la detección de las características locales en diferentes posiciones, se necesitan kernels con aprendizaje  $k_{ij}^{(l)}$ , es decir, las conexiones de los pesos entre el mapa de características  $i$ -ésimo en la capa l-1 y el mapa de características  $j$ -ésimo en la capa l. En la capa de convolución “l” se calcularán las activaciones  $\mathbf{A}_j^{(l)}$  creando subconjuntos de características extraídas de forma espacial en el mapa de características detonato como  $\mathbf{A}_i^{(l-1)}$  de la capa anterior l-1 pero convolucionando los kernels. Se tiene también que “f” representa la función de activación que servirá para no activar una sola neurona, ahora será un arreglo matricial o tensorial de varias neuronas a la vez en un mapa de características que pasará a la siguiente capa.

La operación de convolución se basa en tener 2 matrices, una matriz corresponderá a la imagen de entrada (las imágenes son pixeles con un valor de intensidad y estos en conjunto forman un arreglo matricial de intensidades, es decir, una matriz de números), mientras que la otra corresponderá al kernel. Existen 3 modos de convolución, en esencia, no afecta a las operaciones de convolución pero, si afectará que salidas son las que se mantendrán una vez realizada la operación de convolución.

### Convolución tipo *full*

Esta convolución es implementada añadiendo *padding* a las matrices para poder hacer un “barrido” de toda la matriz por la cual pasará el kernel. El *padding* es el nombre que recibe rellenar los cuatro lados de una matriz con 0’s para aumentar su tamaño. Considerando que se tiene un largo (L) y un ancho (H) y un tamaño de kernel (k, que suele ser cuadrado el kernel), la convolución tipo full tendrá como salida el tamaño L+K-1 (para el largo) y H+K-1 (para el ancho).

### Convolución tipo *same*

Esta convolución no añade ningún tipo de relleno y tendrá el mismo tamaño que la imagen de entrada original.

**Convolución tipo *Valid***

Esta convolución será la que se utilizará en el entrenamiento de las RNC. La imagen tendrá un filtro de menor tamaño a esta, este filtro con números aleatorios hará una multiplicación elemento por elemento (como si de un producto punto se tratase) y finalmente se suman estas multiplicaciones, el valor de este proceso se almacenará en una nueva matriz (la matriz de convolución), acomodando de tal forma que la matriz de salida sea la conjunción de las características que el kernel "decidió" incluir. La figura 2.37 y 2.38 muestran este proceso. Las dimensiones serán 3 en

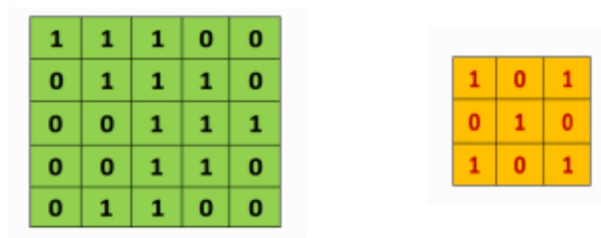


Figura 2.37: Matriz de entrada (izquierda) y kernel (derecha)[29]

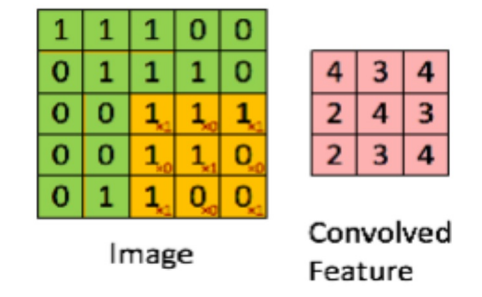


Figura 2.38: Matriz de entrada con el kernel en convolución (izquierda) y la matriz convolucionada (derecha) en convolución tipo valid[29]

la capa de salida, las formas de calcular estas dimensiones se muestran considerando:

- Una imagen de tamaño  $W_1 \times H_1 \times D_1$
- Número de filtros (K)
- Tamaño de filtros (F)
- Paso o salto (S) (Este valor indica si al momento de recorrer el filtro lo hará desplazandose 1 pixel a la vez o más)
- *Padding* (P)

Tanto K,F,S y P son considerados hiperparámetros también.  
 El resultado será una "imagen" de tamaño  $W_2 \times H_2 \times D_2$  donde:

$$W_2 = \frac{W_1 - F + 2P}{S + 1}$$

$$H_2 = \frac{H_1 - F + 2P}{S + 1}$$

$$D_2 = K$$

Siempre se tendrá un largo, un ancho y una "profundidad", se coloca entre comillas porque no hace referencia a que es una imagen en 3 dimensiones de largo, ancho y alto. La tercera dimensión hace referencia a los mapas de características, por ejemplo, una imagen a color tendrá una profundidad  $D_1$  de 3 pues 3 son los mapas de características que compondrán a esa imagen (Rojo, Verde y Azul o RGB). Cabe recalcar que para imágenes que tengan una profundidad, por ejemplo en imágenes médicas, la tercera dimension también se considerará el numero de mapas de características, pues, se considerará que la imagen es bidimensional y para volverla de 3 dimensiones se requirió reunir varios mapas de características para formar ese "grosor". Las imagenes anteriormente descritas recibirán el nombre de "hiperespectrales".

**Capa de "agrupamiento" o *Pooling***

Su función es reducir el tamaño espacial de la representación para reducir la gran cantidad de parámetros y cálculos en la red y, por tanto, controlar el sobreajuste. [30] El proceso necesita de la salida de la capa de convolución, un paso (S) y el número de "máscaras" que se utilizarán (una máscara será una delimitación matricial que tendrá un tamaño pero no contendrá ningún valor, pero tampoco 0's, es como tener unicamente el perímetro de dicha matriz que tomará en cuenta los valores de la capa de salida delimitados por esta máscara). Las máscaras no pueden sobrepasar el tamaño de las salidas de la capa de convolución porque en el *Pooling* no se utiliza el *Padding*. Existen varias formas de realizar el *Pooling* pero la que se utilizará en este trabajo será el de tipo *Average Pooling*.

La idea básica del *Average Pooling* es tomar como referencia el tamaño de la máscara y los valores que estan dentro de esta se sumarán y promediarán, haciendo que se genere una nueva matriz que contendrá los resultados de estos promedios de toda la salida de la capa de convolución. Esto reducirá el tamaño de la imagen creando submuestréos. [31] La figura 2.39 muestra de forma sencilla como es el proceso con una máscara de 2x2 y un paso de 2. Finalmente, después de repetir

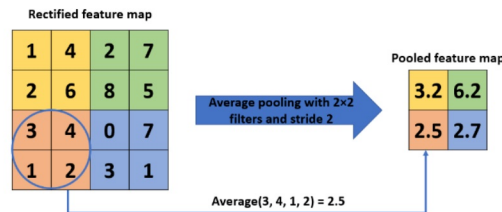


Figura 2.39: Ejemplo de la operación *Average Pooling*[31]

los procesos de la capa de convolución, pasar por una función de activación y llegar a una capa

de *Pooling*, se pasa a una capa completamente conectada no sin antes haber sido "aplanados" los mapas de características, que convierten todas las dimensiones a un solo vector que servirán como las características ya procesadas y seleccionadas por la RNC que pasarán a alimentar una arquitectura de RNA. El proceso de RNA ya se comentó en la sección 2.5.4.

## 2.6. Uso de RNA para diagnóstico clínico, casos de estudio con problemas que involucran hierro

El uso de ML y RNA es una tecnología que se ha desarrollado desde hace más de 20 años, en el campo médico han aparecido varias propuestas que tienen en consideración la cuantificación de hierro. De acuerdo con Espinoza y colaboradores [32], se realizó una propuesta para diagnóstico de anemia ferropénica, megaloblástica y hemolítica. Se tomaron en consideración 27 factores para determinar si un paciente tenía un tipo especial de anemia. Estos 27 factores fueron considerados como las características de entrada para la RNA con 90 muestras de entrenamiento, 30 con anemia ferropénica, 30 con anemia megaloblástica y las 30 restantes con anemia hemolítica, como se tienen en consideración 3 tipos de anemias, es un problema de clasificación.

Su hipótesis se basó en 27 características para la capa de entrada, una capa oculta con 40 neuronas, la tercera capa que también es oculta con 27 neuronas y la capa de salida con una neurona. Cada tipo de anemia se consideró como una red neuronal, es decir, se propusieron 3 hipótesis. Entre los resultados de entrenamiento se obtuvo un 92% de precisión en sus predicciones. El conjunto de validación presentó oscilaciones entre sus respectivos pacientes y sus precisiones, ya que estaban entre un 86% y un 99% de ser diagnosticados correctamente con la anemia correcta. Se presenta entonces, una propuesta que involucra el uso de RNA para el diagnóstico que toma en consideración al hierro en el cuerpo humano. La figura 2.40 muestra la arquitectura utilizada para resolver el trabajo de clasificación.

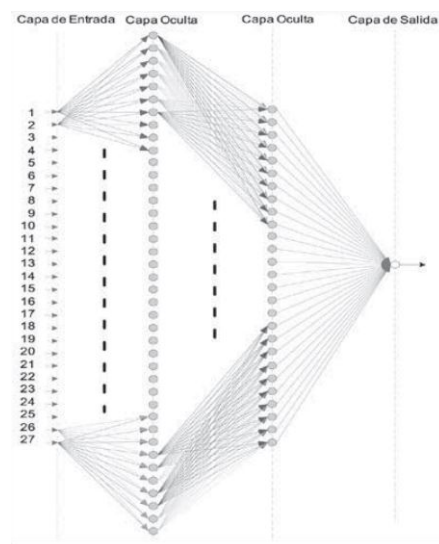


Figura 2.40: Arquitectura de RNA para el diagnóstico de anemia. [32]

### 2.6.1. La Integración de Redes Neuronales en la medición de hierro por RM

El uso del Deep Learning para segmentación automática es uno de los objetos de estudio más prolíferos en el campo médico, pues, segmentar de forma precisa y automática para casos que requieren precisión como la detección de un tumor se vuelven tareas que requieren de una precisión alta y un error de prácticamente cero. Para la medición de hierro se tiene como estudio previo el análisis del miocardio en secuencias  $T2^*$  para cuantificación de hierro. Los autores toman en consideración a 210 pacientes con problemas de sobrecarga de hierro, algunos con talasemias y otros con hemocroatosis hereditaria. Se hace uso de 1 secuencia  $T2^*$  en multi-eco y multicortes para obtener imágenes de  $192 \times 256$  píxeles, un tiempo de eco de 2.0 ms con espaciados de 2.26ms y campo de visión de  $35 \times 35$ , el proceso de uso de Deep Learning fue basado en la arquitectura U-Net la cual consta de 4 capas convolucionales que codifican las imágenes y 4 capas convolucionales que la decodifican, haciendo uso de imágenes de  $256 \times 256 \times 10$  y un total de clasificación de 7 clases de acuerdo con el número de segmentos que se pueden obtener del miocardio para determinar que segmento es el más afectado por el exceso de hierro.[34] Asimismo, una vez segmentadas las imágenes de RM con su respectiva clasificación, se hizo uso del ajuste propuesto por Wood et al. [12]. Se hicieron comparaciones entre 2 expertos del miocardio para segmentación manual y el algoritmo de segmentación automática, teniendo como resultados una cantidad de verdaderos positivos más alta entre uno de los expertos y el modelo de DL que entre ambos expertos, la figura 2.41 muestra claramente los resultados obtenidos mediante un diagrama "bullseye" o digrama de dianas en los que se representa la precisión y efectividad de parte de los expertos contra el modelo de DL para segmentar, lo que supone este resultado es la eficacia del modelo para la segmentación, asimismo, se tomaron como métricas los coeficientes de similitud (DSC) y el error angular(AE) que eran representativos y el DSC era mayor al 75 % lo que representaba, según los autores una segmentación excelente, por otra parte, el AE fue muy bajo, lo que representó un error en el ángulo poco significativo del modelo de DL y los realizados de forma manual. Los trabajos presentados no son propiamente de cuantificación de hierro en tejido hepático con RM y RNC pero, sirven como parteaguas de lo que se ha realizado en el ambito clínico y lo que se ha realizado con el uso de la tecnología de machine learning. En el siguiente capítulo se unirá todo lo descrito para una propuesta nueva que involucra todas las herramientas vistas en el marco teorico.

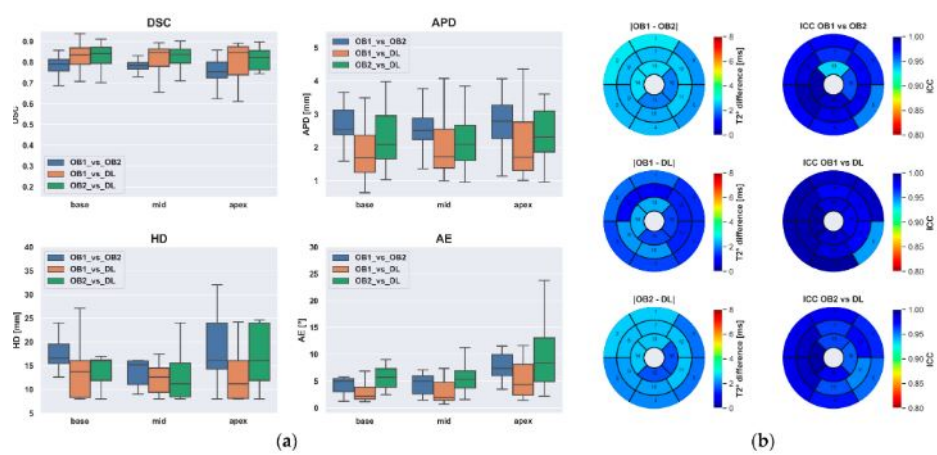


Figura 2.41: Variación de observaciones mediante comparación de segmentación de los 2 expertos y el modelo de DL. En (a) un diagrama de cajas que presenta las métricas de segmentación y en (b) diagramas de diana que muestran las diferencias en segmentación de acuerdo al modelo de la *American Health Association* (paneles izquierdos) y el índice de correlación entre los expertos y el modelo de DL (paneles derechos)[34]

## 2.7. Métodos estadísticos para interpretación de resultados

Una vez realizado el entrenamiento y haber obtenido las predicciones, es necesario utilizar métodos estadísticos para medir correlaciones, distribuciones, visualizaciones de las mediciones para poder determinar si el entrenamiento es óptimo y resuelve el problema.

### 2.7.1. Pruebas de distribución

Antes de realizar análisis, pruebas o gráficos, es necesario estar seguro de tener una distribución de datos normalizada, de lo contrario se pueden tener correlaciones erróneas o sesgadas.

- Prueba de Shapiro Wilks:** Esta prueba, plantea una hipótesis nula en la que se considera que la distribución de la muestra es normal. Considerando un nivel de significancia de 0.05 se tiene por otro lado una hipótesis alternativa que considera a la distribución como no normal. Si el valor  $p$  obtenido es mayor al de significancia, no se rechaza la hipótesis nula. Es necesario tener cuidado tanto en esta prueba como en las que tengan como hipótesis nula como distribución normal ya que si la significancia es alta, no se puede rechazar pero es necesario contrastar la parte numérica con un método gráfico para comprobar si existe una normalidad en las variables.[49] El método consiste realizando un orden entre las mediciones de menor a mayor, calcular varianza y media. Se sigue la siguiente ecuación:

$$W = \frac{\sum_{i=1}^n \sigma_i Y_i^2}{\sum_{i=1}^n (Y_i - \hat{Y})^2} \quad (2.115)$$

Donde  $Y_i$  son los datos de la muestra y  $\hat{Y}$  es la media. [50]

- Prueba de Smirnov-Kolmogorov:** La prueba de Smirnov-Kolmogorov o prueba  $Z$  de Kolmogorov toma en consideración las diferencias mayores absolutas entre funciones de distribución teóricas y las que se midieron.[51] Esta prueba presenta un estadístico que afirma si la distribución tiene las características esperadas o no, no es propiamente una prueba que demuestra la distribución normal pero se puede adaptar a la misma para poder utilizarla. Se considera el uso de estas pruebas cuando se tiene una población mayor a 50 muestras. El estadístico a utilizar se define como:

$$D = \sup_{1 \leq i \leq n} |\hat{F}_n(x_i) - F_0(x_i)| \quad (2.116)$$

Con  $x_i$  el  $i$ -ésimo valor observado de la muestra.  $\hat{F}_n(x_i)$  es un estimador de probabilidad de observar los valores menores o iguales a la  $i$ -ésima muestra observada, finalmente, se tiene que  $F_0(x_i)$  la probabilidad de observar valores menores o iguales a la  $i$ -ésima muestra cuando la hipótesis nula (que menciona que los datos no siguen la distribución pensada) es verdadera.[52]

### 2.7.2. Pruebas de correlación y test de prueba

Una vez demostrada la parte paramétrica o no paramétrica de los datos obtenidos por la RNC, se procede a realizar test de prueba y pruebas de correlación. Se consideran los siguientes test para probar.

#### Test de Wilcoxon

Esta prueba no paramétrica compara datos que no siguen una distribución normal, por lo que no es posible realizar pruebas paramétricas como la prueba  $t$  o test *t de student*. Aunado a la carencia de distribución normal, el test de Wilcoxon tiene la característica de funcionar cuando se tiene un tamaño reducido de población o mediciones para determinar si se puede o no aproximar

a una distribución normal o con pocos sesgos. Cuidado, el hecho de tener un tamaño reducido de mediciones no implica que funcione a la perfección para muestras muy pequeñas o no representativas para una población en especial, es necesario utilizar más estadísticos que apoyen a este test y se garanticen resultados. Asimismo, este test tiene una ventaja sobre otros la cual es no considerar estudios anteriores que digan que tipo de distribución tiene, lo que implica que el no asumir la normalidad de las poblaciones no es problema para la prueba de Wilcoxon.

Entre sus características relevantes se tiene:

- Esta prueba compara si las diferencias entre pares de datos siguen una distribución simétrica entorno a un valor. Si dos muestras provienen de una misma población, es casi intuitivo que las diferencias entre cada par de observaciones se distribuya entorno al 0.
- Utiliza datos ordenados, por lo que si los datos de una población no se pueden ordenar, este test no es útil.
- Es una prueba más robusta que la t de student al tener menor probabilidad de rechazar la hipótesis nula (Es decir, que las diferencias entre 2 muestras no es significativa)

Para que el test de Wilcoxon tenga efecto, se deben cumplir los siguientes criterios:

- Los datos tienen que ser dependientes.
- Los datos tienen que tener la posibilidad de ser ordenados.
- Se puede omitir asumir que la distribución de las muestras es normal.
- Este test trabaja con medianas, no medias.
- Si se tienen valores atípicos, no hay normalidad o el tamaño de población es pequeño este test es más recomendable que la t de student.

Este estadístico se calcula como:

$$W = \min(W_+, w_-) \quad (2.117)$$

Con  $W_+$  la suma de rangos con signo positivo y  $w_-$  la suma de los rangos con signo negativo. Una vez calculado esto, también se recomienda utilizar un tamaño de efecto (Prueba d de Cohen). [53]

### Coefficiente de correlación de Pearson

Este coeficiente de correlación es un índice que mide el grado de covarianza entre distintas variables que, se supone, tienen una relación lineal. Los valores del coeficiente oscilan entre 0 y 1. Se calcula mediante la ecuación:

$$r_{xy} = \frac{\sum Z_x Z_y}{N} \quad (2.118)$$

Es decir, se calcula la media de los productos cruzados de las puntuaciones estandarizadas de las variables X e Y (es decir, las varianzas) para observar que tanto la medición se separa de la media. Un valor de 1 arroja una correlación fuerte y positiva, mientras que el valor de 0 muestra una nula correlación entre las 2 variables. [54]

### Prueba d de Cohen

La d de Cohen se trata de una cantidad entre 0 y 1 que nos indica que tan "fuerte" es el estadístico que se obtiene para las variables que se comparan o testean. Se considera una significancia baja o un efecto bajo si la d es de 0.2, medio si es de 0.5 y muy significativa a partir de 0.8. [55] Esta prueba considera el uso de las medias y una desviación estándar combinada de la forma:

$$d = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_{comb}} \quad (2.119)$$

### 2.7.3. Gráfico de Bland-Altman

Un gráfico de Bland-Altman o *gráfico de diferencia media* es una herramienta gráfica utilizada para evaluar la concordancia entre dos métodos de medición y se ha vuelto una técnica estándar en estudios de métodos comparativos. El gráfico permite visualizar la relación entre dos métodos de medición al mostrar las diferencias entre sus resultados frente a la media de ambos métodos. Este tipo de gráficos y su análisis es útil al introducir nuevas propuestas de medición que se comparan con métodos de referencia establecidos.

Para realizar este gráfico se tienen que obtener 2 conjuntos de datos cuyos métodos de medición hayan medido la misma variable en las mismas muestras. Al tener RM y valores de hierro se cumple que se tienen las mismas muestras y se hacen mediciones de hierro de 2 formas distintas, el estándar de Pespectum y los resultados obtenidos de la RNC.

Para cada par de mediciones se tiene que calcular la media de los dos métodos:

$$Media = \frac{X_1 + X_2}{2}$$

Con  $X_1$  la medición del método 1 y  $X_2$  la medición del método 2.

Después se calcula la diferencia entre ambos métodos:

$$Dif = X_1 - X_2$$

Para la creación del gráfico se colocan los valores de la media en el eje horizontal, mientras que la diferencia de las mediciones en el eje vertical.

Una característica relevante del gráfico de Bland-Altman es que tiene que tener marcados 2 límites y estos serán la media de las diferencias " $\bar{d}$ " y la desviación estándar de las diferencias " $s_d$ ", teniendo entonces que los límites se marcarán como  $\bar{d} \pm 1,96 * s_d$ . Los límites representan el intervalo dentro del cual se espera que caiga el 95 % de las diferencias entre las mediciones de los dos métodos.

La media de las diferencias indica el sesgo sistemático entre los dos métodos. También se cuenta en los gráficos de Bland-Altman con límites de acuerdo los cuales representan la variabilidad aceptable entre los dos métodos. Además, observando la dispersión de los puntos, se pueden identificar tendencias, si las diferencias se incrementan con el valor de la media, podría indicar un sesgo proporcional. Si los puntos se dispersan más a medida que aumenta la media, puede sugerir una variabilidad mayor en mediciones altas.

# Capítulo 3

## Métodos

Para lograr cuantificar el hierro usando resonancia magnética se diseñó una metodología que consta de 3 partes.

- Preprocesamiento de datos: Antes de crear las arquitecturas de red convolucional o hacer uso de estas, es necesario saber si las imágenes de RM tienen artefactos.
- Diseño y entrenamiento de arquitectura de CNN: En esta parte es necesario recalcar que las arquitecturas de la literatura no se utilizaron ya que suelen ser muy complejas que no se adapta a cualquier equipo de computo.
- Uso de estadísticos: Es necesario hacer uso de la estadística para ver correlaciones entre los valores de hierro obtenidos por el estándar de referencia y las predicciones y concluir si el objetivo se ha alcanzado.

### 3.1. Base de datos y preprocesamiento

Se estudió una base de datos con imágenes hepáticas, las cuales, se obtuvieron mediante RM abdominal sin contraste mediante el protocolo de imágenes "LiverMultiScan de la empresa Perspectum©(Oxford, Reino Unido). Las tomas de RM fueron hechas en un escáner marca Siemens®Skyra de 3T, haciendo adquisición de 4 cortes transversales en la zona de la vena porta hepatitis en el hígado de 62 individuos. Se obtuvieron los tiempos de relajación  $T_1$ ,  $T_2^*$  y la fracción de grasa de densidad protónica (PDFF). Estos valores se calcularon mediante la secuencia de eco de gradiente múltiple. El formato de la imagen de RM fue guardada en formato DICOM. Los cálculos de hierro realizados por Perspectum©se considerarán como el estándar de oro para este estudio, pues, los valores a comparar con los resultados del trabajo fueron obtenidos mediante esta empresa.

Las RM de los pacientes tuvieron inhomogeneidades y artefactos, como se presenta en la figura 3.1, por lo que se realizó una corrección de inhomogeneidades y normalización de intensidades para facilitar los procesos computacionales de la RNC.

#### 3.1.1. Corrección de inhomogeneidades

Para hacer correcciones de inhomogeneidades, se emplea el algoritmo "N4 Bias Field Correction" el cual hace uso de las matrices que componen la imagen (matrices de intensidades o de valor de los voxels), además de una matriz que contiene el campo de sesgos y ruido, el cual se asume sigue una distribución de Gauss. [35]

El algoritmo N4 se utiliza para corregir las variaciones en la intensidad de la señal de la imagen

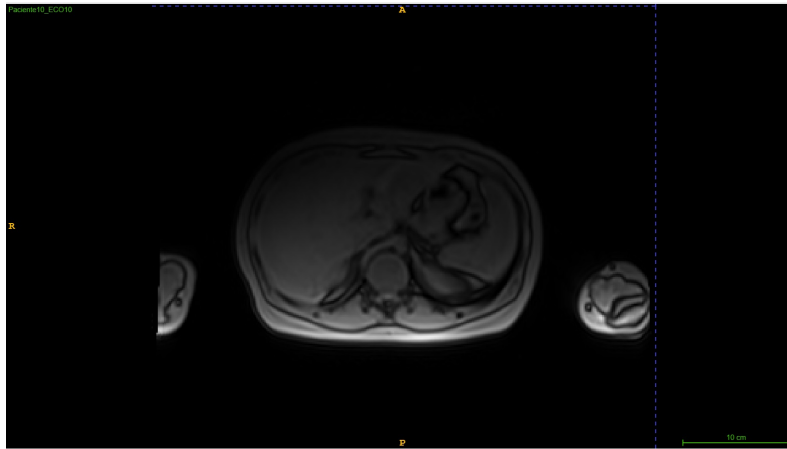


Figura 3.1: RM de un paciente con secuencia eco-gradiente en tiempo de relajación  $T_1$ , en corte transversal.

causadas por las inhomogeneidades del campo magnético estático. Estas inhomogeneidades pueden deberse a diferencias en las propiedades magnéticas de los tejidos, imperfecciones en el campo magnético del equipo o artefactos de movimiento.

El enfoque de N4 también se basa en la no uniformidad que utiliza información local y global para estimar el campo de inhomogeneidades y luego se aplica a la imagen original para corregir las variaciones. El algoritmo es robusto y eficiente, y ha demostrado ser efectivo en la corrección de inhomogeneidades de imágenes de RM. [35]

De manera general, los pasos para el algoritmo N4 son:

- Sesgo de inhomogeneidades: Se hace una corrección del sesgo que son las variaciones en el campo magnético estático y puede hacer que los tejidos parezcan más brillantes u oscuros de lo que realmente son.
- Generación de imágenes de sesgo: Se generan imágenes de sesgo utilizando un enfoque de pirámide multiresolución. El algoritmo divide la imagen original en diferentes escalas y genera imágenes de sesgo en cada escala.
- Segmentación: Se realiza una segmentación en cada escala de las imágenes de sesgo generadas. Esto conlleva realizar una división de la imagen en diferentes regiones o tejidos para estimar el campo de inhomogeneidades de forma precisa.
- Normalización: En esta etapa, se normalizan las imágenes de sesgo utilizando la información de la segmentación. Esto se traduce en suavizar y corregir las variaciones de intensidad dentro de cada región.
- Estimación del campo de inhomogeneidades: A partir de las imágenes de sesgo normalizadas, se estima el campo de inhomogeneidades utilizando métodos de estimación no paramétricos. El campo de inhomogeneidades representa las variaciones en la intensidad de la señal en cada punto de la imagen.
- Corrección del sesgo: Utilizando el campo de inhomogeneidades estimado, se corrige el sesgo de inhomogeneidades en la imagen original. Esto implica dividir la imagen original por el campo de inhomogeneidades para compensar las variaciones en la intensidad.
- Iteraciones: El algoritmo realiza iteraciones para refinar y mejorar la corrección del sesgo. En cada iteración, se generan nuevas imágenes de sesgo, se estima el campo de inhomogeneidades y se corrige el sesgo de la imagen original.

- Convergencia: El algoritmo se repite hasta que se alcanza la convergencia, es decir, cuando la corrección del sesgo deja de mejorar significativamente.
- Postprocesamiento: Una vez que se alcanza la convergencia, se puede realizar un postprocesamiento adicional si es necesario, como aplicar filtros de suavizado o mejorar los bordes de la imagen corregida.

Para el uso de este algoritmo, se emplea el lenguaje de programación Python™, y la librería oficial de “*Advanced Normalization Tools*” o ANTs. [36]

La librería oficial de python se llama como “*antspyx*”. Ésta tiene disponible la corrección de inhomogeneidades descrita previamente haciendo el importe del recurso “*ants*” que lee, corrige y guarda la nueva imagen ya postprocesada. Cabe mencionar que la forma en la que el algoritmo trabaja es realizando evaluaciones tanto de forma visual como de forma cuantitativa, es decir, el algoritmo N4 crea histogramas de intensidades de las imágenes del antes y después de emplear el algoritmo que le permite ver de forma estadística si las inhomogeneidades han sido mitigadas, un histograma más estrecho y centrado sugiere una mejor homogeneidad. Además, se utilizan coeficientes de variación de las intensidades en regiones que deberían ser homogéneas, un coeficiente menor al que se tenía antes de la corrección sugiere una mejor homogeneidad. Otro cálculo relevante para saber si el algoritmo fue correcto es haciendo uso del coeficiente señal-ruido y la relación de contraste-ruido. Si las métricas aumentan después de la corrección se tiene una mejoría en la homogeneidad. Finalmente, se hacen mapas de diferencia para restar los valores de las intensidades de la imagen corregida con la imagen original para visualizar y cuantificar regiones donde se han hecho correcciones de inhomogeneidades.

La imagen sin corrección de inhomogeneidades se puede observar en la figura 3.2, mientras que el resultado final del algoritmo se puede apreciar en la figura 3.3.

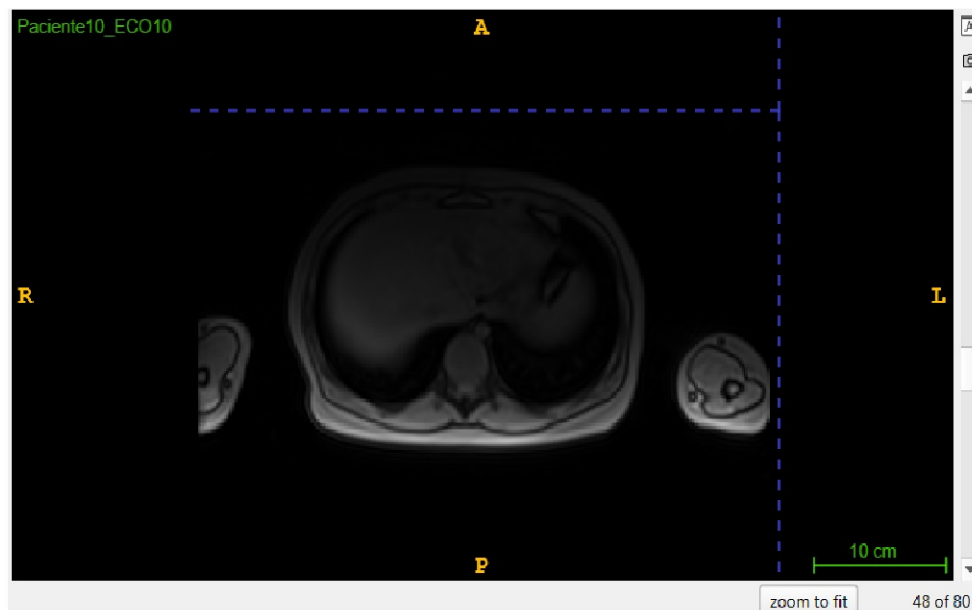


Figura 3.2: RM de un paciente sin corrección de inhomogeneidades . Visualización desde el software ITK-Snap. [37]

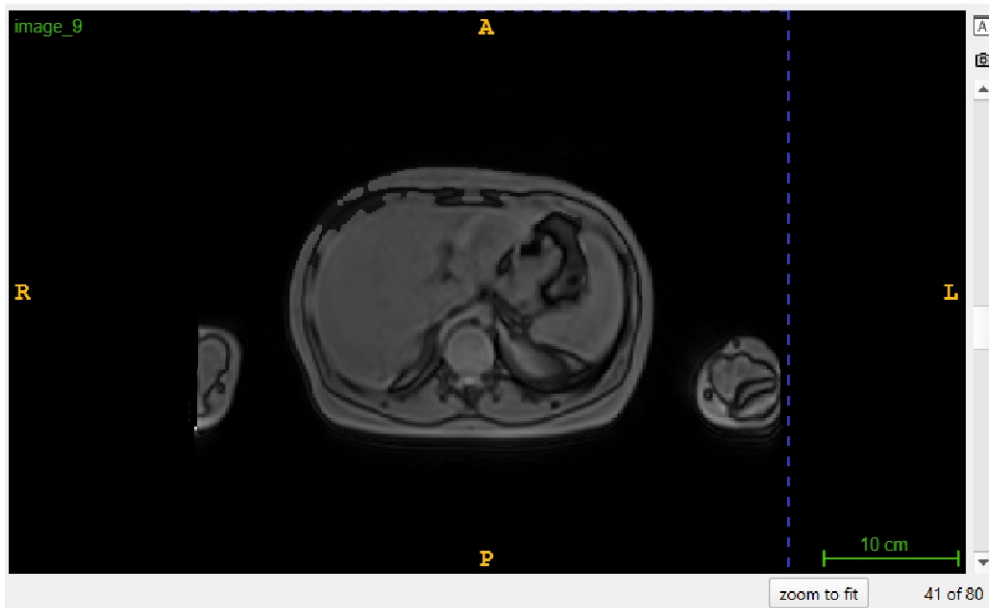


Figura 3.3: Resultado de la aplicación del algoritmo N4 para corrección de inhomogeneidades.

### 3.1.2. Normalización

Una vez aplicado el algoritmo N4 y realizada la corrección de inhomogeneidades, se procedió a realizar una normalización de intensidades.

La normalización empleada es la normalización "z-score" o de *estandarización* [42], la cual consiste en hacer modificaciones de intensidades de un conjunto de datos de tal forma que la media de las intensidades es 0 y la desviación estandar es 1. [38] Esto se realiza siguiendo la ecuación:

$$N_i = \frac{(X_i - \mu)}{\sigma} \quad (3.1)$$

Para el caso de las imágenes de RM, la intensidad de los voxels puede variar debido a diferentes factores como la posición del paciente [39], la configuración del escáner[40], y la calidad de la imagen [41]. La normalización *Z-score* permite estandarizar las intensidades. Además, el uso de esta normalización en las imágenes de RM ayuda a reducir el sesgo introducido por las diferencias en la intensidad de la imagen entre diferentes adquisiciones o entre diferentes pacientes. Por ende, aplicar la normalización es una buena práctica en el procesamiento de imágenes de RM porque estandariza las intensidades de píxeles, reduciendo sesgo y mejorando la precisión de algoritmos de procesamiento de imágenes.

## 3.2. Diseño e implementación de red neuronal convolucional

Se hace uso de la plataforma "Google Colaboratory" para diseñar y entrenar la CNN, al tratarse de un entorno virtual, el hardware proporcionado utiliza un disco duro de 107GB, 12.7GB de memoria RAM y una tarjeta gráfica NVIDIA Tesla T4 con 16GB de memoria RAM GDDR6 con ECC. Además, al hacer uso del lenguaje de programación Python, se mencionan las librerías utilizadas y una breve explicación de cada una para la solución al problema planteado.

- Numpy: Numpy es una librería de Python que destaca en el análisis de datos en masa y los cálculos numéricos. Ésta hace uso de "arreglos" que engloba datos del mismo tipo en

dimensiones que tienen como delimitación la memoria RAM disponible del usuario. Es útil para hacer que las imágenes de RM se vuelvan un cubo (“*tensor*”) que ahora se pueden procesar para poder tomarse como entradas de la red neuronal, la librería Numpy hace este “puente” entre la imagen original y el arreglo de números que la RNC necesita para comenzar el entrenamiento[43].

- Nibabel: Nibabel es utilizada para leer, escribir y cargar archivos de imágenes comunes como las extensiones “*DICOM*”, “*Nifti*” y “*ANALYZE*”. Una vez que se cargan los archivos de imagen de RM, se obtienen los datos de la imagen en forma de arreglo NumPy y se manipulan los datos [44].
- Multiprocessing: Una paquetería que permite la creación de procesos mediante una interfaz de programación de aplicaciones (API). Ésta permite aprovechar al máximo el rendimiento del equipo que hará el entrenamiento y realizar procesos en paralelo para reducir tiempos de aprendizaje y solo usando los recursos que necesita[45].
- Tensorflow: Tensorflow es una librería que facilita la creación de modelos de aprendizaje automático que se puede ejecutar en cualquier entorno. Incluye optimizadores, convoluciones en 2 y 3 dimensiones, desconexión de neuronas y modelos de deep learning[46].
- scikit-learn: Esta librería realiza tareas de ML como regresión, clasificación y clusterización. Asimismo, la validación mediante pliegues (*k-fold*) está disponible en esta librería [47].
- *Garbage collector*: La interfaz *garbage collector* (*gc*), como su nombre lo indica, sirve para recoger y clasificar datos que se han usado, fueron usados una vez o nunca fueron usados, haciendo que se libere memoria que puede ser útil para el proceso de entrenamiento, mejorando así la capacidad computacional y no ocupando espacio innecesario [48].

### 3.2.1. Arquitectura de RNC

La RNC inicialmente parte de una arquitectura simple con entradas de tamaño 62x320x280x80 siendo la primera entrada el número de imágenes, la segunda y tercera entrada el tamaño de largo y ancho, siendo el último valor el número de mapas de características que ingresan inicialmente a la RNC. Después, estas imágenes entran en 2 capas convolucionales con 32 y 64 filtros respectivamente de tamaño 3x3x3. Después de cada convolución se hace uso de una capa de pooling en forma *Average* de tamaño 2x2x2. Una vez saliendo de la última capa de pooling, se pasa a una capa completamente conectada de 10 neuronas que después pasan por última neurona cuya salida correspondió a la predicción de la concentración del hierro. La capa completamente conectada tiene desconexión de neuronas (*Dropout*). Se hace uso del descenso de gradiente estocástico, el optimizador Adam y la validación por pliegues (*K-fold*). Del conjunto de 62 RM, el conjunto de validación se tomaba como  $\frac{1}{k}$  del conjunto de entrenamiento por cada pliegue, se utilizó  $k = 5$  para tener 50 resonancias para entrenamiento y 12 para validación en cada pliegue.

La figura 3.4 muestra el diseño esquemático de la primera RNC, mientras que el diagrama de flujo de la figura 3.5 muestra de forma muy general los procesos y modificaciones de dimensiones que van sufriendo las imágenes hasta llegar a la capa de salida. La red inicial consta de 2 capas de convolución y una capa completamente conectada con 10 neuronas. Las modificaciones para la propuesta de una nueva hipótesis (en el contexto de redes neuronales) fue añadir más capas convolucionales y modificar el número de neuronas de la capa completamente conectada.

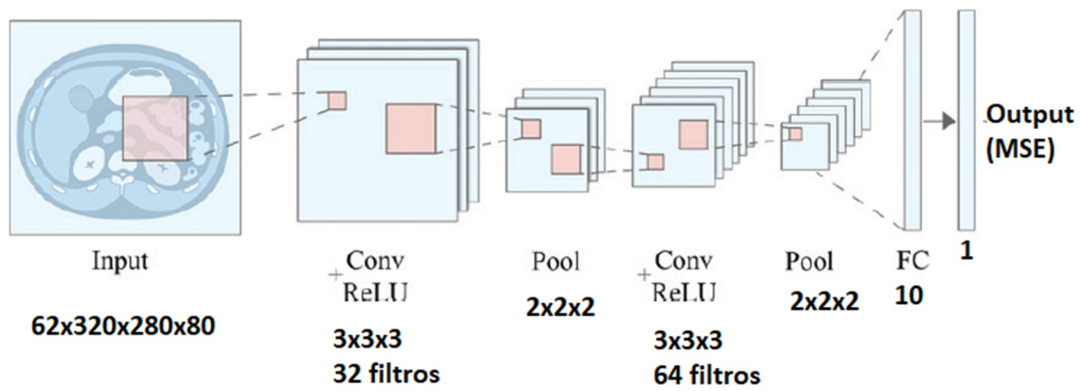


Figura 3.4: Arquitectura de RNC inicial, se muestran las imágenes de entrada, capas de convolución, pooling y red neuronal completamente conectada.

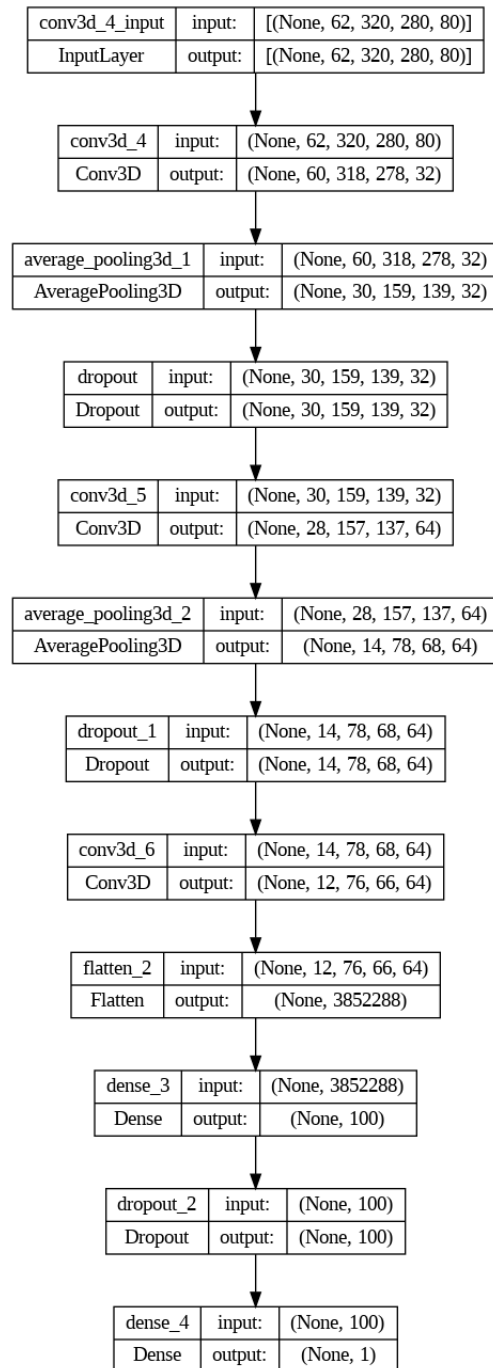


Figura 3.5: Arquitectura de RNC inicial en forma de diagrama de flujo, para esta arquitectura se tienen en la capa de la RNA (capa completamente conectada) 3852288 características.



## Capítulo 4

# Resultados

### 4.1. Resultados de los métodos estadísticos

Capas de convolución	Neuronas de la RNA	Pearson	Valor p	¿Significativa?	d-Cohen
1	10	0.3	$1,5x10^{-2}$	Si	0.64
1	15	0.13	$2,83x10^{-1}$	No	0.27
1	20	0.25	$4,6x10^{-2}$	Si	0.52
1	25	0.14	$2,69x10^{-1}$	No	0.28
2	10	0.007	$9,54x10^{-1}$	No	0.01
2	15	0.14	$2,6x10^{-1}$	No	0.29
2	20	0.18	$1,59x10^{-1}$	No	0.36
2	25	0.21	$8,9x10^{-2}$	No	0.44
3	10	0	$9,99x10^{-1}$	No	0
3	15	0.38	$1x10^{-3}$	Si	0.83
3	20	0.4	$1x10^{-3}$	Si	0.88
3	25	0.44	$2x10^{-4}$	Si	0.99
4	10	0.42	$4x10^{-4}$	Si	0.95
4	15	0.47	$1x10^{-4}$	Si	1.066
4	20	0.44	$2x10^{-4}$	Si	1
4	25	0.78	$3,6x10^{-8}$	Si	2.54
5	10	0.45	$2x10^{-4}$	Si	1.009
5	15	0.39	$1x10^{-4}$	Si	0.85
5	20	0.28	$2,4x10^{-2}$	Si	0.59
5	25	0.35	$5x10^{-3}$	Si	0.75

Tabla 4.1: Tabla de configuraciones de RNC, coeficiente de Pearson, valor p, significancia y d de Cohen

Se realizó la prueba de *Smirnov – Kolmogorov* para mostrar si los datos son o no paramétricos, se presenta en la figura 4.1. Este gráfico utiliza el valor “p” para determinar si la distribución es normal o no, el valor p fue de 0,00035 lo que indica que no es paramétrica la distribución.

Por último se realizó un gráfico de Bland-Altman junto con un gráfico de dispersión para cada una de las salidas de cada RNC para observar de manera gráfica cada una de las predicciones obtenidas por las hipótesis, además, en la tabla 4.1 se pueden ver las modificaciones realizadas

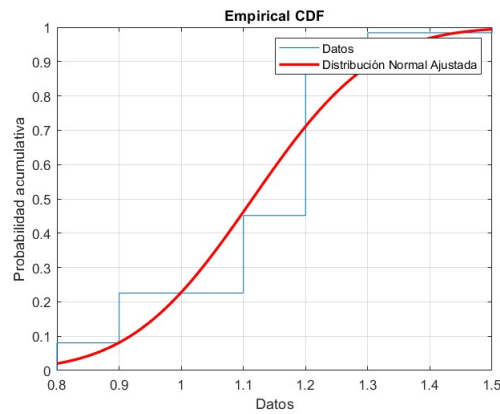


Figura 4.1: Prueba de Smirnov-Kolmogorov.

a la red neuronal principal y como estos cambios repercutieron en los estadísticos utilizados. La tabla 4.2 muestra el coeficiente de variación, el coeficiente de reproducibilidad, y los valores de las líneas horizontales (media de las diferencias, límites superior e inferior) obtenidos de cada gráfico de Bland-Altman.

Las figuras 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, 4.19, 4.20 y 4.21 muestran cada una de las configuraciones descritas por la tabla 4.1. Se proporciona una representación visual de la diferencia entre dos mediciones en el eje y, es decir, las mediciones hechas por Perspectum y las mediciones hechas por las RNCs. Asimismo, se toma en consideración el promedio de las dos mediciones en el eje x. Se considera también un límite de  $\pm 1,96 * s_d$  para la existencia de errores de medición y la consideración del 95 % de los datos que deben estar en ese rango.

## Resultados

### 4.1 Resultados de los métodos estadísticos

Arquitectura	Media de las diferencias	Desviación estándar(SD) de las diferencias	Media +1.96*SD	Media-1.96*SD	Coefficiente de variación (CV)	Coefficiente de reproductibilidad (CR)
1	0.13	0.34	0.80	-0.55	32.75	0.68
2	0.25	0.62	-0.11	18.94	0.37	
3	0.36	0.21	0.76	-0.04	21.95	0.40
4	0.25	0.19	0.62	-0.11	18.94	0.37
5	-0.06	1.16	2.20	-2.33	100.91	2.27
6	0.15	0.17	0.47	-0.18	15.93	0.32
7	0.17	0.15	0.48	-0.13	15.05	0.30
8	0.19	0.15	0.48	-0.11	14.78	0.30
9	0.01	0.15	0.32	-0.29	13.86	0.30
10	0.14	0.15	0.44	-0.16	14.45	0.30
11	0.14	0.15	0.43	-0.15	14.20	0.30
12	0.13	0.15	0.41	-0.16	13.80	0.28
13	0.12	0.15	0.41	-0.16	13.94	0.29
14	0.12	0.14	0.40	-0.16	13.39	0.28
15	0.11	0.14	0.39	-0.17	13.53	0.28
16	0.02	0.09	0.21	-0.16	8.60	0.19
17	0.11	0.14	0.39	-0.16	13.31	0.28
18	0.10	0.14	0.39	-0.19	13.71	0.29
19	0.09	0.16	0.40	-0.21	14.67	0.30
20	0.06	0.15	0.36	-0.24	14.21	0.30

Tabla 4.2: Tabla de métricas obtenidas de los gráficos de Bland-Altman para cada arquitectura de red neuronal

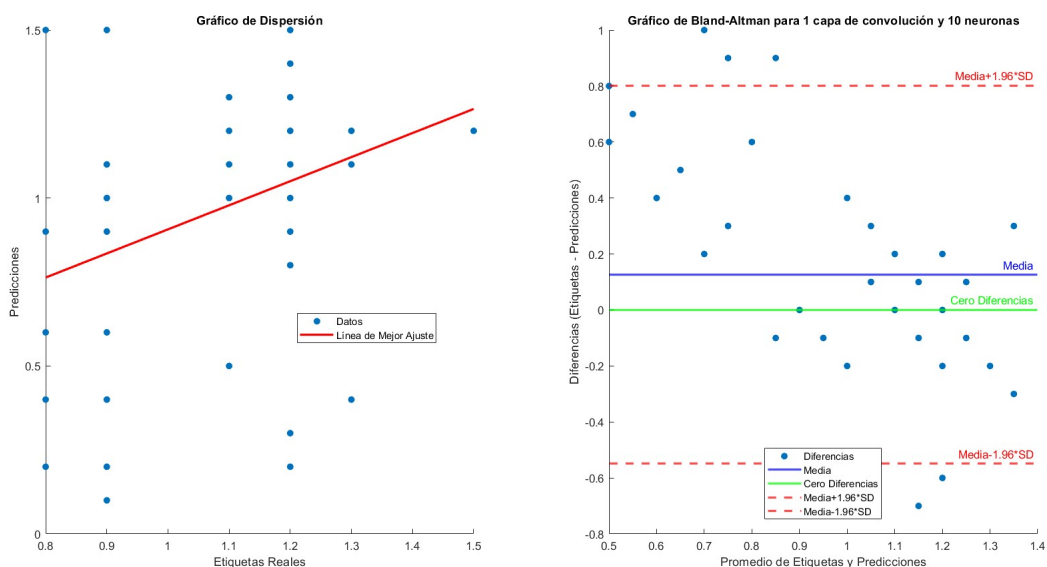


Figura 4.2: Gráfico de dispersión y Bland-Altman para una capa de convolución y 10 neuronas en la capa completamente conectada.

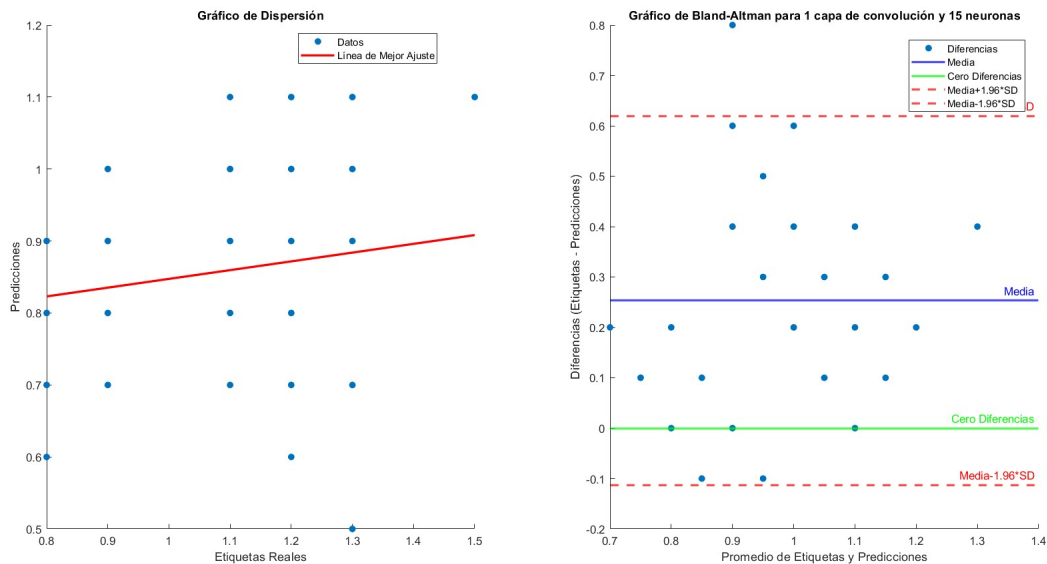


Figura 4.3: Gráfico de dispersión y Bland-Altman para una capa de convolución y 15 neuronas en la capa completamente conectada.

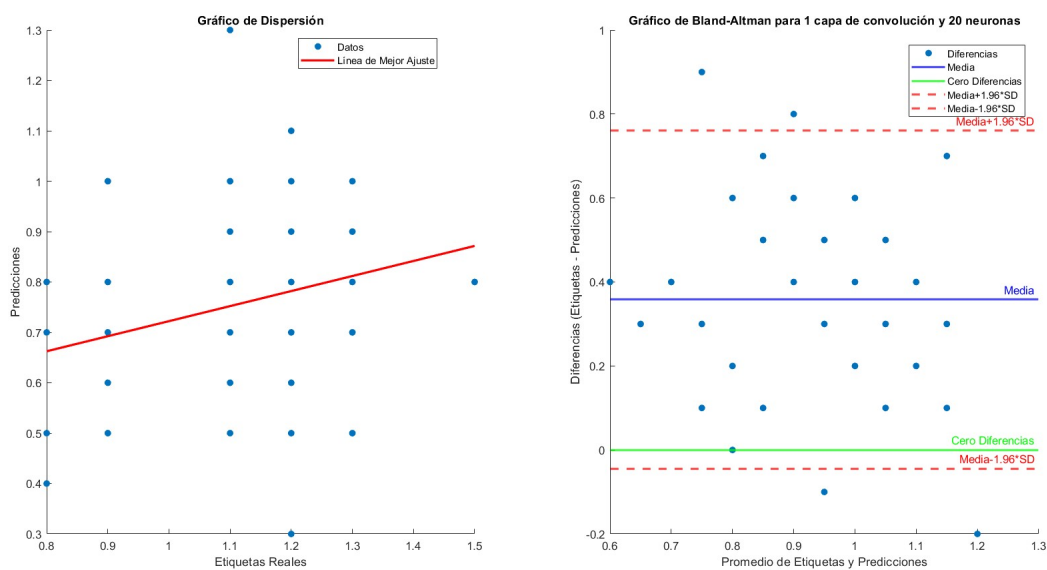


Figura 4.4: Gráfico de dispersión y Bland-Altman para una capa de convolución y 20 neuronas en la capa completamente conectada.

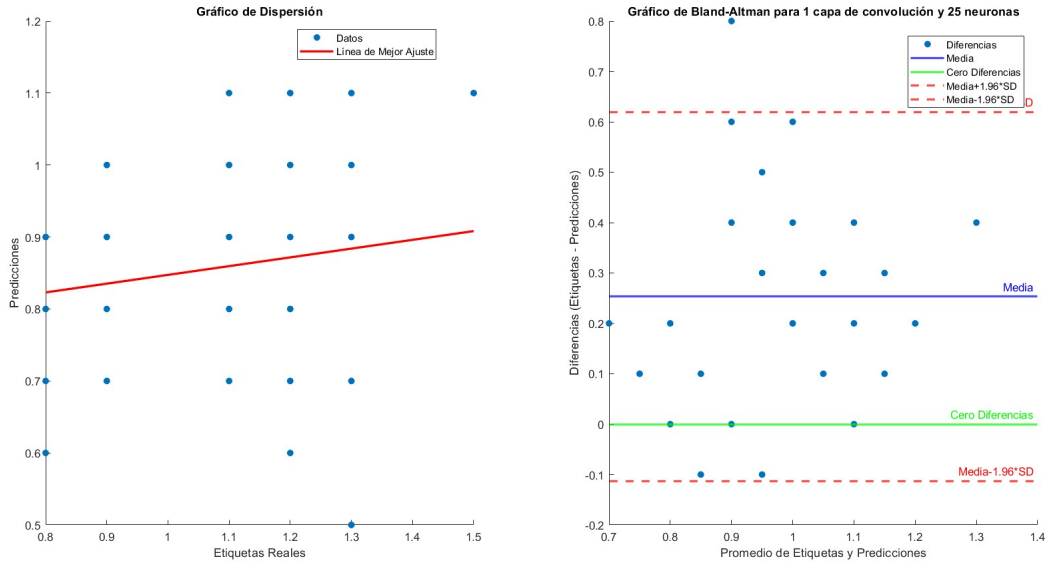


Figura 4.5: Gráfico de dispersión y Bland-Altman para una capa de convolución y 25 neuronas en la capa completamente conectada.

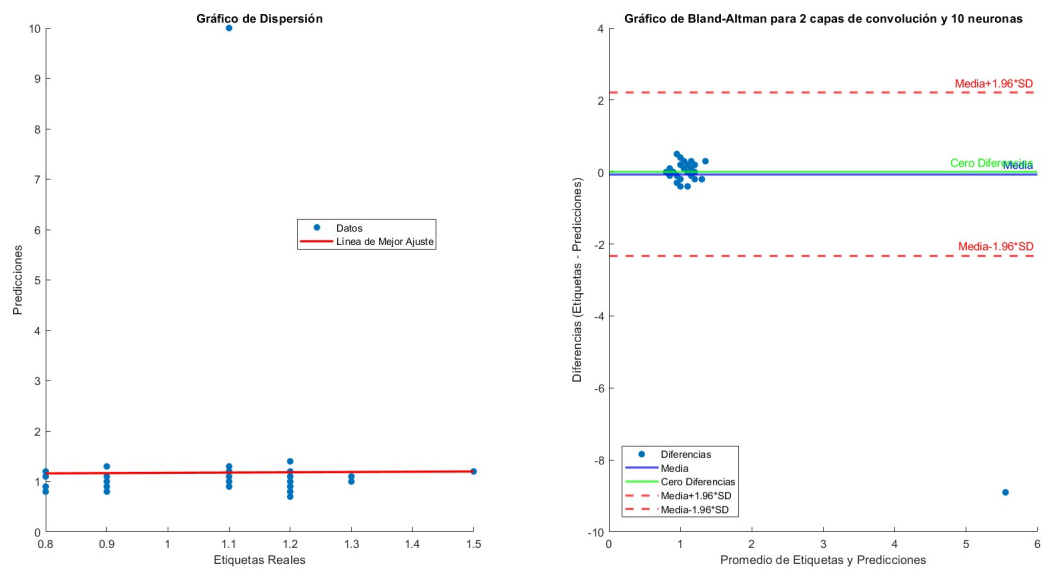


Figura 4.6: Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 10 neuronas en la capa completamente conectada.

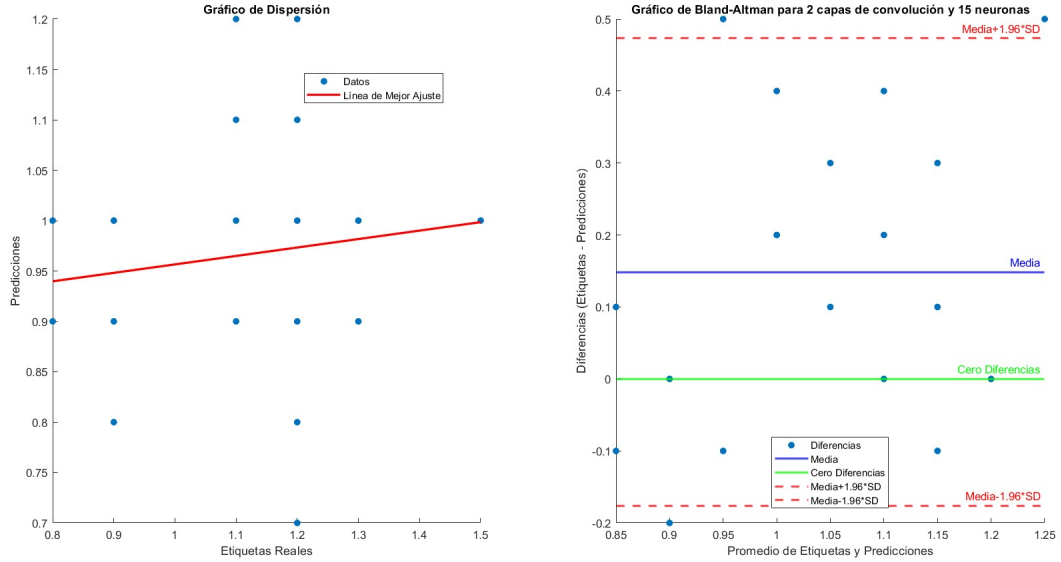


Figura 4.7: Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 15 neuronas en la capa completamente conectada.

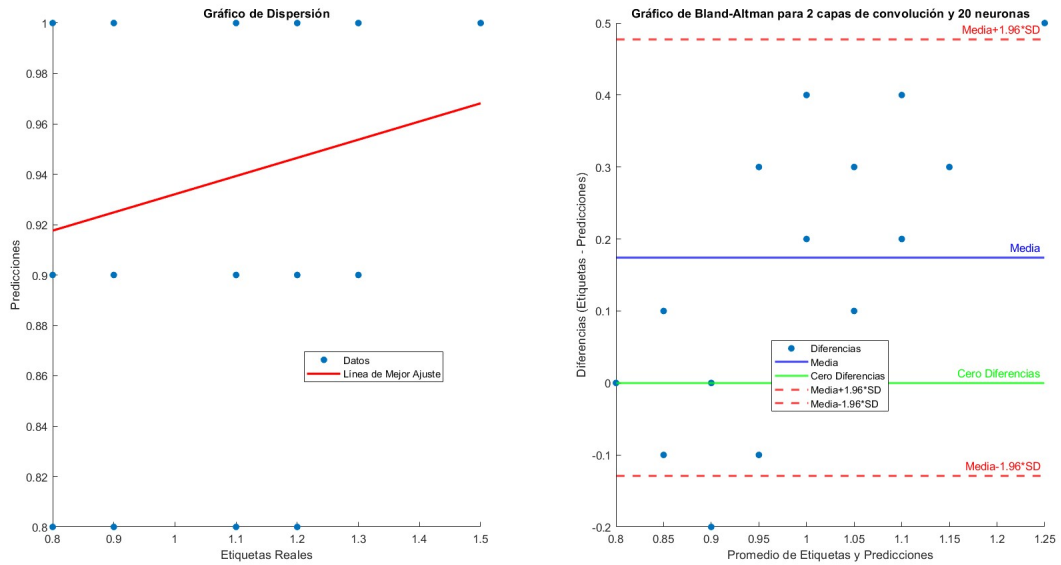


Figura 4.8: Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 20 neuronas en la capa completamente conectada.

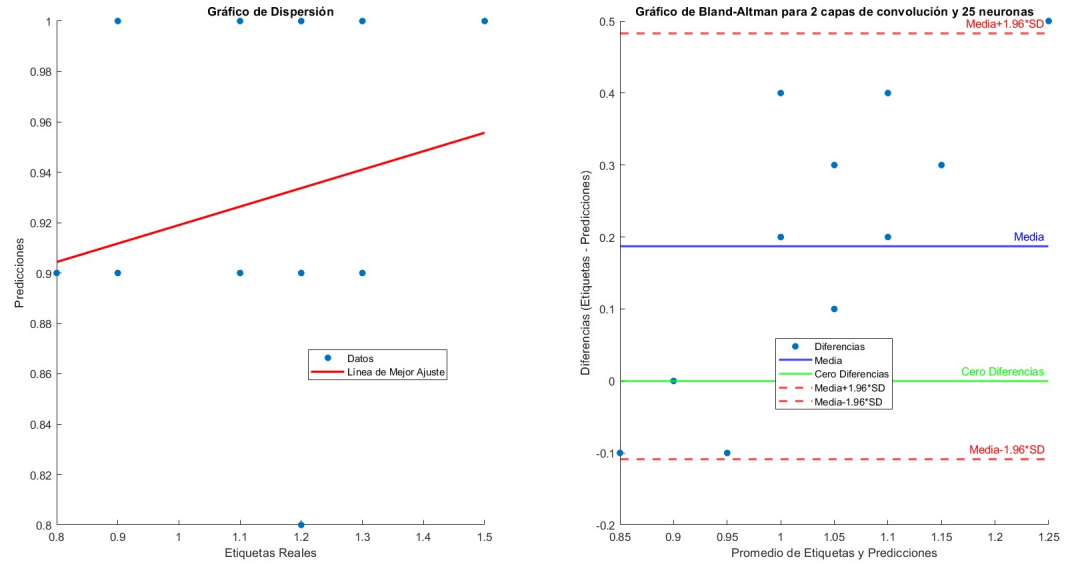


Figura 4.9: Gráfico de dispersión y Bland-Altman para 2 capas de convolución y 25 neuronas en la capa completamente conectada.

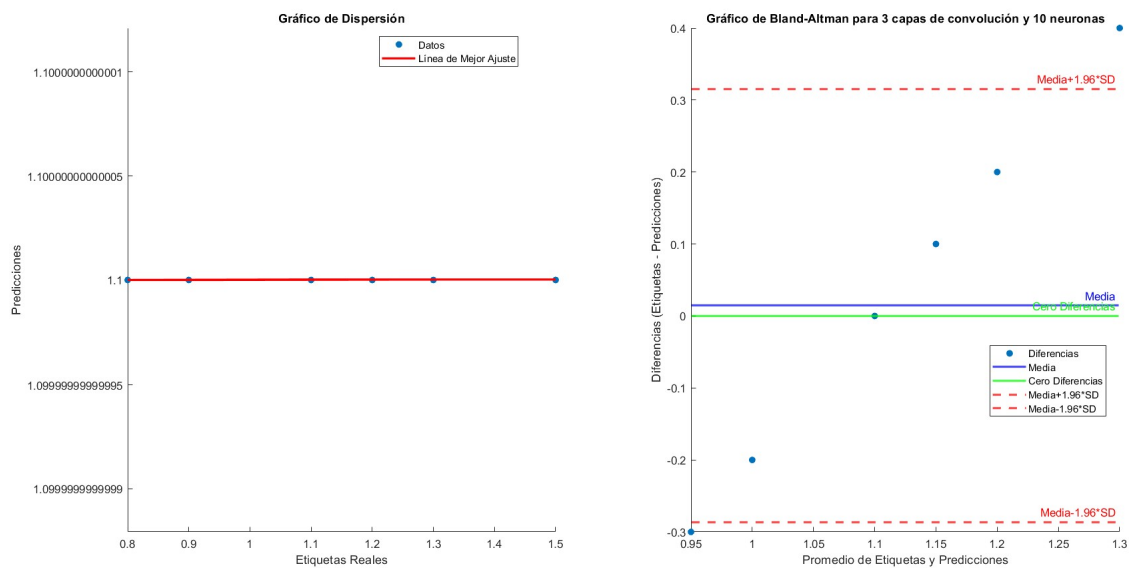


Figura 4.10: Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 10 neuronas en la capa completamente conectada.

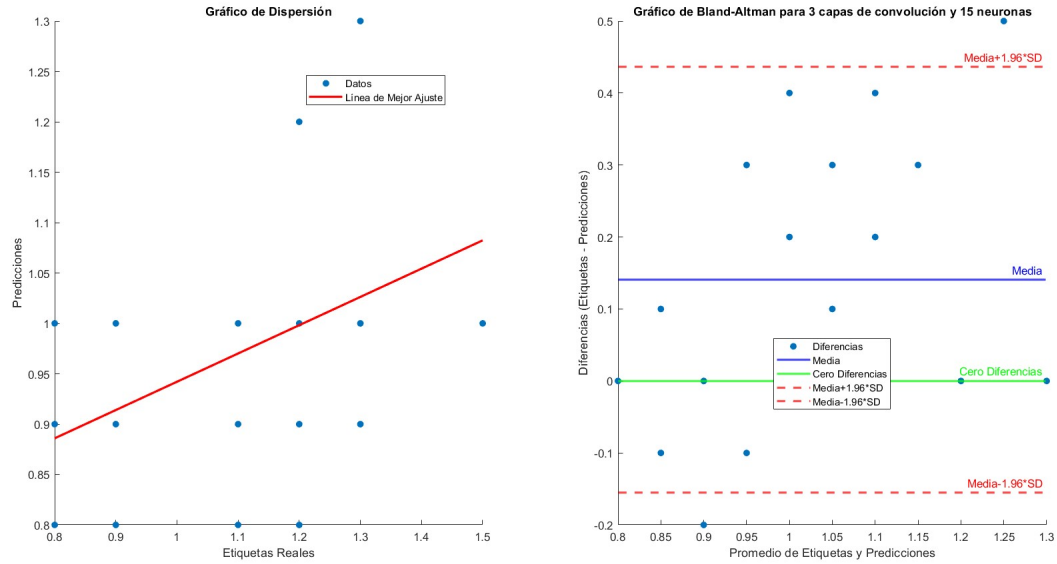


Figura 4.11: Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 15 neuronas en la capa completamente conectada.

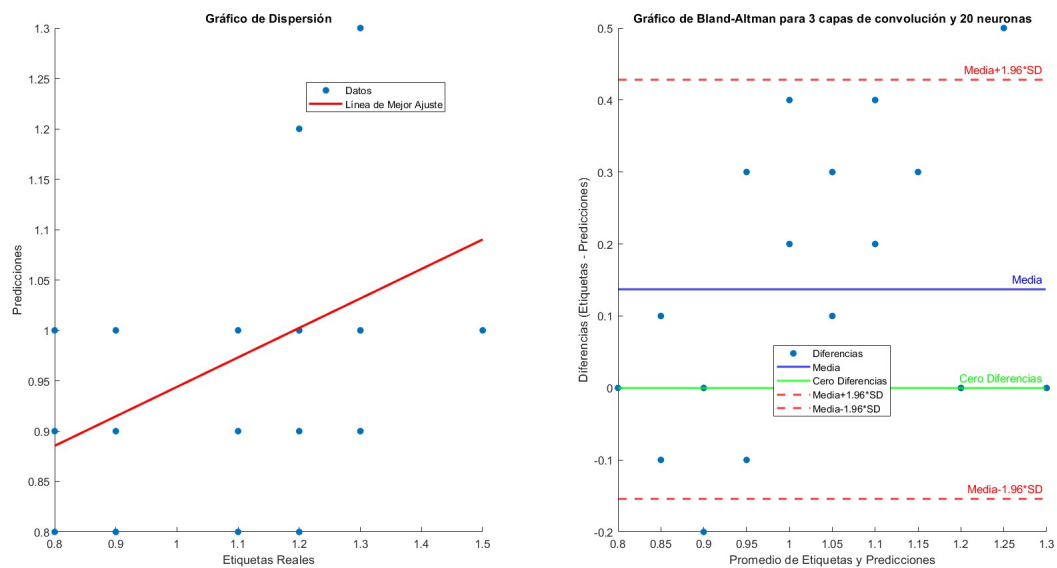


Figura 4.12: Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 20 neuronas en la capa completamente conectada.

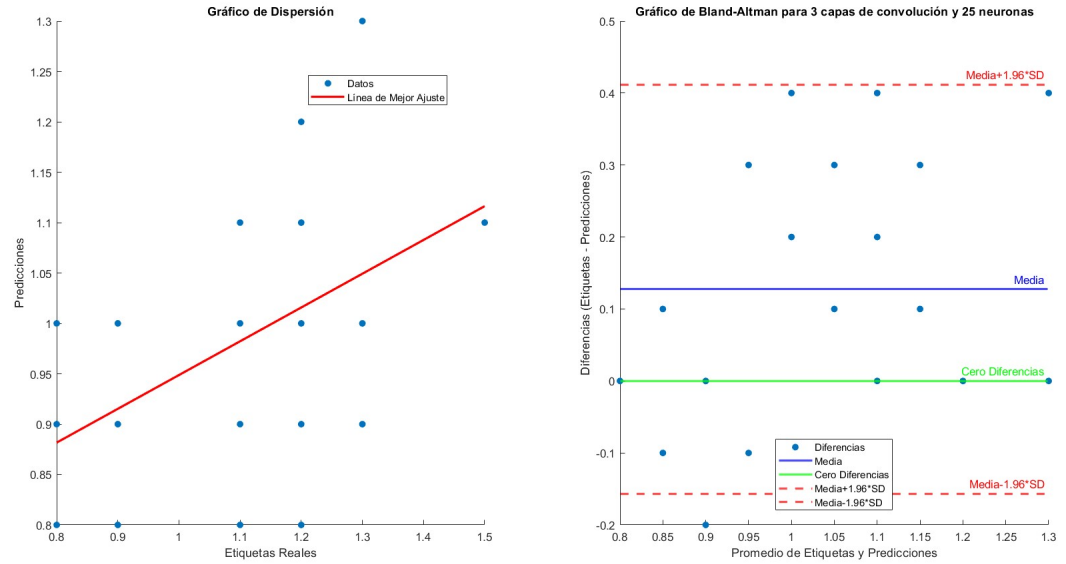


Figura 4.13: Gráfico de dispersión y Bland-Altman para 3 capas de convolución y 25 neuronas en la capa completamente conectada.

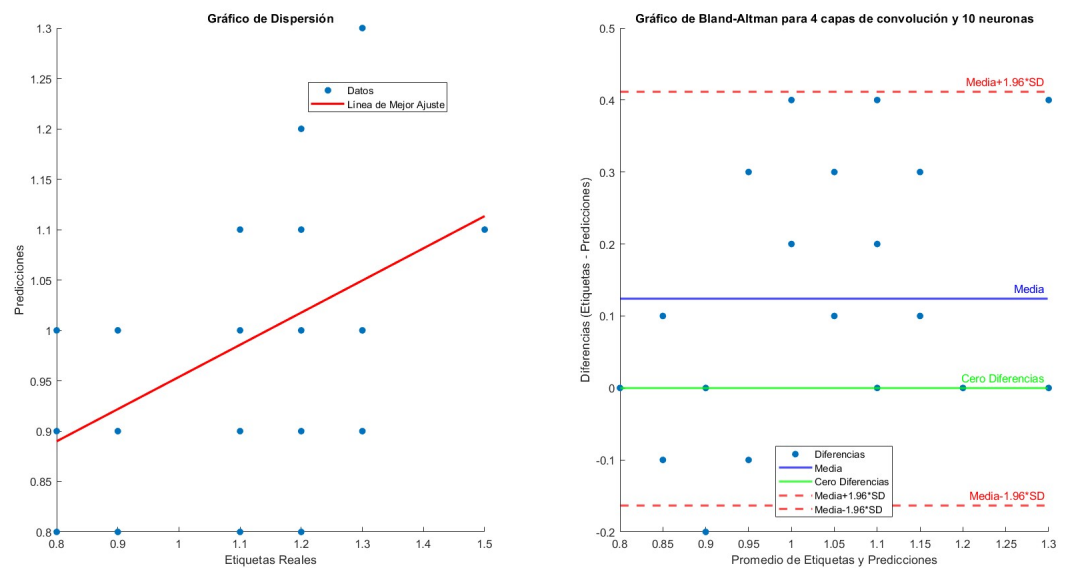


Figura 4.14: Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 10 neuronas en la capa completamente conectada.

## Resultados

### 4.1 Resultados de los métodos estadísticos

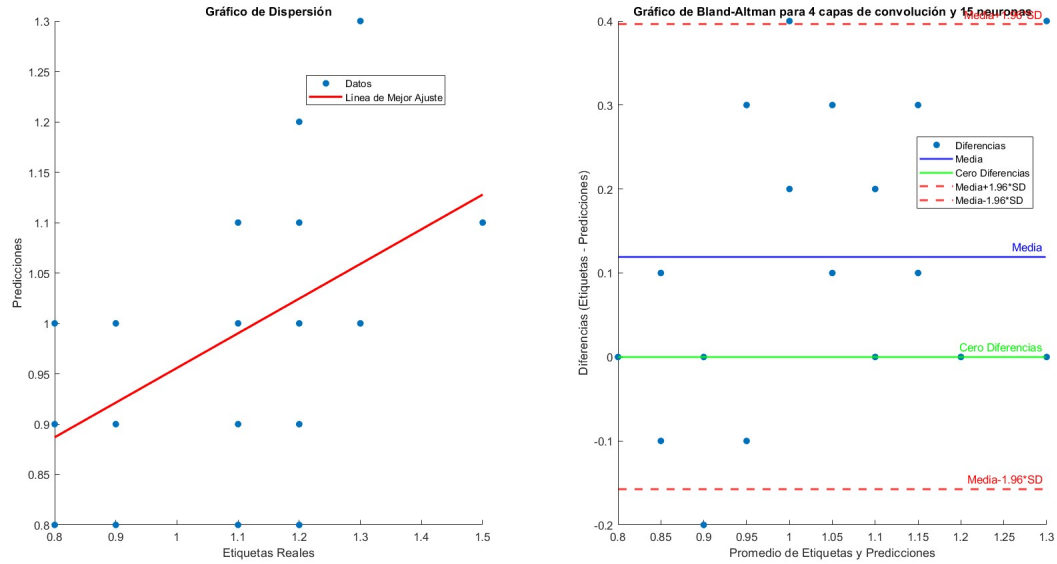


Figura 4.15: Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 15 neuronas en la capa completamente conectada.

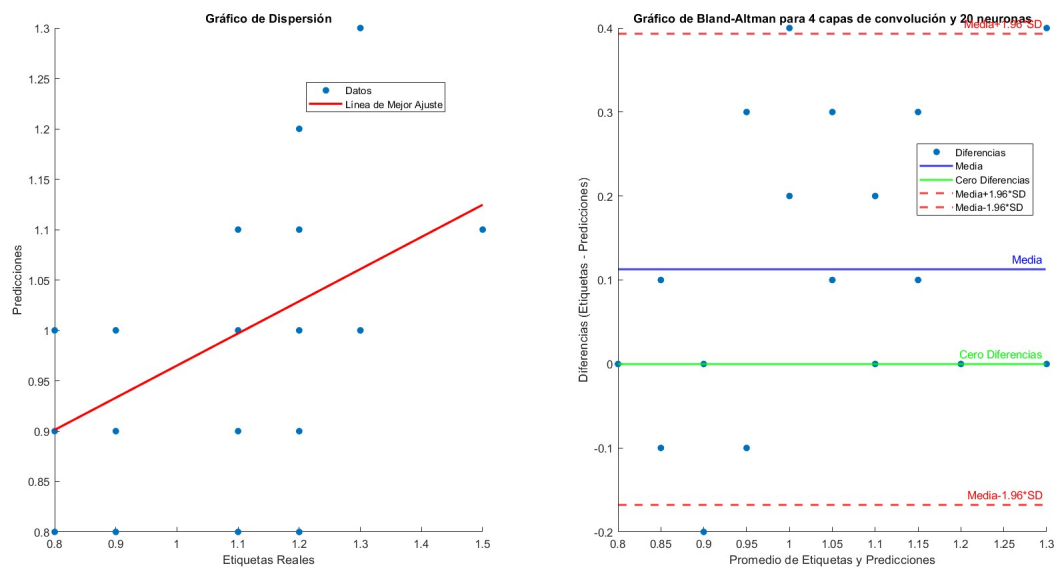


Figura 4.16: Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 20 neuronas en la capa completamente conectada.

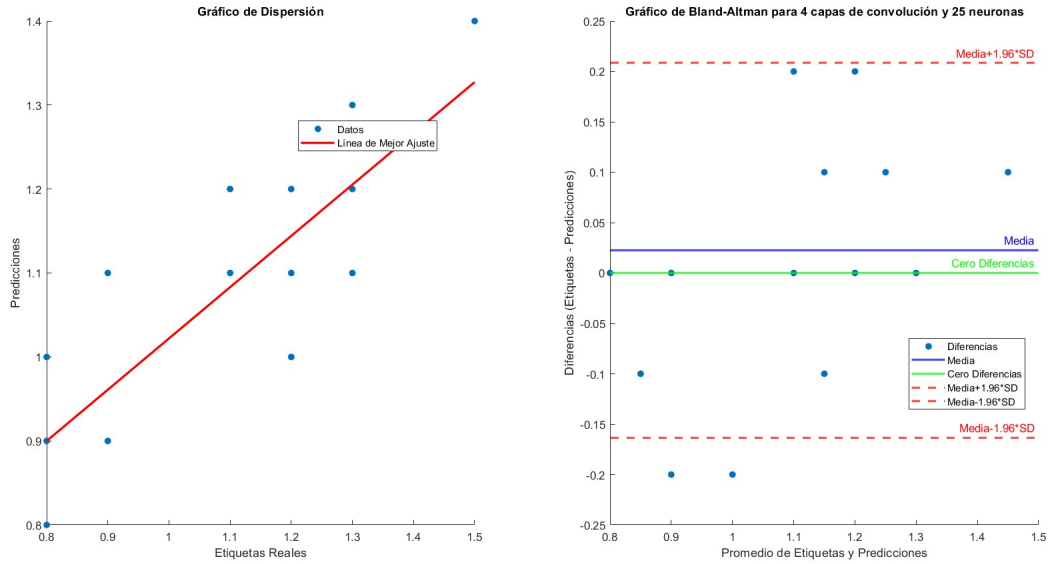


Figura 4.17: Gráfico de dispersión y Bland-Altman para 4 capas de convolución y 25 neuronas en la capa completamente conectada.

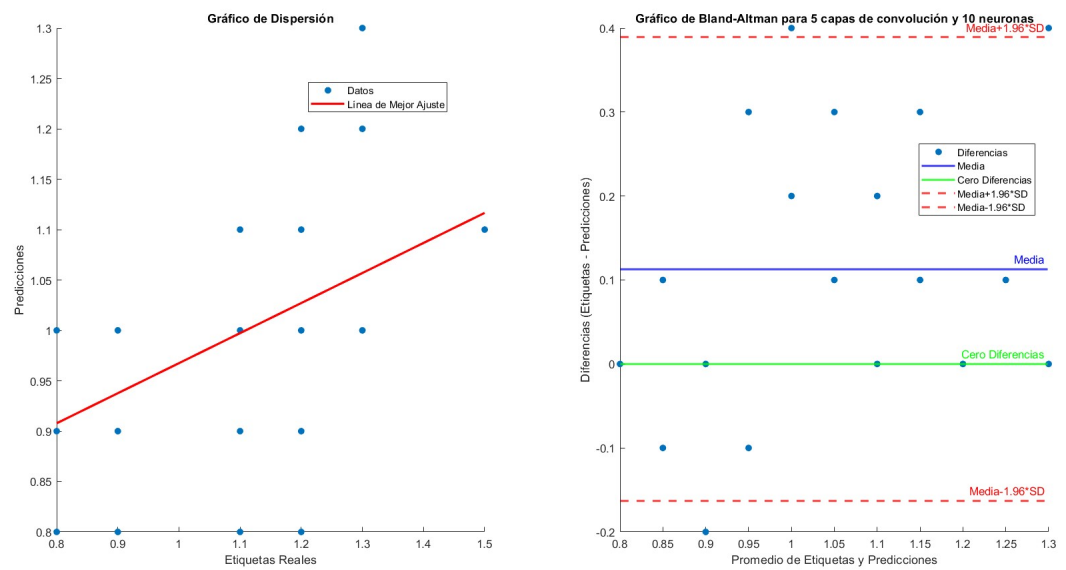


Figura 4.18: Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 10 neuronas en la capa completamente conectada.

## Resultados

### 4.1 Resultados de los métodos estadísticos

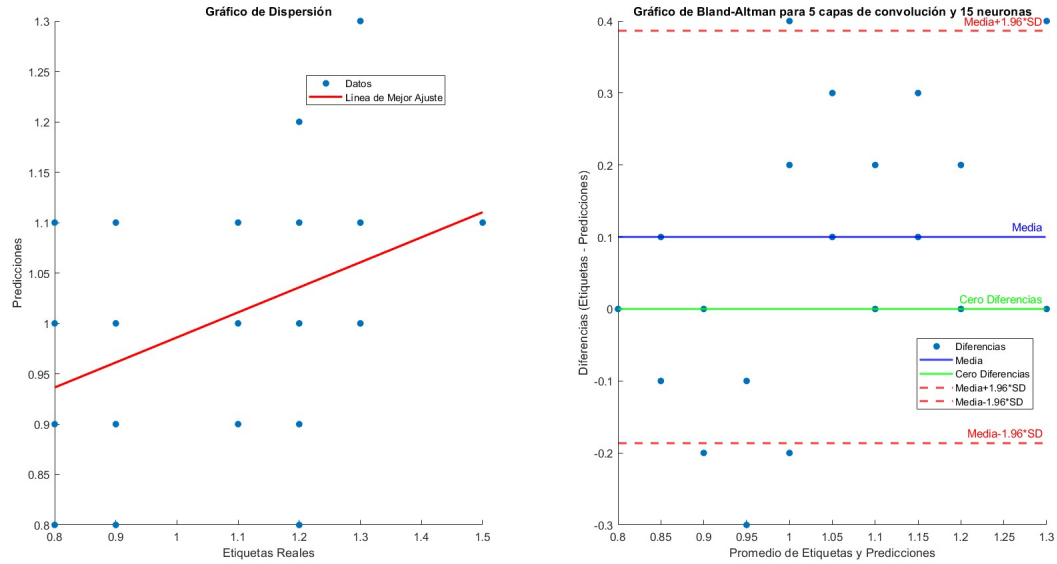


Figura 4.19: Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 15 neuronas en la capa completamente conectada.

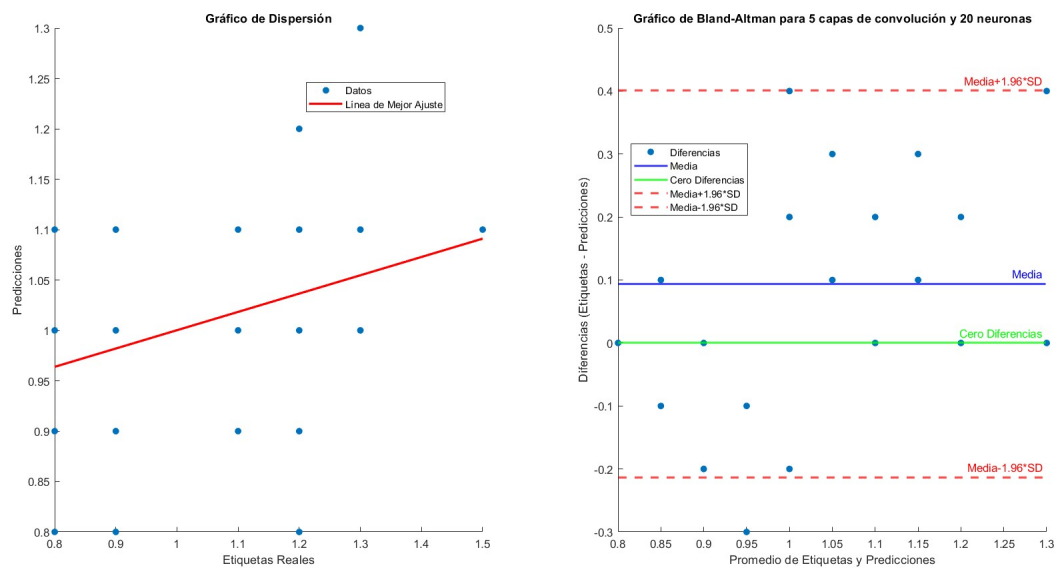


Figura 4.20: Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 20 neuronas en la capa completamente conectada.

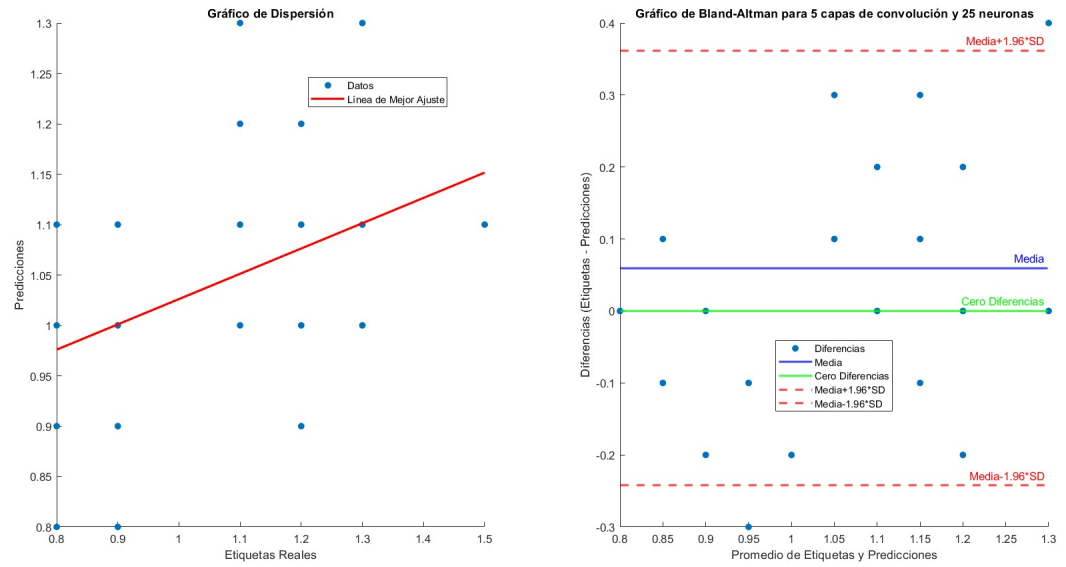


Figura 4.21: Gráfico de dispersión y Bland-Altman para 5 capas de convolución y 20 neuronas en la capa completamente conectada.

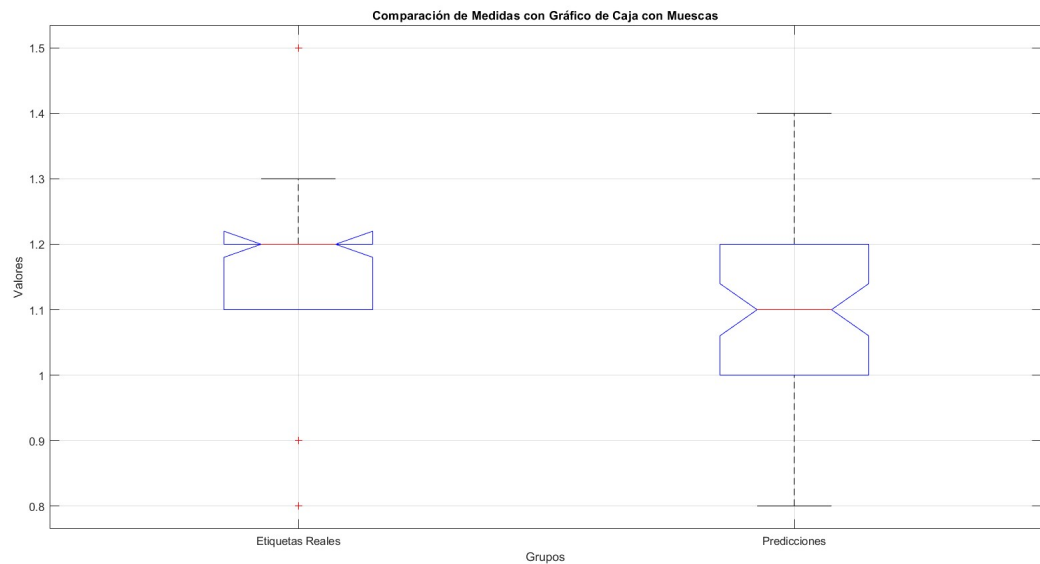


Figura 4.22: Diagrama de cajas, comparación de etiquetas reales y predicciones.



# Capítulo 5

## Discusión

### 5.1. Normalización y corrección de inhomogeneidades

El uso de algoritmos para la corrección de inhomogeneidades permite a las RNC un aprendizaje más robusto, es decir, al haber corregido las imágenes se tiene una mayor capacidad de la CNN para detectar de forma espacial patrones y poder realizar el trabajo de predicción.

### 5.2. Arquitecturas de RNC

Las configuraciones de RNC probadas variaron en el número de capas de convolución y en el número de neuronas en la capa completamente conectada. La tabla 4.1 resume los coeficientes de correlación de Pearson, los valores p y los valores de d de Cohen para las diferentes configuraciones de la red neuronal. Los resultados indican que las configuraciones más complejas tienden a obtener mejores resultados en términos de correlación y significancia.

Redes de una capa de convolución: Las configuraciones con una sola capa de convolución presentaron coeficientes de correlación relativamente bajos, oscilando entre 0.007 y 0.3. Las configuraciones con 10 y 20 neuronas en la capa completamente conectada fueron las únicas que mostraron significancia estadística, con valores p de 0.015 y 0.046, respectivamente, y valores de d de Cohen de 0.64 y 0.52, indicando un efecto moderado. Sin embargo, las configuraciones con 15 y 25 neuronas no alcanzaron significancia estadística (valores p  $>0.05$ ) y presentaron efectos pequeños (d de Cohen  $<0.3$ ).

Redes de dos capas de convolución: Estas configuraciones no mostraron significancia estadística en ninguna de las configuraciones probadas, con coeficientes de correlación variando entre 0.007 y 0.21, y valores p mayores a 0.05. Los valores de d de Cohen, aunque superiores a las redes de una capa, permanecieron en el rango de efectos pequeños a moderados (0.01 a 0.44).

Redes de tres capas de convolución: Las configuraciones con 15, 20 y 25 neuronas mostraron coeficientes de correlación altos (0.38, 0.4 y 0.44, respectivamente) y fueron estadísticamente significativas con valores p muy bajos ( $<0.005$ ). Los valores de d de Cohen fueron considerablemente más altos (0.83 a 0.99), indicando efectos grandes.

Redes de cuatro capas de convolución: Estas configuraciones mostraron un rendimiento aún mejor, con coeficientes de correlación de hasta 0.78 y valores p extremadamente significativos (0.000000036). Los valores de d de Cohen también aumentaron, con el más alto (2.54) observado

en la configuración con 25 neuronas, indicando un efecto muy grande.

Redes de cinco capas de convolución: Aunque estas configuraciones también mostraron significancia estadística, sus coeficientes de correlación fueron ligeramente inferiores a las redes de cuatro capas. Sin embargo, los valores de  $d$  de Cohen permanecieron altos, lo que sugiere que agregar más capas no necesariamente mejora el rendimiento.

### 5.3. Métricas obtenidas de Bland-Altman

Los gráficos de Bland-Altman proporcionan una manera efectiva de evaluar la consistencia y la reproducibilidad entre las predicciones del modelo y los valores reales. Los datos resumidos en la tabla 4.2 incluyen la media de las diferencias, la desviación estándar (SD) de las diferencias, los límites de media  $\pm 1.96*SD$ , el coeficiente de variación (CV) y el coeficiente de reproducibilidad (CR).

Una media cercana a cero indica que no hay un sesgo sistemático significativo entre las predicciones del modelo y los valores reales. En este estudio, las configuraciones de redes con cuatro capas de convolución mostraron medias de diferencias bajas, lo que sugiere un bajo sesgo. La red de 4 capas y 25 neuronas tuvo una media de 0.12. Además, una menor desviación estándar de las diferencias indica una mayor precisión del modelo. Las configuraciones con menor desviación estándar fueron las redes de 4 y 5 capas, lo que sugiere que estos modelos son más precisos.

Los límites superiores e inferiores más estrechos se observaron en configuraciones con 4 y 5 capas de convolución. Para la configuración de 4 capas y 25 neuronas mostró límites de acuerdo de 0.41 y -0.16, lo que indica que el 95% de las diferencias entre las predicciones y los valores reales están dentro de este rango. El coeficiente de variabilidad (CV) bajo indica menor variabilidad relativa en las predicciones. Las configuraciones con 4 capas de convolución tuvieron CVs consistentemente bajos, con el menor CV (13.39) en la configuración con 25 neuronas. El coeficiente de reproductibilidad (CR) proporciona una medida directa de reproducibilidad. Las configuraciones con CRs más bajos fueron nuevamente las redes de 4 y 5 capas, destacando su alta reproducibilidad.

Los diagramas de dispersión de las predicciones del modelo frente a los valores reales muestran visualmente la relación lineal entre estos dos conjuntos de datos. Un diagrama de dispersión con puntos cercanos a la línea de identidad ( $y = x$ ) indica una buena concordancia entre las predicciones y los valores reales.

Para los diagramas de dispersión que tenían configuraciones de 1 y 2 capas de convolución mostraron una mayor dispersión de puntos alrededor de la línea de identidad, indicando una menor precisión y una mayor variabilidad en las predicciones. La correlación más baja y la significancia estadística ausente o baja se reflejan en la mayor dispersión observada. En las configuraciones de 3 capas se observó una mejor alineación de los puntos con la línea de identidad, especialmente en las configuraciones con 15, 20 y 25 neuronas. Esto coincide con los coeficientes de correlación más altos y los valores de  $d$  de Cohen elevados, sin embargo, dadas las pocas diferencias que existen entre cada arquitectura de 3 capas de convolución junto con las 2 primeras arquitecturas de 4 capas (10 y 15 neuronas en la capa completamente conectada) se muestran gráficos que visualmente son similares pero estadísticamente diferentes al igual que los coeficientes obtenidos de cada gráfico se tienen diferencias.

Las configuraciones de 4 y 5 capas mostraron la mayor concentración de puntos cerca de la línea de identidad.

En particular, la configuración de 4 capas y 25 neuronas mostró una baja dispersión de puntos con respecto a la línea de ajuste y a los otros gráficos, corroborando sus altos coeficientes de correlación y bajos valores de CV y CR. La significancia estadística alta se refleja en la cercanía de los puntos a la línea de identidad, indicando muy poca variabilidad no explicada.

De acuerdo con el diagrama de cajas y bigotes de la figura 4.22 se tiene que las etiquetas reales presentan distintas partes aisladas, es decir, valores que se encuentran significativamente alejados del resto de los datos. Esto muestra que es necesario robustecer la arquitectura de RNC para poder aprender y predecir correctamente los valores atípicos del conjunto de datos. Cabe aclarar que no se asume tener un subajuste pero sí se muestra que se puede mejorar el modelo de ML para una adaptación más adecuada al problema.



## Capítulo 6

# Conclusiones

El presente estudio se centró en la cuantificación de hierro mediante resonancia magnética utilizando redes neuronales convolucionales (CNN). Se evaluaron múltiples configuraciones arquitectónicas, variando el número de capas de convolución y la cantidad de neuronas en la capa completamente conectada. Los resultados obtenidos de estas configuraciones fueron analizados a través de diversas métricas estadísticas, incluyendo el coeficiente de correlación de Pearson, valor  $p$ ,  $d$  de Cohen, así como métricas derivadas de los gráficos de Bland-Altman como el coeficiente de variación (CV) y el coeficiente de reproducibilidad (CR).

Las configuraciones más complejas de las redes neuronales convolucionales, específicamente aquellas con 4 capas de convolución y 25 neuronas en la capa completamente conectada, demostraron ser las más efectivas para la tarea de cuantificación de hierro. Esta configuración obtuvo un coeficiente de correlación de Pearson de 0.78, indicando una fuerte relación lineal entre las predicciones del modelo y los valores reales. El valor  $p$  extremadamente bajo (0.000000036) confirmó la significancia estadística de esta relación, mientras que el  $d$  de Cohen de 2.54 indicó un efecto muy grande. Con este resultado se logró uno de los objetivos específicos el cual era alcanzar un valor de correlación mayor a 0.75 y que sea significativa la correlación.

El análisis de los gráficos de Bland-Altman respaldó estos resultados, mostrando que la configuración de 4 capas y 25 neuronas tenía límites de acuerdo estrechos y un bajo coeficiente de variación (CV), lo que sugiere una alta precisión y reproducibilidad en las predicciones. Estos hallazgos fueron consistentes con la mayor concentración de puntos cerca de la línea de identidad en los diagramas de dispersión, indicando una concordancia entre las predicciones del modelo y los valores reales.

La capacidad de predecir con precisión y consistencia los niveles de hierro a partir de imágenes de resonancia magnética tiene importantes implicaciones clínicas. Las configuraciones óptimas de CNN identificadas en este estudio podrían ser utilizadas para mejorar la monitorización y el diagnóstico de condiciones relacionadas con la sobrecarga de hierro, como la hemocromatosis y la talasemia. La alta reproducibilidad y precisión de las predicciones obtenidas con la configuración de 4 capas y 25 neuronas sugieren que este modelo es particularmente adecuado para aplicaciones clínicas donde la exactitud es crucial.

Aunque los resultados obtenidos son prometedores, es importante reconocer ciertas limitaciones del estudio. La variabilidad en la calidad de las imágenes de resonancia magnética y la necesidad de un mayor conjunto de datos para validar los hallazgos son aspectos que deben ser considerados en investigaciones futuras. Además, explorar técnicas adicionales como el aumento de datos, la regularización avanzada y la implementación de arquitecturas más sofisticadas podría mejorar aún más el rendimiento de las redes neuronales convolucionales en esta tarea.

Este estudio ha demostrado que la optimización de la arquitectura de redes neuronales convolucionales puede llevar a mejoras significativas en la cuantificación de hierro mediante resonancia magnética. La configuración de 4 capas de convolución y 25 neuronas en la capa completamente conectada se destacó como la más efectiva, proporcionando predicciones precisas y reproducibles, lo que subraya su potencial para aplicaciones clínicas críticas. Futuras investigaciones deberán enfocarse en ampliar estos hallazgos y explorar nuevas técnicas para seguir mejorando la precisión y la aplicabilidad de estos modelos en el campo de la medicina.

## Apéndice A

# Ecuaciones de Bloch

Las ecuaciones de Bloch son un conjunto de ecuaciones diferenciales, las cuales, describen los cambios en la magnetización debido al proceso de excitación y relajación. Estas ecuaciones describen lo que detecta la bobina receptora del resonador magnético. De manera clásica, las ecuaciones se presentan como:

$$\frac{d\mathbf{M}}{dt} = \gamma \mathbf{M} \times \mathbf{B} = \gamma [(M_y B_z - M_z B_y) \hat{i} + (M_z B_x - M_x B_z) \hat{j} + (M_x B_y - M_y B_x) \hat{k}]$$

De manera general, se considera al campo magnético  $\mathbf{B}$  como un campo estático a lo largo de la dirección z y otro campo que rota en el plano transversal al eje z, es decir:

$$B_x = B_1 \cos(\omega t) \quad B_y = B_1 \sin(\omega t) \quad B_z = B_0$$

Tomando en consideración la derivada temporal de cada componente de la magnetización, haciendo uso de las consideraciones anteriormente descritas para el campo magnético y tomando en cuenta los tiempos de relajación  $T_1$  y  $T_2$ , se tiene:

$$\begin{aligned} \frac{dM_x}{dt} &= \gamma (M_y B_0 + M_z B_1 \sin(\omega t)) - \frac{M_x}{T_2} \\ \frac{dM_y}{dt} &= \gamma (M_x B_1 \cos(\omega t) - M_x B_0) - \frac{M_y}{T_2} \\ \frac{dM_z}{dt} &= \gamma (M_x B_1 \sin(\omega t) + M_y B_1 \cos(\omega t)) - \frac{M_z - M_0}{T_1} \end{aligned}$$

Para la solución de las ecuaciones, se establecen condiciones específicas, por ejemplo, que al apagar o dejar de emitir pulsos de RF, el campo  $B_1$  se vuelve 0 y las soluciones serán:

$$\begin{aligned} M_x(t) &= [M_x(0) \cos(\omega_0 t) + M_y(0) \sin(\omega_0 t)] * \exp\left(\frac{-t}{T_2}\right) \\ M_y(t) &= [M_y(0) \cos(\omega_0 t) - M_x(0) \sin(\omega_0 t)] * \exp\left(\frac{-t}{T_2}\right) \\ M_z(t) &= M_z(0) \exp\left(\frac{-t}{T_1}\right) + M_0 [1 - \exp\left(\frac{-t}{T_1}\right)] \end{aligned}$$

Considerando que el sistema inicialmente está en equilibrio y con un pulso de RF de 90 a lo largo del eje x positivo, las magnetizaciones para el eje x y z serán 0, mientras que la magnetización a lo largo del eje y será de  $M_y(0) = M_0$ , las magnetizaciones entonces serán:

$$M_x(t) = M_0 \sin(\omega_0 t) * \exp\left(\frac{-t}{T_2}\right)$$

$$M_y(t) = M_0 \cos(\omega_0 t) * \exp\left(\frac{-t}{T_2}\right)$$

$$M_z(t) = M_0 [1 - \exp\left(\frac{-t}{T_1}\right)]$$

Las magnetizaciones en el plano xy se pueden ver en forma de exponenciales en el plano complejo, haciendo el resultado como:

$$M_{xy}(t) = M_0 \exp(i\omega_0 t) * \exp\left(\frac{-t}{T_2}\right)$$

Estas ecuaciones explican que la magnetización en el plano xy oscila con la frecuencia de Larmor mientras se tiene un decaimiento con una constante de relajación  $T_2$ , además, la magnetización en el eje z va de 0 a la magnetización inicial, es decir,  $M_0$ . [5]

## Apéndice B

# Transformada de Fourier

El uso de la transformada de Fourier para RM es convertir un mapa de posiciones a una "conjugación" del espacio mediante una función que permita realizar este intercambio de espacios. En una dimensión, la transformada de Fourier para RM se puede describir como:

$$s(\kappa) = \int_{-\infty}^{\infty} dx \rho(x) e^{-i2\pi\kappa x}$$

La densidad de los spines en el espacio de posiciones se transforma en un espacio asociado  $\kappa$ , también conocido como espacio de frecuencias la cual se relaciona con la señal recibida del resonador magnético. [6]

De manera general, la transformada de Fourier en una dimensión se puede ver como:

$$H(\kappa) \equiv \mathbf{F}(h(x)) = \int_{-\infty}^{\infty} dx h(x) e^{-i2\pi\kappa x}$$

Siendo su inversa:

$$h(x) \equiv \mathbf{F}^{-1}(H(x)) = \int_{-\infty}^{\infty} d\kappa H(\kappa) e^{+i2\pi\kappa x}$$

Para saber si la transformada inversa es la correcta, se demuestra a continuación tomando en consideración la delta de Dirac, la cual, en los espacios  $\kappa$  y  $x$  se representan de la siguiente forma:

$$\delta(\kappa - \kappa_0) = \int_{-\infty}^{\infty} dx e^{-i2\pi(\kappa - \kappa_0)x}$$

$$\delta(x - x_0) = \int_{-\infty}^{\infty} d\kappa e^{+i2\pi\kappa(x - x_0)}$$

Colocando la transformada de Fourier en la transformada inversa, se obtiene:

$$h(x) = \int_{-\infty}^{\infty} dx' h(x') \int_{-\infty}^{\infty} d\kappa e^{-i2\pi\kappa(x' - x)} = \int_{-\infty}^{\infty} dx' h(x') \delta(x - x') = h(x) \quad \square$$

Finalmente, si se conocieran todos los espacios  $\kappa$ , se podría determinar una densidad de señales de la forma:

$$\rho(x) = \int_{-\infty}^{\infty} d\kappa s(\kappa) e^{i2\pi\kappa x}$$



## Apéndice C

# Regresión lineal

En el ámbito de las redes neuronales, la regresión lineal es un algoritmo que trata de encontrar una ecuación que relacione variables dependientes e independientes, en un sentido matemático, es de interés conocer sobre una variable a la cual se le llamará "y" con respecto a lo que se conoce de un conjunto de datos "x". Se tiene una variable "x" que "explica" lo que le ocurre a una variable "responsable" "y", en el contexto de la regresión, a la variable "x" también se le conoce como "variable de predicción" porque con base en la información de la "x" se podría "predecir" o estimar alguna información de la variable "y", pero esto solo es una estimación.

Para el primer caso de 2 dimensiones, se considera el uso de datos pareados, es decir, a cada  $x_n$  le corresponderá un  $y_n$ , para poder estimar el valor del conjunto y a partir del conjunto x, es necesario utilizar el metodo de minimos cuadrados, este método parte de los textos *Principia* de Newton que postula el movimiento de los planetas al rededor del Sol como órbitas elípticas siendo el Sol un foco del elipse. Recurriendo a geometría cartesiana, se puede representar la elipse mediante una ecuación algebrarica de segundo grado que, aunque es cuadrática, sus coeficientes son lineales. El número de coeficientes depende del sistema de coordenadas que se escoja. En términos astronómicos, entre más observaciones se tengan, mucho mejor. En principio, tener muchas observaciones permitiría que determinar la órbita de los planetas sea muy sencillo, pero, se tiene que tomar en consideración también que las observaciones tienen errores de medición de tipo experimental, popr lo que es necesario y esencial tanto aprovechar la información de las observaciones "n" como de los errores que se tienen para realizar la mejor estimación posible a partir de un número de parámetros.

Partiendo de la ecuación de la elipse:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = 0$$

Donde los coeficientes  $a_j$  tienen que ser encontrados o estimados para el ajuste, y los valores  $x_j$  son los necesarios para la ecuación de la elipse, usualmente se usa el valor de 1 a menos que se tengan degeneraciones en la elipse.

Del lado izquierdo de la igualdad será 0 si se tiene un ajuste "perfecto", pero como se menciona anteriormente, existen errores de medicion denotados como  $\epsilon_i$  que corresponden a cada observación, como se requiere que los errores sean pequeños y al mismo tiempo se tomen en consideración los errores positivos y negativos, se utiliza el cuadrado de estos, es decir,  $\epsilon^2$ . Al tener un conjunto de errores al cuadrado se pueden acumular para tener un error general. La idea general del metodo de minimos cuadrados es elegir los coeficientes  $a_j$  de tal forma que los errores sean mínimos, es decir:

$$SS = \sum_{i=1}^n \epsilon_i^2$$

El caso lineal, entonces parte de la suposición en la que se considera que el ajuste se obtiene

mediante una recta que es decrita por la ecuación:

$$y = a + bx$$

Dado un conjunto de datos pareados, se escogen "a" y "b" de tal forma que se haga minima la suma de los errores al cuadrado, es decir:

$$SS = \sum_{i=1}^n (y_i - a - bx_i)^2$$

Tomando las derivadas parciales para "a" y "b", se obtiene:

$$\frac{\partial SS}{\partial a} = -2 \sum_{i=1}^n (y_i - a - bx_i)$$

$$\frac{\partial SS}{\partial b} = -2 \sum_{i=1}^n x_i (y_i - a - bx_i)$$

Para encontrar el mínimo, se igualan ambas ecuaciones con 0, es decir: obtiene:

$$-2 \sum_{i=1}^n (y_i - a - bx_i) = 0 \wedge -2 \sum_{i=1}^n x_i (y_i - a - bx_i) = 0$$

Se obtienen dos ecuaciones lineales simultaneas donde se desconocen los valores de "a" y "b" por lo que, en escencia, se puede resolver el sistema de ecuaciones, mediante una notación que simplifica la suma de los  $x_i$  y dividiendo ambas ecuaciones por "n", se obtienen unas ecuaciones conocidas como *ecuaciones normales*, las cuales, se describen como:

$$a + b\bar{x} = \bar{y} \wedge a\bar{x}$$

Multiplicando la primera por  $\bar{x}$  y restando se tienen las soluciones:

$$b = \frac{\bar{x}\bar{y} - \bar{x} * \bar{y}}{\bar{x}^2 - (\bar{x})^2} \wedge a = \bar{y} - b\bar{x}$$

Se tiene una ecuación llamada *linea de regresión muestral* que toma en consideración toda la información de las muestras y se tratan de ajustar los coeficientes "a" y "b" para reducir el error, se tiene entonces la ecuación:

$$y - \bar{y} = b(x - \bar{x})$$

## C.1. Regresión múltiple

Una vez establecida la regresión lineal o regresión lineal simple, se debe considerar el caso general con diversos regresores, pues, el uso de la regresión y ajuste a una linea en el contexto de las RNC implica que se tiene solo un vector de muestras con una simple característica, por ende, la consideración de varias características viene dada por el uso de arreglos matriciales, para los cuales, también existe la regresión pero con múltiples características.

Se considera un vector  $\mathbf{X}$  cuya media aritmética serpa  $E\mathbf{X}$ , por lo que, la media de la  $i$ -ésima coordenada  $X_i$  será  $E(X_i) = (E\mathbf{X})_i$  y la varianza  $\text{var}(\mathbf{X})$  para una matriz de covarianza. Además, se

usan "p" regresores de la forma  $x_1, x_2, \dots, x_p$  cada uno correspondiente a un parámetro  $\beta_1, \beta_2, \dots, \beta_p$ . Con un conjunto de muestras de tamaño "n", se dan las ecuaciones de la forma:

$$y_i = \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon_i \text{ con } i = 1, \dots, n$$

El término de la derecha de la ecuación anterior se coloca como un producto matricial, haciendo que el sistema de ecuaciones se vuelva una ecuación matricial de la forma:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon$$

Para el establecimiento de las ecuaciones normales para esta regresión se sustituye unicamente por convención la  $\mathbf{X}$  por  $\mathbf{A}$ , además, se quitan las letras rellenas entendiendo que todo el tratamiento es con matrices. Partiendo de la ecuación:

$$y_i = \sum_{j=1}^p a_{ij}\beta_j + \epsilon_i$$

Se calcula la probabilidad para una distribución normalizada de la forma:

$$L = \frac{1}{\sigma^n (2\pi)^{\frac{1}{2}n}} \prod_{i=1}^n \exp \left\{ -\frac{1}{2} (y_i - \sum_{j=1}^p a_{ij}\beta_j)^2 / \sigma^2 \right\} = \frac{1}{\sigma^n (2\pi)^{\frac{1}{2}n}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^n (y_i - \sum_{j=1}^p a_{ij}\beta_j)^2 / \sigma^2 \right\}$$

Se obtiene la probabilidad en forma logaritmica:

$$L = \log(L) = c - n \log(\sigma) - \frac{1}{2} \left[ \sum_{i=1}^n (y_i - \sum_{j=1}^p a_{ij}\beta_j)^2 \right] / \sigma^2$$

Tomando la derivada con respecto de los parámetros  $\beta$  e igualando a 0, se tiene:

$$\sum_{j=1}^p \left( \sum_{i=1}^n a_{ir} a_{ij} \right) \beta_j = \sum_{i=1}^n a_{ir} y_i$$

Introduciendo la matriz tamaño  $p \times p$   $C = (c_{ij})$  de modo que  $C = A^T A$  llamada *matriz de información* la cual es simétrica, se tiene:

$$c_{ij} = \sum_{k=1}^n (A^T)_{ij} A_{kj} = \sum_{i=1}^n a_{ki} a_{kj} \longrightarrow \sum_{j=1}^p c_{rj} \beta_j = \sum_{i=1}^n a_{ir} y_i = \sum_{i=1}^n (A^T)_{ri} y_i$$

En notación matricial, se tiene:

$$(C\beta)_r = (A^T y)_r \text{ con } (r = 1, 2, \dots, p)$$

$$C\beta = A^T y, \quad C = A^T A$$

Las ecuaciones anteriores son las *Ecuaciones normales*, análogas a el caso de la regresión lineal, solo que en este caso se tienen múltiples regresores.



## Apéndice D

### Código de RNC

```
[ ]: import numpy as np
import nibabel as nib
import multiprocessing
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv3D, AveragePooling3D, Dropout, Flatten, Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import Sequence
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split
from multiprocessing import Pool
import gc # Módulo de recolección de basura

class MRIRegressor(Sequential):
    def __init__(self, input_shape_mri):
        super().__init__()
        self.input_shape_mri = input_shape_mri
        self.build_model()

    def build_model(self):
        self.add(Conv3D(16, kernel_size=(3, 3, 3), activation='relu',
            input_shape=self.input_shape_mri))
        self.add(AveragePooling3D(pool_size=(2, 2, 2)))
        self.add(Conv3D(32, kernel_size=(3, 3, 3), activation='relu'))
        self.add(AveragePooling3D(pool_size=(2, 2, 2)))
        self.add(Conv3D(64, kernel_size=(3, 3, 3), activation='relu'))
        self.add(AveragePooling3D(pool_size=(2, 2, 2)))
        self.add(Dropout(0.25))
        self.add(Flatten())
        self.add(Dense(25, activation='relu'))
        self.add(Dropout(0.5))
        self.add(Dense(1, activation='linear'))
        self.compile(loss='mean_squared_error', optimizer=Adam())

class DataGenerator(Sequence):
```

```

def __init__(self, image_paths, labels, batch_size):
    self.image_paths = image_paths
    self.labels = labels
    self.batch_size = batch_size

def __len__(self):
    return int(np.ceil(len(self.image_paths) / self.batch_size))

def load_image(self, image_path):
    return nib.load(image_path).get_fdata()

def __getitem__(self, index):
    batch_image_paths = self.image_paths[index * self.batch_size:(index + 1) *
    self.batch_size]
    batch_labels = self.labels[index * self.batch_size:(index + 1) * self.
    batch_size]

    # Cargar imágenes en paralelo
    with Pool(processes=multiprocessing.cpu_count()) as pool:
        batch_images = pool.map(self.load_image, batch_image_paths)

    batch_images = np.array(batch_images)
    batch_labels = np.array(batch_labels)

    # Liberar memoria de las imágenes cargadas
    del batch_image_paths
    gc.collect()

    # Aquí puedes realizar cualquier preprocesamiento adicional que necesites

    return batch_images, batch_labels

# Cargar las rutas de las imágenes y las etiquetas
image_paths = []
labels = []

for i in range(62):
    img_path = f"/content/image_{i+1}.nii.gz"
    img_data = nib.load(img_path).get_fdata()
    image_paths.append(img_path)
    #labels.append(img_label)

image_paths = np.array(image_paths)
labels = np.array([1.3, 1.2, 1.2, 1.2, 1.5, 1.2, 1.2, 1.2, 1.1, 1.1, 1.1, 1.2, 1.
-2, 1.1, 1.2, 1.2, 1.3, 1.2,
                    1.2, 1.1, 1.3, 1.1, 1.1, 1.1, 1.1, 1.1, 1.2, 1.2, 1.3, 1.2, 1.
-1, 1.1, 1.2, 1.2,

```

```
1.2, 0.9, 1.2, 1.2, 0.9, 1.2, 0.9, 1.2, 1.2, 1.2, 0.9, 1.1, 0.
→8, 0.8, 0.9, 1.1, 1.2, 0.8,
0.9, 1.2, 1.2, 1.2, 0.9, 0.9, 1.3, 0.8, 0.8, 0.9])

# Número de epochs para el entrenamiento
num_epochs = 5

# Número de folds para validación cruzada
num_folds = 2
kf = KFold(n_splits=num_folds, shuffle=True, random_state=42)

# Crear el generador de datos
batch_size = 1

# Realizar la validación cruzada
for fold, (train_index, test_index) in enumerate(kf.split(image_paths)):
    print(f"Fold {fold+1}/{num_folds}")

    # Dividir los datos en conjunto de entrenamiento y conjunto de prueba
    train_image_paths, temp_image_paths = image_paths[train_index],
→image_paths[test_index]
    train_labels, temp_labels = labels[train_index], labels[test_index]

    # Dividir el conjunto de prueba en conjunto de validación y conjunto de
→prueba
    val_image_paths, test_image_paths, val_labels, test_labels =
→train_test_split(
    temp_image_paths, temp_labels, test_size=0.5, random_state=42)

    # Crear el generador de datos de entrenamiento, validación y prueba
    train_generator = DataGenerator(train_image_paths, train_labels, batch_size)
    val_generator = DataGenerator(val_image_paths, val_labels, batch_size)
    test_generator = DataGenerator(test_image_paths, test_labels, batch_size)

    # Crear el modelo
    estimator = MRIRegressor(input_shape_mri=(320, 280, 80, 1))

    # Entrenar el modelo
    history = estimator.fit(train_generator, epochs=num_epochs)

    # Obtener las predicciones del conjunto de entrenamiento
    train_predictions = estimator.predict(train_generator)

    # Imprimir las predicciones del conjunto de entrenamiento junto con el
→número de imagen correspondiente
    for i, prediction in enumerate(train_predictions):
```

```

    print(f"Fold {fold+1} - Image {train_index[i]} - Train Prediction:␣
→{prediction[0]}")

    # Obtener las predicciones del conjunto de validación
    val_predictions = estimator.predict(val_generator)

    # Imprimir las predicciones del conjunto de validación junto con el número␣
→de imagen correspondiente
    for i, prediction in enumerate(val_predictions):
        print(f"Fold {fold+1} - Image {val_image_paths[i]} - Validation␣
→Prediction: {prediction[0]}")

    # Evaluar en el conjunto de prueba
    test_loss = estimator.evaluate(test_generator)
    print(f"Fold {fold+1} - Test Loss: {test_loss}")

    # Obtener las predicciones del conjunto de prueba
    test_predictions = estimator.predict(test_generator)

    # Imprimir las predicciones del conjunto de prueba junto con el número de␣
→imagen correspondiente
    for i, prediction in enumerate(test_predictions):
        print(f"Fold {fold+1} - Image {test_index[i]} - Test Prediction:␣
→{prediction[0]}")

    # Liberar memoria de las predicciones y del modelo
    del history, estimator
    gc.collect()

# ... Resto del código para graficar los costos ...

```

```

Fold 1/10
Epoch 1/5
55/55 [=====] - 833s 15s/step - loss: 10.4599
Epoch 2/5
55/55 [=====] - 834s 15s/step - loss: 0.5795
Epoch 3/5
55/55 [=====] - 832s 15s/step - loss: 0.4162
Epoch 4/5
55/55 [=====] - 834s 15s/step - loss: 0.7318
Epoch 5/5
55/55 [=====] - 846s 15s/step - loss: 0.4747
55/55 [=====] - 352s 6s/step
Fold 1 - Image 1 - Train Prediction: 1.100344181060791
Fold 1 - Image 2 - Train Prediction: 1.1118555068969727
Fold 1 - Image 3 - Train Prediction: 0.941167950630188
Fold 1 - Image 4 - Train Prediction: 0.9214986562728882

```

---

Fold 1 - Image 6 - Train Prediction: 1.13202702999115  
Fold 1 - Image 7 - Train Prediction: 1.0068929195404053  
Fold 1 - Image 8 - Train Prediction: 1.0181164741516113  
Fold 1 - Image 9 - Train Prediction: 1.1439670324325562  
Fold 1 - Image 10 - Train Prediction: 1.1060980558395386  
Fold 1 - Image 11 - Train Prediction: 1.2069565057754517  
Fold 1 - Image 12 - Train Prediction: 0.8990291953086853  
Fold 1 - Image 13 - Train Prediction: 0.9911298751831055  
Fold 1 - Image 14 - Train Prediction: 1.0689995288848877  
Fold 1 - Image 15 - Train Prediction: 1.187883734703064  
Fold 1 - Image 17 - Train Prediction: 0.9306713342666626  
Fold 1 - Image 18 - Train Prediction: 1.0634901523590088  
Fold 1 - Image 19 - Train Prediction: 0.9834346175193787  
Fold 1 - Image 20 - Train Prediction: 0.8329680562019348  
Fold 1 - Image 21 - Train Prediction: 0.9485089182853699  
Fold 1 - Image 22 - Train Prediction: 1.0860635042190552  
Fold 1 - Image 23 - Train Prediction: 1.0621287822723389  
Fold 1 - Image 24 - Train Prediction: 1.174258828163147  
Fold 1 - Image 25 - Train Prediction: 0.9965569376945496  
Fold 1 - Image 26 - Train Prediction: 1.0962668657302856  
Fold 1 - Image 27 - Train Prediction: 1.134458065032959  
Fold 1 - Image 28 - Train Prediction: 1.1658694744110107  
Fold 1 - Image 29 - Train Prediction: 1.052941918373108  
Fold 1 - Image 30 - Train Prediction: 1.0182439088821411  
Fold 1 - Image 31 - Train Prediction: 1.039124608039856  
Fold 1 - Image 32 - Train Prediction: 1.1195639371871948  
Fold 1 - Image 33 - Train Prediction: 1.0937126874923706  
Fold 1 - Image 34 - Train Prediction: 0.9861059188842773  
Fold 1 - Image 35 - Train Prediction: 1.1477932929992676  
Fold 1 - Image 36 - Train Prediction: 1.111242651939392  
Fold 1 - Image 37 - Train Prediction: 1.0184999704360962  
Fold 1 - Image 38 - Train Prediction: 1.0719623565673828  
Fold 1 - Image 39 - Train Prediction: 1.1009745597839355  
Fold 1 - Image 40 - Train Prediction: 0.996763288974762  
Fold 1 - Image 41 - Train Prediction: 1.0446184873580933  
Fold 1 - Image 42 - Train Prediction: 1.097940444946289  
Fold 1 - Image 43 - Train Prediction: 0.9699492454528809  
Fold 1 - Image 44 - Train Prediction: 1.0311660766601562  
Fold 1 - Image 45 - Train Prediction: 0.940589427947998  
Fold 1 - Image 46 - Train Prediction: 0.9397971034049988  
Fold 1 - Image 47 - Train Prediction: 1.0681700706481934  
Fold 1 - Image 49 - Train Prediction: 1.1025865077972412  
Fold 1 - Image 51 - Train Prediction: 0.9525517225265503  
Fold 1 - Image 52 - Train Prediction: 1.078430414199829  
Fold 1 - Image 53 - Train Prediction: 1.0116314888000498  
Fold 1 - Image 54 - Train Prediction: 1.0792750120162964  
Fold 1 - Image 55 - Train Prediction: 1.2147514820098877  
Fold 1 - Image 58 - Train Prediction: 1.1640151739120483

---

```
Fold 1 - Image 59 - Train Prediction: 1.084426999092102
Fold 1 - Image 60 - Train Prediction: 0.9879496097564697
Fold 1 - Image 61 - Train Prediction: 1.0993574857711792
3/3 [=====] - 19s 4s/step
Fold 1 - Image /content/image_51.nii.gz - Validation Prediction:
1.1092497110366821
Fold 1 - Image /content/image_49.nii.gz - Validation Prediction:
1.011376142501831
Fold 1 - Image /content/image_58.nii.gz - Validation Prediction:
1.0630266666412354
4/4 [=====] - 26s 6s/step - loss: 0.1606
Fold 1 - Test Loss: 0.16060671210289001
4/4 [=====] - 33s 8s/step
Fold 1 - Image 0 - Test Prediction: 0.5691360831260681
Fold 1 - Image 5 - Test Prediction: 1.1208280324935913
Fold 1 - Image 16 - Test Prediction: 1.1244784593582153
Fold 1 - Image 48 - Test Prediction: 1.072830319404602
Fold 2/10
Epoch 1/5
55/55 [=====] - 839s 15s/step - loss: 9.5024
Epoch 2/5
55/55 [=====] - 838s 15s/step - loss: 0.5557
Epoch 3/5
55/55 [=====] - 844s 15s/step - loss: 0.6212
Epoch 4/5
55/55 [=====] - 834s 15s/step - loss: 0.5052
Epoch 5/5
55/55 [=====] - 849s 15s/step - loss: 0.6258
55/55 [=====] - 353s 6s/step
Fold 2 - Image 0 - Train Prediction: 0.5417196750640869
Fold 2 - Image 1 - Train Prediction: 0.6671227216720581
Fold 2 - Image 2 - Train Prediction: 0.6242839097976685
Fold 2 - Image 3 - Train Prediction: 0.7333148717880249
Fold 2 - Image 4 - Train Prediction: 0.6817893385887146
Fold 2 - Image 5 - Train Prediction: 0.8179256916046143
Fold 2 - Image 6 - Train Prediction: 0.7271395325660706
Fold 2 - Image 7 - Train Prediction: 0.6583150625228882
Fold 2 - Image 8 - Train Prediction: 0.6675533652305603
Fold 2 - Image 10 - Train Prediction: 0.8329864740371704
Fold 2 - Image 11 - Train Prediction: 0.8432228565216064
Fold 2 - Image 14 - Train Prediction: 0.7585396766662598
Fold 2 - Image 15 - Train Prediction: 0.8204461932182312
Fold 2 - Image 16 - Train Prediction: 0.610771656036377
Fold 2 - Image 17 - Train Prediction: 0.7609144449234009
Fold 2 - Image 18 - Train Prediction: 0.707026481628418
Fold 2 - Image 19 - Train Prediction: 0.6276566982269287
Fold 2 - Image 20 - Train Prediction: 0.6651602983474731
Fold 2 - Image 21 - Train Prediction: 0.6359614133834839
```

---

---

```
Fold 2 - Image 22 - Train Prediction: 0.790976881980896
Fold 2 - Image 23 - Train Prediction: 0.6642508506774902
Fold 2 - Image 24 - Train Prediction: 0.7512412071228027
Fold 2 - Image 26 - Train Prediction: 0.7515703439712524
Fold 2 - Image 27 - Train Prediction: 0.7182791829109192
Fold 2 - Image 28 - Train Prediction: 0.6830971837043762
Fold 2 - Image 29 - Train Prediction: 0.7027146816253662
Fold 2 - Image 31 - Train Prediction: 0.7514444589614868
Fold 2 - Image 32 - Train Prediction: 0.724604606628418
Fold 2 - Image 33 - Train Prediction: 0.7794671058654785
Fold 2 - Image 34 - Train Prediction: 0.6106209754943848
Fold 2 - Image 35 - Train Prediction: 0.6978763341903687
Fold 2 - Image 36 - Train Prediction: 0.7903832197189331
Fold 2 - Image 37 - Train Prediction: 0.6501695513725281
Fold 2 - Image 38 - Train Prediction: 0.6849154233932495
Fold 2 - Image 39 - Train Prediction: 0.7524739503860474
Fold 2 - Image 40 - Train Prediction: 0.7327619791030884
Fold 2 - Image 41 - Train Prediction: 0.6908444166183472
Fold 2 - Image 42 - Train Prediction: 0.6656616926193237
Fold 2 - Image 43 - Train Prediction: 0.7486035823822021
Fold 2 - Image 44 - Train Prediction: 0.7218056917190552
Fold 2 - Image 45 - Train Prediction: 0.7500560879707336
Fold 2 - Image 46 - Train Prediction: 0.7237718105316162
Fold 2 - Image 48 - Train Prediction: 0.6280878782272339
Fold 2 - Image 49 - Train Prediction: 0.6891083121299744
Fold 2 - Image 50 - Train Prediction: 0.7414628267288208
Fold 2 - Image 51 - Train Prediction: 0.6314345002174377
Fold 2 - Image 52 - Train Prediction: 0.6928001642227173
Fold 2 - Image 53 - Train Prediction: 0.702358603477478
Fold 2 - Image 54 - Train Prediction: 0.7817494869232178
Fold 2 - Image 55 - Train Prediction: 0.7692937850952148
Fold 2 - Image 56 - Train Prediction: 0.7437039613723755
Fold 2 - Image 57 - Train Prediction: 0.7701848149299622
Fold 2 - Image 58 - Train Prediction: 0.803380012512207
Fold 2 - Image 60 - Train Prediction: 0.746911346912384
Fold 2 - Image 61 - Train Prediction: 0.6737882494926453
3/3 [=====] - 20s 5s/step
Fold 2 - Image /content/image_31.nii.gz - Validation Prediction:
0.7529263496398926
Fold 2 - Image /content/image_26.nii.gz - Validation Prediction:
0.6964754462242126
Fold 2 - Image /content/image_60.nii.gz - Validation Prediction:
0.6265802979469299
4/4 [=====] - 27s 5s/step - loss: 0.1223
Fold 2 - Test Loss: 0.12225031852722168
4/4 [=====] - 27s 5s/step
Fold 2 - Image 9 - Test Prediction: 0.7865618467330933
Fold 2 - Image 12 - Test Prediction: 0.7441160678863525
```

---

---

Fold 2 - Image 13 - Test Prediction: 0.7978748679161072  
Fold 2 - Image 25 - Test Prediction: 0.6723049879074097  
Fold 3/10  
Epoch 1/5  
56/56 [=====] - 849s 15s/step - loss: 8.6986  
Epoch 2/5  
56/56 [=====] - 859s 15s/step - loss: 0.3808  
Epoch 3/5  
56/56 [=====] - 867s 15s/step - loss: 0.4567  
Epoch 4/5  
56/56 [=====] - 861s 15s/step - loss: 0.2978  
Epoch 5/5  
56/56 [=====] - 848s 15s/step - loss: 0.2418  
56/56 [=====] - 364s 6s/step  
Fold 3 - Image 0 - Train Prediction: 0.9424448013305664  
Fold 3 - Image 1 - Train Prediction: 1.042947769165039  
Fold 3 - Image 2 - Train Prediction: 0.9705491065979004  
Fold 3 - Image 4 - Train Prediction: 1.2667996883392334  
Fold 3 - Image 5 - Train Prediction: 0.9691344499588013  
Fold 3 - Image 6 - Train Prediction: 0.9447005987167358  
Fold 3 - Image 7 - Train Prediction: 0.9456454515457153  
Fold 3 - Image 9 - Train Prediction: 1.0092408657073975  
Fold 3 - Image 10 - Train Prediction: 0.9864752292633057  
Fold 3 - Image 11 - Train Prediction: 1.0331776142120361  
Fold 3 - Image 12 - Train Prediction: 0.7823343276977539  
Fold 3 - Image 13 - Train Prediction: 0.9248290061950684  
Fold 3 - Image 14 - Train Prediction: 0.8943363428115845  
Fold 3 - Image 15 - Train Prediction: 1.0454977750778198  
Fold 3 - Image 16 - Train Prediction: 0.908698558807373  
Fold 3 - Image 18 - Train Prediction: 0.9686357975006104  
Fold 3 - Image 19 - Train Prediction: 0.794092059135437  
Fold 3 - Image 20 - Train Prediction: 0.8278387784957886  
Fold 3 - Image 21 - Train Prediction: 0.8489910364151001  
Fold 3 - Image 22 - Train Prediction: 0.9346439838409424  
Fold 3 - Image 23 - Train Prediction: 0.8966317176818848  
Fold 3 - Image 24 - Train Prediction: 1.0892794132232666  
Fold 3 - Image 25 - Train Prediction: 0.5970866680145264  
Fold 3 - Image 26 - Train Prediction: 0.9329863786697388  
Fold 3 - Image 27 - Train Prediction: 0.8837407827377319  
Fold 3 - Image 28 - Train Prediction: 1.0553985834121704  
Fold 3 - Image 29 - Train Prediction: 1.0039349794387817  
Fold 3 - Image 30 - Train Prediction: 0.7879006862640381  
Fold 3 - Image 31 - Train Prediction: 0.9363067150115967  
Fold 3 - Image 32 - Train Prediction: 0.9046393632888794  
Fold 3 - Image 35 - Train Prediction: 0.9496966600418091  
Fold 3 - Image 36 - Train Prediction: 1.0558571815490723  
Fold 3 - Image 37 - Train Prediction: 1.1177386045455933  
Fold 3 - Image 38 - Train Prediction: 0.8876776695251465

---

---

```
Fold 3 - Image 39 - Train Prediction: 0.9829143285751343
Fold 3 - Image 40 - Train Prediction: 0.924953818321228
Fold 3 - Image 41 - Train Prediction: 0.9430292844772339
Fold 3 - Image 42 - Train Prediction: 0.9153579473495483
Fold 3 - Image 43 - Train Prediction: 0.9853160381317139
Fold 3 - Image 44 - Train Prediction: 1.0324020385742188
Fold 3 - Image 45 - Train Prediction: 0.863500714302063
Fold 3 - Image 46 - Train Prediction: 0.8212292194366455
Fold 3 - Image 47 - Train Prediction: 0.9187288284301758
Fold 3 - Image 48 - Train Prediction: 0.8347920179367065
Fold 3 - Image 49 - Train Prediction: 0.9946029186248779
Fold 3 - Image 50 - Train Prediction: 1.0054471492767334
Fold 3 - Image 51 - Train Prediction: 0.7153335809707642
Fold 3 - Image 52 - Train Prediction: 1.0823097229003906
Fold 3 - Image 53 - Train Prediction: 1.0298293828964233
Fold 3 - Image 55 - Train Prediction: 1.0992287397384644
Fold 3 - Image 56 - Train Prediction: 0.9104949235916138
Fold 3 - Image 57 - Train Prediction: 0.9453094005584717
Fold 3 - Image 58 - Train Prediction: 0.9325782060623169
Fold 3 - Image 59 - Train Prediction: 0.9553601741790771
Fold 3 - Image 60 - Train Prediction: 1.0142583847045898
Fold 3 - Image 61 - Train Prediction: 1.029368281364441
3/3 [=====] - 21s 5s/step
Fold 3 - Image /content/image_18.nii.gz - Validation Prediction:
0.8261871337890625
Fold 3 - Image /content/image_35.nii.gz - Validation Prediction:
0.8155635595321655
Fold 3 - Image /content/image_34.nii.gz - Validation Prediction:
0.9103389978408813
3/3 [=====] - 21s 5s/step - loss: 0.0678
Fold 3 - Test Loss: 0.06780143827199936
3/3 [=====] - 19s 5s/step
Fold 3 - Image 3 - Test Prediction: 0.8778729438781738
Fold 3 - Image 8 - Test Prediction: 0.983242392539978
Fold 3 - Image 17 - Test Prediction: 0.9067320823669434
Fold 4/10
Epoch 1/5
39/56 [=====>...] - ETA: 4:17 - loss: 5.4605
```

---



# Bibliografía

- [1] Smith, N. B., Webb, A. (2010). *Introduction to medical imaging*. Cambridge University Press.
- [2] Nishimura, D. G. (2013). *Principles of magnetic resonance imaging*.
- [3] 09-03 Capítulo Nueve: Fundamentos de las Características de la Imagen: La imagen de RM. (n.d.). Resonancia-magnetica.org. <http://www.resonancia-magnetica.org/ch/09-03.html>
- [4] Frank, L., Novelline. (1997). *Squire's fundamentals of radiology*.
- [5] McRobbie, D. W., Moore, E. A., Graves, M. J., Prince, M. R. (2017). *MRI from picture to proton*. Cambridge University Press.
- [6] Brown, R. W., C. Norman Cheng, Y., Haacke, E. M., Thompson, M. R., Venkatesan, R. (2014). *Magnetic resonance imaging: Physical principles and sequence design* (R. W. Brown, Y.-C. N. Cheng, E. M. Haacke, M. R. Thompson, R. Venkatesan, Eds.; 2nd ed.). Wiley-Blackwell.
- [7] Ganong, W. F. (2016). *Fisiología médica - 25a: Edición*. McGrawHill.
- [8] GOLFEYZ S, LEWIS S, WEISBERG IS (2018). *Hemochromatosis: pathophysiology, evaluation and management of hepatic iron overload with a focus on MRI*. *Expert Rev Gastroenterol Hepatol* 12:767-778.
- [9] *Pruebas de hierro*. (n.d.). Medlineplus.gov. <https://medlineplus.gov/spanish/pruebas-de-laboratorio/pruebas-de-hierro/>
- [10] *Hemocromatosis*. (2023, March 21). MayoClinic.org. <https://www.mayoclinic.org/es/diseases-conditions/hemochromatosis/diagnosis-treatment/drc-20351448>
- [11] Department of health and human services. National Institutes of health. *Manual Biopsia de Hígado*. (Julio, 2010) NIH Publication No. 10-47315.
- [12] Wood, J. C., Enriquez, C., Ghugre, N., Tyzka, J. M., Carson, S., Nelson, M. D., Coates, T. D. (2005). *MRI R2 and R2\* mapping accurately estimates hepatic iron concentration in transfusion-dependent thalassemia and sickle cell disease patients*. *Blood*, 106(4), 1460–1465. <https://doi.org/10.1182/blood-2004-10-3982>
- [13] Stark, D. D., Bass, N. M., Moss, A. A., Bacon, B. R., McKerrow, J. H., Cann, C. E., Brito, A., Goldberg, H. I. (1983). *Nuclear magnetic resonance imaging of experimentally induced liver disease*. *Radiology*, 148(3), 743–751. <https://doi.org/10.1148/radiology.148.3.6192464>
- [14] St. Pierre, T. G., Clark, P. R., Chua-anusorn, W., Fleming, A. J., Jeffrey, G. P., Olynyk, J. K., Potrakul, P., Robins, E., Lindeman, R. (2005). *Noninvasive measurement and imaging of liver iron concentrations using proton magnetic resonance*. *Blood*, 105(2), 855–861. <https://doi.org/10.1182/blood-2004-01-0177>

- [15] Hankins, J. S., McCarville, M. B., Loeffler, R. B., Smeltzer, M. P., Onciu, M., Hoffer, F. A., Li, C.-S., Wang, W. C., Ware, R. E., Hillenbrand, C. M. (2009). *R2\* magnetic resonance imaging of the liver in patients with iron overload*. *Blood*, 113(20), 4853–4855. <https://doi.org/10.1182/blood-2008-12-191643>
- [16] Alústiza Echeverría, J. M., Barrera Portillo, M. C., Guisasola Iñiguez, A., Ugarte Muñoz, A. (2017). *Diagnóstico y cuantificación de la sobrecarga férrica mediante resonancia magnética*. *Radiología*, 59(6), 487–495. <https://doi.org/10.1016/j.rx.2017.07.003>
- [17] Gandon, Y., Olivie, D., Guyader, D., Aubé, C., Oberti, F., Sebille, V., Deugnier, Y. (2004). *Non-invasive assessment of hepatic iron stores by MRI*. *Lancet*, 363(9406), 357–362. [https://doi.org/10.1016/s0140-6736\(04\)15436-6](https://doi.org/10.1016/s0140-6736(04)15436-6)
- [18] ST. PIERRE TG, CLARK PR, ET. AL. (2005). *Noninvasive measurement and imaging of liver iron concentrations using proton magnetic resonance*. *Blood*. doi: 10.1182/blood-2004-01-0177
- [19] ZHOU SK, GREENSPAN H, SHEN D. *Deep Learning for Medical Image Analysis*. Academic Press. 2017.
- [20] MANCILLA E. (20 de diciembre de 2022). *Machine learning: definición, métodos y ejemplos*. invigate. <https://blog.invgate.com/es/machine-learning>
- [21] Real Academia Española. (s.f). Heurístico. En *Diccionario de la lengua española*. Recuperado el 30 de marzo de 2024, de <https://dle.rae.es/heur%C3%ADstico>
- [22] Zhou, S. K., Greenspan, H., Shen, D. (Eds.). (2024). *Deep learning for medical image analysis*. (2nd ed.). Academic Press.
- [23] (2019, abril 1). *Juejin.cn*. Recuperado el 30 de marzo de 2024, de <https://juejin.cn/post/6844903810859548680>
- [24] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (s/f). *Dropout: A simple way to prevent neural networks from overfitting*. *Jmlr.org*. Recuperado el 30 de marzo de 2024, de <https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf>
- [25] Lee, T., Gasswint, G., Henning, E. 15 de Diciembre de 2021. *Momentum*. Cornell.edu. Recuperado el 30 de marzo de 2024, de <https://optimization.cbe.cornell.edu/index.php?title=Momentum>
- [26] Kingma, D. P., Ba, J. (2014). *Adam: A method for stochastic optimization*. In arXiv [cs.LG]. <http://arxiv.org/abs/1412.6980>
- [27] Hubel, H., Wiesel, T. N. (1962). *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex*. *The Journal of Physiology*, 160(1), 106–154. <https://doi.org/10.1113/jphysiol.1962.sp006837>
- [28] *Intuitive guide to convolution*. (n.d.). Betterexplained. Recuperado el 30 de marzo de 2024, de <https://betterexplained.com/articles/intuitive-convolution/>
- [29] UJJWALKARN. (2016, August 10). *An intuitive explanation of Convolutional Neural Networks*. The Data Science Blog. <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>
- [30] 05.7 *Redes Neuronales Convoluciones — Introducción al Aprendizaje Automático*. (n.d.). Upm.Es. [https://dcain.etsin.upm.es/carlos/bookAA/05.7\\_RNN\\_Convoluciones\\_CIFAR10\\_INFORMATIVO.html](https://dcain.etsin.upm.es/carlos/bookAA/05.7_RNN_Convoluciones_CIFAR10_INFORMATIVO.html)
- [31] Gholamalinezhad, H., Khosravi, H. (2020). *Pooling methods in Deep Neural Networks, a review*. arXiv [cs.CV]. <http://arxiv.org/abs/2009.07485>

- [32] Muñoz, E., Espinoza, J., Rivera, L. (2012). *Un sistema basado en redes neuronales artificiales para diagnóstico de anemia ferropénica*. Revista de investigación de Sistemas e Informática, 9(1), 35–43. <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/5613>
- [33] Beliveau, V., Nørgaard, M., Birkl, C., Seppi, K., Scherfler, C. (2021). *Automated segmentation of deep brain nuclei using convolutional neural networks and susceptibility weighted imaging*. Human Brain Mapping, 42(15), 4809–4822. <https://doi.org/10.1002/hbm.25604>
- [34] Martini, N., Meloni, A., Positano, V., Latta, D. D., Keilberg, P., Pistoia, L., Spasiano, A., Casini, T., Barone, A., Massa, A., Ripoli, A., Cademartiri, F. (2022). *Fully automated regional analysis of myocardial T2\* values for iron quantification using deep learning*. Electronics, 11(17), 2749. <https://doi.org/10.3390/electronics11172749>
- [35] Tustison, N. J., Avants, B. B., Cook, P. A., Zheng, Y., Egan, A., Yushkevich, P. A., Gee, J. C. (2010). N4ITK: improved N3 bias correction. IEEE Transactions on Medical Imaging, 29(6), 1310–1320. <https://doi.org/10.1109/TMI.2010.2046908>
- [36] ANTs: Advanced Normalization Tools (ANTs). (n.d.). <https://github.com/ANTsX/ANTs?tab=readme-ov-file#readme>
- [37] Yushkevich, P., Hao, J., Ravikumar, S., Pouch, A., Aggarwal, A., Kaczmarczyk, L. (2020). ITK-SNAP. [Itksnap.org](http://itksnap.org)
- [38] zscore. (n.d.). Mathworks.com. <https://la.mathworks.com/help/stats/zscore.html>
- [39] Despotović, I., Goossens, B., Philips, W. (2015). MRI Segmentation of the Human Brain: Challenges, Methods, and Applications. Computational and Mathematical Methods in Medicine.
- [40] Haacke, E. M., Brown, R. W., Thompson, M. R., Venkatesan, R. (1999). Magnetic Resonance Imaging: Physical Principles and Sequence Design. John Wiley Sons.
- [41] Krupa K, Bekiesińska-Figatowska M. Artifacts in magnetic resonance imaging. Pol J Radiol. 2015 Feb 23;80:93-106. doi: 10.12659/PJR.892628. PMID: 25745524; PMCID: PMC4340093.
- [42] Blanco, J. I. (2023, April 28). "¿Por qué la normalización es clave e importante en Machine Learning y Ciencia de Datos?" Medium. <https://jorgeiblanco.medium.com/por-qu%C3%A9-la-normalizaci%C3%B3n-es-clave-e-importante-en-machine-learning-y-ciencia-de-datos-4595f15d5be0>
- [43] NumPy -. Numpy.org. recuperado el 10 de marzo de 2023, de <https://numpy.org/>
- [44] Neuroimaging in python — NiBabel 5.3.0.dev5+g773e3c40 documentation. Nipy.org. Recuperado el 10 de marzo de 2023, de <https://nipy.org/nibabel/gettingstarted.html>
- [45] multiprocessing — Paralelismo basado en procesos — documentación de Python - 3.9.18. Python.org. Recuperado el 10 de marzo de 2023, de <https://docs.python.org/es/3.9/library/multiprocessing.html>
- [46] TensorFlow. TensorFlow. Recuperado el 10 de marzo de 2023, de <https://www.tensorflow.org/?hl=es-419>
- [47] Scikit-learn. Scikit-learn.org. Recuperado el 10 de marzo de 2023, de <https://scikit-learn.org/stable/>
- [48] gc — Garbage Collector interface. Python Documentation. Recuperado el 10 de marzo de 2023, de <https://docs.python.org/3/library/gc.html>

- [49] AnestesiaR. (14 de enero de 2015). No todo es normal. Manejo de datos no normales. AnestesiaR. <https://anestesiार.org/2015/no-todo-es-normal-manejo-de-datos-no-normales/>
- [50] Flores, C., Flores, K., PRUEBAS PARA COMPROBAR LA NORMALIDAD DE DATOS EN PROCESOS PRODUCTIVOS: ANDERSON-DARLING, RYAN-JOINER, SHAPIRO-WILK Y KOLMOGÓROV-SMIRNOV. Portal Amelica.Societas. Revista de Ciencias Sociales y Humanísticas. 27 de mayo de 2021. <http://portal.amelica.org/amelijatsRepo/341/3412237018/index.html>
- [51] Prueba de Kolmogorov-Smirnov para una muestra. (4 de Agosto de 2023). Ibm.com. <https://www.ibm.com/docs/es/spss-statistics/saas?topic=tests-one-sample-kolmogorov-smirnov-test>
- [52] H. A. Prueba de Bondad de Ajuste de Kolmogorov-Smirnov (KS). Ulpgec.Es. [https://www2.ulpgc.es/hege/almacen/download/5/5015/Complemento3Prueba\\_deBondad\\_deAjuste\\_deKolmogorovSmirnov.pdf](https://www2.ulpgc.es/hege/almacen/download/5/5015/Complemento3Prueba_deBondad_deAjuste_deKolmogorovSmirnov.pdf)
- [53] Rodrigo, J. A. Prueba de los rangos con signo de Wilcoxon. Cienciadedatos.net. | [https://cienciadedatos.net/documentos/18prueba\\_de\\_los\\_rangos\\_con\\_signo\\_de\\_wilcoxon](https://cienciadedatos.net/documentos/18prueba_de_los_rangos_con_signo_de_wilcoxon)
- [54] COEFICIENTE DE CORRELACIÓN LINEAL DE PEARSON. Personal.Us.Es. <https://personal.us.es/vararey/adatos2/correlacion.pdf>
- [55] Vista de Tamaño de efecto, potencia de la prueba, factor de Bayes y meta-análisis en el marco de la crisis de reproducibilidad de la ciencia. El caso de la diferencia de medias -con muestras independientes- (primera parte). (01 de Septiembre de 2020). Uba.ar. <https://ojs.econ.uba.ar/index.php/CIMBAGE/article/view/2055/2788>