Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Uso de arreglos de redes neuronales convolucionales para diagnóstico de distintas etapas del espectro de Alzheimer

Tesis presentada al

Posgrado en Física Aplicada

como requisito parcial para la obtención del grado de

Maestro en Ciencias

por

Lic. René Eduardo Rodríguez Pérez

Asesorado por

Dr. Benito de Celis Alonso

Dr. Javier Miguel Hernández López

Puebla Pue.
1 de julio de 2024

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Uso de arreglos de redes neuronales convolucionales para diagnóstico de distintas etapas del espectro de Alzheimer

Tesis presentada al

Posgrado en Física Aplicada

como requisito parcial para la obtención del grado de

Maestro en Ciencias

por

Lic. René Eduardo Rodríguez Pérez

Asesorado por

Dr. Benito de Celis Alonso

Dr. Javier Miguel Hernández López

Puebla Pue.
1 de julio de 2024

Título: Uso de arreglos de redes neuronales convolucionales para diagnóstico de distintas etapas del espectro de Alzheimer
Estudiante: Lic. René Eduardo Rodríguez Pérez

COMITÉ

Dr. Moreno Barbosa Eduardo
Presidente

Dr. Arredondo Velázquez Juan Moisés
Secretario

Dr. Velázquez Castro Jorge
Vocal

Dr. Benito de Celis Alonso
Asesor

Dr. Javier Miguel Hernández López
Asesor

A mis padres René y Guadalupe

Agradecimientos

Agradezco a mis asesores el Dr. Benito de Celis y Javier Hernández por su apoyo y orientación durante la maestría, así como al Dr. Moisés Arredondo por impartir la materia más pesada que he tomado hasta ahora pero también por ayudarme con muchas dudas que surgieron durante este proyecto.
También agradezco a mis padres por siempre creer en mí y motivarme a seguir mis metas; por brindarme su infinito amor, compañía y apoyo moral a lo largo de este camino. De igual manera quiero agradecer a Verónica Pérez por su apoyo incondicional en esta etapa, por motivarme a continuar cuando estaba muy cansado y cuidarme cuando yo no tenía tiempo para hacerlo. Finalmente, quiero agradecer a Gerardo Pérez por su amistad y apoyo en momento difíciles.

Gracias a todas y a todos.

Índice general

Resumen

El uso de imágenes médicas para el diagnóstico de enfermedades es una de las prácticas más comunes en la medicina. Sin embargo, existen condiciones médicas que no son visibles ante el ojo experto del personal médico. Es por esto que se ha perseguido el desarrollo de técnicas de análisis que puedan agilizar el proceso de diagnóstico, así como asistir en la identificación temprana de las mismas.

En este proyecto se propone un modelo de red neuronal convolucional para la discriminación de tres categorías: pacientes con la enfermedad de Alzheimer, una etapa temprana de esta enfermedad (daño cognitivo leve), así como sujetos de prueba sanos. La metodología utilizada consiste en el uso de parches de tres vistas obtenidos a partir de las 6 regiones de interés más afectadas en el desarrollo de esta enfermedad (ínsulas, hipocampos y amígdalas). Cada uno de los parches son analizados por redes convolucionales correspondientes a cada región del cerebro, para finalmente determinar en conjunto la categoría a la que pertenece el paciente en cuestión. Para lograr esto, se propone el uso del teorema de Bayes para conseguir una clasificación global con base en el resultado individual de cada parche.

A pesar de que el modelo propuesto obtuvo resultados inferiores a la media de exactitud reportada en el estado del arte, logra mantenerse sobre el valor mínimo registrado en otros artículos usando una arquitectura simple. Además se ha encontrado que el uso conjunto de las regiones de hipocampos e ínsulas proveen una mejor clasificación de las tres categorías, en comparación del uso simultáneo de las seis regiones o de regiones pares; siendo especialmente sensible para detectar pacientes con daño cognitivo leve.

Abstract

The use of medical images for disease diagnosis is one of the most common practices in medicine. However, there are medical conditions that are not visible to the expert eye of medical persona. This is why the development of analysis techniques has been pursued to streamline the diagnosis process, as well as assist in the early identification of these diseases.

In this project, a convolutional neural network model is proposed for the discrimination of three categories: patients with Alzheimer's disease, an early stage of this disease (mild cognitive impairment), as well as healthy test subjects. The methodology used consists of the use of three-view patches obtained from the 6 most affected regions in the development of this disease (insulas, hippocampi and amygdalas). Each of the patches is analyzed by convolutional networks corresponding to each brain region, to finally determine together the category to which the patient in question belongs. To achieve this, the use of Bayes' theorem is proposed to obtain a global classification based on the individual result of each patch.

Although the proposed model obtained results lower than the average accuracy reported in the state of the art, it manages to remain above the minimum value recorded in other articles using a simple architecture. In addition, it has been found that the joint use of the hippocampus and insula regions provides better classification of the three categories, compared to the simultaneous use of the six regions or pair regions; being especially sensitive to detect patients with mild cognitive impairment.

Keywords: Alzheimer, CNN, Diagnosis, IA, hipocampi, insula, amygdala.

Capítulo 1

Introducción

Justificación

A pesar de no tener una gran tasa de incidencia, en comparación a otras enfermedades, la enfermedad de Alzheimer conlleva una serie de complicaciones que dificultan la vida de quien la padece. Esta enfermedad tiene la particularidad de que el deterioro del cerebro de los pacientes inicia hasta un par de décadas antes de que se presente su sintomatología característica, por lo que cuando es diagnosticada, ya se encuentra en una etapa avanzada.
Dadas las características y el reto que implica la detección y el diagnóstico de esta enfermedad, es que cobra importancia el hecho de lograr un diagnóstico temprano y preciso. De esta manera es posible comenzar un acompañamiento médico para manejar y ralentizar la enfermedad en sus primeras fases permitiéndole al paciente tener una mejor calidad de vida.
En la actualidad existen distintas pruebas para el diagnóstico de Alzheimer como las pruebas de estado mental, neuropsicológicas, exámenes de líquido cefalorraquídeo y el uso de imágenes médicas para descartar otro tipo de padecimientos. No obstante existen propuestas de herramientas para el diagnóstico temprano de esta enfermedad con ayuda de modelos de aprendizaje de máquina como las redes neuronales convolucionales, los cuales son principalmente utilizados para el análisis de imágenes estructurales de resonancia magnética del cerebro por medio de distintas estrategias como el uso de regiones de interés. Sin embargo, al investigar, existen pocos trabajos en los que se utilicen imágenes funcionales de resonancia magnética en estado de reposo, las cuales contienen información relevante sobre la actividad espontánea del cerebro sin realizar alguna tarea, así como la correlación entre áreas con una misma funcionalidad independientemente de su ubicación. Por lo tanto surge la pregunta de si es posible obtener mejores resultados al analizar este tipo de imágenes, ya que estas tienen la propiedad de contener información sobre la actividad cerebral espontánea sin algún tipo de estímulo.

Hipótesis

Es posible obtener mejores resultados de clasificación usando imágenes funcionales de resonancia magnética en estado de reposo con redes neuronales convolucionales simples, en comparación con los reportados en el estado del arte usando imágenes estructurales y arquitecturas complejas para la enfermedad de Alzheimer.

Objetivo principal

En este proyecto de tesis se busca el proponer un modelo de red neuronal convolucional que, con base en el análisis de regiones de interés, nos permita identificar pacientes pertenecientes a tres categorías: Saludable, con daño cognitivo leve y con Alzheimer avanzado. Igualando o superando los resultados reportados en el estado del arte.

Objetivos secundarios

Para el desarrollo de dicha propuesta es necesario considerar los siguientes objetivos secundarios:

- Revisión bibliográfica sobre el uso de redes neuronales convolucionales para la estadificación de Alzheimer, estrategias utilizadas y regiones de interés de mayor trascendencia durante el desarrollo de este padecimiento.
- Obtener una base de datos de imágenes funcionales de resonancia magnética en estado de reposo y realizar su preprocesado básico.
- Realizar segmentación de las regiones de interés, dividirlas en subimágenes y etiquetarlas según la clase a la cual pertenece el paciente.
- Programación de una red neuronal convolucional para cada una de las regiones de interés.
- Determinar la estrategia de unificación de los resultados de clasificación de cada una de las redes para obtener la clasificación final del paciente.
- Determinar qué métricas son las óptimas para evaluar el desempeño del arreglo de redes.
- Análisis de resultados.

Resumen de capítulos

En el capítulo 2 se ahondará sobre la relevancia del diagnóstico temprano de la enfermedad de Alzheimer, así como sus implicaciones en la vida diaria de quienes la padecen, además se abordarán los principios físicos que son utilizados para la obtención de imágenes estructurales de resonancia magnética, así como las imágenes funcionales en estado de reposo. Aunado a esto se presentan las bases matemáticas para el funcionamiento de algoritmos de machine learning, específicamente las redes neuronales convolucionales y el uso de estos modelos para el diagnóstico de esta enfermedad y sus etapas.
En el capítulo número 3 se habla de las herramientas, las bases de datos y las estrategias utilizadas en el desarrollo del proyecto. Así como el procesado de la información y la experimentación para determinar la arquitectura con mejor desempeño durante el entrenamiento; el método de combinación de clasificación para cada región de interés y las métricas para evaluar el desempeño de la propuesta.
El capítulo 4 contiene los resultados de todo el proceso mencionado en el capítulo anterior, los resultados de clasificación para distintos arreglos de redes, así como los resultados de evaluación del modelo.
En el capítulo 5 se discuten los resultados obtenidos a lo largo del proceso, indagando sobre las causas del comportamiento obtenido de cada uno de los experimentos realizados. Además de comparar los resultados obtenidos con los reportados en otros trabajos para determinar si se cumplió con el objetivo del proyecto.
Finalmente en el capítulo 6 se hace una breve recapitulación de lo discutido anteriormente y se ahonda en el trabajo a futuro para este proyecto.

Abreviaturas

AD Enfermedad de Alzheimer por sus siglas en inglés *Alzheimer´s Disease*. 13

BOLD Dependiente del nivel de oxigenación, *Blood Oxygen Level Dependent*. 21

BP Propagación hacia atrás, *Back propagation*. 34

CN Sujeto de control sano por las siglas en inglés de *Cognitive Normal*. 13

EMC Daño cognitivo leve temprano ,*Early mild cognitive impairment*. 13

FC Completamente conectada, *Fully connected*. 44

fMRI Imagen funcional de resonancia magnética, *functional magnetic resonance imaging*. 55

LMCI Daño cognitivo leve tardío, *Late Mild Cognitive Impairment*. 13

MCI Daño cognitivo leve, *Mild Cognitive Impairment*. 13

MMSE Examen breve de estado mental, *Mini mental state exam*. 14

MRI Imagen por resonancia magnética, *Magnetic resonance imaging*. 15

RF Radio frecuencia. 17

RNCs Redes neuronales convolucionales. 24

rs-fMRI Imagen funcional de resonancia magnética en estado de reposo, *resting state functional magnetic resonance imaging*. 21

SMC Problemas relevante de memoria, *Significant Memory Concern*. 13

TR Tiempo de relajación. 20

Índice de figuras

Índice de tablas

Capítulo 2

Marco teórico

El planteamiento de este proyecto de tesis comienza con la revisión del estado del arte de la enfermedad de Alzheimer (AD) dentro del marco que estaremos trabajando. En este primer capítulo se presentan las secciones correspondientes a información relevante sobre esta enfermedad, los principios físicos de la obtención de imágenes de resonancia magnética y funcionamiento de las redes neuronales convolucionales

2.1. Enfermedad de Alzheimer

El Alzheimer es un padecimiento neurodegenerativo provocado por el daño a las células cerebrales, este daño se presenta principalmente en las regiones del cerebro relacionadas con funciones de memoria, lenguaje y razonamiento; sin embargo, eventualmente dicho daño cerebral puede extenderse a zonas encargadas de controlar funciones como el caminar e incluso básicas como la deglución. Es debido a esto que esta enfermedad es potencialmente mortal, siendo 8 años el tiempo promedio de vida luego de su diagnóstico para los pacientes mayores de 65 años, aunque algunos llegan a vivir hasta 20 años. A pesar de que no fue hasta los años 70 que se reconoció al Alzheimer como causa de demencia, fue en el año de 1901 que se tiene el primer registro de esta enfermedad por parte del psiquiatra Alois Alzheimer en una paciente de 51 años; luego de su muerte, al revisar su cerebro se encontraron alteraciones en las células, las cuales serían encontradas más tarde en otros pacientes que padecían la misma condición.

La principal sintomatología de este padecimiento consiste en deterioro crónico de memoria, déficits en habilidades de lenguaje y orientación espacio - temporal, alteración de comportamiento y finalmente pérdida de autonomía para realizar tareas diarias. Gracias al estudio de esta enfermedad se ha logrado reconocer dos tipos principales: Familiar y esporádico [1]. El primero representa entre el 1 y el 5 % de los casos de AD y, como lo sugiere su nombre, es referente a la predisposición genética que poseen ciertos individuos a desarrollar dicho padecimiento. Por otro lado, el 95 % de los casos totales corresponden al tipo esporádico, que incluye a los casos de desarrollo tardío por parte de pacientes mayores de 65 años sin antecedentes de familiares con la misma enfermedad. Por lo que, si un individuo desarrolla este padecimiento, es altamente probable que se trate del tipo esporádico. Aunado a esto, se trata de una enfermedad progresiva que puede provocar cambios en el cerebro hasta un par de décadas antes de la aparición de la sintomatología mencionada anteriormente. Por lo cual se presenta en distintas fases clasificadas según su nivel de avance, la cuales comúnmente se denominan como un continuo: CN (Cognitivo Normal), SMC (Problemas relevantes de memoria), EMC (Deterioro cognitivo leve temprano), MCI (Deterioro cognitivo leve), LMCI (Deterioro cognitivo leve tardío) y finalmente AD (Enfermedad de Alzheimer).

2.1.1. Prevalencia de Alzheimer

Esta enfermedad representa uno de los tipos de demencia con mayor prevalencia a nivel mundial. Se estima que alrededor de 50 millones de personas padecen de esta condición y este número aumentará a más del doble para el año 2050 [2]. Tan sólo en 2023 se estima que habían alrededor de 6.7 millones de estadounidenses mayores de 65 años que padecían esta enfermedad, lo cual se traduce en que una de cada nueve personas mayores de 65 años padecen esta enfermedad y estas cifras aumentan con conforme la edad de la población [3]. No obstante, esta enfermedad no es exclusiva de las personas con edades avanzadas, sino que también afecta a personas menores de 65 años, se calcula que alrededor de doscientos mil personas entre 30 a 64 años viven con este padecimiento [4].
En particular para México, en 2021 se tenían alrededor de un millón trescientos mil pacientes con esta enfermedad, afectando de igual manera a mujeres y hombres con una tasa de incidencia del 10 % para personas mayores de 65 años y 47 % para personas mayores de 85 años. Esta tasa de incidencia incrementa un 0.7 % por año para personas mayores de 60 años y un 9 % anual para mayores de 80 años [5].

2.1.2. Fisiología

Después de años de estudios se ha logrado un conocimiento más profundo sobre los mecanismos que suceden durante su desarrollo, como lo son placas neuríticas y ovillos neurofibrilares. Estas formaciones están relacionadas con la acumulación del péptido beta amiloide y la hiperfosforilación de la proteína Tau en los microtúbulos del citoesqueleto de las células cerebrales. Según la teoría amiloide del Alzheimer, la sobreproducción del péptido beta se debe a la interrupción de procesos que regulan la escisión proteolítica de su proteína precursora [6], su acumulación provoca la formación de placas que interrumpen el transporte nucleocitoplasmático entre neuronas provocando así la muerte celular.
Además de los procesos mencionados, están la inflamación y atrofia del cerebro, esto es debido a que la presencia de las proteínas Tau y Beta-Amiloide actúan como un antígeno, activando las células inmunes del cerebro, las cuales se encargan de eliminar estas proteínas tóxicas y los restos de las neuronas muertas. Sin embargo, puede presentarse la situación en la que la tasa de producción de estas proteínas sea tan alta que supere la capacidad de las células inmunes, provocando así la inflamación. Esto a su vez reduce la capacidad del cerebro de metabolizar glucosa, la cual es imperativo para la actividad neuronal [7].

2.1.3. Diagnóstico

Existe una amplia variedad de herramientas para ayudar con el diagnóstico de la enfermedad de Alzheimer, de las cuales la más usada y aceptada es el examen breve de estado mental (MMSE por sus siglas en inglés) ya que, gracias a este test, es posible estimar el grado de daño cognitivo leve de un paciente [8]. El MMSE consiste en un conjunto de preguntas que suman una puntuación máxima de 30 puntos, los cuales están divididos en distintas categorías a evaluar como: Orientación espacial y temporal, atención y cálculo matemático, lenguaje y construcción visual, entre otros. A pesar de permitir un registro del deterioro cognitivo del paciente a lo largo del tiempo, se trata de una herramienta complementaria para el diagnóstico por parte del personal experto, el cual se considera imprescindible.
Además del uso del MMSE para deteminar el diagnóstico de un paciente, es conveniente contar con imágenes médicas que asistan al personal experto. Entre la amplia gama de modalidades en imágenes, existe una que es particularmente útil para la visualización de tejido blando en el cuerpo y que alcanza un nivel alto de detalle: La resonancia magnética.

2.2. Imágenes de resonancia magnética

A pesar de ser unas de las técnicas de imagen más recientes, obteniendo sus primeras imágenes en 1973 y cuyos creadores obtuvieron el premio Nobel de medicina en 2003; el uso de imágenes por resonancia magnética (MRI) ha tenido un fuerte impacto en el área de imagenología e investigación. Esto es debido a que ofrece ventajas, especialmente en la visualización de tejidos blandos en el cuerpo, las principales ventajas son:

- No requiere de radiación ionizante.
- La imágenes obtenidas pueden ser en tres o dos dimensiones.
- Permite obtener imágenes de tejido suave con buen contraste.
- Ofrece la posibilidad de obtener resoluciones espaciales de hasta 1 mm.
- Es una técnica no invasiva por lo que los efectos de penetración son despreciables, a diferencia de técnicas de imagenología como los rayos X y la tomografía computarizada que utilizan radiación ionizante [9].

Una de la cosas que caben destacar sobre esta técnica, es que el nivel de señal espacial obtenida depende de dos factores principales:

- Las propiedades físicas del tejido en cuestión.
- La concentración de núcleos de hidrógeno, esto es debido a que moléculas como el agua y lípidos pueden ser encontradas en todo el cuerpo.

Es gracias a estas ventajas, que es considerada una de las técnicas de imagenología más sofisticadas y completas. Sin embargo, para comprender su funcionamiento y otras técnicas derivadas, es necesario ahondar en los fenómenos físicos que dan origen a estas imágenes.

2.2.1. Origen de la señal de resonancia magnética

Esta técnica aprovecha la alta presencia de protones en el cuerpo para obtener señales cuadrimensionales (espacial y temporal). Partiendo de esto, se hablará sobre las propiedades de esta partícula subatómica. Como cualquier partícula subatómica, el protón posee un momento angular ($\vec{P}$) del cual deriva su momento magnético, una propiedad vectorial que nos indica la intensidad de una fuente de campo magnético. Estas dos propiedades están relacionadas por medio de la constante giromagnética (γ) por la siguiente expresión:

$$|\vec{\mu}| = \gamma|\vec{P}| \tag{2.1}$$

Como resultado de la física cuántica, se sabe que algunos parámetros físicos pueden presentarse únicamente en valores discretos. Uno de estos parámetros es el momento angular del protón, el cual posee un único valor y por lo tanto el momento magnético hereda estas mismas propiedades. En el estado natural del cuerpo humano, los momentos magnéticos de los protones se encuentran ordenados aleatoriamente. Sin embargo, al introducir a un paciente a un campo magnético externo intenso B_0 generalmemte de entre 1.5 a 3 Teslas; para estas intensidades de campo magnético los momentos magnéticos del cuerpo se alinearían con un ángulo $\theta = 54{,}7$ con respecto al campo externo en sentido paralelo o antiparalelo [10]. Siendo el primero, el estado de menor energía o más favorable en comparación con el antiparalelo. La diferencia de energía entre ambos estados está dada por:

$$\Delta E = \frac{\gamma h B_0}{2\pi} \tag{2.2}$$

Esto da como resultado la obtención de una magnetización neta, cuya intensidad depende de la cantidad de momentos magnéticos individuales alineados en cada sentido. Al someterse a un campo magnético externo, el momento de cada protón es forzado a alinearse a él, provocando una torca sobre el momento magnético. Esta fuerza está dada de la siguiente manera:

$$\vec{\tau} = \vec{\mu} \times \vec{B_0} \tag{2.3}$$

La cual a su vez produce un movimiento de precesión sobre el plano xy, sin embargo, los protones en el átomo se encuentran ordenados de manera aleatoria, por lo que las componentes de sus momentos magnéticos sobre este mismo plano, se cancelan entre sí permitiendo que sólo persista la componente en el eje z como se muestra en la figura (2.1) [1], permitiendo un solo momento magnético neto para cada átomo. Como se ha mencionado, al someter un átomo a un campo

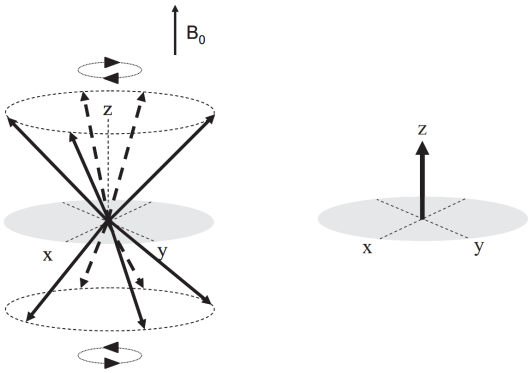

Figura 2.1: **Momento magnético neto** . Los momentos magnéticos de varios átomos en una molécula en precesión se cancelan para obtener un momento magnético neto de la molécula.

externo, la componente en el eje z de éste, puede alinearse únicamente de manera paralela y antiparalela al sentido del campo $\vec{B_0}$. Siendo estos sus únicos dos estados, uno de ellos con energía mínima por lo que, para hacer que un átomo pase de un estado de energía mínima al siguiente habría de excitarse. En la creación de imágenes por resonancia magnética esto se hace por medio del uso de radio frecuencias con energía específica.
Partiendo de la relación de energía y frecuencia de De Broglie $\Delta E = hf$ y al igualar la energía entre los dos estados del protón de la ecuación [2.2], se obtiene la energía específica de radio frecuencia necesaria para ese cambio de estado

$$hf = \Delta E = \frac{\gamma h B_0}{2\pi}$$

$$f = \frac{\gamma B_0}{2\pi}$$

Que también puede escribirse como:

$$\omega = \gamma B_0 \tag{2.4}$$

A la ecuación (2.4) se le conoce como ecuación de Larmor y es gracias a ella que se puede modificar la alineación del momento magnético de los átomos. A continuación se abordará el cómo a partir de esta variación en la energía se puede obtener una señal.

[1] (2011). Magnetización de un átomo precesando. En N.Smith, A.Webb, Introduction to medical imaging. Cambridge University Press, pag. 211

2.2.2. Uso de radio frecuencia

El uso de radio frecuencias en la resonancia magnética se hace con el fin de crear una magnetización transversal sobre el plano xy. Esto se logra gracias al envío de un pulso de radio frecuencia, cuya componente de campo magnético del pulso (B_1) esté orientada a 90° con respecto al campo externo $\vec{B_0}$. Esta componente B_1 actúa sobre el momento magnético del átomo de la misma forma en que lo hace el campo externo, lo cual provoca que el momento neto rote hacia el plano xy como se puede observar en la figura (2.2) [2].

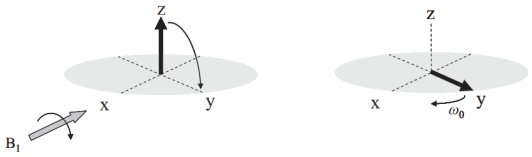

Figura 2.2: **Momento magnético antes y después de un pulso de** 90°. Se utiliza un pulso de RF con la componente de su campo magnético a 90° con respecto al B_0. Luego de detener el pulso, el la magnetización de la molécula precesa alrededor del eje z a la misma frecuencia de Larmor.

Es gracias a la alineación del momento magnético sobre el plano, que la señal puede ser recibida por antenas de radio frecuencia. Las antenas de radio frecuencia (RF) pueden ser, en su presentación más simple, un par de bobinas colocadas a 90° una con respecto a la otra. Al atraversarlas un flujo de campo magnético no constante en tiempo, por la ley de inducción de Faraday, obtendremos una señal eléctrica proporcional al campo magnético $\vec{B_0}$ aplicado y al número de protones en el medio. Cabe destacar que la posición en la que el pulso de RF colocó al momento magnético es un punto inestable, por lo que al terminar el pulso, el momento magnético neto del átomo regresará al estado de mínima energía.

Tiempos de relajación

La disminución de energía mencionada anteriormente es descrita por dos fenómenos principales, los cuales corresponden al tiempo de relajación longitudinal y transversal. Estos son referentes a la componente en el eje z de la magnetización y a la componente perpendicular de la misma respectivamente (Figura 2.3)[3], en esta figura se puede observar el decaimiento de la magnetización en cada eje, las cuales se obtienen al resolver las ecuaciones de Bloch sobre los ejes x, y y z como se ve en la ecuación (2.5).

$$\begin{aligned} M_x &= M_0 cos(\alpha) e^{-\frac{t}{T2}} \\ M_y &= M_0 sen(\alpha) e^{-\frac{t}{T2}} \\ M_z &= M_0 cos(\alpha) + (M_0 - M_0 cos(\alpha)) \left(1 - e^{-\frac{t}{T1}}\right) \end{aligned} \tag{2.5}$$

Algo que es importante de señalar es que dichos tiempos no están correlacionados entre sí.

- Tiempo de relajación $T1$: Está relacionado con el tiempo de relajación longitudinal y se define como el tiempo en el que tejido recupera el 63 % de la magnetización en el eje z. Además este mecanismo de relajación es referente a la interacción con los demás protones vecinos y

[2](2011). Orientación de la magnetización luego de un pulso de RF. En N.Smith, A.Webb, Introduction to medical imaging. Cambridge University Press, pag. 212

[3](2014). Decaimiento de señal. En C.Arévalo, L.Carrizales, M.Landrove, Verificación de la Distribución de Dosis en el Isocentro de un Gammaknife Mediante Dosimetría de Gel de Agarosa. Instituto Venezolano de Investigaciones Científicas, pag. 41

las fluctuaciones locales del campo magnético. La variación de este parámetro entre tejidos provoca el efecto de contraste en la imagen generada, ya que tejidos con un tiempo T1 más cortos serán más brillantes con respecto a los que poseen tiempos más largos como el agua. Este tipo de contraste es especialmente útil para visualizar tejido con grasa como el cerebro. Esto es debido a la estructura molecular del tejido y el tamaño de las moléculas que lo forman, ya que esto facilitará o entorpecerá la dispersión de energía [11].

- Tiempo de relajación $T2$: El tiempo T2 o de relajación transversal, se define como el tiempo que tarda el tejido para perder el 63 % de la magnetización en el plano xy [12]. Este parámetro es resultado de las interacciones spin-spin con los protones cercanos. Cabe destacar que este parámetro no cuantifica la pérdida de energía como el tiempo T1, sino que se trata de disminución en la coherencia de fase de los spin de los protones en el sistema [13].

- Tiempo de relajación $T2^*$: Similar al tiempo $T2$, con la particularidad de que no sólo contempla el desfase por interacciones spin-spin del tejido, sino también el desfase debido a las inhomogeneidades locales del campo magnético provocadas principalmente por cambios en la suceptibilidad magnética del tejido.

Es por esto, que estas técnicas son utilizadas según las necesidades del estudio a realizar. Para el caso de las imágenes $T1$, se utilizan para la detección de enfermedades en las cuales la concentración de agua en alguna región del paciente cambian, mientras que las imágenes $T2$ y $T2^*$ son usadas para cuantificar las concentraciones de hierro en el hígado, por ejemplo.

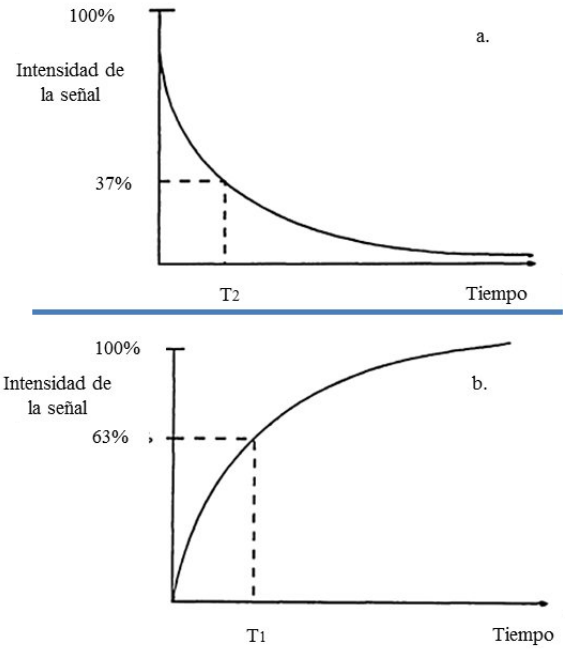

Figura 2.3: **Tiempos T1 y T2**. Se muestran las curvas de decaimiento de la señal para los casos correspondientes al tiempo T1 y T2.

Todo lo anterior es referente al comportamiento de moléculas de agua dentro de un campo magnético intenso, esto es más sencillo debido a la distribución electrónica de la molécula. Sin embargo, para moléculas como los lípidos esto es distinto debido a la geometría de la molécula, que a su vez hace que no sea posible omitir la contribución del momento magnético de los electrones. Esta

contribución actúa disminuyendo el campo magnético que se aplica sobre la molécula, por lo que el campo neto (B_{Neto}) sobre la molécula es:

$$B_{Neto} = B_0(1 - \sigma)$$

Donde σ es la constante de apantallamiento proporcional a la contribución de los electrones de la molécula. Y por lo tanto la frecuencia de resonancia es:

$$\omega = \gamma B_0(1 - \sigma) \tag{2.6}$$

De esta manera es posible distinguir entre la señal de dos o más tejidos distintos.
En el cuerpo humano existen tejidos que contienen una mayor densidad de protones, como lo son el tejido adiposo y el líquido cefalorraquídeo, con tiempos de relajación longitudinal o T1 más cortos. Por otro lado, los tejidos con menor densidad de protones como lo son el agua y tejidos dañados, tienen tiempos de relajación transversal más largos.
Existen secuencias de RF que se pueden usar para visualizar distintos tejidos con mayor contraste, estas diferencias dan como resultado distintas ponderaciones que permiten una mejor visualización de distintos tejidos. Algunas de las ponderaciones más comunes son las T1, T2, T2*. Para los fines de este trabajo, se restringirá el uso a la ponderación T1 debido a que es la indicada para poder visualizar de mejor manera el tejido cerebral.

2.2.3. Señal espacial de resonancia magnética

Una vez entendido el principio físico de la señal de resonancia magnética, habría también que hablar sobre cómo es posible distinguir entre señales provenientes de distintas ubicaciones del cuerpo. Con el fin de lograrlo, se llegó a la conclusión de que el campo magnético en el interior del resonador debería de variar espacialmente para que, de esta manera, cada región del paciente tuviera una frecuencia de resonancia específica. Gracias a esto, es posible usar secuencias de RF para obtener señales provenientes de cada una de las regiones del resonador y, para lograrlo se integraron bobinas de gradiente que inducen un campo magnético lineal dado por:

$$\nabla B_z = G_x + G_y + G_z \tag{2.7}$$

De manera en que no haya contribución extra de las bobinas de gradiente en el isocentro del resonador, por lo que el único campo en esa posición será el del imán y la correspondiente al campo local $B_{z(Tejido)}$, generado por la magnetización del tejido o lo que es lo mismo:

$$B_z = B_0 + zG_z + B_{z(Tejido)} \tag{2.8}$$

Por lo que, de la ecuación [2.4] la frecuencia de resonancia para este nuevo campo inhomogéneo es:

$$\omega_z = \gamma B_z = \gamma(B_0 + zG_z + B_{z(Tejido)}) \tag{2.9}$$

Espacio K

Las señales obtenidas por las antenas son codificadas en el dominio de frecuencias espaciales, este dominio se expresa por medio de dos variables, las cuales no se pueden visualizar directamente, sino que hace falta aplicar la transformada inversa de Fourier para descomponer este espacio de frecuencias en una imagen interpretable. Algo que resulta relevante es que la información codificada en este espacio depende de la técnica de adquisición utilizada.

2.2.4. Secuencias de imágenes

Aunque se ha abordado la forma en que se puede obtener señales a partir de la resonancia magnética, en la práctica, la adquisición de estas imágenes es tardada con respecto a los estándares clínicos, por lo que se ha recurrido al desarrollo de prácticas que permitan una adquisición más rápida, es decir nuevas secuencias.

Secuencia gradiente eco

Como se ha hablado anteriormente, la señal máxima se consigue cuando se tiene un pulso de alineación $\alpha = 90°$. Sin embargo, al enviar un pulso de noventa grados, el tiempo de repetición también aumenta para lograr que los protones vuelvan a su estado relajado antes de ser excitados nuevamente, en consecuencia el proceso de la adquisición de imágenes es más lento. Es por esto que se propuso reducir este ángulo α para agilizar el proceso.

$$I(x,y) \propto \rho(x,y)\frac{\left(1-e^{-\frac{TR}{T1}}\right)sen\alpha}{1-e^{-\frac{TR}{T1}}cos\alpha}e^{-\frac{TE}{T2^*}} \tag{2.10}$$

Para el caso de una imagen axial, la intensidad de los voxeles está dada por la relación de la expresión (2.10). Esta expresión muestra que existen tres factores principales que afectan la intensidad de la señal: La densidad de protones del tejido, el ángulo α; el tiempo esperado entre cada pulso de radio frecuencia o tiempo de relajación (TR), y el tiempo de relajación T1 en conjunto; y finalmente la combinación del tiempo de eco, así como el tiempo de relajación $T2^*$. Al variar estos parámetros, es posible ponderar la imagen a adquirir, ya sea T1 o $T2^*$ o destacar la concentración de protones en un tejido [10].

Secuencia spin eco

Las secuencias spin eco fueron creadas para suplir una de las principales desventajas de la secuencia gradiente eco, y es que, permiten obtener imágenes ponderadas para distintos valores T2. Después del primer pulso de radio frecuencia los protones irán desfasándose debido a la interacción con sus vecinos, en este tipo de secuencias se usa un segundo pulso de RF de 180° para invertir la precesión de los protones y hacerlos entrar en fase nuevamente, a esto es lo que se le llama eco de spin. Esto se ve reflejado en la posibilidad de obtener imágenes con contraste de tejidos con distintos tiempos T2.

Secuencia EPI

A pesar de tratarse de buenas técnicas de adquisición de imágenes en poco tiempo, las secuencias anteriores no son lo suficientemente rápidas para ciertas aplicaciones como el estudio de la perfusión de un tejido, angiografías por resonancia magnética, resonancia cardiaca y resonancia magnética funcional. Para este tipo de aplicaciones se crearon las secuencias rápidas, la primera de ellas fue la secuencia eco planar. Esta secuencia funciona con un único pulso de radio frecuencia seguido de un tren de ecos, con los cuales es posible llenar del espacio K completo. A esto se le llama secuencia single shot. Si el tren de ecos generado no es lo suficientemente largo, entonces en cada pulso llenarán de forma alternante líneas del espacio K.
Algunas de las ventajas de estas secuencias es que son especialmente sensibles al contraste $T2^*$, además de que son capaces de cubrir grandes extensiones del cuerpo en un tiempo de adquisición muy corto. Sin embargo, sus desventajas también son relevantes, ya que son imágenes de baja resolución, además de ser sensibles al movimiento y a variaciones del campo magnético.

Otra de las capacidades de la resonancia magnética que podemos destacar es que además

de obtener imágenes anatómicas o estructurales de distintas partes del cuerpo, es posible configurar los escáneres para poder detectar otro tipo de fenómenos fisiológicos como el flujo de sangre, activación de zonas del cerebro, etc.

2.2.5. Imágenes funcionales por resonancia magnética

A pesar de representar sólo el 2 % del peso de una persona, el cerebro realiza alrededor del 20 % del metabolismo del cuerpo, donde dicha energía es principalmente utilizada para el funcionamiento de señalización eléctrica de este órgano. El cerebro, a diferencia de otros órganos, no cuenta con depósitos para almacenar energía; por lo que depende de la entrega de glucosa por parte del torrente sanguíneo y su metabolización por medio de la oxidación para continuar con su actividad.

Es gracias a esta necesidad de energía para la actividad neuronal, que la concentración de sangre oxigenada en la región activa disminuye, lo que a su vez aumenta la concentración desoxihemoglobina. La diferencia en las propiedades de susceptibilidad magnética de la desoxihemoglobina y oxihemoglobina, radican en la presencia o ausencia de enlaces con oxígeno. Si la hemoglobina carece de oxígeno, el hierro presente en ella estará en estado ferroso, por lo que tendrá cuatro electrones desapareados. Estos electrones generarán un momento magnético que la hará comportarse como un material paramagnético. Por otro lado, si ahora la hemoglobina está unida a un oxígeno, la hemoglobina compartirá uno de sus electrones con la molécula de oxígeno, perdiendo el momento magnético neto que poseía [14]. Dicho fenómeno provocará gradientes de señal, dando como resultado el contraste entre regiones; a esto se le conoce como el efecto Blood Oxygen Level Dependent (BOLD por sus siglas en inglés). Estos cambios en la señal a su vez alterarán el tiempo de decaimiento $T2^*$.
Esta técnica tiene dos aplicaciones principales:

- Resonancia funcional basada en tareas:
  Esta técnica cuenta con tres variantes principales: Tareas en bloques, basado en un evento y una combinación de ambos. El primero consiste en la presentación de varias tareas o condiciones en un paradigma repetitivo, que puede tener o no episodios de descanso para tener referencia del cerebro únicamente con actividad espontánea. Para el caso de la técnica basada en eventos consiste en la presentación de una sola tarea realizada bajo distintas situaciones separadas por un intervalo de descanso.

- Resonancia funcional en estado de reposo (rs-fMRI):
  El objetivo principal de esta modalidad es la cuantificación de la conectividad funcional entre distintas regiones del cerebro, esto por medio de la correlación de patrones de activación simultáneas dentro de una misma ventana temporal, de esta manera se determinan áreas con una misma funcionalidad; y con base en el nivel de correlación de estas regiones, se pueden identificar redes de funcionales asociadas con ciertas regiones del cuerpo.
  Las principales ventajas con respecto a la resonancia basada en tareas, es que se trata de una técnica más sensible para medir la conectividad del cerebro [15]; debido a que los cambios espontáneos de la señal BOLD tienen la particularidad de tener fluctuaciones en la amplitud del orden de 0.01-0.1 Hz [16]. Además de revelar información de la activación de todo el cerebro en lugar de regiones específicas. Por lo cual esta técnica tiene la posibilidad de ser usada para monitorear el avance de enfermedades o desordenes mentales [17].

Como se puede observar, las imágenes de rs-fMRI tiene el potencial de obtener información relevante con respecto a la funcionalidad del cerebro y el desarrollo de anomalías, por lo que se presenta como una herramienta para el seguimiento de enfermedades como el Alzheimer [18].

rs-fMri para el estudio de Alzheimer

Es gracias a la información que aportan las imágenes en estado de reposo que, al revisar en el estado del arte, es posible encontrar una gran variedad de técnicas de análisis para estudiar el desarrollo de la enfermedad de Alzheimer, como la extracción de mapas neuronales [18], análisis de patrones en las imágenes [19], etc. Esto es posible gracias a los cambios en la oxigenación en regiones afectadas, lo cual es consecuencia de la muerte celular y la degradación de la actividad neuronal.
En particular para el análisis de patrones en las imágenes funcionales de estado de reposo, una de las técnicas más usadas es el uso de modelos de aprendizaje máquina, ya que, gracias a su gran capacidad de detección de patrones, son capaces de obtener buenos resultados en su clasificación.

2.3. Redes Neuronales

El cerebro humano posee una gran cantidad de habilidades que no pueden ser igualadas por ninguna unidad de procesamiento artificial, como por ejemplo, la habilidad de reconocer objetos específicos entre una gran cantidad de objetos relacionados incluso en condiciones no favorables como en una habitación con poca iluminación. Sin embargo, la habilidad más sorprendente de nuestro cerebro es, sin duda, la capacidad de aprender nuevas habilidades; esto es posible gracias a una compleja y extensa red interconectada de unidades de procesamiento individuales llamadas neuronas, las cuales son capaces de comunicarse de manera casi inmediata con otras por medio de pulsos eléctricos. Es por esto, que uno de los objetivos de la investigación en el área de matemáticas es la creación de algoritmos que permitan dotar de la habilidad de aprendizaje a máquinas con mayor capacidad de procesamiento: las computadoras.

2.3.1. Aprendizaje de máquina

El desarrollo de las herramientas más sofisticadas que conocemos en la actualidad, comenzó con algoritmos más sencillos mucho antes, incluso, de la creación de las primeras computadoras. Esta primera etapa del desarrollo de algoritmos de aprendizaje es llamada comúnmente Aprendizaje de máquina tradicional. El objetivo principal de este tipo de técnicas es mejorar el desempeño de un algoritmo con el pasar del tiempo.
En el contexto del aprendizaje de máquina es común encontrar distintas clasificaciones de los algoritmos usados en esta área, sin embargo, la clasificación comunmente usada es:

- Aprendizaje supervisado
- Aprendizaje no supervisado
- Aprendizaje semi supervisado
- Aprendizaje por refuerzo

Cada una de esta clasificaciones son usada principalmente para ciertas aplicaciones según el objetivo. A continuación se profundizará en dichas clasificaciones y se mencionarán algunos de los algoritmos clasificados

Aprendizaje supervisado

Este tipo de algoritmos consisten en la aproximación de una función, de manera en que luego del entrenamiento del algoritmo, se obtendrá una función que logre estimar de mejor manera la relación entre la información de entrada y la salida esperada. Esto es, dado un conjunto de características $\mathbf{X}$ tal que $\mathbf{X} = \{x_1, x_2, \cdots, x_n\}$ con una salida o etiqueta $\mathbf{Y}$ correspondiente, se busca aproximar

una función $\mathbf{F}$ tal que $\mathbf{F}(\mathbf{X}) = \mathbf{Y}$
Esto se realiza con el fin de que el algoritmo pueda aprender patrones y de esta forma sea capaz de predecir la información $\mathbf{Y}_{new}$ a partir de un conjunto de características $\mathbf{X}_{new}$
Por lo tanto se trata principalmente de modelos predictivos para clasificación o predicción donde se requiere de datos etiquetados, es decir, que cada $\mathbf{X}$ de entrenamiento tenga su correspondiente $\mathbf{Y}$ para que el modelo aprenda la relación entre ellos. Algunos ejemplos son los modelos de vecinos más cercanos, Naive Bayes, árboles de decisión, regresión lineal, máquinas de soporte vectorial (SVM por sus siglas en inglés) y las redes neuronales.

Aprendizaje no supervisado

En el caso de este tipo de algoritmos no es necesario contar con conjuntos de características etiquetadas, ya que no buscan establecer una relación directa entre las x_i y una $\mathbf{Y}$ particular, sino que buscan detectar patrones en las características de un conjunto de datos. Es común el uso de este tipo de algoritmos en casos donde no se sabe con seguridad qué tipo de información se busca obtener de los datos.
Las principales aplicaciones de este tipo de algoritmos son el reconocimiento de patrones y modelos descriptivos, ya que pretenden establecer reglas de correspondencia para sintetizar y agrupar conjuntos de datos. Unos de los ejemplos son los algoritmos de agrupación K-medias y reglas de asociación.

Aprendizaje semisupervisado

Este tipo de aprendizaje puede considerarse una combinación de los dos anteriores ya que a pesar de poseer datos etiquetados, no descarta la información implícita que puede ser obtenida de los datos.

Aprendizaje por reforzamiento

Estos algoritmos aprenden directamente de la información obtenida en tiempo real, con el fin de tomar decisiones que maximicen la recompensa o minimicen la penalización del algoritmo. Esto lo realiza de forma iterativa hasta probar cada uno de los escenarios posibles, todo esto gracias a una señal de reforzamiento o retroalimentación en un proceso de decisión de Markov. Este método forma parte de la rama de la Inteligencia Artificial.
Los principales aplicaciones de estos algoritmos pueden ser la automatización de la operación de herramientas. Entre los algoritmos más usados están: aprendizaje-Q, diferencias temporales y redes profundas adversarias.

Además de la clasificación mencionada anteriormente, es posible clasificar estos algortimos por similaridad en cuanto sus procesos y aplicaciones. Con base a esta clasificación sería posible determinar cuál sería el algoritmo que mejor pueda funcionar para nuestro trabajo.

- Algoritmos de regresión
  En este tipo modelan la relación entre las características de la información que obtienen y recibe retroalimentación iterativa al cuantificar el error de las predicciones que realiza.

- Aprendizaje basado en características
  Estos algoritmos escogen las características que pueden resultar más representativas de la información obtenida. Al obtener nueva data, la comparan con ayuda de una métrica de similaridad para poder determinar el conjunto al que mejor se adapte ese nuevo dato y realizar una predicción.

- Algoritmos de árboles de decisión
  En estos algoritmos plantean modelos de decisiones con base en las características de los datos

y sus posibles resultados, esto con el fin de formar subconjuntos homogéneos al explorar todas las posibilidades posibles.

- Algoritmos Bayesianos
  Los algoritmos Bayesianos son los que ocupan explícitamente el teorema de Bayes para realizar clasificación y regresión.

- Algoritmos de clustering
  Estos métodos ocupan las características de los datos para así organizar la data en grupos de mayor similaridad.

- Algoritmos de aprendizaje por reglas de asociación
  Se usan para extraer relaciones entre las variables de la data para establecer reglas. Ayuda principalmente a encontrar relaciones implícitas en la información.

- Algoritmos de redes neuronales artificiales
  Comúnmente son usados para el reconocimiento de patrones para clasificación y regresión, además de tener una gran versatilidad para ser usados en distintos problemas.

- Algoritmos de aprendizaje profundo
  Estos podrían considerarse como una versión mejorada de los algoritmos de redes neuronales artificiales, ya que se usan para el análisis de data analógica como imágenes, entre otras.

- Algoritmos de reducción de dimensión
  Estos algoritmos buscan simplificar o sintetizar de la información en la data de entrada, normalmente es usado como un pre procesamiento de la información para usarse en algún otro algoritmo.

- Ensambles de algoritmos
  La principal característica de este tipo de algoritmos el es uso de varios modelos débiles, como son llamados en este contexto, entrenados independientemente. Esto es con el fin de que al combinar sus predicciones puedan obtener mejores resultados de los que podría obtener cada uno por separado.

Dada esta breve revisión de los algoritmos de aprendizaje, es posible vislumbrar el tipo de herramienta que sería más útil para los fines de esta investigación.

Dentro de las herramientas del aprendizaje profundo, se encuentra uno de los algoritmos más usados y con mejores resultados en el campo de análisis de imágenes: Las redes neuronales convolucionales (RNCs). Las RNCs presentan una amplia variedad de ventajas sobre otro tipo de algoritmos en aplicaciones de visión de computadora, como:

El número de parámetros; los algoritmos de redes neuronales artificiales, a pesar de su gran versatilidad y poder de análisis tienen ciertas desventajas. El número de parámetros entrenables en estos modelos provoca que consuman una gran cantidad de recursos computacionales durante su entrenamiento, sin embargo, en el caso de las RNCs disminuyen el número de parámetros a modificar sin perder calidad en los resultados; esto es posible gracias a los elementos de convolución, en los cuales se profundizarán en futuras secciones [20].

Ayudan a conservar relaciones espaciales en la imagen; las imágenes no son más que un conjunto de tensores que contienen distintas intensidades en cada una de sus entradas y a cada una de estas entradas se les llama característica, por lo que se trata de elementos de alta dimensionalidad. Las RNCs permiten analizar cada una de las matrices en el tensor y conservar las relaciones espaciales de la información extraída con respecto a la imagen original, por lo que son herramientas muy útiles para tareas como la identificación de objetos y segmentación.

Es por esto que estos modelos resultan especialmente útiles para aplicaciones de visión de computadora, en particular para el análisis de imágenes médicas.

Antes de profundizar en las maravillas y aplicaciones de las redes neuronales convolucionales, es necesario estudiar más a fondo los cimientos matemáticos sobre los cuales descansan este tipo de algoritmos, con el fin de ser capaces de entender los procesos detrás de su aprendizaje.

Tarea de clasificación

En el contexto del aprendizaje de máquina, la clasificación se refiere a la tarea en la que un modelo dado intenta predecir la clase a la cual pertenece una muestra dada con base en las características de la misma. Como se trató anteriormente en la sección de estilos de aprendizaje, en el área de aprendizaje supervisado, las muestras o elementos de las bases de datos $\mathbf{X}$, se consideran como un conjunto de valores tales que $\mathbf{X} = (x_1, \ldots, x_n)$. Este conjunto de elementos de nuestra base de datos puede ser representado en el espacio de características donde se asocia una dimensión a cada una de las n características. Según la distribución de los conjuntos de puntos en este espacio, la base de datos puede ser linealmente separable, no linealmente separable o no separable. El primer caso corresponde a los conjuntos de datos para los cuales existe un hiperplano que sea capaz de separar correctamente las clases en el espacio. El caso de los no linealmente separables, es el tipo de base de datos en los cuales un hiperplano es no es suficiente para separar las clases, por lo que se utiliza una sábana en el hiperespacio para delimitar las clases. Finalmente, las bases de datos no separables son las bases de datos cuyas clases comparten espacios cercanos al grado de no poder ser discriminados ni usando sábanas en el hiperespacio. A la intesersección de las superficies en el hiperespacio que son usadas para separar las clases con el espacio de características se les llama *frontera de decisión* En la figura (2.4) [4] se pueden observar ejemplos de estos casos.

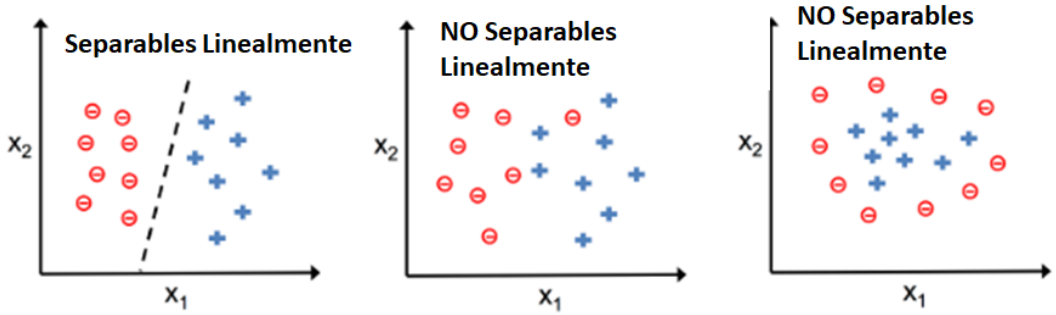

Figura 2.4: **Tipos de bases de datos según sus distribuciónes en el espacio de características**. Clases linealmente separables: Corresponden a una distribución en la cual es posible separar las clases por medio de hiperplanos (Primer caso). Clases no separables linealmente: Estas son distribuciones las cuales no es posible separar las clases por medio de hiper planos sino por medio de superficies más complejas (Tercer caso) o por ningún tipo de superficie en el hiperespacio (Segundo caso).

2.3.2. Redes Neuronales Convolucionales

La forma en cómo percibimos el mundo que nos rodea siempre ha sido una de las más interrogantes en la historia de la humanidad, a lo largo de los siglos se han intentado dar respuestas desde distintas disciplinas. Junto con el desarrollo de la ciencia y la experimentación se comenzó a

[4] Distribuciones de datos en el espacio de características. Recuperado de https://miro.medium.com/v2/resize:fit:1868/1*-wmejyYl552E3Ueoosfo8Q.png (18 de marzo de 2024).

indagar sobre la naturaleza del cerebro y su funcionamiento. En el siglo XIX ya había ciertos avances respecto a su estudio, sin embargo, la forma en que las células nerviosas se interconectaban aún era un misterio. Fue hasta que Camillo Golgi demostró la forma en que esto sucedía por medio de su técnica usando plata refinada. Más adelante en 1887 Santiago Ramón y Cajal, usando esa misma técnica, pudo realizar observaciones importantes que lo llevaron a proponer que las neuronas, células individuales con su propia pared celular y polaridad, se interconectan para propagar impulsos eléctricos; lo que les permitió ganar el premio nobel de medicina en 1906.

Desde entonces ha crecido el deseo de la humanidad por dotar a las computadoras de las habilidades de procesamiento y reconocimiento de patrones, con la diferencia que estas, al tener una mayor capacidad de cómputo que el cerebro humano, serían capaces de realizar tareas aún más complejas. Es por esto que se ha buscado la forma de mimetizar la forma en que estas células procesan la información. No fue hasta 1948 que Wallter Pitts y Warren McCulloch publicaron su artículo *A logical calculus of the ideas immanent in nervous activity* con base en las ideas propuestas por Alan Turing sobre cómo describir la funcionalidad del cerebro [21].
Finalmente en 1958 Frank Rosenblatt, basado en el trabajo mencionado anteriormente, propuso su modelo matemático del perceptrón, en el cual ahondaremos a continuación.

El perceptrón

Basado en la modelación matemática de una neurona biológica, el preceptrón presenta la oportunidad de imitar la forma en que la información llega a la célula y cómo,esta a su vez, se activa para enviar un estímulo o señal hacia otra para continuar con la propagación de la información. En 1958 Frank Rosenblatt, en su artículo *The perceptron: A probabilistic model for information storage and organization in the brain*, comienza estableciendo las bases de su modelo matemático al explicar la forma en que los seres vivos percibimos la luz. De acuerdo con Rosenblatt, el funcionamiento

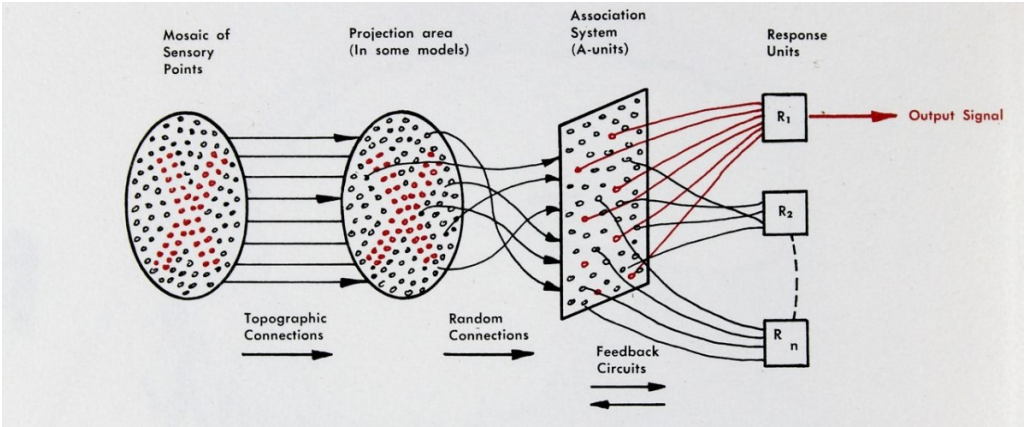

Figura 2.5: **Organización de un perceptrón**. Organización de las distintas zonas que procesan la información que ingresa a una neurona de la corteza visual.

de un foto perceptrón o de una célula perteneciente a la corteza visual del cerebro (Figura 2.5) [5] consiste en distintos momentos del procesamiento de la información:

- Comienza con la llegada de la información a las regiones sensoriales, también llamadas puntos S. Estos puntos se caracterizan por tener respuestas "todo o nada", es decir, sus potenciales de acción se presentan únicamente en un valor discreto o no se presenta. Este tipo de activaciones

[5](1958). Organización de un perceptrón. En F.Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review Vol. 65, No. 6 pp. 386-408

son muy comunes en los estudios de neurología donde se asume que existe un umbral mínimo sobre el cual la señal de la neurona será transmitida.

- Las señales transmitidas son recibidas por las células de asociación dentro del área de proyección, esta área en ocasiones es omitida en algunos modelos. Al conjunto de puntos S que transmiten información a una célula de asociación en particular, se les llaman puntos de origen, estos pueden realizar el papel de excitar o inhibir las células de asociación. Aquí se realiza una breve síntesis de las señales, donde si la suma algebráica de éstas iguala o supera un umbral, la célula transmitirá una señal a la siguiente capa. Cabe destacar que la información de los puntos sensoriales es sintetizada en n menor número de puntos en el área de proyección.

- Luego de transmitir las señales, la información pasa al área de asociación por medio de conexiones aleatorias, de manera en que el número de conexiones es el mismo al de células activas en la capa de proyección.

- Finalmente, las respuestas R_1, R_2, $\cdots$, R_n actúan como células con un conjunto de puntos sensoriales aleatorio asociado.

Es posible notar en la imagen (2.5) que existe un flujo establecido de la información, la cual se propaga hacia adelante, mientras que hay retroalimentación entre las áreas de asociación y la de respuesta. Esta retroalimentación puede ser de tipo excitatoria o inhibitoria. De esta manera es que el área de asociación puede irse ajustado, de forma en que las respuestas de salida pueden ser más apropiadas [22].

De esta manera es que se puede simplificar el procesamiento de la información por parte de una neurona y, con base en esto, Roseblatt propuso el modelo matemático de una neurona. Este modelo contempla una neurona con múltiples entradas de información, las cuales no comparten el mismo nivel de relevancia, por lo que cada una de estas entradas debe de ir acompañada por un factor de ponderación. Luego de esto, la suma de estos términos pasa por una función que comprueba si dicho valor supera o no un umbral dado para así transmitir su información. Esta última parte es la que identificamos anteriormente como la unidad de respuesta que da origen a la señal de salida para así recibir retroalimentación.
Todo este proceso puede ser sintetizado en la figura (2.6) [6] donde el papel de comprobación del valor de la suma ponderada es llevado a cabo por medio de la función de activación, en la cual ahondaremos más adelante. Matemáticamente lo expresado anteriormente puede escribirse de la siguiente forma:

$$P = \sum_{i=1}^{n} x_i w_i + b \tag{2.11}$$

Donde los x_i son cada una de las entradas de la neurona, así como los w_i son los pesos de cada una de ellas y finalmente el término b es el correspondiente al sesgo o *bias* el cual agrega un grado de libertad extra a nuestro modelo para así hacerlo más versátil y adaptable al problema en cuestión. Si nuestra neurona está realizando una tarea de regresión, este será el resultado de la predicción hecha con base a los valores de entrada de cada una de las características de nuestra muestra.

En otro caso, si se está realizando una tarea de clasificación, es necesario un paso extra ya que una vez que ha obtenido la suma algebráica de las entradas y los pesos es posible verificar si la neurona será activada dado su valor resultante, esto con ayuda de la función de activación que mejor resulte para la tarea.

$$y = g(P) \tag{2.12}$$

[6] Perceptrón. Recuperado de https://www.wikiwand.com/es/Perceptr%C3%B3n (18 de marzo de 2024). 18

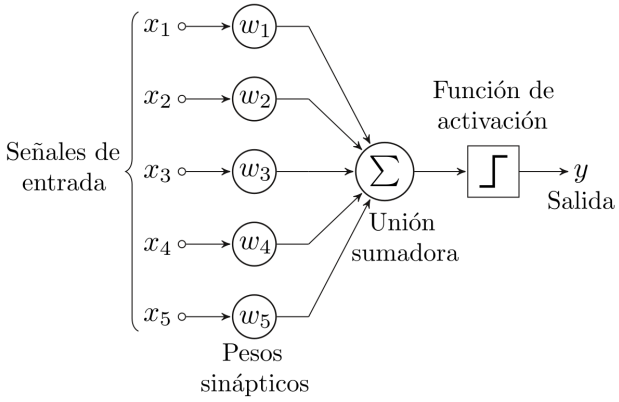

Figura 2.6: **Modelo matemático de perceptrón**. Este diagrama muestra la forma en que se procesa la información de entrada en una neurona y su activación, en este caso se tiene una entrada de 4 características.

Donde y es la salida del modelo y la función $g(P)$ es la función de activación mencionada anteriormente.

Este simple modelo es capaz de resolver problemas de clasificación linealmente separables, lo que significa que es capaz de distinguir la clase a la que pertenecen distintas muestras de una base de datos siempre y cuando estas puedan ser separadas por una frontera lineal en el espacio de características. Sin embargo, la gran mayoría de problemas de clasificación buscan discriminar entre clases cercanas entre ellas en dicho espacio, por lo que esta propuesta no es suficiente para poder abordar estos casos. Debido a esto es que se desarrolló otra metodología con mejores capacidades para tratar estos casos: El perceptrón multicapa.

2.3.3. Perceptrón multicapa

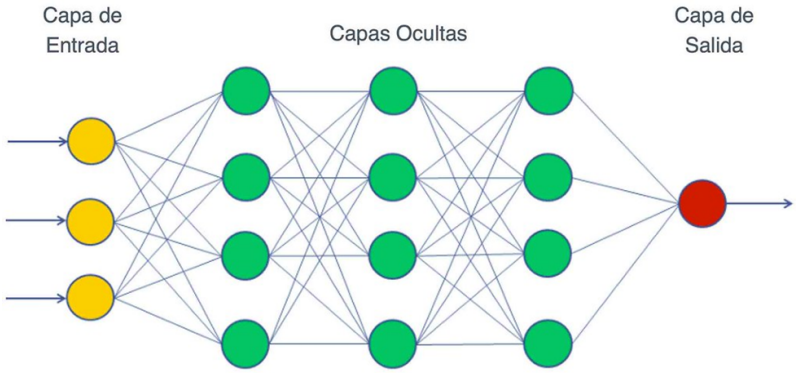

Figura 2.7: **Perceptrón multicapa**. Este diagrama muestra los principales elementos de un perceptrón multicapa. Cada uno de los nodos verdes y rojos representa una neurona como el perceptrón, las líneas muestran el flujo de la información y representan los pesos usados para ponderar las sumas algebráicas en las neuronas de la siguiente capa.

Para poder resolver problemas más complejos se ha propuesto de modelos más sofisticados como lo son las redes neuronales de avance o *Feed Foward Neural Networks*, las cuales son llamadas

de esta manera debido a que el flujo de la información a través de ella sigue un sólo sentido directamente desde las entradas hacia las salidas sin ningún tipo de loop. Estos modelos consisten en el flujo de la información a través de un conjunto de neuronas como las descritas en la sección anterior ordenadas en capas como se puede observar en la figura (2.7)[7]. En cada una de estas capas intermedias, también llamadas **capas ocultas**, las neuronas reciben las salidas de cada una de las neuronas de la capa anterior y transmiten sus propias salidas a cada una de las neuronas de la capa siguiente, por lo cual también son conocidas como capas completamente conectadas. El añadir un mayor número de capas a nuestro modelo puede hacerlo más versátil para poder generar fronteras de decisión más complejas, de manera en que se puedan obtener los mejores resultados posibles. Sin embargo, un mayor número de capas no es, por sí sola, la causa de una mejor predicción en modelos más complejos; esto es gracias al papel de las funciones no lineales de activación usadas en cada capa.
Existe una gran variedad de funciones de activación entre las cuales destacan las siguientes por ser las más usadas:

- **Función Sigmoide**
  Al igual que todas las funciones de activación mencionadas en esta sección, se trata de una función continua cuyo rango es el intervalo (0,1), esta propiedad es útil particularmente para ser usada en la capa de salida en los modelos de perceptrón multicapa, ya que la probabilidad de algún suceso se encuentra en este mismo intervalo. Esta función es comúnmente utilizada para clasificación binaria
  $$Y(z) = \frac{1}{1 + e^{-z}} \tag{2.13}$$
- **Función Tangente Hiperbólica**
  Esta función tiene un comportamiento similar a la función sigmoide, sin embargo, tiene un rengo de (-1, 1), por lo que al incluir al cero en este intervalo, puede ayudar a que el modelo pueda obtener resultados con un menor margen de error más rápido durante el entrenamiento de nuestro modelo.
  $$Y(z) = \frac{e^{-z} - e^{-z}}{e^{-z} + e^{-z}} \tag{2.14}$$
- **Unidad lineal rectificada**
  La función de unidad lineal rectificada o ReLU en inglés, posee un comportamiento notablemente distinto, ya que como se muestra en la ecuación (2.15), no cuenta con puntos de saturación, lo que le dota de un comportamiento favorable durante el entrenamiento del algoritmo, sin embargo, cabe destacar que su uso se restringe a las capas ocultas o internas de un modelo multicapa. Esto es debido a que su rango está limitado a valores mayores o iguales a cero, además su salida no es simétrica al asignar a un solo valor una gran cantidad de valores de su dominio.
  $$Y(z) = max(0, z) \tag{2.15}$$
- **Función Softmax**
  Esta función es usada principalmente para ejercicios de clasificación multiclase, aunque también puede ser usada para clasificación binaria. Esto es posible gracias a su capacidad de normalización de las activaciones de las neuronas pertenecientes a la capa anterior, obteniendo como salida un vector de probabilidad de clasificación para cada clase a discriminar, por lo que las entradas de dicho vector siempre sumarán 1, ayudando así con su interpretabilidad.
  $$S(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{2.16}$$

A grandes rasgos se ha abordado sobre lo que son los modelos de perceptrón multicapa, sin embargo, no ha ahondado sobre la forma en que dichos modelos procesan la retroalimentación del ambiente.

[7]Perceptrón Multicapa. Recuperado de https://aprendeia.com/que-es-el-perceptron-simple-y-multicapa/ (18 de

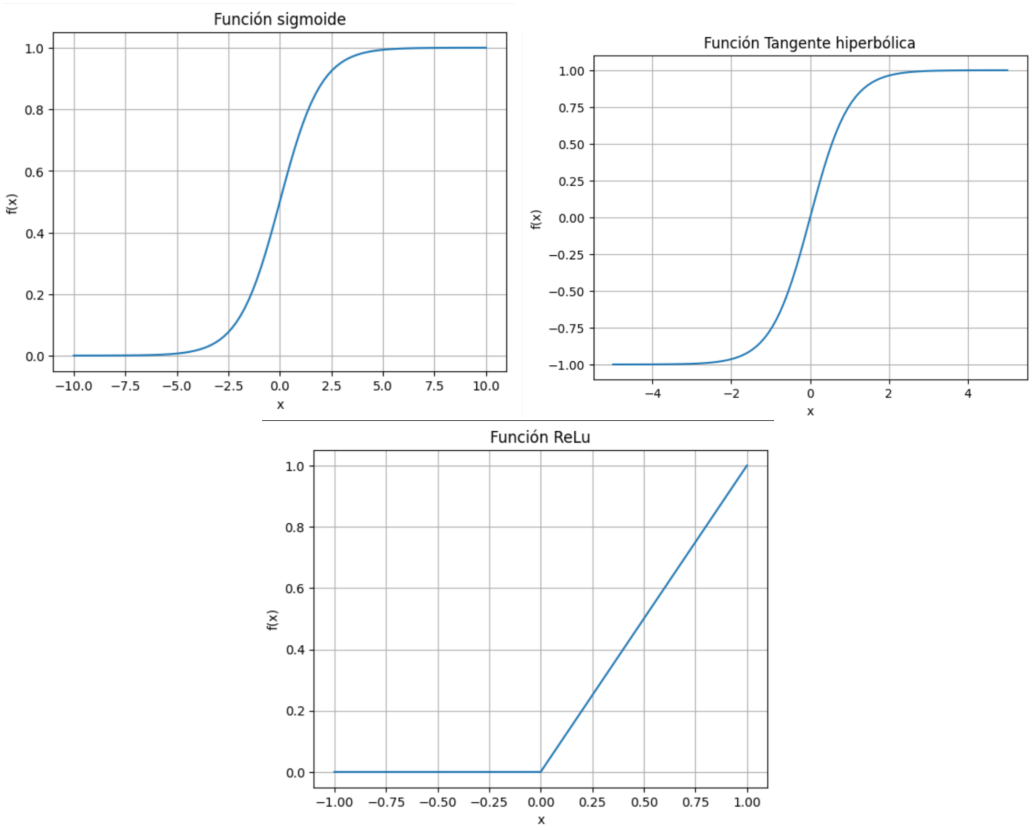

Figura 2.8: **Funciones de activación**. Se puede observar el comportamiento de las funciones de activación más comunes dentro de un intervalo dado. Nótese la diferencia en la pendiente de las funciones sigmoide y tangente hiperbólica cerca del origen, siendo menor la de la sigmoide, por lo cual esta alcanzará primero la saturación, es decir, los cambios en el valor de la función serán cada vez menores conforme el valor de entrada se aleje del origen. Finalmente la función ReLu no se satura.

Funciones de costo

Como se ha mencionado, para lograr que el modelo escogido aprenda a predecir las clasificaciones dado un conjunto de muestras de entrenamiento, son necesarios los siguientes requisitos: El conjunto de entrenamiento debe contar con las etiquetas de cada una de las muestras, es decir, se debe saber con anterioridad la clase a cual pertenece cada una. Además es necesario contar con retroalimentación para saber si la predicción de clasificación realizada por el modelo fue correcta o incorrecta. Para poder cuantificar el grado de error o diferencia entre la predicción hecha (Y_i) y la etiqueta ($\hat{Y}_i$) de nuestra i-ésima muestra, dada cierta configuracion de pesos del modelo. Para esto es necesario agregar a nuestro modelo un ingrediente más: La función de costo. Las funciones de costo más usadas son:

- **Error cuadrático medio**
  El error cuadrático medio (MSE por sus siglas en inglés) se encarga de calcular la distancia promedio entre cada una de las predicciones de etiquetas de cada muestra y su valor real. Esta función suele expresarse sin el término de un medio que se muestra en la ecuación (2.17), sin embargo, se puede agregar para así simplificar la expresión a la hora de calcular su derivada y reducir la carga computacional. Una de las desventajas de esta métrica es que al utilizar el cuadrado del error del modelo es posible que para valores menores a 1 y muy grandes, se tenga una estimación errónea por parte de esta función.

$$C = \frac{1}{2n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.17}$$

- **Error absoluto medio**
  Esta función de activación tiene un comportamiento similar a la función anterior, ya que calcula la diferencia absoluta promedio, sin embargo, esta función penaliza con menor severidad los valores grandes de error, evitando una estimación errónea del error del modelo.

$$C = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \tag{2.18}$$

- **Cross entropy**
  Para el caso de la función de costo de entropía cruzada existen dos versiones, se trata de la función de costo binaria (Ecuación 2.19) y la categórica (Ecuación 2.20), las cuales corresponden a problemas de clasificación binaria y multiclase respectivamente. Algo que cabe destacar de esta función, es el hecho de que es capaz de calcular la entropía total entre distribuciones de probabilidad, estos conceptos provienen de la teoría de información, en la cual la entropía mide la incertidumbre de una fuente de información. En el contexto de las redes neuronales artificiales, esta función se encarga de cuantificar la diferencia entre distribuciones de probabilidad dado un conjunto de eventos.

$$C = -\frac{1}{n}\sum_{i=1}^{n}[\hat{Y}_i log\,(Y_i) + \left(1 - \hat{Y}_i\right) log(1 - Y_i)] \tag{2.19}$$

$$C = -\frac{1}{n}\sum_{i=1}^{n}\hat{Y}_i log(Y_i) \tag{2.20}$$

Estas son sólo algunas de las opciones de funciones existentes que se pueden usar para cuantificar el error en las predicciones del modelo escogido, sin embargo, es posible definir nuevas funciones según las necesidades de nuestro modelo y el problema en el que se esté trabajando. Estas propuestas deben cumplir con las siguientes condiciones:

marzo de 2024).

- **Ser diferenciable**: Esta condición es una de las más importantes y en la siguiente sección se hablará más al respecto.
- **Ser una función convexa**: Esto es, la función debe contar con un mínimo global, aunque, si bien no todas las funciones de costo populares son convexas, todas ellas tienen al menos mínimos locales

Una vez que hemos hablado sobre las herramientas que se tienen para cuantificar el desempeño del modelo, se puede proceder con el componente más importante en el proceso de aprendizaje de una red neuronal, incluso para modelos más complejos. En la siguiente sección se abordarán los dos momentos principales en el flujo de la información dentro de una red neuronal: La propagación hacia adelante y propagación hacia atrás.

Algoritmo de propagación hacia adelante

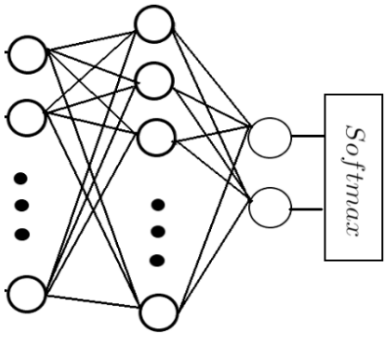

Figura 2.9: **Perceptrón multicapa de ejemplo**. Este diagrama muestra el modelo utilizado como ejemplo en esta sección. Tiene una capa de entrada para 169 características más un sesgo, en la capa oculta se encuentran 5 neuronas y dos en la capa de salida. La función de costo usada es Entropía cruzada para clasificación multiclase, Softmax y Relu como funciones de activación en capa de salida y ocultas respectivamente.

Para simplificar la explicación de estos dos momentos se propone el siguiente ejemplo. Dada la red neuronal que se muestra en la figura(2.9) la cual es una red neuronal de 3 capas incluyendo a la capa de entrada y salida. A esta red ingresan dos datos de entrada, los cuales se llamará $x_1, x_2, \cdots, x_{169}$, recordando la ecuación (2.11), el término b representa el sesgo que se agrega a nuestro modelo, sin embargo, para agilizar la programación, es posible agregar un término llamado x_0, el cual será multiplicado por un peso extra w_0 de manera en que esta combinación cumplirá el papel del sesgo b, este término será agregado en cada uno de las capas. Por lo que las entradas pueden ser escritas como el vector:

$$X = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{169} \end{pmatrix} \tag{2.21}$$

Cada una de las conexiones entre los nodos de la figura (2.9) representa los pesos asociados a cada activación de la capa anterior, estos pesos pueden escribirse de forma más simple como una matriz de pesos de dimensiones *Neuronas de capa anterior* x *Neuronas de capa siguiente*, esto se puede apreciar de mejor manera en la ecuación 2.22, la cual muestra la primer matriz de pesos entre la primera y la segunda capa. En ella se puede observar el conjunto de w_{0j} con $j \in \{1, 2, \ldots, 5\}$, los

cuales corresponde al valor de los sesgos mencionados anteriormente.

$$W_{170\times 5} = \begin{pmatrix} w_{0,1} & w_{0,2} & \dots & w_{0,5} \\ w_{1,1} & \ddots & \dots & w_{1,5} \\ \vdots & \vdots & \vdots & \vdots \\ w_{169,1} & \dots & \dots & w_{169,5} \end{pmatrix} \tag{2.22}$$

De manera en que al operar $W^T X$ se obtienen las sumas ponderadas de cada una de las entradas para cada neurona de la segunda capa, a estas sumas se denotarán como $z_j^{(l)}$, donde l es la capa de la red a la que se propagará la información y j representa el número de la neurona de la capa a la que se propagará la información. El conjunto de $z_j^{(l)}$ de la capa l se pueden representar como un vector $Z^{(l)}$, por lo que, para el caso del ejemplo se tiene lo siguiente:

$$W_{170\times 5}^T X = \begin{pmatrix} \sum_{i=0}^{170} w_{1i}x_i \\ \sum_{i=0}^{170} w_{2i}x_i \\ \vdots \\ \sum_{i=0}^{170} w_{5i}x_i \end{pmatrix} = Z^{(2)} \tag{2.23}$$

Recordando que es altamente común el uso de funciones de activación f en la salida de cada neurona, estas funciones pueden ser algunas de las que ya se han mencionado en la sección anterior, de manera que la activación de la neurona sea una expresión no lineal. Por esto utilizará la función ReLu de la ecuación (2.15)

$$a_j^{(l)} = ReLu(z_j^{(l)}) \tag{2.24}$$

Por lo cual, a sería la activación de la neurona j en la capa l y de igual manera, el vector de activaciones de la capa l será $A^{(l)}$. Para evitar perder generalidad en nuestra notación, se define lo siguiente:

$$A^{(1)} \equiv X$$

Por lo cual, a manera de resumen, es posible expresar la propagación de la información hacia adelante para la capa 2, como:

$$Z^{(2)} = W^{T(1)} A^{(1)}$$
$$A^{(2)} = f(Z^{(2)})$$

Y finalmente para la capa 3, se puede expresar como a continuación:

$$Z^{(3)} = W^{T(2)} A^{(2)}$$
$$A^{(3)} = f(Z^{(3)})$$

A simple vista es fácil detectar el patrón del algoritmo de propagación hacia adelante, por lo que es posible escribir como regla para las activaciones de una red de l capas la ecuación (2.25)

$$\begin{aligned} Z^{(l+1)} &= W^{T(l)} A^{(l)} \\ A^{(l+1)} &= f(Z^{(l+1)}) \end{aligned} \tag{2.25}$$

En el caso ejemplo, la capa de salida es la tercera, por lo que el vector $A^{(3)}$ es la predicción realizada por nuestro modelo, de manera que en este caso particular $A^{(3)} = Y$ el vector de predicciones, las cuales serán comparadas con las etiquetas de nuestras muestras de entrenamiento para así calcular el error de la red por medio de la función de costo.
De esta manera es posible sintetizar el flujo hacia adelante de la información, a continuación se profundizará en la segunda fase para el aprendizaje de redes neuronales artificiales.

Algoritmo de propagación hacia atrás

El algoritmo de propagación hacia atrás o back propagation fue propuesto por primera vez en la década de los 60 y finalmente cobró relevancia luego del artículo *Learning representations by back-propagating errors* por Rumelhart et al. Este algoritmo propuesto es de gran ayuda para entrenar modelos complejos con un gran número de parámetros, sin embargo, algunas de sus desventajas es que puede ser lento en comparación a otras alternativas, además de que puede converger a minimos locales no óptimos. A pesar de las desventajas que pueda implicar su uso, frente a otras alternativas como el método de diferencias finitas, Algoritmo de Levenberg-Marquardt, etc. BP representa una buena opción debido a tener mayor precisión, requerir menor recurso computacional y ser más rápido.
Para comenzar con es este algoritmo, es necesario definir una función de costo. Una vez que la función escogida haya cuantificado el grado de error del modelo, es necesario comunicar de alguna manera a las capas anteriores que los valores de los parámetros o pesos utilizados para intentar predecir la clase de las muestras de entrenamiento no fueron los mejores. Este método tiene por objetivo el ajustar nuestro modelo de manera en que cada uno de sus parámetros adquieran los valores necesarios para que el error obtenido sea mínimo. Es posible intuir con base en los cursos elementales de cálculo, esto se reduce a un simple problema de minimización de la función de costo, sin embargo, en la práctica, esto resulta altamente no trivial.
Para este tipo de modelos la función de costo, independientemente de la que haya sido escogida, se trata de una superficie que se encuentra en un espacio donde cada una de las dimensiones está asociada a un parámetro de la red, por lo que, para encontrar el punto en el espacio en el cual la función en cuestión toma su valor mínimo habría que derivarla con respecto a cada uno de estos parámetros,no obstante, los modelos de este tipo pueden llegar a tener varios miles de parámetros. Además debido a la complejidad de estas superficies, en la mayoría de los casos resulta imposible encontrar un mínimo global y se debe aspirar a encontrar el menor mínimo local de una región. De esta manera, por medio del gradiente de la función de costo, se puede guiar al modelo para llevarlo hasta el mínimo local más cercano, a esto se le conoce como algoritmo de optimización de gradiente descendente. Ahora que se tiene una idea de cómo se logrará que el modelo pueda mejorar sus predicciones, es momento de abordar el método para lograrlo.

Para simplificar la explicación del algoritmo de BP se continuará con el ejemplo anterior de la red neuronal de la imagen (2.9). Para el proceso de la propagación del error se utilizará un caso más concreto, las especificaciones de la red serán las siguientes: En ella se utilizará la función de costo de entropía cruzada multiclase (Ecuación 2.20), además la capa de salida de la red tendrá la función de activación Softmax (Ecuación 2.16).
Una vez establecidas estas especificaciones es posible realizar la propagación del error.
Para realizar el algoritmo de back propagation se inicia calculando la derivada de la función de activación. Por la regla de la cadena se obtiene el gradiente de la función con respecto a cada uno de los parámetros de la red, por lo que para obtener la derivada con respecto a la segunda matriz de pesos como:

$$\frac{\partial J}{\partial W^{(2)}} = \frac{\partial J}{\partial Z_3}\frac{\partial Z_3}{\partial W^{(2)}} \tag{2.26}$$

Por lo que a continuación se presenta el desarrollo para obtener dicha derivada, se inicia con la derivada la función de activación en la capa de salida para utilizarla más adelante.

Derivada función Softmax

La función softmax, para una sola muestra, puede ser expresada como:

$$Y_i = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}}$$

Debido a que la función softmax es una función vectorial, la derivada completa se puede expresar como su Jacobiano:

$$\frac{\partial Y}{\partial Z} = \begin{pmatrix} \frac{\partial Y_1}{\partial z_1} & \frac{\partial Y_1}{\partial z_2} & \cdots \\ \frac{\partial Y_2}{\partial z_1} & \frac{\partial Y_2}{\partial z_2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{2.27}$$

Por lo que la derivada (2.27) puede reescribirse de la siguiente manera:

$$\frac{\partial Y}{\partial Z} = \frac{\partial Y_i}{\partial z_j} \tag{2.28}$$

Para i = (1,2,$\cdots$, *Número neuronas de salida*) y para j = (1,2,$\cdots$, *Número neuronas de capa anterior*). Es posible observar que se tienen dos posibles casos al calcular esta derivada dependiendo de los valores de i y j:

- **Caso 1:** $i \neq j$
  Se puede sacar el término j-ésimo de la suma del denominador de la función Softmax

$$\frac{\partial Y_i}{\partial z_j} = \frac{\partial}{\partial z_j}\left[\frac{e^{z_i}}{e^{z_j} + \sum_{k\neq j}^{N} e^{z_k}}\right] \tag{2.29}$$

  Siguiendo la regla de la derivada numerador y denominador, se tiene:

$$\frac{\partial}{\partial z_j} e^{z_i} = 0$$

$$\frac{\partial}{\partial z_j}(e^{z_j} + \sum_{k\neq j}^{N} e^{z_k}) = e^{z_j}$$

  Por lo que siguiendo la regla de la derivada de un cociente:

$$\frac{\partial Y_i}{\partial z_j} = -\frac{e^{z_i} e^{z_j}}{(e^{z_j} + \sum_{k\neq j}^{N} e^{z_k})^2} \tag{2.30}$$

  Volviendo a integrar el j-ésimo término a la suma y separándolo como el producto de dos cocientes, se tiene:

$$\frac{\partial Y_i}{\partial z_j} = -\frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \frac{e^{z_j}}{\sum_{k=1}^{N} e^{z_k}}$$

  Desde esta expresión es posible observar que se trata de la salida de la función activación en cada neurona:

$$\frac{\partial Y_i}{\partial z_j} = -Y_i Y_j \tag{2.31}$$

- **Caso 1:** $i = j$
  Es posible sacar el término i-ésimo de la suma del denominador de la función Softmax

$$\frac{\partial Y_i}{\partial z_i} = \frac{\partial}{\partial z_i}\left[\frac{e^{z_i}}{e^{z_i} + \sum_{k\neq i}^{N} e^{z_k}}\right] \tag{2.32}$$

  Procediendo de manera similar, se pueden calcular las derivadas del cociente:

$$\frac{\partial}{\partial z_i} e^{z_i} = e^{z_i}$$

$$\frac{\partial}{\partial z_i}(e^{z_i} + \sum_{k \neq i}^{N} e^{z_k}) = e^{z_i}$$

Ahora usando la regla de la derivada de un cociente

$$\frac{\partial Y_i}{\partial z_i} = \frac{(e^{z_i} + \sum_{k \neq i}^{N} e^{z_k})e^{z_i} - e^{z_i}(e^{z_i})}{(e^{z_i} + \sum_{k \neq i}^{N} e^{z_k})^2} \tag{2.33}$$

Reduciendo términos semejantes y nuevamente reincorporando término i-ésimo a la suma y descomponiendo como el producto de dos cocientes.

$$\frac{\partial Y_i}{\partial z_i} = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \frac{\sum_{k \neq i}^{N} e^{z_k}}{\sum_{k=1}^{N} e^{z_k}} \tag{2.34}$$

Es útil recordar que todas las probabilidades de salida de la función softmax suman 1, por lo que si a la suma en el numerador del segundo cociente le falta el término correspondiente a la probabilidad de salida de la i-ésima neurona, entonces se puede reescribir como $(1 - Y_i)$

$$\frac{\partial Y_i}{\partial z_i} = (1 - Y_i) \tag{2.35}$$

Una vez que hemos calculado la derivada de la función de activación de la capa de salida, ahora se debe calcular la derivada de la función de costo.

Derivada de la función de costo

Partiendo de la función de costo.

$$J(W) = -\frac{1}{m} \sum_{i=1}^{M} \sum_{j=1}^{N} \hat{Y}_j^{(i)} log(Y_j^{(i)})$$

Donde m es el número de muestras que evaluará la función de costo y N es el número de neuronas en la capa de salida de la red y además:

$$Y^{(i)} = f(Z^{(3)})$$

Además f es la función de activación de la tercera capa, es decir la función Softmax.
Para simplificar el proceso de la derivación de esta función se puede escribir su derivada como:

$$\frac{\partial J}{\partial z_k} = -\frac{1}{m} \sum_{i=1}^{M} \frac{\partial}{\partial z_k} \left[\hat{Y}_k^{(i)} log(Y_k^{(i)}) + \sum_{j \neq k}^{N} \hat{Y}_j^{(i)} log(Y_j^{(i)}) \right] \tag{2.36}$$

$$\frac{\partial J}{\partial z_k} = -\frac{1}{m} \sum_{i=1}^{M} \left[\frac{\hat{Y}_k^{(i)}}{Y_k^{(i)}} \frac{\partial}{\partial z_k} Y_k^{(i)} + \sum_{j \neq k}^{N} \frac{\hat{Y}_j^{(i)}}{Y_j^{(i)}} \frac{\partial}{\partial z_k} Y_j^{(i)} \right]$$

Dados los resultados de la derivada de la función Softmax de las ecuaciones (2.31) y (2.35), sustituyendo en la expresión anterior, se obtiene:

$$\frac{\partial J}{\partial z_k} = \frac{1}{m} \sum_{i=1}^{M} \left[-\frac{\hat{Y}_k^{(i)}}{Y_k^{(i)}} Y_k^{(i)}(1 - Y_k^{(i)}) - \sum_{j \neq k}^{N} \frac{\hat{Y}_j^{(i)}}{Y_j^{(i)}} (-Y_j^{(i)} Y_k^{(i)}) \right]$$

Reduciendo términos semejantes, factorizando los $Y_k^{(i)}$ de la suma y el otro término:

$$\frac{\partial J}{\partial z_k} = \frac{1}{m}\sum_{i=1}^{M}\left[-\hat{Y}_k^{(i)} + Y_k^{(i)}\left(\hat{Y}_k^{(i)} + \sum_{j\neq k}^{N}\hat{Y}_j^{(i)}\right)\right]$$

Por lo que los términos en el paréntesis pueden sumarse para tener una suma completa de cada una de las entradas del vector de etiquetas para cada la muestra en turno. Cabe destacar que la suma de dichas entradas es 1 ya que la muestra pertenece a una sola clase.

$$\frac{\partial J}{\partial z_k} = \frac{1}{m}\sum_{i=1}^{M}\left(Y_k^{(i)} - \hat{Y}_k^{(i)}\right)$$

Finalmente, esta expresión corresponde a una sola neurona, por lo que cubrir el vector de salida de todas las neuronas, la expresión quedaría:

$$\frac{\partial J}{\partial Z^{(3)}} = \frac{1}{m}\sum_{i=1}^{M}\left(Y^{(i)} - \hat{Y}^{(i)}\right) \tag{2.37}$$

Y finalmente calculando el tercer factor de la derivada, recordando $Z^{(3)} = W^{(2)T}A_i^{*(2)}$, donde el asterisco en el superíndice denota que el vector de activaciones tiene concatenado el 1 del sesgo de la segunda capa de la red, por lo que la derivada es:

$$\frac{\partial Z^{(3)}}{\partial W^{(2)}} = A_i^{*(2)} \tag{2.38}$$

Ya que es la derivada de un conjunto de polinomios. Por lo que al sustituir las ecuaciones (2.37) y (2.38) en la expresión (2.26), después de reducir términos semejantes se obtiene que la derivada de la función de costo con respecto a la segunda matriz de pesos es:

$$\frac{\partial J}{\partial W^{(2)}} = \frac{1}{m}\sum_{i=1}^{M}\left(Y^{(i)} - \hat{Y}^{(i)}\right)A_i^{*(2)}$$

Donde $A^{*(2)}$ es el vector de activaciones de la capa 2 y el subíndice indica que se trata de la i-ésima muestra. Finalmente, se define la diferencia $(Y^{(i)} - \hat{Y}^{(i)}) \equiv \delta^{(3)}$ como el error en la capa 3 para la i-ésima muestra, por lo tanto, la derivada es:

$$\frac{\partial J}{\partial W^{(2)}} = \frac{1}{m}\sum_{i=1}^{m}\left[\delta^{(3)}A_i^{*(2)}\right] \tag{2.39}$$

Es decir, la derivada de la capa 3 depende directamente del error de salida de la red. Cabe destacar que $\delta^{(3)}$ es un vector que contiene los errores de cada una de las neuronas de salida.
Este solo fue el desarrollo para calcular el valor de la derivada de la función de costo con respecto a la segunda matriz de pesos, por lo que aún hace falta calcular la derivada con respecto a la primera matriz de pesos. Es posible observar que se puede obtener de manera similar una expresión como la ecuación (2.26).

$$\frac{\partial J}{\partial W^{(1)}} = \frac{\partial J}{\partial Z_3}\frac{\partial Z^{(3)}}{\partial W^{(1)}} \tag{2.40}$$

Donde $Z^{(3)} = W^{(2)T}A^{*(2)}$ y a su vez $A^{*(2)} = f(W^{(1)T}A^{*(1)})$ donde f es la función de activación de la capa 2, la cual se ha definido como la función ReLu, la cual se define por partes como:

$$\frac{\partial}{\partial x}ReLu(x) = \begin{cases} 0 & si \quad x \leq 0 \\ \\ 1 & si \quad x < 0 \end{cases} \tag{2.41}$$

Además es posible observar que los primeros dos factores de la derivada son los mismos que para la derivada anterior, por lo que sólo habrá que calcular el segundo:

$$\frac{\partial Z^{(3)}}{\partial W^{(1)}} = W^{(2)T} \frac{\partial}{\partial W^{(1)}} ReLu(W^{(1)T} A_i^{*(1)})$$

Suponiendo que el argumento de la función ReLu es mayor a cero, por lo que el resultado de esta derivada es:

$$\frac{\partial Z^{(3)}}{\partial W^{(1)}} = W^{(2)T} A_i^{*(1)} \tag{2.42}$$

Por lo que al sustituir las derivadas correspondientes en la ecuación (2.40) se obtiene la expresión:

$$\frac{\partial J}{\partial W^{(1)}} = \frac{1}{m} \sum_{i=1}^{M} \left(Y^{(i)} - \hat{Y}^{(i)} W^{(2)T} A_i^{*(1)} \right) \tag{2.43}$$

Sin embargo, se puede observar que la diferencia se trata del error de la capa 3, ahora se definirá el error de la capa dos como $\delta^{(2)} \equiv \delta^{(3)} W^{(2)T}$, por lo que finalmente

$$\frac{\partial J}{\partial W^{(1)}} = \frac{1}{m} \sum_{i=1}^{m} \left(\delta^{(2)} A_i^{*(1)} \right) \tag{2.44}$$

Si la red contara con más capas sería más notorio que las derivadas de la función de costo tiene la misma tendencia a lo largo de las capas, por lo que es posible definir como regla para el error en la propagación hacia atrás lo siguiente:

$$\delta^{(l)} = \delta^{(l+1)} W^{(l)} \tag{2.45}$$

Cabe destacar que la forma encontrada para cada una de las derivadas de la función del costo depende directamente de función utilizada, así como las funciones de activación usadas en cada capa de la red. Sin embargo, para una configuración de red dada, es posible encontrar patrones similares al caso del ejemplo tratado.
Pareciera que con calcular las derivadas de nuestra función de costo es suficiente para que la red pueda aprender, aunque también hace falta actualizar los parámetros del modelo para que, en cada iteración del proceso antes mencionado, éstos puedan tomar valores que nos lleven a mejores resultados con menor error.

2.3.4. Algoritmos de optimización

Una vez calculado el gradiente de la función de costo con respecto a cada uno de los parámetros entrenables del modelo, es necesario actualizarlos de manera que al utilizar estos nuevos valores, el error de las predicciones sea menor.

Gradiente descendente

El proceso de la actualización de los pesos de cada una de las capas, en el algoritmo de gradiente descendente, se lleva a cabo por medio de la siguiente regla:

$$\Delta W^{(l)} = -\alpha \frac{\partial J}{\partial W^{(l)}} \tag{2.46}$$

Donde α es la constante llamada constante de aprendizaje o *learning rate* de la red, esta constante determina el tamaño del paso que el gradiente dará en dirección del mínimo local más cercano.
Recordando la breve explicación al inicio de esta sección, sobre la superficie que representa la función de costo en el espacio de parámetros, es fácil visualizar este proceso de minimización

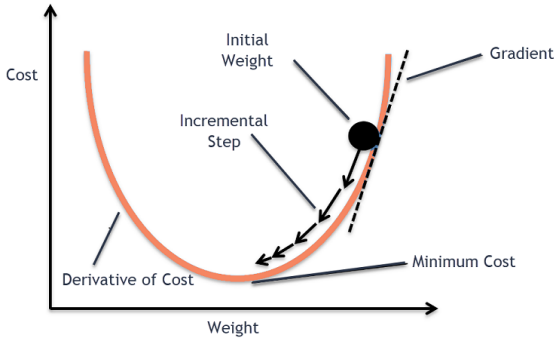

Figura 2.10: **Algoritmo de gradiente descendente**. En esta figura se muestra un ejemplo sencillo del funcionamiento del algoritmo de gradiente descendente y cómo la actualización del peso guía a la función a converger a un valor mínimo de costo.

con un pequeño ejemplo. Suponiendo que se tiene una función de costo que depende de una sola variable, con la forma que se muestra en la figura (2.10) [8]. Partiendo del primer valor que toma este único parámetro, al que se le llamará w_0 y calculando el gradiente de la función de costo en ese punto, se obtiene la tangente de la función que indicará la dirección en la que se encuentra el mínimo local más cercano. Para acercarlo a él, se tomará el valor de w_0 y se le restará el gradiente de la función multiplicado por la constante de aprendizaje. De manera en que, si el gradiente es negativo, se moverá hacia la derecha y en caso contrario se moverá a la izquierda. De igual forma, entre menor sea el gradiente, los pasos con los que se moverá serán más pequeños para permitir una convergencia más suave. De esta manera es que el modelo puede converger en un mínimo local e ir corrigiendo los valores de los parámetros para así reducir su error.

Algoritmo de momento adaptativo

A diferencia del algoritmo de gradiente descendente que trabaja con una tasa de aprendizaje fija para todos los parámetros, el algoritmo Adam o adaptative momentum adapta el ritmo de aprendizaje de cada parámetro con base en el valor de su gradiente y un nuevo término llamado momento. Estas dos a diferencias permiten que la función de costo converja a un valor mínimo más rápido, con más estabilidad y con posibilidad de obtener menor error. La regla de actualización de parámetros en su forma más básica es:

$$W_{t+1}^{(l)} = W_{t+1}^{(l)} - \left(\frac{\alpha m_t}{\sqrt{v_t + \epsilon}} \right) \tag{2.47}$$

Donde los términos v_t y m_t se expresan como:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla J \tag{2.48}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)(\nabla J)^2 \tag{2.49}$$

Donde el término m_t es la parte correspondiente al momentum, en el cual se puede pensar como un análogo del momentum físico de mecánica, el cual aumenta según el gradiente acelerándolo y ayudando a evitar estados metaestables, asegurando que la función converja al mínimo local óptimo. Por otro lado, el término v_t es el encargado de regular la tasa de aprendizaje de cada parámetro, mejorando la estabilidad de la convergencia. Como podemos observar, esta regla de actualización depende de la historia del gradiente para cada iteración del modelo [23].

[8] Gradiente descendente. Recuperado de https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/ (18 de marzo de 2024)

Estos algoritmos representan una gran herramienta para tareas de clasificación y regresión con cualquier tipo de modelos, sin importar la cantidad de datos y características que estos tengan; sin embargo, es importante señalar que aún se encuentran limitados por el poder computacional disponible, ya que para redes profundas o con un gran número de capas ocultas, el recurso computacional necesario irá creciendo. Además, debido a su forma de procesar la información, este tipo de redes neuronales no son el mejor método de análisis para imágenes ya que, principalmente, no hacen uso de la valiosa información espacial que éstas contienen. Es por esto que se desarrollaron otros modelos con particularidades que permitan un mejor aprovechamiento de las propiedades que tienen las imágenes, estos modelos son las redes neuronales convolucionales (RNCs) de las cuales se abordarán más a fondo en la siguiente sección.

2.3.5. Redes neuronales convolucionales

Al igual que sucedió con el modelo del perceptrón, las redes neuronales convolucionales poseen un gran número de virtudes y ventajas sobre otros modelos. Sin embargo, al desarrollar herramientas tan sofisticadas como estas, la complejidad de los problemas a los que se aplican también ha crecido, por lo que estos instrumentos por sí solos no son suficientes para afrontar los nuevos retos. Es por esto que se desarrollaron las redes neuronales convolucionales, las cuales están basadas en el funcionamiento de la corteza visual de los animales.
El desarrollo de este modelo matemático comenzó en 1958 cuando Hubel y Weiesel [24] al momento de estudiar la relación entre la pupila y la corteza cerebral de un gato, en este experimento mostraban distintas formas en diferentes ángulos y con diferentes niveles de brillo. Este experimento llevó al descubrimiento de lo que llamaron la célula selectiva de dirección, la cual es capaz de identificar los bordes de una figura plana o con distintos ángulos. Más tarde, en 1984, en Japón [25] se propuso un modelo neurocognitivo basado en el concepto de campo receptivo, el cual fue nombrado Neocognitrón. Éste fue creado para suplir las deficiencias que tenían otros modelos, como la baja capacidad para reconocer patrones en situaciones de cambio de posición de algún objeto o distorción de imágen; además de identificar patrones basados en la similaridad geométrica. El funcionamiento del neocognitrón consiste en descomponer una imagen en sub características para así procesarlas, esto por medio de sus dos neuronas llamadas elemento S que es el encargado de extraer las características de la imagen y elemento C de antideformación. En el elemento S intervienen dos parámetros: El campo receptivo y el parámetro del umbral. El primero se encarga de determinar el número de conexiones de entrada, y el segundo determina el grado de reacción de las subcaracterísticas. Un ejemplo del funcionamiento conjunto de esto, es que al observar un objeto en movimiento, el elemento S sólo se encargará de extraer las características más relevantes del objeto y evitará cualquier interferencia visual; mientras que el elemento C escoge las características sensoriales correspondientes al campo receptivo. Finalmente, en 1989, Yann LeCun creó la primer red neuronal convolucional llamada LeNet, para el reconocimiento de letras escritas a mano.

Este tipo de modelo ofrece una gran variedad de ventajas para el análisis de imágenes con respecto a las capacidades de otras alternativas, como las máquinas de soporte vectorial, redes neuronales profundas, etc [26]. Entre las ventajas más destacables que ofrecen estan:

- La estructura de las RNCs está especializada en el aprovechamiento de la información espacial que contiene la imagen.
- Son capaces de reconocer objetos incluso después de ser rotados, esto es gracias a que pueden reconocer bordes de objetos.
- Gracias a la forma en la que este tipo de redes neuronales extraen y procesan la información, es posible condensarla reduciendo el número de características de la imagen original.

Además de poder ser usadas para tareas como regresión, clasificación, es posible especializarlas en tareas de detección de objetos y segmentación [27].

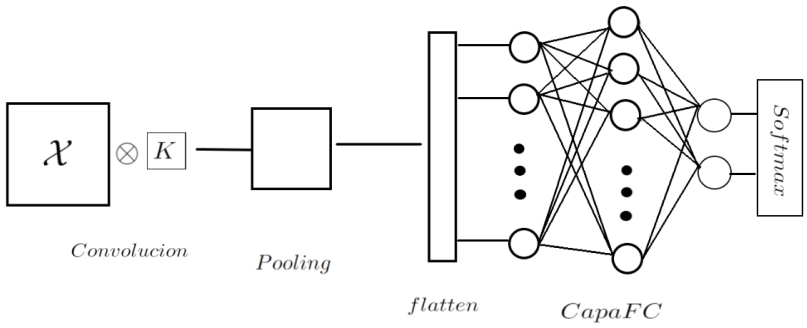

Figura 2.11: **RNC ejemplo**. Esta será la RNC de referencia para el ejemplo de esta sección. Está formada por una capa de convolución con un solo filtro, una capa de pooling, una capa flatten y una capa FC, la cual es igual al perceptrón multicapa ejemplo de la sección anterior.

La estructura básica de una RNC está formada por los elementos que se muestran en la figura (2.14), en ella es posible identificar los siguientes componentes:

Capa de convolución

Las imágenes no son más que un tensor tres dimensiones: Largo, ancho y número de canales. Donde este último valor depende si se trata de una imagen en escala de grises o una imagen RGB respectivamente. Estos tensores poseen distintos valores de intensidad en cada una de sus entradas, haciendo que en conjunto representen información visual. En este contexto se les llamará *mapas de características* al conjunto de matrices que forma parte de estos tensores.

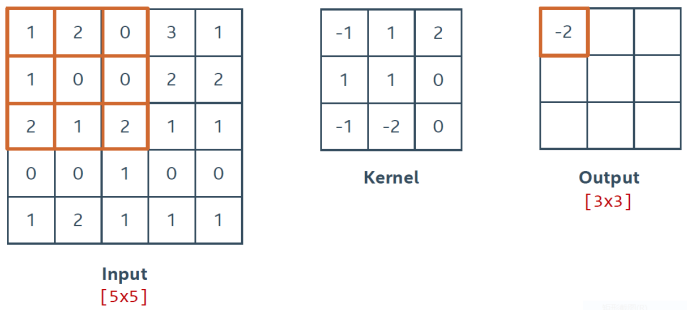

Figura 2.12: **Convolución**. El proceso de convolución consiste en la operación de un filtro sobre un mapa de características realizando un producto punto de sus entradas con las del mapa de entrada y cuyo resultado será asignado a una entrada del mapa de características de salida con un menor tamaño.

Es por esto que en la capa de convolución se busca de extraer información de estos valores de intensidad, para así obtener representaciones de la imagen de entrada en distintos niveles de abstracción. La condensación de dicha información se lleva a cabo gracias a la convolución de filtros

o kernels, a lo largo y ancho de cada uno de los mapas de características de entrada. Sin embargo, al operar sobre el mapa de características éste irá reduciendo su tamaño; por lo que es relevante el cuidar las dimensiones de los mapas de características a lo largo de este proceso (Véase figura 2.12) [9]. Esto es posible gracias a herramientas como el *stride* y el *padding*.

- **Stride**: El valor de Stride indica el tamaño del paso con el que el kernel se moverá sobre los mapas de características. El stride escogido puede ser horizontal, vertical o incluso una combinación de ambos, según las necesidades del problema en cuestión. La importancia de este valor radica en que según la configuración de stride escogida, esta puede afectar las dimensiones de la salida de la capa de convolución, llegando incluso a disminuir la carga computacional del modelo. Además el valor de stride utilizado afecta directamente el campo de visión de la red, esto debido a que, al escoger un valor mayor nuestro modelo tomará en cuenta las características globales de la imagen en lugar de características sutiles, perdiendo información potencialmente útil.
- **Padding**: Como se ha mencionado, si no se usa cuidadosamente la operación de convolución, es posible que llegar a condensar la información de entrada de manera en que resulte contraproducente para su análisis, ya que en cada capa de convolución, la imagen de entrada irá reduciendo su tamaño y en consecuencia puede ir perder información relevante. Además en modelos con varias capas de convolución, esto puede provocar que nuestra imagen reduzca su tamaño considerablemente. Es por esto que se crea el padding, que permite mantener un tamaño deseado en mapas de características de salida. Este valor señala la cantidad de filas y columnas de ceros que rodearán a la imagen de entrada, para que de esta manera el mapa de características de salida pueda conservar su tamaño original o tener las dimensiones requeridas por el modelo. A la hora de operar sobre la imagen, los filtros suelen operar un menor número de veces los pixeles ubicados en las orillas, lo cual provoca la pérdida de esta información. Por lo que, además de llevar un control sobre las dimensiones de los mapas de características, el padding puede ayudar a que los pixeles de los bordes queden reubicados en una zona más alejada de la orilla, para así operar sobre ellos el mismo número de veces que en los pixeles centrales.
  Al hablar de esta herramienta, es importante mencionar el tipo de padding que se puede utilizar y cuál será su consecuencia en el mapa de características de salida:
  - Padding valid:
    Este tipo de padding no agrega filas y columnas de ceros alrededor de nuestra imagen, por lo que el mapa de características obtenido será de menor tamaño que la entrada. Este padding es especialmente útil para los casos en el que la imagen de entrada sea lo suficientemente grande como para que la reducción de tamaño no cause algún problema, así como los casos en los que se planea reducir la carga computacional del modelo.
  - Padding same:
    Como lo dice su nombre, este tipo de padding agrega el número de filas y columnas de ceros necesarias, para que el tamaño del mapa de características sea igual al de la imagen de entrada. Se recomienda utilizar este tipo de padding cuando es importante conservar la información espacial del la imagen original.
  - Padding full:
    Finalmente, este tipo de padding agrega el número de filas y columnas de ceros necesarias para que las dimensiones del mapa de características sea mayor al de entrada. Esto es importante cuando se requiere conservar la información espacial de la imagen, y cuando se busca que la salida de la convolución tenga dimensiones específicas. Sin embargo, esto puede llegar a tener consecuencias negativas en el desempeño de nuestro modelo ya que aumenta la carga computacional y puede introducir ruido artificial a los mapas de características.

[9] Convolución. Recuperado de https://programmerclick.com/article/2605330897/ (18 de marzo de 2024) 31

Es importante escoger cuidadosamente el tipo de padding a utilizar, ya que si se utiliza en exceso puede afectar negativamente el desempeño de la red [28].

Al variar estos dos valores se puede cambiar el tamaño del mapa de características resultante de nuestra capa de convolución, en la ecuación (2.50) se puede observar esta relación para poder determinar las dimensiones de salida.

$$D_{Salida} = \frac{(D_{Entrada} - K + 2 * P)}{S} + 1 \tag{2.50}$$

Donde D_{Salida} es el tamaño de salida del largo o ancho en cuestión.
$D_{Entrada}$ es el tamaño original del largo o ancho en cuestión.
K es el tamaño del kernel utilizado.
P es la cantidad de columnas o filas de ceros agregadas al mapa de características de entrada.
S representa el valor de stride escogido para la convolución.
Es importante mencionar que el número de mapas de características que se obtendrán a la salida de esta capa es igual al número de filtros usados, es decir, un mapa de característica por cada filtro operado sobre la entrada de la capa de convolución.

Capa de submuestreo o Pooling

Las capas submuestreo o pooling, como también se les conoce, son generalmente usadas luego de una capa de convolución. Sirven principalmente para reducir el tamaño de los mapas de características, y así disminuir la carga computacional del modelo mientras conservan las características más representativas. La forma en que funciona esta capa es la siguiente: Luego de definir el tamaño de la ventana que se utilizará, la cual es equivalente a un filtro de la capa de convolución con un valor dado de stride y no ocupará padding. Esta ventana se moverá a lo largo y ancho de los mapas de características operando sobre los mismos, la operación realizada depende del tipo de submuestreo utilizado, que su vez depende de los propósitos del modelo en cuestión; entre los tipos de pooling usados frecuentemente están los siguientes:

- **Max Pooling**
  Este tipo de submuestreo tomará las entradas del mapa de características dentro de la ventana, y determinará el valor máximo de ellas; luego lo asignará a la entrada correspondiente en el mapa de características de salida. Además es necesario guardar la ubicación del valor máximo de la ventana cada vez que el filtro opere sobre el mapa de entrada, este paso es necesario para cuando se realice la propagación hacia atrás del error. Esta técnica es especialmente útil para identificar rasgos importantes en la imagen como bordes, lo cual es importante en tareas de reconocimiento de objetos. Además, esto lo hace invariante ante pequeños desplazamientos[29] , ya que los valores máximos seguirán siendo los mismos. Sin embargo, por su naturaleza, es sensible al ruido en los valores de los mapas de características [30], y puede provocar pérdida de información espacial como relaciones entre los valores cercanos de las matrices.

- **Mean Pooling**
  En este caso, para disminuir el número de características, se utilizarán los valores dentro de la ventana para calcular la media de estos, y asignarlo a la entrada correspondiente del mapa de características de salida. Esta estrategia permite conservar mejor la información espacial de entrada y detectar patrones globales, además de reducir el ruido de entrada. No obstante, esta misma ventaja puede hacer que se pase por alto información relevante como los bordes de objetos.

A pesar de ser las dos técnicas de submuestreo más utilizadas en las RNCs del estado del arte, es posible definir otras operaciones para reducir el número de características según convenga para los propósitos del modelo.

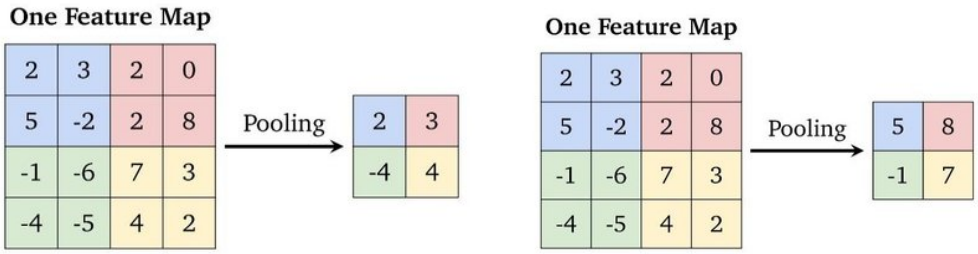

Figura 2.13: **Técnicas de pooling**. Se muestran dos técnicas comunes de pooling, en ambas se ocupan un valor de stride de 2 con una ventana de 2×2. Mean pooling (Arriba), Max pooling (Abajo)

Al igual que en la capa de convolución, los valores de stride y el tamaño de la ventana utilizada, poseen relevancia en cuanto los resultados del submuestreo realizado (Véase figura 2.13) [10]. En el caso del stride puede lograr que la ventana se sobreponga en varias regiones del mapa de características, provocando que sea capaz de transmitir información más relevante [31] y el tamaño de la ventana determinará la cantidad de información que pueda llegar a omitir al condensar en el submuestreo.

Capa completamente conectada

Finalmente, luego del proceso de extracción de características por parte de las capas de convolución y pooling, los valores en el tensor de mapas de características resultante se reorganizarán en forma de vector, el cual actuará como el vector de características con el que trabajará la red completamente conectada (FC por sus siglas en inglés) descrita en la sección de perceptrón multicapa, y finalmente dar como resultado la clasificación de la imagen de entrada.

Ahora que se conoce más a fondo el cómo están conformadas las redes neuronales convolucionales, es momento de ahondar en la forma en que este tipo de redes aprende y clasifica con base en las imágenes que se le presentan.

2.3.6. Propagación hacia adelante RNCs

Como ya se ha discutido superficialmente en la sección de capa de convolución, el tipo de información de entrada para las RNCs son imágenes, las cuales pueden ser en escala de grises o a color. En el primer caso, estas imágenes pueden ser representadas por una simple matriz cuyas entradas son los valores de intensidad de cada pixel. En el caso de las imágenes RGB, se trata de un tensor que alberga tres matrices cuyas entradas son los valores de intensidad de cada uno de los colores, de manera que al juntarlas generan la imagen a color. Para poder simplificar la explicación del funcionamiento de una RNC, supondremos una imagen en escala de grises como entrada a nuestro modelo de ejemplo el cual se puede observar en la imagen (2.14). Esta red consta de una capa de convolución, una de pooling y en la capa completamente conectada tenemos a la red ejemplo de la sección de perceptrón multicapa, ya que se ha comprendido cómo funciona, con la diferencia de que la capa de entrada tendrá un mayor número de elementos.
Sea $\mathcal{X}$ la imagen original de tamaño 28×28 en escala de grises, la cual ingresa a la capa de convolución. Esta capa cuenta con filtro K de tamaño 3×3 y un sesgo que, al igual que en la capa completamente conectada, nos ayudará a darle más flexibilidad al modelo. En este ejemplo se considerará un padding del tipo *valid*, un valor de stride de 1 y la capa de convolución tendrá

[10] Técnicas de Pooling. Recuperado de https://blog.naver.com/kkang9901/221776454163 (18 de marzo de 2024)

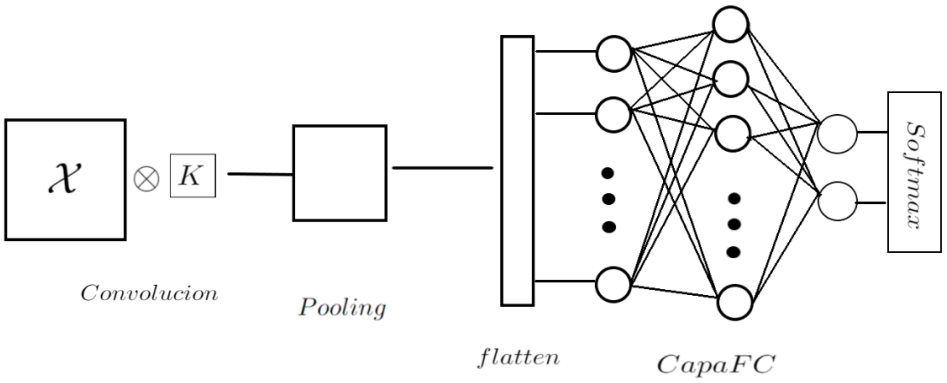

Figura 2.14: **RNC ejemplo**. La arquitectura de la red ejemplo de esta sección consiste en una capa de convolución con un solo filtro, una capa de pooling, una capa flatten y finalmente una capa completamente conectada igual al perceptrón multicapa ejemplo.

como función de activación a la función ReLu. Por lo que la convolución del filtro en la imagen de entrada se puede expresar como:

$$\mathcal{X} \otimes K = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,28} \\ x_{2,1} & x_{22} & \cdots & x_{2,28} \\ \vdots & \vdots & \ddots & \vdots \\ x_{28,1} & \cdots & \cdots & x_{28,28} \end{bmatrix} \otimes \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} + b$$

Donde los x_{ij} y los k_{ij} son los valores de las intensidades en escala de grises y las entradas del filtro respectivamente, finalmente, el b corresponde al bias o sesgo de la capa de convolución. Es importante señalar que estos k_{ij} serán los responsables de extraer las características de la imagen de entrada, por lo cual a lo largo del aprendizaje de nuestro modelo, estos pesos se irán actualizando con el fin de encontrar los valores que mejor ayuden a extraer características útiles para lograr obtener mejores resultados.
Con base en la ecuación (2.50), el mapa de características que obtendremos después de esta convolución será de dimensiones 26×26. Desarrollando cada una de las operaciones que el filtro realiza sobre la imagen, se obtendrá un total de 676 ecuaciones, correspondientes al resultado de cada paso del kernel.

$$\mathcal{X} \otimes K = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,26} \\ z_{2,1} & z_{2,2} & \cdots & z_{2,26} \\ \vdots & \vdots & \ddots & \vdots \\ z_{26,1} & \cdots & \cdots & z_{26,26} \end{bmatrix}$$

donde cada uno de los z_{ij} son las entradas de la matriz Z^* ya que incluye el sesgo de esta capa.

Estos a su vez pueden escribirse como a continuación:

$$\begin{aligned}
z_{1,1} &= k_{11}x_{1,1} + k_{12}x_{1,2} + k_{13}x_{1,3} + k_{21}x_{2,1} + k_{22}x_{2,2} + k_{23}x_{2,3} + k_{31}x_{3,1} + k_{32}x_{3,2} + k_{33}x_{3,3} + b \\
z_{1,2} &= k_{11}x_{1,2} + k_{12}x_{1,3} + k_{13}x_{1,4} + k_{21}x_{2,2} + k_{22}x_{2,3} + k_{23}x_{2,4} + k_{31}x_{3,2} + k_{32}x_{3,3} + k_{33}x_{3,4} + b \\
&\vdots \\
z_{26,26} &= k_{11}x_{24,24} + k_{12}x_{24,25} + k_{13}x_{24,26} + k_{21}x_{25,24} + k_{22}x_{25,25} + k_{23}x_{25,26} + k_{31}x_{26,24} + k_{32}x_{26,25} \\
&\quad + k_{33}x_{26,26} + b
\end{aligned} \tag{2.51}$$

En este ejemplo se escogió ReLu como la función de activación f de la capa de convolución, por lo que para este caso $\mathcal{A}^{(0)*} = ReLu(Z^{(0)*})$. La salida de la capa de convolución es un tensor con una dimensión extra al de entrada, esta cuarta dimensión corresponde al número de filtros que operaron sobre sobre el mapa de entrada, por lo que, para generalizar lo mencionado:

$$f(\mathcal{X} \otimes K) = \mathcal{A}^{(0)} \tag{2.52}$$

Una vez obtenida la activación de la capa de convolución continuará a la capa de submuestreo, como se mencionó en la sección de capa de pooling, la forma de realizar el submuestreo depende de la técnica escogida, en este caso se trata de la técnica de max pooling con una ventana de 2×2 y un stride de 2. Por lo que nuevamente usando la ecuación (2.50), se puede calcular que el tamaño del mapa de características luego del submuestreo será de 13×13. Este nuevo mapa contendrá las características más relevantes del mapa convolucionado.

$$\mathcal{A}^{(1)} = Pooling(\mathcal{A}^{(0)}) \tag{2.53}$$

Luego de la capa de pooling sigue la capa completamente conectada, sin embargo, para poder ingresar la información a esta última parte, es necesario que realizar un aplanamiento de los mapas de características, esto consiste reorganizar las entradas de un tensor a un vector que será la entrada de la capa FC. Esto suele realizarse tomando cada una de las filas de los mapas de características y concatenándolos uno tras otro. No obstante, al igual que varias de las técnicas dentro de esta área, es posible definir alguna técnica que nos pueda ser de ayuda con respecto a los objetivos de nuestro modelo.

$$\mathcal{A}^{(1)}_{Flatten} = \left(a^1_1, a^1_2, \cdots, a^1_{169}\right) \tag{2.54}$$

Por lo que finalmente, este vector entrará a la capa completamente conectada con un proceso de similar al descrito anteriormente.

2.3.7. Propagación hacia atrás RNCs

Una vez que se obtiene la retroalimentación al comparar la salida del modelo y la etiqueta de la muestra de entrada, es necesario propagar el error hacia atrás para ir actualizando los parámetros de la red. Iniciando con la capa completamente conectada es posible calcular el gradiente de la función de costo para cada matriz de pesos.
Una vez que calculado el gradiente de la función de costo hasta la primera matriz de pesos como ya se ha realizado antes, se tienen los resultados de la ecuaciones (2.44) y (2.39).
Luego de esto, es posible calcular el gradiente de la función de costo con respecto al kernel y el sesgo de la capa de convolución, por lo cual se puede escribir la regla de la cadena de la siguiente manera:

$$\frac{\partial J}{\partial A^{*(1)}} = \frac{\partial J}{\partial Z^{(3)}} \frac{\partial Z^{(3)}}{\partial A^{*(2)}} \frac{\partial A^{(2)}}{\partial Z^{(2)}} \frac{\partial Z^{(2)}}{\partial A^{*(1)}} \tag{2.55}$$

En la ecuación (2.55) se han escrito las derivadas **necesarias hasta la entrada de la capa FC**. Por lo que se desarrollarán cada una de las derivadas hasta este punto. De la ecuación (2.37) se

tiene el resultado de la primera derivada, por lo que la siguientes es:

$$\begin{aligned}\frac{\partial Z^{(3)}}{\partial A^{*(2)}} &= \frac{\partial}{\partial A^{*(2)}}(W^{(2)}A^{*(2)}) \\ &= W^{(2)}\end{aligned} \tag{2.56}$$

Recordando que $A^{(2)} = ReLu(Z^{(2)})$, de la ecuación (2.41) se obtiene la derivada como una función a trozos, en consecuencia, se utilizará el caso en el que $Z^{(2)} > 0$, por lo que la siguiente derivada es:

$$\begin{aligned}\frac{\partial A^{(2)}}{\partial Z^{(2)}} &= \frac{\partial}{\partial Z^{(2)}} ReLu(Z^{(2)}) \\ &= 1\end{aligned} \tag{2.57}$$

Y finalmente, la última derivada dentro de la capa FC es:

$$\begin{aligned}\frac{\partial Z^{(2)}}{\partial A^{*(1)}} &= \frac{\partial}{\partial A^{*(1)}}(W^{(1)}A^{*(1)}) \\ &= W^{(1)}\end{aligned} \tag{2.58}$$

Por lo tanto, usando la ecuación (2.45), el valor de la derivada hasta la entrada de la capa FC para una sola muestra es:

$$\begin{aligned}\frac{\partial J}{\partial K}_{FC} &= (Y - \hat{Y})W^{(2)}W^{(1)} \\ &= \delta^{(3)}W^{(2)}W^{(1)} \\ &= \delta^{(2)}W^{(1)} \\ &= \delta^{(1)}\end{aligned} \tag{2.59}$$

Como se puede observar en la ecuación anterior, la derivada de la función de costo hasta la entrada de la capa completamente conectada es igual al error en la capa de entrada. Cuando se tenía una red densa este paso no era necesario, ya que carecía de sentido calcular el error en la información de entrada, sin embargo; debido a que ahora esta información proviene de la convolución y pooling de la primera parte de la red, es necesario propagar el error a dichas secciones.
Nótese que el error en la capa de entrada debe ser un vector con el mismo tamaño que el vector de entrada a esta capa y, de acuerdo con la figura (2.14), la matriz de pesos $W^{(1)}$ tiene dimensiones 170×5 mientras que el vector de error $\delta^{(2)}$ tiene dimensiones 5×1, por lo que al realizar esta operación no se obtendría un vector de las dimensiones necesarias para propagarlo a la capa de pooling. Esto se debe que la matriz $W^{(1)}$ contiene también los pesos asociados al sesgo de la capa de entrada, esta información al no provenir de la capa anterior no es necesario propagarla, por lo que en este paso se omitirá la fila correspondiente al sesgo y obteniendo así un vector δ^1 con las dimensiones correctas 169×1. A continuación se reorganizará este vector en un tensor con las misma dimensiones del mapa de características que se aplanó en la propagación hacia adelante.

$$\delta^{(1)} \longrightarrow \delta^{(1)}_{Reshape} \tag{2.60}$$

Para continuar con la propagación a través de la capa de pooling, es necesario considerar el tipo de submuestro que se realizó, el tamaño de la ventana utilizada y el stride, para que de esta manera sea posible aumentar el número de características para regresar a las dimensiones del tensor que entró a la capa de pooling (Ver figura 2.15) [11].

[11] BP para max pooling. Recuperado de https://mukulrathi.com/demystifying-deep-learning/conv-net-backpropagation-maths-intuition-derivation/
BP para mean pooing. Recuperado de https://medium.com/dejunhuang/learning-day-55-back-propagation-in-cnn-pooling-layer-c980e76780a7 (18 de marzo de 2024)

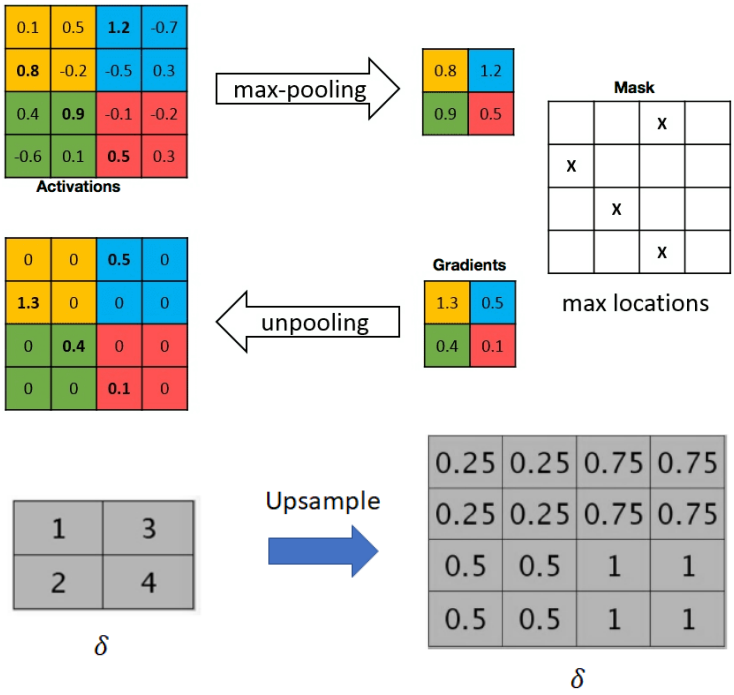

Figura 2.15: **Back propagation para técnicas de pooling**. A la hora de propagar el error hacia atrás es necesario deshacer el submuetreo realizado durante la propagación hacia adelante, éste proceso depende del tipo de pooling realizado. Max pooling (Arriba), Mean pooling (Abajo)

- **Propagación Mean Pooling**
  Sabiendo que cada una de las entradas del mapa de características de salida corresponde a la media de los valores dentro de la ventana utilizada, es posible descomponer cada una de las entradas de los mapas del tensor de error en pequeñas submatrices del tamaño del kernel usado en la propagación hacia adelante. Estas submatrices contendrán el valor de la entrada de la que provienen dividido entre el ancho del kerner usado al cuadrado. Esto puede observarse de forma más clara en la figura.

- **Propagación Max Pooling**
  Para este tipo de submuestreo se realizará algo parecido al anterior, con la diferencia de que las submatrices tendrán ceros en sus entradas con excepción de las ubicaciones donde se encontraban los máximos en el momento de la propagación hacia adelante. Esto se puede observar en la figura.

En esta capa no existen pesos que actualizar por lo cual no es necesario realizar ninguna derivada para el gradiente, así que de esta forma se habrá propagado el tensor de errores a través de la capa de pooling; por lo que se tiene que:

$$\delta^{(1)}_{Reshape} \longrightarrow \delta^{(1)}_{Unsample} \tag{2.61}$$

Por lo cual en esta fase del proceso de propagación hacia atrás, el gradiente de la función de la función de costo con respecto a la información de entrada en la capa FC es:

$$\frac{\partial J}{\partial A^{*(1)}} = \delta^{(1)}_{Unsample} \tag{2.62}$$

Finalmente se ha propagado el error hasta la entrada de la capa de pooling, por ende, corresponde calcular la derivada de la función de costo con respecto al kernel y el sesgo de la capa de convolución.

Definiendo $A^{(0)} = Relu(Z^{(0)})$ a la activación de la capa de convolución, donde $Z^{(0)}$ es la salida de la capa de convolución, por lo que nuevamente utilizando la ecuación (2.41) y escogiendo el caso en el que el argumento de la función es mayor que cero.

$$\frac{\partial A^{(0)}}{\partial K} = \frac{\partial A^{(0)}}{\partial Z^{(0)}} \frac{\partial Z^{(0)}}{\partial K}$$

Es posible entonces escribir la derivada de la función de costo con respecto al kernel como:

$$\frac{\partial J}{\partial K} = \frac{\partial J}{\partial A^{*(1)}} \frac{\partial A^{(0)}}{\partial Z^{(0)}} \frac{\partial Z^{(0)}}{\partial K} \tag{2.63}$$

Como la segunda derivada del lado derecho es igual a 1, podemos reescribir la derivada de J de la siguiente manera:

$$\frac{\partial J}{\partial K} = \frac{\partial J}{\partial Z^{(0)}} \frac{\partial Z^{(0)}}{\partial K} \tag{2.64}$$

Donde $\frac{\partial J}{\partial Z^{(0)}} = \delta^{(1)}_{Unsample}$ Debido a que la derivada 2.64 de la función de costo con respecto al kernel depende de cada una de las entradas del filtro, esta derivada puede escribirse de la siguiente forma:

$$\frac{\partial J}{\partial K} = \begin{bmatrix} \frac{\partial J}{\partial k_{11}} & \frac{\partial J}{\partial k_{12}} & \frac{\partial J}{\partial k_{13}} \\ \frac{\partial J}{\partial k_{21}} & \frac{\partial J}{\partial k_{22}} & \frac{\partial J}{\partial k_{23}} \\ \frac{\partial J}{\partial k_{31}} & \frac{\partial J}{\partial k_{32}} & \frac{\partial J}{\partial k_{33}} \end{bmatrix} \tag{2.65}$$

Por lo que escribiendo la expresión correspondiente a cada entrada de la matriz, se tiene:

$$\frac{\partial J}{\partial k_{mn}} = \sum_{i=1}^{26} \sum_{j=1}^{26} \frac{\partial J}{\partial z^{(0)}_{ij}} \frac{\partial z^{(0)}_{ij}}{\partial k_{mn}} \tag{2.66}$$

Por lo que desarrollando las derivadas de J, se tiene el siguiente conjunto de ecuaciones:

$$\frac{\partial J}{\partial k_{11}} = \left(\frac{\partial J}{\partial z^{(0)}_{1,1}} \frac{\partial z^{(0)}_{1,1}}{\partial k_{11}}\right) + \left(\frac{\partial J}{\partial z^{(0)}_{1,2}} \frac{\partial z^{(0)}_{1,2}}{\partial k_{11}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{2,1}} \frac{\partial z^{(0)}_{2,1}}{\partial k_{11}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{26,26}} \frac{\partial z^{(0)}_{26,26}}{\partial k_{11}}\right)$$

$$\frac{\partial J}{\partial k_{12}} = \left(\frac{\partial J}{\partial z^{(0)}_{1,1}} \frac{\partial z^{(0)}_{1,1}}{\partial k_{12}}\right) + \left(\frac{\partial J}{\partial z^{(0)}_{1,2}} \frac{\partial z^{(0)}_{1,2}}{\partial k_{12}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{2,1}} \frac{\partial z^{(0)}_{2,1}}{\partial k_{12}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{26,26}} \frac{\partial z^{(0)}_{26,26}}{\partial k_{12}}\right)$$

$$\vdots$$

$$\frac{\partial J}{\partial k_{33}} = \left(\frac{\partial J}{\partial z^{(0)}_{1,1}} \frac{\partial z^{(0)}_{1,1}}{\partial k_{33}}\right) + \left(\frac{\partial J}{\partial z^{(0)}_{1,2}} \frac{\partial z^{(0)}_{1,2}}{\partial k_{33}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{2,1}} \frac{\partial z^{(0)}_{2,1}}{\partial k_{33}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{26,26}} \frac{\partial z^{(0)}_{26,26}}{\partial k_{33}}\right)$$

Con base el conjunto de ecuaciones (2.51) , es posible calcular sin problema las segundas derivadas de cada paréntesis, reduciendo las ecuaciones anteriores de la siguiente manera

$$\frac{\partial J}{\partial k_{11}} = \left(\frac{\partial J}{\partial z^{(0)}_{1,1}} x_{1,1}\right) + \left(\frac{\partial J}{\partial z^{(0)}_{1,2}} x_{1,2}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{2,1}} x_{2,1}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{26,26}} x_{24,24}\right)$$

$$\frac{\partial J}{\partial k_{12}} = \left(\frac{\partial J}{\partial z^{(0)}_{1,1}} x_{1,2}\right) + \left(\frac{\partial J}{\partial z^{(0)}_{1,2}} x_{1,3}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{2,1}} x_{2,2}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{26,26}} x_{24,25}\right)$$

$$\vdots$$

$$\frac{\partial J}{\partial k_{33}} = \left(\frac{\partial J}{\partial z^{(0)}_{1,1}} x_{3,3}\right) + \left(\frac{\partial J}{\partial z^{(0)}_{1,2}} x_{3,4}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{2,1}} x_{4,3}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{26,26}} x_{26,26}\right)$$

No obstante de las expresiones anteriores, es posible observar que se trata de el resultado de la convolución de la imagen de entrada con el error que había sido propagado a través de las capas. Por lo tanto, la derivada de la función de costo con respecto al kernel es:

$$\frac{\partial J}{\partial K} = \mathcal{X} \otimes \delta^{(1)}_{Unsample} \tag{2.67}$$

Para el caso de la derivada de J con respecto al sesgo B de la capa de convolución, el procedimiento es similar, usando los mismos argumentos se puede llegar a la siguiente expresión

$$\frac{\partial J}{\partial B} = \frac{\partial J}{\partial Z^{(0)}} \frac{\partial Z^{(0)}}{\partial B} \tag{2.68}$$

La cual puede ser escrita de forma análoga a la ecuación (2.66)

$$\frac{\partial J}{\partial b} = \sum_{i=1}^{26} \sum_{j=1}^{26} \frac{\partial J}{\partial z^{(0)}_{ij}} \frac{\partial z^{(0)}_{ij}}{\partial b} \tag{2.69}$$

Al calcular la derivada de cada elemento $z^{(0)}_{ij}$ con respecto al sesgo b, se obtiene como resultado 1, por lo cual las expresiones para las ecuaciones para las derivadas de J son:

$$\frac{\partial J}{\partial b} = \left(\frac{\partial J}{\partial z^{(0)}_{1,1}}\right) + \left(\frac{\partial J}{\partial z^{(0)}_{1,2}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{2,1}}\right) + \cdots + \left(\frac{\partial J}{\partial z^{(0)}_{26,26}}\right)$$

$$= \sum_{i=1}^{26} \sum_{j=1}^{26} \frac{\partial J}{\partial z^{(0)}_{ij}}$$

Por lo tanto

$$\frac{\partial J}{\partial b} = \sum_{i=1}^{26} \sum_{j=1}^{26} \delta^{(1)}_{Unsample(ij)} \tag{2.70}$$

Una vez que se ha calculado el gradiente de la función de costo con respecto a cada uno de los parámetros de la red, es posible utilizar la técnica que se haya elegido para actualizarlos, ya sea gradiente descendente, Adam, etc.
De esta manera es como modelos tan complejos como las redes neuronales convolucionales reducen el error con cada iteración a lo largo de su entrenamiento.

2.3.8. RNCs para la estadificación de Alzheimer

En el estado del arte existen una amplia variedad de técnicas de análisis de imágenes médicas para la estadificación de esta enfermedad, ya que como se ha mencionado a lo largo de este trabajo, no se trata de una tarea trivial. En particular para resolver este problema por medio de las redes neuronales artificiales, no basta con la capacidad de estas herramientas, sino que además es necesario utilizar distintas variaciones de este modelo para obtener resultados aceptables.
Existen modelos de redes usadas con imágenes 3D de resonancia magnética con 12 capas de convolución recurrentes para clasificación binaria entre las clases AD y MCI [32], mientras que otros trabajos utilizan técnicas como el transfer learning, que consiste en el uso de un modelo previamente entrenado con una base de datos para identificar ciertas cualidades, luego de esto se utiliza una nueva base de datos para enseñarle a identificar cualidades de estas nuevas muestras, en el este caso de este artículo la red ocupada fue la VGG 16 [33], aunque existen muchas opciones de redes preentrenadas disponibles.
Esta técnica es espacialmente útil en los casos en los que no se cuenta con una cantidad suficiente

de muestras para entrenar un modelo.
En otros trabajos, además de realizar variaciones en la arquitectura de los modelos, se proponen estrategias de preprocesado extra al tratamiento básico, para discriminar entre la información con la que aprenderá la red y la que podría interferir negativamente en dicho proceso [40].
En [34] los autores proponen la segmentación de materia gris, dividida en sub imágenes llamadas parches para entrenar su modelo. Gran parte de los trabajos de clasificación de Alzheimer y sus etapas están concentrado en la discriminación de pacientes con enfermedad avanzada y en alguna otra de sus etapas, es decir, clasificación binaria [35]. Esto es debido a que muchas de las características de las distintas etapas de AD no son separables, es decir, algunas de ellas son compartidas, por lo que la clasificación se vuelve una tarea particularmente complicada [36] [37] [35]. Además del uso de RNCs convencionales, se han utilizado alternativas como el uso de redes neuronales convolucionales residuales [38], Graph convolutional networks y Auto-encoder networks [39]. Además, en [36] se propone la extracción de redes cerebrales de conectividad funcional a partir de las imágenes de f-MRI, con el fin de alimentar las redes neuronales (RNs) propuestas. En la tabla (2.1) se presenta un breve resumen de estos trabajos y sus resultados.

Trabajo	Tipo de clasificación	Grupos clasificados	Accuracy	Método	Tipo de imagen
Basaia et al. (2019b)	Binaria	AD-CN	99.2	RNC 3D 2capas conv	MRI
		AD-MCI	75.4		
		MCI - CN	85.9		
Jain et al. (2019b)	Multiclase y binaria	AD-MCI-CN	95.73	Transfer learning VGG16 2D	MRI
		AD-CN	99.14		
		AD-MCI	99.3		
		MCI - CN	99.22		
Initiative (2018b)	Binaria	NC -AD	84.5	Multimodal and Multiscale Deep Neural Network	18F-FDG-PET
		NC- AD/MCI	85.5		
		NC-EMCI/LMCI/AD	85.9		
		NC -AD	81.9	Multimodal and Multiscale Deep Neural Network	MRI
		NC- AD/MCI	82.8		
		NC-EMCI/LMCI/AD	82.5		
		NC -AD	84.6	Multimodal and Multiscale Deep Neural Network	MRI + 18F-FDG-PET
		NC- AD/MCI	86		
		NC-EMCI/LMCI/AD	86.4		
Ramzan et al. (2019e)	Multiclase	AD-MCI-LMCI-EMCI-SMC-CN	97.61 Promedio	Resnet-18 entrenada desde cero	rs-fMRI
		AD-MCI-LMCI-EMCI-SMC-CN	98.13 Promedio	Resnet-18 Pesos congelados	
		AD-MCI-LMCI-EMCI-SMC-CN	98.10 Promedio	Resnet-18 Fine tunning	
Alorf y Khan (2022c)	Binaria y multiclase	CN-SMC	92.75	Stacked sparse auto-encoders	rs-fMRI
		CN-EMCI	86.79		
		CN-MCI	96.67		
		CN-LMCI	87.81		
		CN-AD	90.89		
		6 CLASES	76.18		
Thakur y Snekhalatha (2022b)	Multiclase	AD-CN-EMCI-LMCI	98.44	ResNet 50 Fine tunning	18F-FDG-PET
Ahmed et al. (2019b)	Binaria	AD-NC	80.4	Clasificador de parches Hipocampo RH+LH Model	MRI
			85.55	Clasificador conjunto	
Ahmed et al. (2020b)	Binaria	AD-CN	93.58	Clasificador de parches de 3 vistas 6 ROIs	MRI
		AD-LMCI	85.51		
		AD-MCI	81.73		

Tabla 2.1: Comparativa de técnicas para la estadificación de Alzheimer

En otros trabajos se ha concluído que el preprocesado de las imágenes desempeña un papel fundamental para una extracción de características más eficiente (Salvi et al., 2021b). En particular para el uso en histopatología [40], se sugiere que el uso de imágenes completas para alimentar RNs representa un factor de desventaja ya que algunas de las características representativas de la imagen pueden perderse al realizar un submuestreo sobre la imagen completa. Esto resulta en un aprendizaje ineficiente, por lo cual se propone el uso de técnicas de selección de parches guiada por segmentación.

Entre los trabajos que presentan clasificación multiclase, encontramos trabajos que proponen un preprocesado usando la segmentación de regiones de interés escogidas con base en el daño que sufren durante el desarrollo de la enfermedad, junto con la partición de imágenes 2D en parches para ser analizados por medio de un conjunto de RNCs [41][42].
Durante la revisión del estado del arte se encontró que gran parte de los trabajos de clasificación de este padecimiento reportan clasificación binaria. En particular, Fathi et al. (2022) menciona en su revisión las siguientes estadísticas :

- Pacientes NC y AD. Con exactitud entre 80 - 99 %
- Pacientes NC y MCI con un resultado de exactitud promedio de 86.88 %.
- Clasificación entre EMCI y LMCI con un rango de exactitud de 62 - 99 %.
- Clasificación entre NC y EMCI, cuyo valor de exactitud oscila entre 64 y 99 % .
- Grupos de clasificación multiclase, los menos reportados, tienen una desviación estándar de exactitud de 13.31, más alto que el de los otros grupos.

Luego de la revisión de 74 artículos, Fathi et al., 2022 recomienda concentrarse en las clasificaciones binarias de los grupos NC/AD y NC/MCI, ya que presentan los resultados de exactitud más altos. Además que la modalidad de imágenes que ha llevado a mejores resultados, son las imágenes estructurales de resonancia magnética seguidas de las imágenes funcionales de esta misma técnica. Finalmente, las mejores clasificaciones fueron obtenidas por redes neuronales convolucionales, superando incluso a técnicas como las redes de codificado automático.

En el siguiente capítulo se abordarán en los métodos de preprocesado escogidos para este proyecto, el modelo propuesto y las métricas para evaluar el desempeño de la propuesta.

Capítulo 3

Métodos

Dada la hipótesis de este proyecto, en el presente capítulo se definirá la metodología que se utilizada para la comprobación o refutación de la hipótesis. Con base en la revisión del estado del arte para las estrategias de clasificación para el Alzheimer en el capítulo anterior, se propone el uso de estrategia de cortes combinada con parches basados en regiones de interés. Estos serán generados a partir de imágenes de rs f-MRI para el análisis conjunto de las tres ROIs mencionadas anteriormente.
Para lograr cubrir toda la información posible de las ROIs, se propone el uso de parches de tres vistas. Con base en estos parches, se realizará una clasificación multiclase de los grupos NC, MCI y AD. Esto con el fin de determinar si el uso de este tipo de imágenes en este método puede generar mejores resultados que al usar imágenes estructurales. A continuación comenzaremos con el preprocesado de las imágenes de la base de datos utilizadas.

3.1. Protocolo

La base de datos utilizada en este proyecto pertenece a la Iniciativa de neuroimagen de la enfermedad de Alzheimer (ADNI por sus siglas en inglés), esta iniciativa es un consorcio de universidades y centros clínicos de Estados Unidos y Canadá. Tiene como fin la creación de un repositorio estandarizado de imágenes de diagnóstico de pacientes en el espectro de esta enfermedad [44]. Por otro lado, debido a la reducida cantidad de pacientes de AD en la base de datos ADNI, se utilizó además la base de datos Resting-state fMRI in dementia patients de la universidad de Harvard [45].

Sujetos

Partiendo de los 395 pacientes con imágenes de resonancia magnética funcional en estado de reposo disponibles, sólo se escogieron aquellos pertenecientes a tres clases (Tabla 3.1): Personas sanas (NC), personas con daño cognitivo leve (MCI) y personas con la enfermedad de Alzheimer (AD). De estos pacientes con edades entre 57 - 96 años; 194 son femeninos y 196 masculinos.
Por otro lado la base de datos rs-fMRI de Harvard sólo cuenta con 9 pacientes AD, 8 pacientes CN y 10 pacientes MCI; en este caso no se especifican el rango de edades ni el género de los pacientes.

Se realizó preprocesado a todos los pacientes disponibles pertenecientes a las clases AD, CN y MCI, para después extraer los cortes donde mejor se aprecian las ROIs; luego de esto se obtuvieron los parches correspondientes a cada una de las regiones de interés. Para realizar el entrenamiento de los modelos propuestos se utilizaron los parches correspondientes a 96 pacientes, 32 de cada clase utilizada; y para la prueba se usaron 42 pacientes, 14 de la clase AD, 13 de CN y 15 de la clase MCI. Todo este proceso de explic más a fondo en las siguientes secciones.

Protocolo de adquisión de imágenes

Para adquirir las imágenes de la base de datos ADNI se utilizó un escáner Philips modelo Achieva, intensidad de campo = 3T, TE = 30.0 ms, TR = 3000.0 ms, Matriz(X,Y) = 64.0,64.0, Ángulo de pulso = 80,0°, número de cortes = 6720.0, Resolución en el plano(X,Y) = (3.3 mm, 3.3 mm) y Grosor de corte = 3.3 mm. En el caso de la base de datos rs-fMRI in dementia patients se utilizó un escáner Siemens Magnetom Allegra, intensidad de campo = 3T, TE = 30.0 ms, TR = 2080.0 ms, Matriz(X,Y) = 64.0,64.0, Ángulo de pulso = 70,0°, número de cortes = 6720.0, Resolución en el plano(X,Y) = (3.3 mm, 3.3 mm) y Grosor de corte = 2.5 mm. En ambos casos las imágenes fueron adquiridas en formato NII.

Fase de enfermedad	Alzheimer	Cognitive normal	Early mild cognitive impairment	Mild cognitive impairment	Late mild cognitive impairment	Significant memory concern
Número de pacientes disponibles	39	113	80	93	45	25

Tabla 3.1: Imágenes de f-MRI disponibles por clase en la base de datos ADNI

Software y hardware

Para la realización del preprocesado de las imágenes se utilizó un procesador Intel(R) Core(TM) i3-8130U CPU @ 2.21 GHz en conjunto con los software MATLAB version: 9.13.0 (R2022b) de The MathWorks Inc. , SPM 12 del Instituto de Neurociencia y Medicina de Forschungszentrum Jülich y la toolbox CONN version 22a.
Para el caso del entrenamiento de las RNCs se utilizó el mismo procesador y el software utilizado fue Jupyter notebook versión 6.4.12 de Anaconda Inc., junto con la paquetería Tensor Flow versión 2.13.0 desarrollada por Google Brain Team. Finalmente, para la optimización de los hiperparámetros de las RNCs se empleó la librería Optuna versión 3.4.0

3.2. Elección de regiones de interés

De acuerdo con el Instituto Nacional del Envejecimiento en Estados Unidos, en la etapa más avanzada de la enfermedad de Alzheimer, el daño cerebral es generalizado por lo que el cerebro posee un tamaño reducido debido a los mecanismos descritos en la primera parte del primer capítulo. Sin embargo, lo que resulta relevante para los fines de esta investigación, es el inicio de la degradación cerebral. Se ha documentado que este proceso inicia en zonas como el hipocampo y la corteza entorrinal [46]. Esta zona se encuentra en el lóbulo temporal y se encarga de conectar al hipocampo con la neocorteza, por lo que este padecimiento comienza afectando las regiones del cerebro relacionadas con tareas como el aprendizaje y la memoria.
Con base en esta premisa se ha demostrado para imágenes estructurales de resonancia magnética, por medio de la distinción del percentil de densidad intracraneal a través de una prueba de permutaciones para 101 regiones de interés, cuya hipótesis era que los percentiles de densidad intracraneal para los grupos de pacientes pertenecientes a las clases AD y CN eran distintas.Los resultados de esta prueba arrojaron 6 regiones de interés (ROI por sus siglas en inglés) con valores P más bajas fueron: Hipocampos, Ínsulas y Amgdalas [47]. Todas estas regiones se presentan en pares (izquierda y derecha).
Una vez que se han definido las regiones de interés a analizar, es necesario continuar con la metodología a utilizar para estudiar dichas regiones, ya que como se ha mencionado anteriormente,

estos modelos aprenden directamente de la información que se les presenta, por lo que es relevante el uso de un preprocesado de esta información para facilitar la extracción de características [40].

Estrategias de extracción de características

Algunas de las estrategias para simplificar la extracción de información en imágenes se muestran en la tabla (3.2), es importante considerar los pros y los contras de cada estrategia para encontrar la que mejores resultados pueda brindar a nuestro proyecto. Se ha encontrado que la estrategia de extracción de características por cortes obtuvo los mejores resultados en la gran mayoría de los artículos revisados, excepto en [43] en el cual se utilizó como estrategia de extracción la combinación de cortes y parches, dando así mejores resultados.

Tabla 3.2: Estrategias de extracción de características

Estrategia	Breve descripción	Ventajas	Desventajas
Basada en voxeles	Se usa como característica el valor de intensidad de cada voxel	Detecta cambios sutiles de información	Gran cantidad de dimensiones
Basada en regiones de interés	Uso de regiones pre escogidas bajo una hipótesis	Trabaja con una región de la imagen	La región puede no contener la información de interés
Basada en cortes	En lugar de trabajar con imágenes 3D, usan cortes 2D	Soluciona los problemas de contar con pocas imágenes 3D y de trabajar con muhas dimensiones.	Tiene un menor desempeño que al usar imágenes 3D
Basada en parches	Se divide la imagen original en subimágenes	Detecta cambios sutiles y evita el problema de trabajar con muchas dimensiones	Si los parches no son obtenidos de una zona relevante, se puede omitir información importante.

Técnica de aprendizaje por parches

La técnica de aprendizaje por parches consiste en la extracción de subimágenes a partir de una imagen original, para la búsqueda de características "finas" en zonas específicas. A pesar de que el aprendizaje basado en parches es aplicado comúnmente en algoritmos de identificación de objetos y segmentación [40], esta técnica ofrece ventajas útiles para su uso en RNs como:

- Si la base de datos a utilizar no cuenta con igual número de elementos que representen cada clase, es posible usar conjuntos aleatorios de igual número de parches para entrenar la RNC y así evitar que la red tenga algún sesgo hacia alguna clase en específico.
- Resolver el problema de la escasez de datos al generar mayor número de subimagenes por cada imagen original de la base de datos. Además de mejorar el desempeño de la red y disminuyendo el tiempo de entrenamiento.

Algo a destacar es que las imágenes a usar deben ser las más representativas por lo que los parches usados deben ser de zonas específicas con información clave. Además, en [41] y [42] se propone del uso de parches de tres vistas, los cuales consisten en parches generados a partir de cortes coronales, axiales y sagitales, planteando el uso conjunto con regiones de interés (RI) para proporcionar a las RNs la información necesaria para un aprendizaje eficiente. Es relevante mencionar de estos trabajos, es que debido al tamaño reducido de las imágenes de entrada, la complejidad de las redes propuestas es mucho menor que a las usadas en la mayoría de los trabajos.

3.3. Análisis de imagen

3.3.1. Preprocesamiento

A dicha base de datos se les realizó la serie de preprocesado por defecto con ayuda de la Functional Connectivity toolbox (CONN) en Matlab. Esta serie de preprocesado por defecto o estándar, consta de los siguientes procesos [48]:

- Realineamiento funcional
  En este proceso todos los cortes son corregistrados y remuestrados con respecto a la primera imagen de cada volumen como referencia utilizando una interpolación de b-ranura, además de reducir la susceptibilidad de distorsión por movimiento al estimar las derivadas del campo de deformación con respecto al movimiento de cabeza.

- Segmentación directa y normalización
  En este procedimiento las imágenes estructurales y funcionales son normalizadas en el espacio estándar MNI y segmentados en materia gris, materia blanca y fluído cerebro espinal. Para la normalización aplica esta segmentación usando la media de la señal BOLD como referencia y para las imágenes estructurales usa el volumen T1 como referencia.

- Suavizado funcional
  Esta fase del preprocesado se realiza por medio de la convolución sobre el volúmen con un kernel Gaussiano de 8mm de ancho completo y medio máximo, con el fin de aumentar el coeficiente señal ruido de la señal BOLD.

Sin embargo, al momento de aplicar esta serie en las 39 imágenes de la clase AD, dos de ellas no pudieron ser preprocesadas, debido a que los archivos estaban dañados. Por lo que sólo se pudieron realizar 37 de ellas, para las demás clases también se presentó esta situación, lo cual no presentaba ningún problema ya que estas clases contaban con un mayor número de pacientes. No obstante, al entrenar modelos de aprendizaje automático es importante equilibrar la cantidad de información utilizada de cada clase, ya que el modelo empleado puede desarrollar algún sesgo hacia las clases mayoritarias perdiendo generalidad; por lo cual se emplearon únicamente 37 pacientes de cada una de las tres clases escogidas.

Para continuar con el preprocesado de la base de datos, se buscó algún software para realizar la segmentación de nuestra base de datos, se intentó con varias herramientas como ITK Snap, MANGO, Brain Suite, Free Surfer etc. Sin embargo, la mayoría de herramientas que cuenta con segmentación automática sólo está disponible para las imágenes estructurales. Y en el caso de ITK Snap la segmentación manual es especialmente complicada y tardada en imágenes funcionales sin dominio de la anatomía del cerebro. Por otro lado Free Surfer sólo está disponible para sistemas LINUX y MacOs; y al lograr la instalación de una máquina virtual, no se contaba con la contraseña del archivo Zip de instalación ni se recibió respuesta al solicitarla, por ende, por cuestiones de tiempo se decidió continuar sin una herramienta de segmentación e intentar delimitar las regiones manualmente.

Para este procedimiento, se utilizó el toolbox de Matlab Statistical Parametric Mapping (SPM 12) y su función de Check Registration para poder visualizar los volúmenes preprocesados de las imágenes estructurales y funcionales en conjunto. Esta herramienta tiene ventaja de que al usar volúmenes corregistrados y al señalar alguna región en dicho volúmen, este mismo punto es señalado en los demás volúmenes mostrados como se puede observar en la figura (3.1).
De esta manera es posible determinar la ubicación de las ROIs escogidas y los cortes en los cuales se pueden apreciar mejor en cada uno de los planos anatómicos. Además de identificar las coordenadas que de los 4 puntos que delimitan cada ROI en el cada plano anatómico. Con ayuda

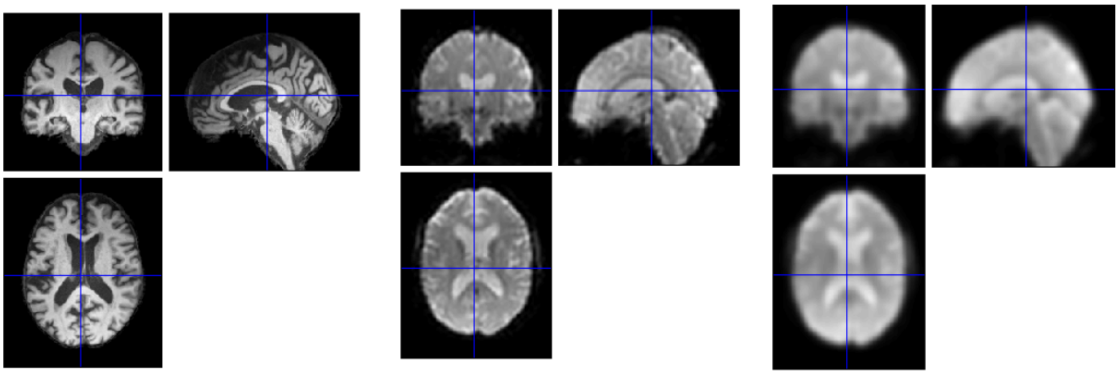

Figura 3.1: **Visualización SPM 12**. La herramienta Check Registration de esta toolbox de Matlab permite la visualización simultánea de distintos volúmenes en los 3 planos anatómicos principales. Se muestra un volúmen estructural (Izquierda), funcional (Centro) y funcional luego del Smoothing con el kernel Gaussiano(Derecha).

de las coordenadas MNI de las ROIs obtenidas con esta herramienta, se escribió un código en matlab para recortar la proyección de cada ROI sobre cada corte anatómico. En la figura (3.2) se muestra muestra el proceso de recorte de la región de interés, en este caso se trata de el corte sagital para aislar la región de la ínsula derecha. Para poder extraer la mayor información posible, se consideraron 2 cortes hacia arriba y hacia abajo con respecto al plano en cuestión, esto debido a que el grosor de los cortes en los volúmenes en r-fMRI es de 3.3mm y se buscaba sacar la mayor cantidad posible de cortes útiles.

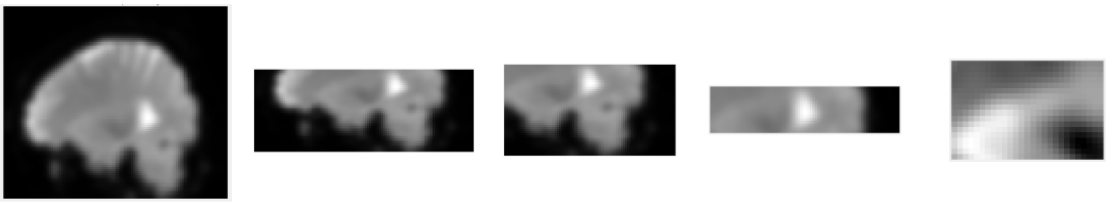

Figura 3.2: **Recorte de región de interés**. Con base en las coordenadas MNI que delimitan cada ROI, se realizaron los cortes para reducir la información ajena a estas regiones; en este caso se trata del corte sagital para la región de la ínsula derecha.

Se realizó un programa en Matlab para realizar modificaciones básicas al archivo .nii, como lo es rotar los cortes con respecto a dos ejes de manera en que la visualización en Matlab coincida con la forma en que los muestra SPM 12. Luego de esto, genera un rectángulo en el cual únicamente se aprecian los órganos de interés para luego cortarlos en subimágenes de 9 x 9 pixeles (figura 3.3), para evitar obtener parches de las zonas sin señal BOLD relevante como los bordes o zonas fuera del cerebro, se utilizó la restricción de revisar el pixel central de cada uno de los parches, si este valor de intensidad es mayor a un valor umbral de 500, será guardado en carpetas según la clase a la que pertenecía cada paciente. El tamaño de estas subimágenes o parches fueron escogidos de estas dimensiones para así obtener el mayor número de parches de cada corte, considerando la resolución que poseen las imágenes funcionales y el tamaño de los órganos.
Además, para probar el desempeño del modelo propuesto, de los 111 pacientes de la base de datos ADNI se separaron 5 pacientes por clase y todas las imágenes disponibles de la segunda base de datos de la Universidad de Harvard, por lo cual la lista de sujetos de prueba consta de 14 pacientes

AD, 13 sujetos control (CN) y 15 pacientes MCI para generar los parches de cada zona.

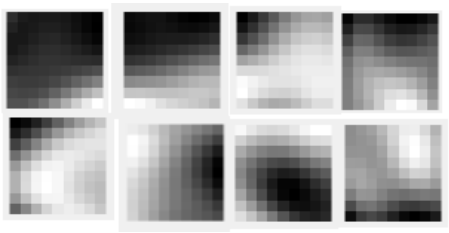

Figura 3.3: **Corte de parches**. A partir de las imágenes de las ROIs aisladas se realizó el corte en parches; en esta figura se muestran los parches obtenidos a partir de la ínsula derecha.

3.3.2. RNCs para clasificación de parches

Con base en la propuesta del análisis de ROIs, se usarán un conjunto de RNCs, de las cuales cada una se encargará del análisis de cada ROI por separado, por lo que cada una de estas redes debería realizar su propio trabajo de clasificación de los parches correspondientes a su región.

Entrenamiento de RNC

Para realizar el entrenamiento de las RNCs individuales, se programó el siguiente proceso en lenguaje Python en Jupyter notebook con ayuda de la librería Tensorflow:

- Data augmentation: Esta técnica consiste en realiza una copia de las muestras originales bajo una serie cambios como rotaciones, desplazamientos, etc. Esta técnica se utiliza principalmente para suplir la escasez de datos cuando se entrena una red desde cero, ya que estos requieren de una gran cantidad de información. Tomando como ejemplo la figura (3.3), de un solo corte de la ROI más grande, se pueden obtener alrededor de 8 parches, por lo que de un solo paciente se pueden obtener alrededor de 100 parches de tres vistas. De esta manera en total para los 96 pacientes, serían alrededor de 9600 parches; sin embargo, si deseamos un modelo más robusto es necesario hacer crecer este conjunto de muestras, especialmente para las ROIs más pequeñas.

- Normalización: Los valores de intensidad de las imágenes son normalizadas para mantener dichos valores en un intervalo y así evitar que nuestro modelo diverja.

- Validación cruzada: Para entrenar cada red neuronal se utilizó el método de validación cruzada, la cual consiste en la división del conjunto de entrenamiento en n subconjuntos de los cuales el primero se utiliza como conjunto de validación y el resto como conjunto de entrenamiento. Luego del primer entrenamiento se usará el segundo conjunto como validación y el resto para entrenamiento, este proceso se repetirá hasta que cada uno de los subconjuntos se haya utilizado como validación; de esta manera se puede obtener un modelo con mayor generalidad. Para este caso el total de imágenes disponibles de la región en cuestión, fue dividido en cuatro subconjuntos para utilizar este método. Cada entrenamiento fue realizado por el método de mini batch o lotes reducidos con un tamaño de 1000 elementos durante 400 épocas. La duración del entrenamiento fue escogido a partir del comportamiento del modelo durante el mismo; ya que para una duración menor, las curvas de aprendizaje seguían mostrando tendencia a crecer, mientras que para duraciones más grandes, dichas curvas tendían a un valor específico.

Arquitectura de RNC

Con el fin de determinar la configuración óptima de hiperparámetros para estas seis redes, se realizaron diversos experimentos variando la arquitectura de las redes, el número y tamaño de filtros, así como distintas funciones de costo y de activación. Luego de varios experimentos, se determinó que la función de costo que ofrecía mejores resultados era la entropía cruzada, al permitir una disminución más rápida del error de las redes. También se encontró que las funciones de activación que permitían mejores resultados eran ReLu en las capas de convolución y ocultas, así como softmax en la capa de salida para la clasificación de los parches.
Se realizaron otros experimentos variando la tasa de aprendizaje de la red, sin embargo, al ser uno de los parámetros más sensibles ya que una ligera variación puede hacerlo converger a un mínimo local distinto al óptimo; se optó por el uso del optimizador Adam por a su capacidad de corregir la tasa de aprendizaje de cada uno de los parámetros.

Finalmente, debido a que la sábana de la función de costo en el espacio de parámetros, no sólo varía con la arquitectura de la red en usada, sino también con la base de datos usada para entrenar, se optó por usar la siguiente arquitectura para las seis redes del arreglo (Figura 3.4):
Una capa de convolución seguida de una capa completamente conectada de dos capas ocultas y finalmente una capa de activación, esto debido a las dimensiones de los parches utilizados, además de continuar con la tendencia de las RNCs que podemos encontrar en la literatura.

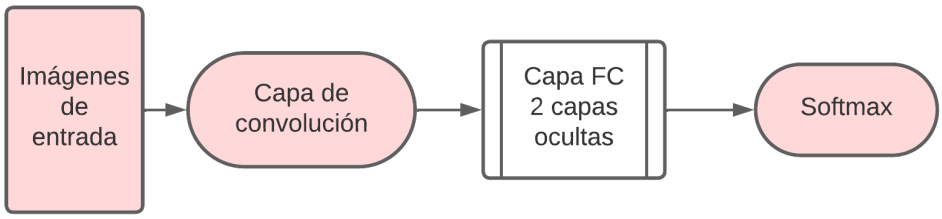

Figura 3.4: **Arquitectura de RNC propuesta**. Esta arquitectura cuenta con una única capa de convolución para prevenir que el tamaño de las imágenes de entrada se encoja demasiado; una capa FC con dos capa ocultas y activación por softmax.

Una vez determinada la arquitectura a utilizar en cada una de las redes, se busca prepararlas para el análisis de su respectiva región. Aún después de haber de haber escogido una arquitectura fija para todas las regiones, existe una larga lista de factores por determinar para cada red individualmente, por lo que sólo se propone variar tres aspectos:

- Número de filtros en la capa de convolución.
- Tamaño del filtro a convolucionar.
- Número de neuronas en las capas ocultas.

Para encontrar el mejor valor estos parámetros, se agilizó el proceso de prueba y error por medio de la librería Optuna de Python. Esta librería permite realizar gran número de entrenamientos durante un número de épocas dado, en cada uno de estos entrenamientos es que varía los parámetros sugeridos, en este caso los de la lista anterior, para así determinar la mejor combinación para cierta arquitectura y para cierto data set. Usando entrenamientos de cien épocas para cien combinaciones distintas en cada una de las seis redes del arreglo, se determinó la combinación de hiperparámetros que mejores resultados permitía.
Después de ajustar los hiperparámetros mencionados anteriormente se debe encontrar la manera

de obtener una clasificación global del paciente cuyos parches fueron clasificados por cada una de las redes. A continuación se presenta una propuesta para llevar a cabo esta tarea.

3.3.3. Clasificación por medio de un arreglo de RNCs

Para realizar las pruebas de desempeño de las redes en conjunto, es necesario evaluar los parches de cada uno de los pacientes de prueba en cada una de las RNC, para que estas realicen una predicción para cada parche de entrada. De esta forma se obtiene un vector de salida, éste contiene las probabilidades de que el parche de entrada pertenezca a cada una de las clases.
Por lo tanto en cada red se obtiene un tensor de probabilidades para cada uno de los parches que ingresan a ellas, sin embargo, todos estos vectores corresponden a fragmentos de un solo paciente (Figura 3.5). En el estado del arte no existe una metodología concreta para poder combinar todas las probabilidades en una sola, es por esto que se propone el uso del teorema de Bayes.

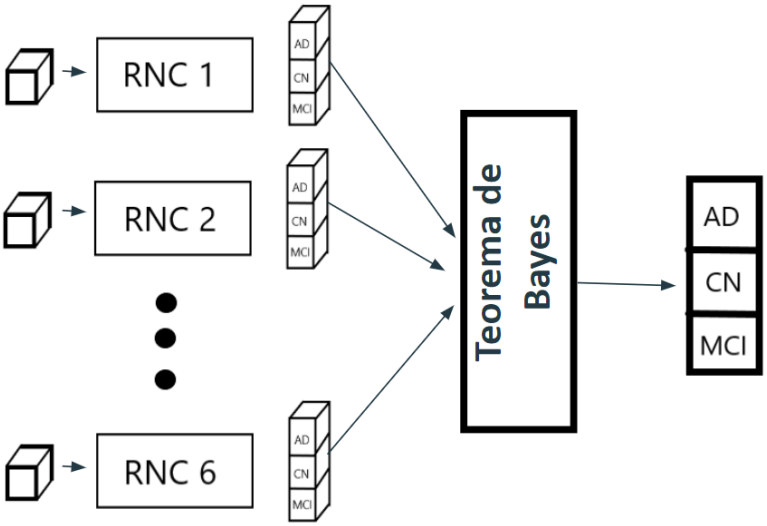

Figura 3.5: **Combinación de probabilidades**. Cada conjunto de parches correspondientes a una ROI obtiene un tensor de probabilidades de dimensiones *Número de clases*$\times 1 \times$ *Número de parches*, los cuales deberán combinarse para obtener la probabilidad global de clasificación del paciente en cuestión.

El teorema de Bayes nos presenta el problema clásico de probabilidad condicional, en su forma más sencilla se expresa de la siguiente forma:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3.1}$$

Donde:

- $P(A|B)$ es la probabilidad del evento A dado que ha ocurrido el evento B.
- $P(B|A)$ representa la probabilidad del evento B dado que ha sucedido A.
- $P(B)$ es la probabilidad total de que suceda el evento B.

El caso tratado aquí se puede plantear de la siguiente manera:
Sean los eventos C_1, C_2, C_3 dependientes del evento I donde, a su vez, I está formado por n eventos tal que $I = \{I_1, I_2, \cdots, I_j, \cdots, I_n\}$, donde I_j es el j-ésimo evento.

Reescribiendo el teorema (ec. 3.1) para el caso planteado del evento C_1

$$P(C_1|I) = \frac{P(I|C_1)P(C_1)}{P(I)} \tag{3.2}$$

A continuación se presenta el desarrollo de cada término de la ecuación anterior:

- $P(I)$:
  Partiendo del hecho de que la probabilidad total del evento I está dada por:

  $$P(I) = P(I|C_1)P(C_1) + P(I|C_2)P(C_2) + P(I|C_3)P(C_3) \tag{3.3}$$

  Sin embargo, como $I = \{I_1, I_2, \cdots, I_n\}$, para el evento C_i se tiene que:

  $$\begin{aligned} P(I|C_i)P(C_i) &= P(I_1|C_i)P(C_i) + \cdots + P(I_n|C_i)P(C_i) \\ &= \left[P(I_1|C_i) + \cdots + P(I_n|C_i)\right] P(C_i) \\ &= \sum_{j=1}^{n} P(I_j|C_i)P(C_i) \end{aligned}$$

  Por lo tanto, la ecuación (3.3) se puede reescribir como:

  $$P(I) = \sum_{j=1}^{n} P(I_j|C_1)P(C_1) + \sum_{j=1}^{n} P(I_j|C_2)P(C_2) + \sum_{j=1}^{n} P(I_j|C_3)P(C_3)$$

  $$P(I) = \sum_{i=1}^{3}\sum_{j=1}^{n} P(I_j|C_i)P(C_i) \tag{3.4}$$

- $P(I|C_1)$:
  El primer término del numerador se puede expresar como la probabilidad de que cada uno de los eventos I_j sucedan si ha sucedido el evento C_1:

  $$\begin{aligned} P(I|C_1) &= P(I_1|C_1)P(I_2|C_1)\cdots P(I_n|C_1) \\ &= \prod_{j=1}^{n} P(I_j|C_1) \end{aligned} \tag{3.5}$$

Por lo que finalmente, al sustituir las ecuaciones (3.4) y (3.5) en la expresión (3.2), se puede reescribir de la siguiente forma:

$$P(C_1|I) = \frac{\prod_{j=1}^{n} P(I_j|C_1)P(C_1)}{\sum_{i=1}^{3}\sum_{j=1}^{n} P(I_j|C_i)} \tag{3.6}$$

De esta forma, es posible determinar la probabilidad de que un paciente pertenezca a una clase, dadas las probabilidades de cada uno de sus parches.
Una vez resuelto el problema de la probabilidad global de un paciente, es posible implementar un arreglo de RNCs (Figura 3.6)

3.4. Métricas de evaluación

Es posible tener una idea del desempeño del arreglo de redes con base en la probabilidad de clasificación de cada paciente o el número total de clasificaciones correctas, sin embargo, es útil usar métricas para determinar si la hipótesis de este proyecto se cumplió; además de si existe alguna relación entre la probabilidad de un paciente de pertenecer a la clase predicha y si la clasificación hecha fue correcta.
A continuación definiremos las métricas utilizadas para estos fines:

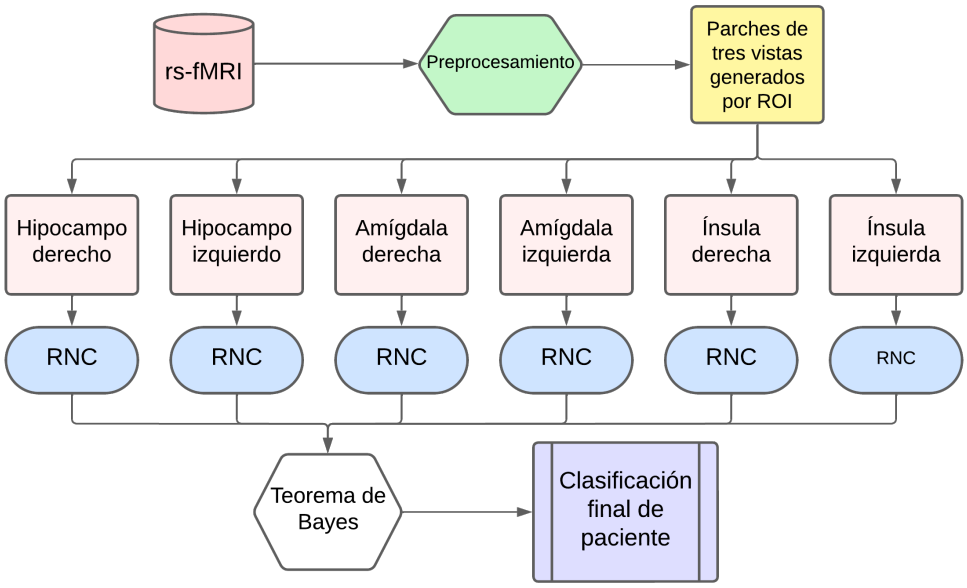

Figura 3.6: **Arreglo de RNCs**. Flujo de la información desde el volumen rs-fMRI hasta la obtención de la clasificación del correspondiente volumen.

3.4.1. Accuracy

El accuracy o exactitud es una de las métricas más utilizadas en trabajos de machine learning, en este trabajo se utilizará para conocer la exactitud del arreglo completo. Se define de la siguiente forma:

$$Accuracy = \frac{\text{Número casos correctos}}{\text{Número casos totales}} \tag{3.7}$$

3.4.2. Sensibilidad

La sensibilidad se encarga de cuantificar el grado en el que el modelo es capaz de identificar una clase correctamente. Esta métrica se define como se muestra a continuación:

$$Sensitivity = \frac{\text{Número verdaderos positivos}}{(\text{Número verdaderos positivos} + \text{Número falsos negativos})} \tag{3.8}$$

3.4.3. U de Mann-Whitney

Este tipo de pruebas para datos no paramétricos sirve para determinar si existe diferencia alguna entre grupos de datos con respecto a alguna característica específica, por medio del uso del rango de los datos. La hipótesis nula de esta prueba es que no existe diferencia significativa entre ambos grupos [49].
Para el caso concreto de este trabajo, será utilizado para conocer si existe alguna diferencia significativa entre los casos de clasificación correcta e incorrecta con respecto a la probabilidad asignada por el arreglo. De esta forma sería posible concluir si el modelo es capaz de discriminar entre una buena o mala predicción, ya que en una red individual la probabilidad de una clasificación representa la seguridad del modelo de que la muestra pertenece a la clase dada.

3.4.4. Q de Cochran

Al igual que la prueba de Mann-Whitney, la Q de Cochran es una prueba para datos no paramétricos, principalmente utilizado para campos categóricos. Esta prueba es utilizada en el ámbito médico para encontrar si existe alguna diferencia significativa entre tres o más mediciones. En este caso se utilizará esta métrica para conocer si existe diferencia significativa entre 3 o más configuraciones de arreglos, con respecto al número de aciertos logrados.

Una vez definidas las métricas a utilizar para evaluar el desempeño del arreglo de RNCs, a continuación se presentarán los resultados obtenidos al implementar la metodología introducida en este capítulo.

Capítulo 4

Resultados

En el este capítulo se presentan los resultados de los experimentos realizados a lo largo del desarrollo de este proyecto. Para iniciar se muestran los resultados utilizados para la determinación de la arquitectura usada en las redes del arreglo.

4.1. Determinación de arquitectura

A continuación se muestran los resultados al variar algunos de los parámetros de una red convolucional, con el fin de determinar la configuración que brinde mejores resultados.

RNC prueba 1

Esta prueba consiste en una RNC con la siguiente arquitectura y elección de parámetros:

- Dos capas de convolución con 50 filtros de 3×3 con función de activación ReLu y padding Same.
- Dos capas densas con 50 neuronas con función de activación ReLu.
- Capa de dropout al 20 %.
- Una capa densa de 50 neuronas con función de activación ReLu
- Capa de salida de 3 neuronas con función de activación Softmax

Como optimizador se utilizó el algoritmo Gradiente descendente con una taza de aprendizaje de 1×10^{-2} y función de costo entropía cruzada multiclase. El entrenamiento se usó validación cruzada con 4 subconjuntos o folds durante 300 épocas utilizando un tamaño de lote de 100 elementos.

	Entrenamiento		Validación	
Fold	Error	Accuracy	Error	Accuracy
1	0.7901	0.6241	0.8190	0.6045
2	0.7902	0.6228	0.8253	0.6003
3	0.7898	0.6270	0.7933	0.6205
4	0.7975	0.6172	0.7978	0.6197

Tabla 4.1: Comportamiento de los valores de pérdida y accuracy durante el entrenamiento y validación.

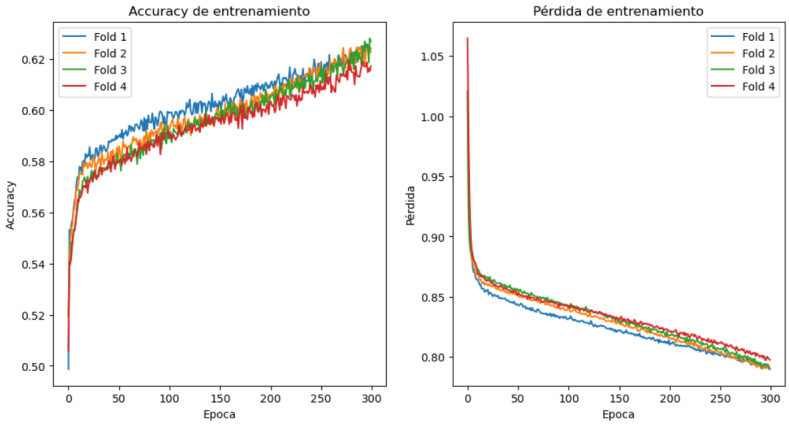

Figura 4.1: **Entrenamiento de la RNC de prueba 1**. Esta figura muestra el desempeño de la red a lo largo de las 300 épocas, así como el seguimiento de la reducción de su error.

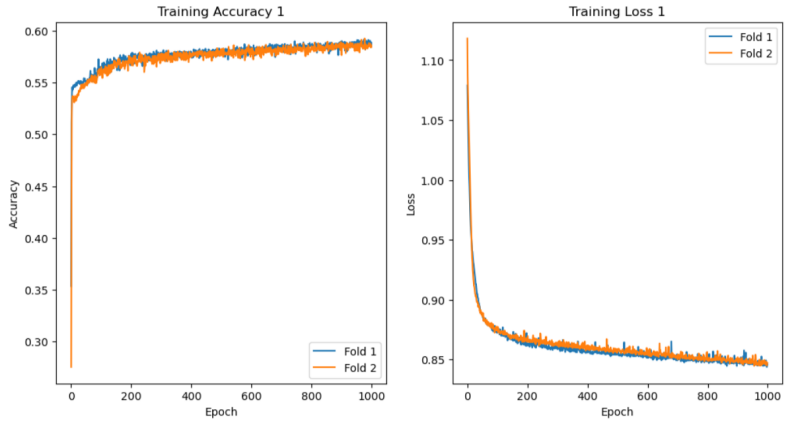

Figura 4.2: **Extensión del entrenamiento prueba 1.** Entrenamiento de la prueba 1 durante mil épocas con dos folds.

En la tabla 4.1 se muestra el comportamiento de la primer prueba luego del entrenamiento con cada fold y en la figura (4.1). Es importante destacar el valor máximo de exactitud alcanzado en el entrenamiento así como su disminución en cada fold. Además en la figura(4.2) se muestra el comportamiento del mismo modelo durante un entrenamiento más largo.

Prueba 2

Para esta segunda prueba se han tomado los mismos parámetros y arquitectura de la prueba 1, sin embargo, sólo se ha variado la tasa de aprendizaje a un valor de 1×10^{-4}, como resultado se han obtenido las variaciones mostradas en la tabla (4.2). Además en la figura (4.3) es posible observar el cambio en el desempeño de la red con sólo este cambio, el cual muestra un comportamiento asintótico en su crecimiento a un valor de accuracy menor con respecto a la prueba anterior.

Prueba 3

Para esta prueba se utilizó la siguiente arquitectura:

- Una capa de convolución con 50 filtros de tamaño 3×3 con función de activación ReLu
- Dos capas densas con 50 neuronas y función de activación ReLu

	Entrenamiento		Validación	
Fold	Error	Accuracy	Error	Accuracy
1	0.8969	0.5430	0.8873	0.5614
2	0.9021	0.5490	0.8932	0.5418
3	0.9065	0.5509	0.9055	0.5409
4	0.8872	0.5512	0.8775	0.5464

Tabla 4.2: Valores de accuracy y error para los conjuntos de validación y entrenamiento de la prueba 1.

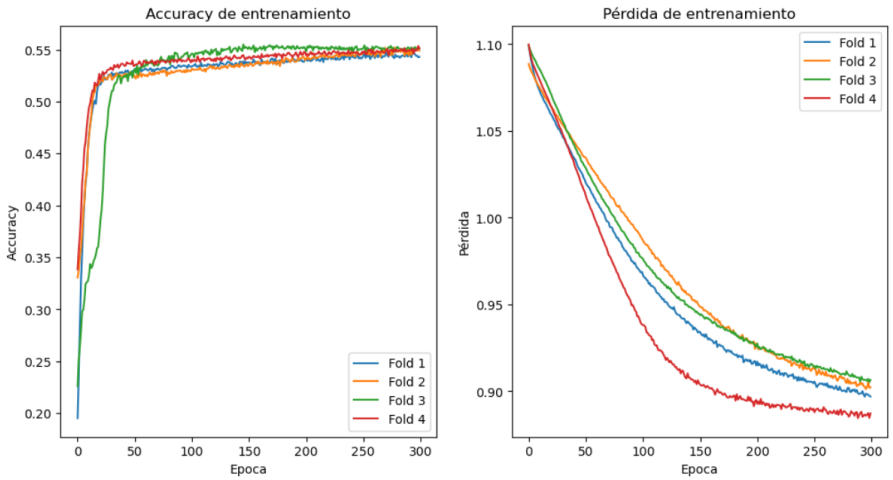

Figura 4.3: **Entrenamiento de la RNC de prueba 2**. Estas gráficas muestra el desempeño de la red a lo largo de las 300 épocas, así como el seguimiento de la reducción de su error.

- Una capa de salida con 3 neuronas con función de activación Softmax.

Como función de costo se utilizó Cross entropy y el optimizador fué Adam, el cual inicia con una tasa de aprendizaje de 0.001 con la ventaja de que este optimizador utiliza el momento adaptativo. Es importante notar que para este caso, en la figura (4.4) muestra una tendencia de aumento en los valores de accuracy el cual podría continuar con un mayor número de épocas; de igual forma se observa una consistencia en los valores de la tabla (4.3) a al final de cada fold.

	Entrenamiento		Validación	
Fold	Error	Accuracy	Error	Accuracy
1	0.3193	0.8629	0.9958	0.7590
2	0.3109	0.8688	0.9390	0.7471
3	0.3269	0.8621	0.8447	0.7521
4	0.3181	0.8720	0.8616	0.7420

Tabla 4.3: Valores de error y accuracy para los conjuntos de entrenamiento y validación de la prueba 3

Prueba 4

Para la prueba número 4 de arquitectura se utilizó la siguiente configuración:

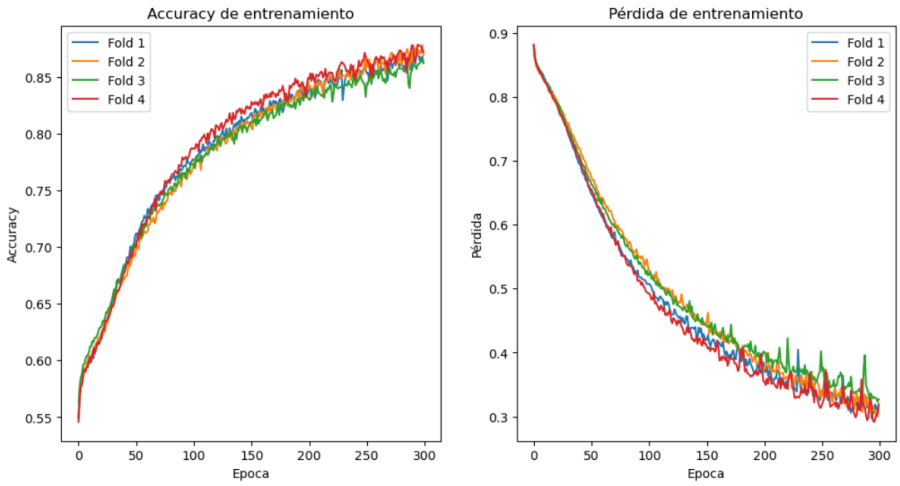

Figura 4.4: **Entrenamiento de la RNC de prueba 3**. Estas gráficas muestra el desempeño de la red de la prueba 3 a lo largo de las 300 épocas, así como los valores del error.

- Dos capas de convolución con 50 filtros de tamaño 3×3 con función de activación ReLu
- Dos capas densas con 50 neuronas cada una con función de activación ReLu
- Capa de dropout al 20 % seguida de una capa densa de 56 neuronas activadas por la función ReLu. El dropuout en las capas densas consiste en la desconección aleatoria de cierto porcentaje de neuronas, de manera en que su activación no se transmita a la siguiente capa.
- Una capa de salida de 3 neuronas con función de activación softmax

El optimizador de esta prueba fue Adam con la función de costo de error cuadrático medio.

	Entrenamiento		Validación	
Fold	Error	Accuracy	Error	Accuracy
1	1.1117	0.3312	1.117	0.3331
2	1.117	0.3301	1.1118	0.3339
3	1.1118	0.3428	1.1114	0.3337
4	1.1116	0.3344	1.1120	0.3332

Tabla 4.4: Comportamiento del modelo correspondiente a la prueba 4 al final del entrenamiento con cada subconjunto.

Prueba 5

En esta prueba la arquitectura utilizada fue:

- Una capa de convolución con 50 filtros de tamaño 3×3 y función de activación ReLu
- Dos capas densas con 50 neuronas y función de activación sigmoide
- Tres neuronas de salida con función de activación softmax

En el entrenamiento se utilizó el optimizador Adam y la función de costo de entropía cruzada para clasificación multiclase.

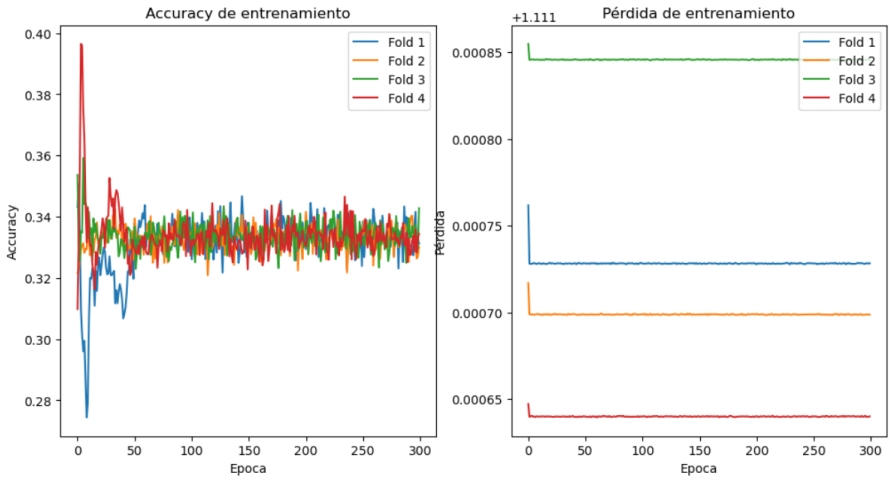

Figura 4.5: **Entrenamiento de la RNC de prueba 4**. Se muestra el entrenamiento de la red para cada uno de los folds.

	Entrenamiento		Validación	
Fold	Error	Accuracy	Error	Accuracy
1	0.4427	0.8070	0.7314	0.7251
2	0.4744	0.8009	0.7273	0.7124
3	0.3863	0.8351	0.7066	0.7433
4	0.4137	0.8228	0.7755	0.7227

Tabla 4.5: Comportamiento del modelo de la prueba 5 al final de cada fold

4.2. Determinación de hiperparámetros de las RNCs

Una vez que se ha escogido una arquitectura específica para todas las redes correspondientes a las ROIs; se utilizó el optimizador para hiperparámetros Optuna para encontrar los valores de hiperparámetros de cada red. Este optimizador funciona de manera que dada una arquitectura y establecidos los parámetros a probar, así como el intervalo en que estos van a variar, se realiza un entrenamiento durante un determinado número de épocas. Luego de obtener los resultados de la métrica escogida, varía nuevamente los hiperparámetros para repetir este proceso n veces; el número de experimentos escogido. Luego de este proceso se muestra la configuración de hiperparámetros que lograron el mejor desempeño de la red durante el entrenamiento con la información procedente de una ROI en específico.
Los parámetros a variar en estos experimentos son: El número de filtros en la capa de convolución, el tamaño de los filtros a operar y el número de neuronas en las capas ocultas de la capa FC.
En la tabla (4.6) se presentan un breve resumen de los resultados del optimizador para cada red.
Dada la arquitectura determinada en esta sección, a continuación se presentan los resultados de los entrenamientos de los arreglos de redes para cada ROI.

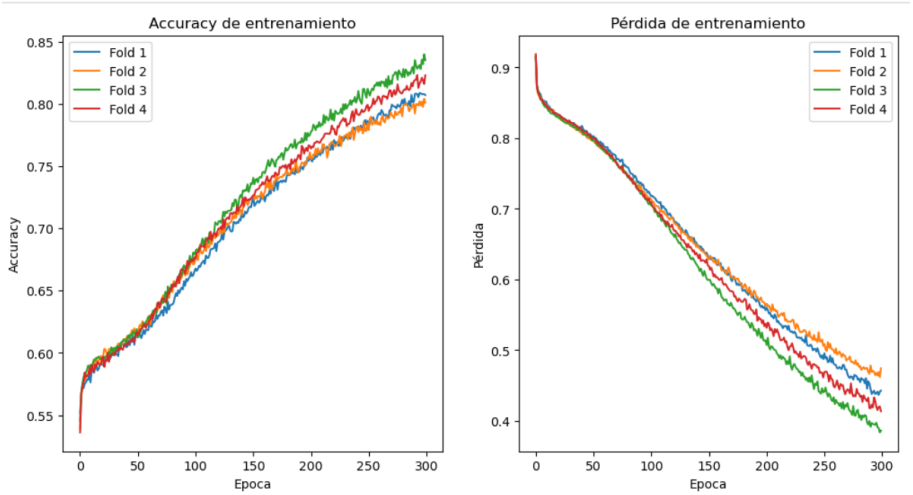

Figura 4.6: **Entrenamiento de la RNC de prueba 5**. Se muestra el entrenamiento de la red para cada uno de los folds.

ROI	Accuracy (%)	Número de filtros	Tamaño de filtro	Número de neuronas en capas ocultas
Hipocampo Derecho	71.97	455	10	98
Hipocampo Izquierdo	70.98	455	9	100
Ínsula Derecha	72.39	360	10	93
Ínsula Izquierda	71.85	403	7	72
Amígdala Derecha	70.94	232	10	88
Amígdala Izquierda	70.34	251	10	89

Tabla 4.6: Resultados de Optuna para cada una de las ROIs.

4.3. Resultados de arreglos de RNCs

Como ya se ha mencionado en este trabajo, dadas las dimensiones de órganos como las amígdalas, es complicado obtener una cantidad suficiente de parches para entrenar correctamente una RNC. Por lo cual la información aportada por el arreglo de RNCs de estos órganos podría influir negativamente en los resultados al usarlo en conjunto con los demás arreglos. Debido a esto es que se propone el uso de un arreglo adicional que utilice los dos órganos restantes: hipocampos e ínsulas. Aunado a esto, con el fin de tener una referencia de la discrepancia en los resultados al usar únicamente un par de ROIs, se propone el uso de tres arreglos extra correspondientes a los hipocampos, ínsulas y amígdalas. Por lo cual se tiene la siguiente lista de arreglos a explorar:

- Hipocampo izquierdo y derecho.
- Ínsula izquierda y derecha.
- Amígdala izquierda y derecha.
- Pares de hipocampos e ínsulas.

Por lo que a continuación se presentan los resultados para cada uno de los arreglos de RNCs.

Resultados ínsulas

El arreglo de RNCs correspondiente a las regiones de las ínsulas izquierda y derecha obtuvo los resultados de la tabla (4.7). Esta tabla se muestra en la parte superior la clase a la cual pertenecen

cada uno de los 42 pacientes de prueba, mientras que en las columnas de clasificación se muestran las clases así como las probabilidades asignadas por parte del arreglo de RNCs. Las clasificaciones asignadas que coinciden con la clase original de cada paciente están escritas en color rojo. Para este caso se tiene un total de 13 aciertos siendo MCI la clase con mayor número de aciertos y las probabilidades asignadas por este arreglo tienen una desviación estándar de 0.2129 con una media de 96.67 %.

Ínsulas						
	AD		CN		MCI	
	Clasificación	Probabilidad	Clasificación	Probabilidad	Clasificación	Probabilidad
1	AD	0.999854017	CN	0.50145171	CN	0.599694681
2	CN	0.999997812	CN	0.625324492	MCI	0.956141606
3	CN	0.964218222	AD	0.999162812	AD	0.941358497
4	MCI	0.984979043	AD	0.989902727	AD	0.999419071
5	AD	0.728861471	AD	0.994386312	MCI	0.999969778
6	MCI	0.999998082	MCI	0.965735997	MCI	0.44049694
7	CN	0.506062725	MCI	0.99321297	MCI	0.868869816
8	CN	0.507837682	MCI	0.96778987	CN	0.922442137
9	CN	0.999170783	MCI	0.999453311	CN	0.515896582
10	AD	0.92709035	MCI	0.679638014	CN	0.993666236
11	CN	0.946664084	CN	0.998474775	MCI	0.808093212
12	CN	0.487592482	AD	0.995265526	AD	0.999868008
13	AD	0.972053958	AD	0.804352243	CN	0.99999998
14	CN	0.679987928			CN	0.999872829
15					MCI	0.980449341

Tabla 4.7: Resultados de clasificación de los 42 sujetos de prueba y sus respectivas probabilidades de pertenecer a la clase predicha.

Resultados hipocampos

En la tabla (4.8) se muestran las clasificaciones del arreglo par correspondiente a los hipocampos izquierdo y derecho, las clasificaciones correctas están escritas en color rojo. La clase con mayor número de aciertos es MCI con 9 de los 17 aciertos totales logrados por este arreglo; además se tiene que las probabilidades tienen una desviación estándar de 0.1925 y una media de 83.13 %.

Resultados Amígdalas

Luego del entrenamiento con los parches de la región de las amígdalas, se obtuvieron los resultados de la tabla (4.9) donde los resultados en rojos son las clasificaciones correctas. Este arreglo obtuvo un total de 13 aciertos de los cuales 8 fueron en la clase AD, la desviación estándar de sus valores de probabilidad es de 0.1429 con una media de 62.92 %; es importante notar que los valores de probabilidad de las clasificaciones tienen una menor desviación estándar en comparación que los otros arreglos a excepción del arreglo de 6 ROIs, sin embargo, el valor alrededor del cual oscilan es más bajo con un máximo de 92.93 %.

Resultados del arreglo ínsula - hipocampo

En la tabla (4.10) se muestran las predicciones para cada uno de los sujetos de prueba luego de analizar sus ínsulas e hipocampos. Para este caso la cantidad de de aciertos por clase es similar, teniendo 27 aciertos en total de los cuales 10 fueron en la clasa AD, 9 en la clase CN y 8 en MCI;

Hipocampos						
	AD		CN		MCI	
	Clasificación	Probabilidad	Clasificación	Probabilidad	Clasificación	Probabilidad
1	MCI	0.644146362	MCI	0.499553463	MCI	0.564806821
2	CN	0.544428261	MCI	0.878780945	CN	0.677199201
3	MCI	0.588621149	MCI	0.549380051	CN	0.81014639
4	AD	0.489376104	MCI	0.930619436	MCI	0.975253017
5	MCI	0.995601463	MCI	0.818975886	MCI	0.931864404
6	AD	0.861094242	AD	0.851196734	MCI	0.388012938
7	MCI	0.961321846	AD	0.96083656	CN	0.993136741
8	AD	0.507738848	AD	0.953873391	CN	0.566389469
9	AD	0.851375757	MCI	0.987228907	AD	0.441233395
10	AD	0.513841504	AD	0.686380781	MCI	0.646429856
11	MCI	0.833239081	MCI	0.992346997	MCI	0.714872177
12	CN	0.796365024	CN	0.857634676	MCI	0.712197915
13	AD	0.939349791	CN	0.944990324	MCI	0.924870664
14	MCI	0.974284092			MCI	0.971537966
15					AD	0.829511045

Tabla 4.8: Resultados de clasificación hecha por los hipocampos para los sujetos de prueba y sus respectivas probabilidades.

para el caso de este arreglo se obtuvo una desviación estándar de los valores de probabilidad de 0.1906 con una media de 97.54 %.

Resultados del arreglo de 6 regiones de interés

Finalmente en la tabla (4.11) se pueden observar los resultados de la clasificación realizada luego del análisis de todas las regiones de interés. Nótese la diferencia del número de clasificaciones correctas logradas con este arreglo con respecto al caso anterior al incluir la aportación de las amígdalas, esto da como resultado un total de 19 aciertos donde la clase MCI es la que cuenta con el mayor número de aciertos. En este arreglo la media de las probailidades asigandas es de 98.66 % con una desviación estándar de 0.1455.

4.4. Evaluación de metodología

Con base en los resultados obtenidos en cada uno de los entrenamientos del conjunto de redes, es posible comenzar con su análisis para determinar si tienen un buen desempeño, así como verificar o rechazar a la hipótesis de esta investigación. Para esto se utilizó las métricas de U de Mann-Whitney y la Q de Cochran.
A continuación en la tabla (4.12) se muestran los resultados de la prueba de Mann-Whitney para conocer si existe relación entre el éxito las clasificaciones realizadas y las probabilidades asignadas a cada paciente. Los resultados de sensibilidad para cada arreglo y clase se presentan en la tabla (4.13), estos valores muestran qué clase puede ser identificada más facilmente por un arreglo dado. Mientras que en la tabla (4.14) se presentan los resultados de la prueba de Cochran, para verificar si existe alguna diferencia significativa entre la clasificación de 3 arreglos distintos; esto es principalmente para conocer si, efectivamente, el uso de arreglos de más de dos regiones resulta más conveniente que los arreglos de pares de órganos.

Amigdalas						
	AD		CN		MCI	
	Clasificación	Probabilidad	Clasificación	Probabilidad	Clasificación	Probabilidad
1	AD	0.581560648	MCI	0.545316495	CN	0.401464333
2	CN	0.799400475	MCI	0.626680097	AD	0.752853669
3	AD	0.637601107	MCI	0.614607487	CN	0.555393545
4	AD	0.683543873	AD	0.796230296	MCI	0.453898756
5	AD	0.693391454	MCI	0.717914354	CN	0.430561878
6	AD	0.872132863	AD	0.712758424	MCI	0.53055818
7	MCI	0.430758834	MCI	0.382574902	CN	0.533511432
8	AD	0.793559137	MCI	0.537360591	CN	0.550105171
9	MCI	0.536356194	MCI	0.722310836	AD	0.740523371
10	MCI	0.704709108	CN	0.840188704	CN	0.904837347
11	AD	0.631908878	MCI	0.641238255	AD	0.6009261
12	CN	0.534220964	AD	0.50744346	AD	0.479751538
13	AD	0.929361536	AD	0.604284639	MCI	0.902495162
14	MCI	0.806228799			MCI	0.674777838
15					CN	0.55686851

Tabla 4.9: Resultados de clasificación por medio del análisis de las amígdalas.

Hipocampo - Insula						
	AD		CN		MCI	
	Clasificación	Probabilidad	Clasificación	Probabilidad	Clasificación	Probabilidad
1	CN AD	0.991118753 0.999962383	CN MCI	0.986693232 0.878493161	MCI CN	0.983562243 0.586839149
3	AD	0.999991021	CN	0.935601088	MCI	0.748949641
4	AD	0.999997212	CN	0.719525868	CN	0.504320427
5	AD	0.930734679	CN	0.998035469	AD	0.506786015
6	CN	0.882529846	CN	0.730785389	MCI	0.999987243
7	AD	0.898951856	AD	0.83155635	AD	0.549239759
8	MCI	0.997440083	AD	0.999563436	MCI	0.978192927
9	AD	0.972774881	CN	0.999999881	CN	0.999188414
10	AD	0.991560794	CN	0.967587856	MCI	0.947123466
11	AD	0.997824238	CN	0.625154098	AD	0.992991588
12	AD	0.496857429	MCI	0.985379906	AD	0.937844012
13	MCI	0.991436955	CN	0.999996222	MCI	0.896488499
14	AD	0.999850743			MCI	0.998424437
15					MCI	0.578316988

Tabla 4.10: Resultados de clasificación a partir del análisis de las regiones de las ínsulas y los hipocampos.

6 Regiones						
	AD		CN		MCI	
	Clasificación	Probabilidad	Clasificación	Probabilidad	Clasificación	Probabilidad
1	MCI	0.861175645	CN	0.95003443	AD	0.999430185
2	AD	0.766003969	CN	0.986995272	MCI	0.795732936
3	MCI	0.956076196	MCI	0.998238192	MCI	0.643030247
4	CN	0.986226085	MCI	0.811716232	CN	0.911162656
5	AD	0.999619038	AD	0.797703557	MCI	0.999814957
6	AD	0.867846102	MCI	0.68704776	MCI	0.999657769
7	CN	0.518009957	AD	0.998618377	CN	0.997673794
8	AD	0.998079586	CN	0.773801813	AD	0.999925274
9	MCI	0.847896753	CN	0.992038459	AD	0.858580758
10	CN	0.999234408	MCI	0.99215464	CN	0.973040395
11	MCI	0.999660498	MCI	0.789332359	MCI	0.807540789
12	AD	0.998843156	CN	0.667865916	MCI	0.9999998
13	CN	0.991822117	CN	0.998197712	AD	0.996132903
14	MCI	0.858810035			MCI	0.999910702
15					MCI	0.99910717

Tabla 4.11: Resultados de clasificación por parte del arreglo de 6 regiones de interés

Accuracy, sensitivity y U de Mann-Whitney						
Modelo		Ínsulas	Hipocampos	Amígdalas	Hipocampo-Ínsula	6 Regiones
Accurcy (%)		30.95	40.47	30.95	64.28	45.23
U de Mann-Whitney	Valor Z	-0.476128	-0.11531	-1.80929	-1.2731	-0.54331
	Valor P	0.6339	0.9082	0.0704	0.1014	0.5869

Tabla 4.12: Valores de accuracy y valor P obtenido de la prueba U de Mann - Whitney

Amigdalas			
	VP	FP	Sensibilidad
AD	8	8	0.5
CN	1	9	0.1
MCI	4	12	0.25
Hipocampos			
AD	6	6	0.5
CN	2	6	0.25
MCI	9	13	0.409090909
Ínsulas			
AD	4	8	0.333333333
CN	3	14	0.176470588
MCI	6	7	0.461538462
Hipocampo - Ínsula			
AD	10	6	0.625
CN	9	5	0.642857143
MCI	8	4	0.666666667
6 Regiones			
AD	5	6	0.454545455
CN	6	7	0.461538462
MCI	8	10	0.444444444

Tabla 4.13: Número de verdaderos positivos, falsos positivos y valores de sensibilidad para cada arreglo de RNCs y cada clase

Modelos a comparar	Valor Q de Cochran	Valor P
Ínsula-Hipocampo / 6 regiones / Amigdalas	8.2222	0.0163
Ínsula-Hipocampo / 6 regiones / Hipocampos	4.8	0.0907
Ínsula-Hipocampo / 6 regiones / Ínsulas	8.4571	0.0145
Amigdalas / Hipocampos / Ínsulas	1.2307	0.5404

Tabla 4.14: Resultados de prueba Q de Cochran en tres grupos de comparación.

Capítulo 5

Discusión de resultados

En este capítulo se discuten los resultados obtenidos, los cuales se resumen a continuación.
Comenzando con los experimentos para determinar la arquitectura usada en el desarrollo de este trabajo, se presentan algunos de los experimentos más relevantes.
Para la prueba número uno, el comportamiento de entrenamiento, muestra un valor de accuracy con comportamiento ascendente que podría indicar que continuará mejorando al ampliar el tiempo de entrenamiento figura(4.1), sin embargo, la reducción del error es lento considerando la duración de este proceso.
En el caso de la segunda prueba, utilizando la misma configuración de red, se evidencía el cambio que se tiene al variar la tasa de aprendizaje. En la figura (4.3) se muestra que este único cambio provoca una ralentización del aprendizaje del modelo o incluso una disminución en la capacidad de la red. Es por esto que resulta conveniente el uso de un optimizador que sea capaz de modificar el ritmo de aprendizaje del modelo.
En la prueba número tres se modificó la arquitectura utilizada en la red, a pesar de tratarse de un modelo más sencillo, en la figura (4.4) se muestra un crecimiento más rápido en los valores de accuracy del modelo, con una tendencia de crecer al aumentar el tiempo de entrenamiento.
Para la prueba número 4, nuevamente se probó una arquitectura similar a la utilizada en la prueba número 1, con la diferencia de utilizar la función de costo del error cuadrático medio; sin embargo, se puede observar tanto en la figura (4.5) como en la tabla 4.4, que este cambio afecta completamente el desempeño del modelo. Finalmente la prueba número 5, se utilizó la misma arquitectura de la prueba 3, cambiando la función de activación de las capas densas por la función sigmoide, logrando así un comportamiento similar a dicha prueba. Sin embargo, esta prueba muestra un ritmo de aprendizaje más lento en comparación de la prueba 3.

Con base en la revisión del desempeño de las arquitecturas probadas, se determinó que la arquitectura con mejor desempeño es la correspondiente a la prueba número 3. Esta a pesar de tener una diferencia entre los valores de accuracy de entrenamiento y validación , lo cual podría sugerir un ligero sobre ajuste, mantiene un comportamiento ascendente durante el entrenamiento. Esto permite inferir que aumentará un poco más en un entrenamiento más largo, además de esto, es importante mencionar que dicha diferencia en los valores de accuracy se presenta en la gran mayoría de experimentos realizados con distintas arquitecturas, por lo que este comportamiento puede deberse a la cantidad de información disponible para realizar el proceso de entrenamiento.
Como ya se mencionó, en el caso de la prueba número uno, existe consistencia entre los valores de accuracy que podría indicar un entrenamiento más conveniente, por lo cual se realizó una prueba extra con la misma arquitectura para un entrenamiento de mil épocas para verificar su comportamiento y determinar si se trata de una mejor opción; sin embargo, los resultados de este entrenamiento mostraron un crecimiento limitado (figura 4.2), por lo tanto se optó por la arquitectura de la prueba 3.

Finalmente, de los resultados de Optuna se puede observar las ligeras variaciones que tiene la función de costo en el espacio de parámetros de cada red, dado el uso de una base de datos en específico.

5.1. Resultados arreglos de RNCs

Dados los resultados en las tablas (4.7-4.11), es posible concluir que ciertos modelos poseen mayor capacidad para discriminar algunas clases de las demás, obteniendo un mayor número de aciertos para esos pacientes. Para comprobar esta diferencia en la capacidad de los modelos para identificar cada clase, hace falta mencionar los resultados de sensibilidad mostrados en la tabla (4.13). Para el caso del arreglo a cargo de las ínsulas, se para la clase MCI un valor de sensibilidad del 46.15 %, mientras que para los hipocampos y amígdalas fue la clase AD; no obstante, en ambos casos el valor máximo de sensibilidad fue de 50 %. Esto se debe a que en ningún caso, el número de clasificaciones correctas de una clase dada supera las veces que el modelo cometió errores al asignar dicha clase a otros pacientes.
En el caso de los arreglos para más de 2 regiones, el que obtuvo mejor desempeño es el modelo Hipocampo-ínsula con un valor de sensibilidad superior al 60 % para las tres clases, siendo MCI la clase con mayor valor con 66.66 %.

A partir de los resultados clasificación, se puede observar que no existe una relación directa entre la probabilidad asignada a cada paciente y el éxito de dicha clasificación. Por lo que para apoyar o refutar esta observación, se utilizó la prueba U de Mann-Whitney, cuya hipótesis nula establece que no existe diferencia significativa entre probabilidad obtenida por los pacientes clasificados correcta e incorrectamente. En consecuencia se obtuvieron los resultados de la tabla (4.12) cuyos valores P de cada arreglo fueron de 63.39 % para el arreglo de ínsulas, 90.82 % para los hipocampos, para el arreglo de Hipocampo-Ínsula es de 10.14 %, mientras que para el arreglo de 6 regiones se obtuvo un valor P de 58.69 %; en el caso del arreglo de las amígdalas fue de sólo 7 %. Lo cual indica que no existe relación entre las probabilidades obtenidas y el éxito de clasificación. Esto podría ser consecuencia de escoger un método no óptimo para la combinación de probabilidades o de la falta de generalidad en las redes; por otra parte, con base en la expresión del teorema de Bayes para RNCs (ec. 3.6) la razón de que su resultado sea elevado es que las probabilidades de una clase específica para cada uno de los parches sea alta, incluso si no es correcta. Esto es consecuencia de la falta de generalidad de las redes, lo cual se mencionó al inicio de este capítulo, provocando que la red cometa errores en la clasificación.

Dados los resultados de clasificación, se recurre a otro tipo de métrica para poder concluir si el uso de los arreglos de 6 regiones y el arreglo hipocampo-ínsula, para este estudio, permitió mejores resultados en comparación con los arreglos de dos regiones. Para esto es que se utilizó la prueba Q de Cochran, cuya hipótesis nula es que no existe diferencia significativa entre tres de los arreglos utilizados con respecto a la cantidad de aciertos de clasificación. La tabla (4.14) muestra los resultados al comparar distintos grupos de arreglos. Se puede observar que al comparar los arreglos de 6 regiones e Ínsula-Hipocampo con el de amígdalas e ínsulas, se obtuvieron valores P de 0.0163 y 0.0145 respectivamente. Y al compararlos con los hipocampos, se obtuvo un valor P de 0.09, lo cual era de esperarse ya que se trata de la principal región del cerebro afectada por esta enfermedad.

Una de las limitaciones de este estudio es la baja disponibilidad de volúmenes de rs-fMRI, ya que

gran parte de las bases de datos públicas sólo cuentan con volúmenes de imágenes estructurales, de las cuales se pueden obtener mayor cantidad de información. Además del recurso computacional con el que se contó, ya que influye directamente en el tiempo de cómputo durante el entrenamiento de las 16 redes usadas en distintas ocasiones en este proyecto.

Finalmente, si hubiera sido posible acceder a herramientas de segmentación automática para poder evitar el factor de error humano al escoger los parámetros arbitrarios de las regiones que delimitaban los órganos de interés, es posible que los resultados de clasificación hubieran mejorado. Aunado a esto, es necesario tomar en cuenta que, debido a la cantidad de prueba correspondientes a cada clase, los resultados presentan cierto sesgo

5.2. Comparativa con trabajos del estado del arte

Dada la modalidad de imagen utilizada en este trabajo y el tipo de clasificación, es complicado comparar los resultados obtenidos con la tabla (2.1) ya que pocos de ellos utilizan el mismo tipo de imagen, además de ocupar estrategias más sofisticadas para mitigar la problemática de la poca cantidad de volúmenes disponibles. No obstante, entre los trabajos que reportan clasificación multiclase se encuentra el uso de transfer learning como en el caso de Jain et al. (2019b) usando la arquitectura de VGG 16 de dos dimensiones, y reporta un valor de accuracy de 95.73 %, usando imágenes estructurales de resonancia magnética para discriminar entre las tres clases utilizadas en este trabajo; por otro lado Ramzan et al. (2019e) realiza la estadificación de 5 etapas de Alzheimer usando imágenes rs-fMRI y entrenando desde cero una ResNet-18, obteniendo un accuracy promedio de 97.61 %. Así mismo Alorf y Khan (2022c) utilizando una red Stacked sparse auto-encoder y rs-fMRI reportan un accuracy de 76.18 % al discriminar entre 6 clases; mientras que Thakur y Snekhalatha (2022b) reportan un valor de accuracy de 98.44 % al diferenciar entre 4 clases al usar transfer learning de una ResNet-50 e imágenes PET. Por lo cual al comparar los resultados de este trabajo con los ya mencionados en específico, se ve totalmente superado, sin embargo, es posible retomar lo reportado por Fathi et al. (2022) luego de su revisión del estado del arte. En este review, los autores mencionan que la media de accuracy reportada para trabajos en clasificación multiclase es de 87.39 % con valores que van desde 47.42 % hasta 99.89 %, por lo cual, el valor más alto de accuracy de 64.28 % obtenido en este trabajo se encuentra por debajo la media reportada por Fathi.
No obstante, es notable destacar la metodología utilizada y la arquitectura de las redes, la cual resulta mucho más simple que las empleadas en la mayoría de los artículos revisados.
Además de las estrategias utilizadas en este trabajo, se propuso una metodología de combinación de probabilidades distinta a la utilizada en trabajos relacionados con el uso de parches, ya que en su mayoría utilizan estrategias de votaciones o simplemente promediando las probabilidades obtenidas.

Por lo tanto, con base en los resultados discutidos en este trabajo, es posible concluir que la hipótesis de esta investigación es rechazada, ya que por medio de la metodología utilizada no es posible obtener resultados superiores a los reportados en el estado del arte con imágenes estructurales; además de la desventaja de poca disponibilidad de bases de datos de rs-fMRI, incluso utilizando técnicas para aumentar el número de muestras, como la técnica de parches y data augmentation. No obstante debe reconocerse que las imágenes rs-fMRI poseen una gran cantidad de información útil para el diagnóstico de este enfermedad, la cual puede ser mejor aprovechada por medio otras técnicas de preprocesado, así como otros modelos de análisis que consideran la baja disponibilidad de este tipo de imágenes.
Además de lo anterior, se encontró que el uso conjunto de las regiones de hipocampos e ínsulas permiten mejores resultados que el uso simultáneo de las 6 regiones de interés, lo cual podría resultar contra intuitivo, no obstante esto está relacionado con el tamaño de las regiones de interés; ya que regiones como las amígdalas tienen dimensiones reducidas en comparación con

Trabajo	Wang et al. (2020)	Mu et al. (2024)	Lien et al. (2023) MobileNet V2	Lien et al. (2023) NASNetMobile	El-Latif et al. (2023)	Arreglo propuesto
Tipo de RNC	3D	2D	2D con 3 canales	2D con 3 canales	2D con 3 canales	2D
Clasificación	Binaria	Multiclase	Binaria MCI-AD	Binaria MCI-AD	Multiclase	Multiclase
Tipo de imagen	MRI	MRI	SPECT	SPECT	MRI	rs-fMRI
Número de parámetros	0.24×10^6	18.1×10^6	3.44×10^6	5.3×10^6	6.5×10^6	12.48×10^6 ($3.1 \times 10^6 porred$)
Número de capas de convolución	14	4	19	56	2	4 (1 por red)
Accuracy (%)	89.8	95.2	68.43	68.77	95.93	64.28

Tabla 5.1: Comparación de resultados obtenidos con el método empleado y obtenidos con arquitecturas livianas en el estado del arte.

las ínsulas e hipocampos. Aunado al tamaño anatómico de estos órganos, también influye la resolución de las rs-fMRI, lo cual provoca que no sea posible obtener gran cantidad de parches de esta región y, en consecuencia, esto afecta a la calidad del entrenamiento de la redes correspondientes.

Finalmente, para contrastar los resultados obtenidos en este trabajo, se incluye en la tabla **??** una breve revisión de algunos modelos de RNCs livianos usados en distintos artículos para la estadificación del Alzheimer. En dicha tabla se puede observar el tipo de RNC usada, el tipo de clasificación hecha, la modalidad de imagen utilizada, así como el número de parámetros entrenables de la red, número de capas de convolución y el valor de accuracy reportado. Como es posible observar a lo largo de este trabajo, estos valores no determinan por completo la arquitectura ni el desempeño de la red, sin embargo, pueden servir como referencia de la complejidad de la red, así como del tiempo de entrenamiento de ellas.

Dados los resultados de accuracy de los distintos modelos, se puede confirmar que los obtenidos por la metodología de este trabajo no son suficientes para superar los resultados de estos modelos livianos. Sin embargo, es necesario notar que a pesar de ser clasificados como modelos sencillos, estos constan de arquitecturas complejas o profundas, además de usar otras modalidades de imágenes. La complejidad de estos modelos se ve reflejado en la cantidad de parámetros entrenables de dichas redes, así como en la cantidad de capas de convolución usadas, siendo la arquitectura más simple la correspondiente a este trabajo.
La siguiente arquitectura más simple es la reportada por El-Latif et al. (2023), formada por dos capas de convolución, dos capas max pooling, así como dos capas FC, teniendo por entrada imágenes MRI de dimensiones $150 \times 150 \times 3$, logrando un 95.93 % de accuracy usando más de 5 veces el número de parámetros que la arquitectura propuesta. Luego de esta se tiene la red propuesta en Wang et al. (2020), la cual consta de un total de 14 capas de convolución distribuidas a lo largo de varios bloques densos y cuenta con la menor cantidad de parámetros de los trabajos comparados.
Además, es relevante mencionar en específico las redes utilizadas por Lien et al. (2023), las cuales son RNCs usadas actualmente en dispositivos móviles, por lo que se trata de redes profundas estudiadas, diseñadas para tener un desempeño óptimo con la menor cantidad de recursos posibles. Usando imágenes SPECT son capaces de obtener hasta un 68.77 % de accuracy a la hora de realizar clasificación binaria entre los pacientes con daño cognitivo leve (MCI) y Alzheimer (AD).

Con base en esta breve revisión de estos resultados obtenidos con redes livianas, es posible observar que a pesar de no conseguir superar los resultados de estas RNCs, la propuesta de este proyecto logra resultados relevantes con respecto a otras alternativas significativamente más complejas como la MobileNet V2 y la NASNetMobile las cuales la superan por poco más del 4 % de accuracy, usando unos cuantos miles de parámetros menos.

Capítulo 6

Conclusiones

Dado el impacto de la enfermedad de Alzheimer, en este trabajo se busca el proponer un modelo de red neuronal convolucional simple, capaz de obtener resultados comparables con el estado del arte al discriminar entre tres categorías; dos de las cuales forman parte del espectro de Alzheimer. La principal diferencia de este estudio es el uso de imágenes funcionales de resonancia magnética en estado de reposo, esto para intentar rechazar o confirmar la hipótesis central del proyecto: Es posible obtener mejores resultados de clasificación usando rs-fMRI con RNCs simples, en comparación con los reportados en el estado del arte usando imágenes estructurales y arquitecturas complejas para la enfermedad de Alzheimer. .

Con base en el estudio simultáneo de distintas combinaciones de regiones, se determinó que el uso conjunto de más de dos regiones de interés influye positivamente de manera significativa en la clasificación de los pacientes de prueba en comparación de los arreglos de dos regiones. Sin embargo, la evidencia indica que los arreglos de dos o más órganos pares no muestran consistencia entre la probabilidad y clase asignada al paciente. Esto es consecuencia de la principal limitación de este proyecto: La baja cantidad de volúmenes disponibles de este tipo de imágenes. No obstante, a pesar de esta limitante, el nivel más alto de exactitud y sensibilidad obtenido en este trabajo, se coloca dentro del rango reportado en el estado del arte, aunque por debajo de la media.
Uno de los logros a destacar de este trabajo es la obtención de estos resultados para clasificación multiclase, utilizando una arquitectura de red considerablemente más simple que las encontradas comúnmente en distintos artículos. Así mismo se propone una metodología matemática más formal para la obtención de una probabilidad global, dadas las probabilidades individuales al usar la técnica de parches, ya que en los trabajos de este tipo se utilizan técnicas como el conteo de votos y promediar las probabilidades obtenidas.
Del mismo modo, en este trabajo, se obtuvo que las regiones con mejores resultados al usarse en conjunto, son los hipocampos e ínsula, siendo estos órganos los más afectados por esta enfermedad, por lo cual sería recomendable poner especial atención en dichas zonas.

De acuerdo con los resultados discutidos en el capítulo anterior, no es posible confirmar la hipótesis de esta investigación, sin embargo, existen otras técnicas de preprocesado que pueden aprovechar de mejor manera la información contenida en las rs-fMRI, como la extracción de redes cerebrales para el análisis de regiones correlacionadas, técnica usada en gran parte de los trabajos que usan esta modalidad de imagen. Por lo tanto se concluye que la técnica usada en este trabajo no fue la óptima para aprovechar el potencial de estas imágenes. Además de esto, es posible afirmar que a pesar de no obtener los resultados deseados, la técnica de parches tiene un gran potencial de análisis de cambios sutiles de información en imágenes; no obstante, es necesario cuidar las dimensiones de estos, para evitar que la tarea de condensación de información en las

capas de convolución resulte contraproducente.
Dadas las alterativas para suplir la falta de información; es importante mencionar que la técnica de transfer learning, a pesar de ser una de las más usadas, es dificil de implementar en conjunto con el uso de parches en lugar de imágenes completas. Esto, principalmente, es debido a las dimensiones de los parches y las imágenes para las que fueron creadas estas arquitecturas; no basta con el uso de padding, ya que no hay forma de asegurar que la información obtenida en las convoluciones tenga relación directa con la contenida por el parche, en especial al usar subimágenes tan reducidas.

Perspectiva a futuro

Dados los resultados de esta tesis, se propone como trabajo a futuro el uso de algún modelo de clasificación que tenga una menor dependencia de la cantidad de datos con el que es entrenado, ya que sigue siendo complicado el tener acceso a más imágenes de esta modalidad; entre los modelos sugeridos para probar están las máquinas de soporte vectorial, las cuales son conocidas por tener resultados aceptables incluso sin grandes bases de datos.
Además de esto, se propone el uso de técnicas que permitan un mejor aprovechamiento de la información en las rs-fMRI, como la extracción de redes cerebrales, el uso de series de tiempo, etc. Finalmente se sugiere la exploración de otros tipos de redes neuronales para el análisis de información en otros niveles de abstracción, como las redes neuronales residuales, de grafos, entre otros.

Bibliografía

[1] Andrade-Guerrero, J., Santiago-Balmaseda, A., Jeronimo-Aguilar, P., Vargas-Rodríguez, I., Cadena-Suárez, A. R., Sánchez-Garibay, C., Pozo-Molina, G., Méndez-Catalá, C. F., Cardenas-Aguayo, M. D. C., Diaz-Cintra, S., Pacheco-Herrero, M., Luna-Muñoz, J., Soto-Rojas, L. O. (2023). Alzheimer's Disease: An Updated Overview of Its Genetics. International Journal of Molecular Sciences, 24(4), 3754. https://doi.org/10.3390/ijms24043754

[2] De Salud, S. (2021, 5 octubre). Enfermedad de Alzheimer, demencia más común que afecta a personas.gob.mx. Recuperado 23 de marzo de 2023, de https://www.gob.mx/salud/es/articulos/enfermedad-de-alzheimer-demencia-mas-comun-que-afecta-a-personas-adultas-mayores?idiom=es.

[3] Rajan, K. B., Weuve, J., Barnes, L. L., McAninch, E. A., Wilson, R. S., Evans, D. A. (2021). Population estimate of people with clinical Alzheimer's disease and mild cognitive impairment in the United States (2020–2060). Alzheimer's Dementia, 17(12), 1966-1975. https://doi.org/10.1002/alz.12362

[4] Hendriks, S., Peetoom, K., Bakker, C., Van Der Flier, W. M., Papma, J. M., Koopmans, R. T., Verhey, F. R., De Vugt, M., Köhler, S., Withall, A., Parlevliet, J. L., Uysal-Bozkir, Ö., Gibson, R. C., Neita, S. M., Nielsen, T. R., Salem, L. C., Nyberg, J., Lopes, M. A., Dominguez, J. C., . . . Ruano, L. (2021). Global Prevalence of Young-Onset Dementia. JAMA Neurology, 78(9), 1080. https://doi.org/10.1001/jamaneurol.2021.2161

[5] 21 de septiembre, Día Mundial del Alzheimer. (s. f.). https://www.insp.mx/avisos/21-de-septiembre-dia-mundial-del-alzheimer

[6] Breijyeh, Z., Karaman, R. (2020). Comprehensive Review on Alzheimer's Disease: Causes and Treatment. Molecules, 25(24), 5789. https://doi.org/10.3390/molecules25245789

[7] 2023 Alzheimer's disease facts and figures. (2023). Alzheimer's Dementia, 19(4), 1598-1695. https://doi.org/10.1002/alz.13016

[8] Tombaugh, T. N., McIntyre, N. E. (1992). The Mini-Mental State Examination: A Comprehensive Review. Journal of the American Geriatrics Society, 40(9), 922–935. https://doi.org/10.1111/j.1532-5415.1992.tb01992.x

[9] Imagen por Resonancia Magnética (IRM). (s.f.). National Institute Of Biomedical Imaging And Bioengineering. https://www.nibib.nih.gov/espanol/temas-cientificos/imagen-por-resonancia-magn%C3%A9tica-irm

[10] Smith, N. B., Webb, A. (2010). Introduction to Medical Imaging: Physics, Engineering and Clinical Applications. Cambridge University Press.

[11] Seic. (2022, 7 marzo). ¿De qué dependen el T1 y el T2? https://ecocardio.com/documentos/biblioteca-preguntas-basicas/preguntas-al-radiologo/916-de-que-dependen-t1-y-t2.html

[12] M, M. G., A, M. F., K, T. A., C, E. B. (2005). CALCULO DE TIEMPOS T1 y T2 IN VITRO. Revista Chilena de Radiología, 11(3). https://doi.org/10.4067/s0717-93082005000300003

[13] Serai, S. D. (2021). Basics of magnetic resonance imaging and quantitative parameters T1, T2, T2*, T1rho and diffusion-weighted imaging. Pediatric Radiology, 52(2), 217-227. https://doi.org/10.1007/s00247-021-05042-7

[14] Gómez, F. J., Manjón, J. V., Mollá, E., Dosdá, R., Robles, M., Luis, M. B. (2001). Mapas de resonancia magnética funcional obtenidos con PC. Radiología (Madr., Ed. Impr.);43(2): 55-61, Mar. 2001. Ilus | IBECS.https://pesquisa.bvsalud.org/portal/resource/pt/ibc-756

[15] Viviano, J. D., Boyer, R., Calarco, N., Gold, J. M., Foussias, G., Bhagwat, N., Stefanik, L., Hawco, C., DeRosse, P., Árgyelán, M., Turner, J. A., Chavez, S., Kochunov, P., Kingsley, P. B., Zhou, X., Malhotra, A. K., Voineskos, A. N. (2018b). Resting-State Connectivity Biomarkers of Cognitive Performance and Social Function in Individuals With Schizophrenia Spectrum Disorder and Healthy Control Subjects. Biological Psychiatry, 84(9), 665-674. https://doi.org/10.1016/j.biopsych.2018.03.013

[16] Van Den Heuvel, M. P., Pol, H. E. H. (2010b). Exploring the brain network: A review on resting-state fMRI functional connectivity. European Neuropsychopharmacology, 20(8), 519-534. https://doi.org/10.1016/j.euroneuro.2010.03.008

[17] Al-Arfaj, H. K., Al-Sharydah, A. M., Al-Suhibani, S. S., Alaqeel, S., Yousry, T. (2023). Task-Based and Resting-State Functional MRI in Observing Eloquent Cerebral Areas Personalized for Epilepsy and Surgical Oncology Patients: A Review of the Current Evidence. Journal Of Personalized Medicine, 13(2), 370. https://doi.org/10.3390/jpm13020370

[18] Vemuri, P., Jones, D. T., Jack, C. o. R. (2012). Resting state functional MRI in Alzheimer's Disease. Alzheimer's Research Therapy, 4(1). https://doi.org/10.1186/alzrt100

[19] Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Mehmood, Z. (2019). A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks. Journal Of Medical Systems, 44(2). https://doi.org/10.1007/s10916-019-1475-2

[20] Mishra, P. (2021, 10 diciembre). Why are Convolutional Neural Networks good for image classification? Medium. https://medium.datadriveninvestor.com/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8

[21] McCulloch, W. S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The Bulletin Of Mathematical Biophysics, 5(4), 115-133. https://doi.org/10.1007/bf02478259

[22] Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6), 386-408. https://doi.org/10.1037/h0042519

[23] Agarwal, R. (2023, 13 septiembre). Complete Guide to the Adam Optimization Algorithm. Built In. https://builtin.com/machine-learning/adam-optimization

[24] Hubel, D. 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

[25] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4), 193-202. https://doi.org/10.1007/bf00344251

[26] Daniel. (2021, diciembre 16). Convolutional Neural Network: definición y funcionamiento. Formación en ciencia de datos | Datascientest.com; DataScientest. https://datascientest.com/es/convolutional-neural-network-es

[27] Ayuso, A. L., de la Fuente López, E. (2022). SEGMENTACIÓN AUTOMÁTICA DE IMÁGENES DE RESONANCIA MAGNÉTICA CEREBRAL MEDIANTE REDES NEURONALES CONVOLUCIONALES. Universidad de Valladolid.

[28] DeepAI. (2020, 25 junio). Padding (Machine learning). DeepAI. https://deepai.org/machine-learning-glossary-and-terms/padding

[29] Leterme, H., Polisano, K., Perrier, V., Alahari, K. (2022). On the Shift Invariance of Max Pooling Feature Maps in Convolutional Neural Networks. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2209.11740

[30] Ng, C., Lo, W. (2019). Effect of Image Distortion on Facial Age and Gender Classification Performance of Convolutional Neural Networks. IOP Conference Series: Materials Science And Engineering, 495, 012029. https://doi.org/10.1088/1757-899x/495/1/012029.

[31] Sousa, M. C. F. (2022, 30 marzo). Visualizing the Fundamentals of Convolutional Neural Networks. Medium. https://towardsdatascience.com/visualizing-the-fundamentals-of-convolutional-neural-networks-6021e5b07f69: :text=Stride,big%20overlap%20between%20receptive%20fields.

[32] Basaia, S., Agosta, F., Wagner, L., Canu, E., Magnani, G., Santangelo, R., Filippi, M. (2019). Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks. NeuroImage: Clinical, 21, 101645. https://doi.org/10.1016/j.nicl.2018.101645

[33] Jain, R., Jain, N., Aggarwal, A., Hemanth, D. J. (2019). Convolutional neural network based Alzheimer's disease classification from magnetic resonance brain images. Cognitive Systems Research, 57, 147-159. https://doi.org/10.1016/j.cogsys.2018.12.015

[34] Initiative, A. D. N. (2018). Multimodal and Multiscale Deep Neural Networks for the Early Diagnosis of Alzheimer's Disease using structural MR and FDG-PET images. Scientific Reports, 8(1). https://doi.org/10.1038/s41598-018-22871-z

[35] Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Mehmood, Z. (2019b). A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks. Journal Of Medical Systems, 44(2). https://doi.org/10.1007/s10916-019-1475-2

[36] Alorf, A., Khan, M. U. G. (2022). Multi-label classification of Alzheimer's disease stages from resting-state fMRI-based correlation connectivity data and deep learning. Computers in Biology and Medicine,15106240. https://doi.org/10.1016/j.compbiomed.2022.106240

[37] Thakur, M., Snekhalatha, U. (2022). Multi-stage classification of Alzheimer's disease from 18F-FDG-PET images using deep learning techniques. Physical And Engineering Sciences In Medicine, 45(4), 1301-1315. https://doi.org/10.1007/s13246-022-01196-2

[38] Ramzan, F., Khan, M. U. G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., Mehmood, Z. (2019). A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting State fMRI and Residual Neural Networks. Journal of Medical Systems, 44(2). https://doi.org/10.1007/s10916-019-1475-2

[39] Hazarika, R. A., Abraham, A., Sur, S. N., Maji, A. K., Kandar, D. (2021). Correction to: Different techniques for Alzheimer's disease classification using brain images: a study. International Journal of Multimedia Information Retrieval, 10(4), 255-255. https://doi.org/10.1007/s13735-021-00222-5

[40] Salvi, M., Acharya, U. R., Molinari, F., Meiburger, K. M. (2021). The impact of pre- and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis. Computers in Biology and Medicine. https://doi.org/10.1016/j.compbiomed.2020.104129

[41] Ahmed, S., Choi, K. H., Lee, J. G., Kim, B. C., Kwon, G., Lee, K. C., Jung, H. G. (2019). Ensembles of Patch-Based Classifiers for Diagnosis of Alzheimer Diseases. DOAJ (DOAJ: Directory of Open Access Journals). https://doi.org/10.1109/access.2019.2920011

[42] Ahmed, S., Kim, B. C., Lee, K. C., Jung, H. G. (2020). Ensemble of ROI-based convolutional neural network classifiers for staging the Alzheimer disease spectrum from magnetic resonance imaging. PLOS ONE, 15(12), e0242712. https://doi.org/10.1371/journal.pone.0242712

[43] Zhang, Y., Qizhi, T., Liu, Y., Liu, Y., He, X. (2022). Diagnosis of Alzheimer's disease based on regional attention with sMRI gray matter slices. Journal of Neuroscience Methods, 365, 109376. https://doi.org/10.1016/j.jneumeth.2021.109376

[44] Petersen, R. C., Aisen, P. S., Beckett, L. A., Donohue, M., Gamst, A., Harvey, D., Jack, C. R., Jagust, W. J., Shaw, L. M., Toga, A. W., Trojanowski, J. Q., Weiner, M. W. (2010). Alzheimer's Disease Neuroimaging Initiative (ADNI): Clinical characterization. Neurology, 74(3), 201-209. https://doi.org/10.1212/wnl.0b013e3181cb3e25

[45] Mascali, D., DiNuzzo, M., Gili, T., Moraschi, M., Fratini, M., Maraviglia, B., Serra, L., Bozzali, M., Giove, F. (2015). Resting-state fMRI in dementia patients. Harvard Dataverse. Recuperado 23 de marzo de 2024, de https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/29352

[46] Hoja informativa sobre la enfermedad de Alzheimer | NIA. (s.f.). National Institute On Aging. https://www.nia.nih.gov/espanol/enfermedad-alzheimer/enfermedad-alzheimer

[47] Reiss, P. T., Stevens, M. H. H., Shehzad, Z., Petkova, E., Milham, M. P. (2010). On Distance-Based Permutation Tests for Between-Group Comparisons. Biometrics, 66(2), 636-643. https://doi.org/10.1111/j.1541-0420.2009.01300.x

[48] Nieto-Castañón, A. (2021). CONN functional connectivity toolbox: RRID SCR009550, release 21. https://doi.org/10.56441/hilbertpress.2161.7292

[49] T-Test, Chi-Square, ANOVA, Regression, Correlation. . . (s.f.). https://datatab.es/tutorial/mann-whitney-u-test

[50] Abrol, A., Bhattarai, M., Fedorov, A., Du, Y., Plis, S. M., Calhoun, V. D. (2020). Deep residual learning for neuroimaging: An application to predict progression to Alzheimer's disease. Journal of Neuroscience Methods, 339, 108701.https://doi.org/10.1016/j.jneumeth.2020.108701

[51] Odusami, M., Maskeliunas, R., Damaševičius, R., Krilavičius, T. (2021). Analysis of Features of Alzheimer's Disease: Detection of Early Stage from Functional Brain Changes in Magnetic Resonance Images Using a Finetuned ResNet18 Network. Diagnostics, 11(6), 1071.https://doi.org/10.3390/diagnostics11061071

[52] D.H. Lu, et al., Multimodal and multiscale deep neural networks for the early diagnosis of Alzheimer's disease using structural MR and FDG-PET images, Sci. Rep. 8 (2018), https://doi.org/10.1038/s41598-018-22871-z.

[53] Liu, M., Cheng, D., Yan, W., & Alzheimer's Disease Neuroimaging Initiative. (2018). Classification of Alzheimer's disease by combination of convolutional and recurrent neural networks using FDG-PET images. Frontiers in neuroinformatics, 12.https://doi.org/10.3389/fninf.2018.00035

[54] Karhunen, J., Raiko, T., Cho, K. (2015). Unsupervised deep learning. En Advances in Independent Component Analysis and Learning Machines (pp. 125–142). Elsevier.

[55] Ebrahimighahnavieh, M. A., Luo, S., Chiong, R. (2020). Deep learning to detect Alzheimer's disease from neuroimaging: A systematic literature review. Computer Methods and Programs in Biomedicine, 187(105242), 105242. https://doi.org/10.1016/j.cmpb.2019.105242

[56] Shen, D., Wu, G., Suk, H.-I. (2017). Deep learning in medical image analysis. Annual Review of Biomedical Engineering, 19(1), 221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

[57] Suk, H.-I., Lee, S.-W., Shen, D., Alzheimer's Disease Neuroimaging Initiative. (2014). Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. NeuroImage, 101, 569–582. https://doi.org/10.1016/j.neuroimage.2014.06.077

[58] Islam, J., Zhang, Y. (2018). Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. Brain Informatics, 5(2). https://doi.org/10.1186/s40708-018-0080-3

[59] Wang, Q., Li, Y., Zheng, C., Xu, R. (2021). DenseCNN: A Densely Connected CNN Model for Alzheimer's Disease Classification Based on Hippocampus MRI Data.AMIA ... Annual Symposium proceedings. AMIA Symposium,2020, 1277–1286.

[60] Lien, W., Yeh, C., Chang, C., Chang, C., Wang, W., Chen, C., & Lin, Y. (2023b). Convolutional Neural Networks to Classify Alzheimer's Disease Severity Based on SPECT Images: A Comparative Study. Journal Of Clinical Medicine, 12(6), 2218. https://doi.org/10.3390/jcm12062218

[61] El-Latif, A. A. A., Chelloug, S. A., Alabdulhafith, M., & Hammad, M. (2023b). Accurate Detection of Alzheimer's Disease Using Lightweight Deep Learning Model on MRI Data. Diagnostics, 13(7), 1216. https://doi.org/10.3390/diagnostics13071216