



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR
INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE MASTER

MASTER UNIVERSITARIO EN INVESTIGACIÓN
EN TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES

Armonización de medidas escalares de resonancia magnética de difusión

Autor:

D. David Martínez Amor

Tutor:

Dr. D. Santiago Aja Fernández

Valladolid, Septiembre 2019

TÍTULO: Armonización de medidas escalares de resonancia magnética de difusión

AUTOR: D. David Martínez Amor

TUTOR: Dr. D. Santiago Aja Fernández

DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería Telemática

Tribunal

PRESIDENTE: Dr. D. Juan Ignacio Arribas Sánchez

VOCAL: Dr. D. Pablo Casaseca de la Higuera

SECRETARIO: Dr. D. Rodrigo de Luis García

FECHA: Septiembre 2019

CALIFICACIÓN:

Resumen del TFM

Las medidas escalares que se derivan de los estudios de resonancia magnética de difusión (Anisotropía Fraccional, Difusividad Media), pueden variar en función de los parámetros de adquisición de los datos. Como consecuencia, datos obtenidos en distintos escáneres pueden no ser compatibles entre ellos, y, por lo tanto, no pueden ser usadas en estudios clínicos multicentro.

En el presente proyecto se plantea el uso de redes convolucionales y técnicas de deep learning para encontrar el mapeo existente entre dos adquisiciones realizadas con diferentes parámetros de tal manera que estas sean comparables.

Palabras clave

Resonancia magnética, difusión, aprendizaje automático, deep learning, CNN, FCN, U-Net

Abstract

Scalar measurements derived from diffusion magnetic resonance studies (Fractional Anisotropy, Mean Diffusivity) may vary depending on data acquisition parameters. As a consequence, data obtained from different scanners may not be compatible with each other, and therefore may not be used in multicenter clinical studies.

In the present project, the use of convolutional networks and deep learning techniques is proposed in order to find the existing mapping between two acquisitions made with different parameters in such a way that they are comparable.

Keywords

Magnetic resonance, diffusion, Aprendizaje Automático, deep learning, CNN, FCN, U-Net

Agradecimientos

Índice general

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Fases	2
1.4	Medios	3
2	Planteamiento del Problema	5
2.1	Biomarcadores cuantitativos de imagen (QIB)	5
2.2	Limitaciones de las medidas de DMRI como QIB	6
2.3	Variabilidad con los parámetros de adquisición	8
3	Teoría	11
3.1	Resonancia de difusión	11
3.1.1	Principio físico de la difusión	11
3.1.2	Diffusion Weighted Imaging (DWI)	12
3.1.3	Diffusion Tensor Imaging (DTI)	13
3.1.4	Medidas escalares	15
3.2	Aprendizaje Automático	17
3.2.1	Bases	18
3.2.2	Algoritmos	22
3.3	Redes neuronales	26
4	Armonización usando Aprendizaje Automático	33
4.1	Planteamiento del problema	33
4.2	Datos	34
4.3	Procedimiento	37
4.4	Resultados	38
4.5	Conclusiones	45
5	Armonización usando redes neuronales	47
5.1	Planteamiento del problema	47
5.2	CNNs	47
5.3	U-Net	49
5.4	Datos	55
5.5	Preprocesado de los datos	55
5.6	Procedimiento	56

5.7	TBSS	64
6	Conclusiones y líneas futuras de trabajo	83
6.1	Conclusiones	83
6.2	Líneas futuras	84

Capítulo 1

Introducción

1.1 Motivación

La resonancia magnética de difusión (DMRI) es una técnica de imagen que permite la cuantificación de la difusión de las moléculas de agua en los tejidos in vivo. Usando diferentes esquemas de adquisición y modelos permite el estudio de propiedades isotrópicas y anisótropas de diferentes tejidos, con especial relevancia de los órganos abdominales y en cerebro [1]. Debido a su capacidad para medir la anisotropía presente en los tejidos, las técnicas de DMRI aplicadas a estudios cerebrales proporcionan una información muy valiosa sobre la organización y la conectividad de las fibras nerviosas en la sustancia blanca. Estas técnicas han resultado especialmente atractivas para la comunidad científica en los últimos años, debido a los hallazgos que relacionan patologías neurológicas con alteraciones de la materia blanca [2, 3, 4].

Desde un punto de vista clínico, la cuantificación de las variaciones estructurales en la sustancia blanca habitualmente se realiza mediante medidas escalares que describen la difusión dentro de un vóxel. Las medidas más usadas basadas en el tensor de difusión (DT) son la Anisotropía Fraccional (FA), el coeficiente de difusión aparente (ADC) o la Difusividad Media (MD). Diversos estudios han puesto en evidencia cómo la variación de los parámetros relacionados con la adquisición puede afectar seriamente a la reproducibilidad y precisión de las medidas derivadas del DT. Esto limita el uso de estas medidas como biomarcadores clínicos, y por ello no pueden ser usadas en estudios clínicos multicentro. Por ello surge la idea de tratar de obtener medidas invariantes al proceso de adquisición.

1.2 Objetivos

El objetivo del presente trabajo es, mediante la aplicación de técnicas de Aprendizaje Automático, establecer un modo de mapear medidas provenientes de adquisiciones con distintos parámetros a un espacio común o entre ellas, de manera que se puedan comparar y sea viable aplicarlas en estudios clínicos con éxito. Para ello nos marcamos los siguientes subobjetivos:

1. **Entender la naturaleza de la difusión:** Para situar el problema, el primer paso es entender el proceso físico que subyace y cómo se generan las medidas escalares de difusión. Entender la naturaleza de los datos es vital para poder elegir y generar un modelo de Aprendizaje Automático con éxito.
2. **Crear un modelo funcional con Aprendizaje Automático:** Desarrollar un modelo que tome como entrada algún tipo de medida escalar de difusión en sus distintas variedades y nos dé a la salida los datos mapeados a un espacio común donde sean comparables.
3. **Comprobar y verificar resultados:** Habrá que comprobar que los resultados numéricos mejoran con distintas validaciones, y como prueba final usar esos datos en un estudio clínico y ver que mejoran respecto a los originales.

1.3 Fases

- Estudio de los datos y preprocesado: Se estudiarán los datos de los que se dispone (pseudo sintéticos y reales), su naturaleza y la casuística particular de las imágenes. Asimismo, se realizarán las técnicas de preprocesado que se consideren necesarias.
- Experimentos con distintos modelos: En esta etapa se planteará el uso de distintos algoritmos que podrían ser de utilidad para el caso en concreto, y se desarrollarán modelos candidatos a realizar el mapeo.
- Selección del modelo final: De entre los modelos propuestos, se evaluarán los que mejor funcionen mediante validación y métricas del error de las predicciones. Para ese modelo se explicará en detalle la arquitectura propuesta y los parámetros con los que se configura.
- Puesta a prueba en un estudio: Para verificar la validez del modelo, usaremos las imágenes generadas por el mismo en un estudio clínico y compararemos los resultados con los de las imágenes originales.

1.4 Medios

Para el presente trabajo se propone el uso de las siguientes tecnologías y herramientas:

- Matlab, para lectura, preprocesado y graficado de los datos, tanto a priori como a posteriori. También para el desarrollo de scripts para shell/bash.
- Python, con las librerías Pandas, Numpy, Scipy, Tensorflow y Keras. Los modelos y arquitecturas de Aprendizaje Automático se desarrollarán en este lenguaje. Las redes neuronales que se propongan harán uso de Tensorflow (y Keras por ende) en su versión GPU para lograr un entrenamiento más rápido.
- MRtrix y FSL, herramientas avanzadas para el análisis de imágenes por resonancia magnética, que nos permitirán calcular medidas escalares, aplicar transformaciones sobre los datos, realizar visualizaciones en 3D de los mismos, y llevar a cabo los estudios clínicos.

Capítulo 2

Planteamiento del Problema

2.1 Biomarcadores cuantitativos de imagen (QIB)

Un biomarcador de imagen es una característica biológica que puede detectarse en una imagen y que es relevante para el diagnóstico. Los diagnósticos basados en información radiológica tradicionalmente se basan en la evaluación cualitativa de información obtenida de diferentes modalidades, como Resonancia Magnética (MRI), TAC o ultrasonidos. Estas técnicas se basan en la inspección visual del contraste entre tejidos y están relacionadas con los llamados biomarcadores cualitativos. Sin embargo, cuando hablamos de biomarcadores de imagen cuantitativos (*Quantitative Imaging Biomarkers*, QIB) se hace referencia a características medibles y cuantificables, extraídas de las propias imágenes y que pueden ser usadas para la evaluación (presencia o no, severidad, gradación) de una enfermedad, lesión o condición crónica, o para estudiar indirectamente indicadores relacionados con el desempeño cognitivo del cerebro. La obtención de biomarcadores cuantitativos en imagen conlleva muchas más tareas que la obtención de marcadores cualitativos, ya que es necesario definir y estandarizar protocolos de adquisición, técnicas específicas de análisis de datos, etc.

Estas características permiten la validación precisa y objetiva de métricas derivadas de las imágenes, basadas en parámetros con relevancia anatómica y fisiológica. En una segunda fase, los marcadores pueden permitir un análisis individualizado de la situación del paciente, planificación del tratamiento óptimo o estudios de respuesta al tratamiento [5]. La principal diferencia entre un biomarcador cualitativo y uno cuantitativo es que el segundo es una característica objetiva y medible, en contraste con la observación cualitativa (y a veces subjetiva) del primero. Las características usadas por los QIBs no tienen por qué ser directamente medibles. La mayor parte de las veces, las propiedades específicas de los tejidos o de las estructuras biológicas son parámetros ocultos que deben extraerse de las imágenes médicas mediante técnicas de análisis, procesado y modelado. Una de

las principales motivaciones detrás del uso de QIB es la llamada biopsia virtual. En algunas ocasiones, los QIBs proporcionan información de un tejido más precisa y más barata que las técnicas de biopsia tradicionales como, por ejemplo, en el caso del hígado. Una biopsia tradicional requiere hospitalización, tiene un coste y un riesgo asociados y solo analiza una pequeña fracción del tejido (1/50000 del tejido total del hígado), lo que puede ser problemático en afectaciones heterogéneas. Además, la biopsia está desaconsejada en pacientes con síndrome mielodisplásico (sangrado incontrolado) y en niños. Ante estos problemas, los QIBs surgen como una alternativa no invasiva y precisa.

Sin embargo, no cualquier medida derivada de imágenes médicas puede ser usada como biomarcador. Para ser un QIB válido, debe ser [5, 6]:

1. **Exacto:** (*accurate*) en términos de teoría de la estimación, se refiere a que el sesgo del error de estimación es pequeño. Consideramos dos clases de exactitud:
 - a) **Técnica:** Existe una correlación medible con una referencia aceptada (phantom, tejido o animal)
 - b) **Clínica:** la característica debe tener capacidad diagnóstica o bien capacidad para gradar una enfermedad.
2. **Preciso:** (*precisse*) debe ser repetible para un mismo sujeto y para distintos sujetos, en la misma sesión y en distintas.
3. **Robusto:** la medida es insensible a los parámetros de adquisición y a las técnicas de estimación.
4. **Reproducible:** baja variabilidad entre plataformas, sitios y equipos.
5. **De utilidad clínica:** la medida solo puede considerarse QIB si demuestra un papel importante en diagnóstico o monitorización de enfermedades. Debe estar sujeta a validación clínica y suponer una mejora sobre las medidas existentes.

Desde un punto de vista práctico, a los QIBs se les exige que se puedan obtener al menor coste posible, en el menor tiempo posible y que sean seguros e inocuos para el paciente.

2.2 Limitaciones de las medidas de DMRI como QIB

De acuerdo con las definiciones anteriores, el uso del DT presenta algunos problemas que limitan su uso como posibles QIB:

1. La inexactitud del modelo de difusión: el modelo de difusión Gaussiana no describe muchas de las situaciones presentes en la sustancia blanca, como el cruce de fibras o la difusión extracelular.
2. Factores de confusión externos: las medidas escalares dependen de un número de factores externos, entre los que están el ruido de adquisición y los parámetros de adquisición.
3. Las técnicas de estimación y análisis: la variabilidad de los algoritmos usados para estimación [7], postprocesado [8] y análisis [7, 9, 10, 11, 12] influye en la variabilidad de los resultados.

La principal limitación del modelo de tensor es la suposición de Gaussianidad, que no permite describir efectos de volumen parcial que comprendan haces con orientaciones no constantes o bien múltiples tractos nerviosos. Dada la complejidad del entramado de fibras nerviosas en la sustancia blanca, este tipo de efectos se producen en un elevado porcentaje de su volumen total (se estima que en torno a 2/3), lo que ha dado pie a la aparición de un conjunto de técnicas que tratan de superar esta limitación [13]. La aproximación más directa consiste en el muestreo de todo el espacio q (el espacio de todas las posibles direcciones de gradiente y del parámetro conocido como *b-value*), para estimar la difusión numéricamente usando técnicas basadas en DFT. Esta aproximación se conoce como Imagen por Espectro de Difusión (DSI, *Diffusion Spectrum Imaging*). El principal problema de este método es la necesidad de muestrear todo el espacio q , lo que lleva asociado altísimos tiempos de adquisición, imposibles de realizar en la práctica clínica. Alternativamente, puede emplearse un esquema de muestreo subóptimo conocido como HARDI (*High Angular Resolution Diffusion Imaging*), para el que el espacio q se muestrea con un único valor de b y un alto número de imágenes de gradiente (en torno a 50-60). Las principales técnicas HARDI son los tensores de orden superior, los modelos multitensores, la deconvolución esférica, *Diffusion Orientation Transform* (DOT), *Kurtosis Imaging* y las técnicas de *Q-Ball imaging* [14, 15]. Aunque no es posible la completa caracterización de la probabilidad de difusión, las técnicas HARDI tratan de describirla direccionalidad de los tractos nerviosos en la sustancia blanca, siendo capaces de proporcionar una estimación de la probabilidad de existencia de un haz de fibras alineado con cada dirección espacial, resolviendo el volumen parcial en cada vóxel [13]. Si bien las técnicas HARDI generalizan el modelo cuadrático a un modelo no paramétrico, siguen estando ligadas a la suposición de un comportamiento exponencial de la señal de difusión. Siendo esta suposición suficientemente precisa en un sentido local [16], no puede extrapolarse a todo el rango posible de gradientes, por lo que las medidas cuantitativas derivadas de los descriptores tipo HARDI se verán condicionadas por el valor concreto del módulo de los gradientes de difusión que se escojan.

La segunda limitación del modelo de DT es la dependencia de los valores escalares con los parámetros de adquisición. La cuantificación de las variaciones estructurales en la sustancia blanca habitualmente se realiza mediante medidas escalares que se obtienen a partir del DT, como es el caso de la Anisotropía Fraccional (FA), coeficiente de difusión

aparente (ADC) o la Difusividad Media (MD). Diversos estudios han puesto en evidencia cómo la variación de los parámetros relacionados con la adquisición puede afectar seriamente a la reproducibilidad y precisión de las medidas derivadas del DT. Entre estos parámetros se encuentran el número de repeticiones de cada escaneo [17], el valor de la sensibilidad de la difusión o *b-value* [18, 19, 20], el número de direcciones de gradiente usadas en el muestreo [21, 22, 18], el esquema empleado en estas direcciones [23], el espaciado o resolución de voxel [24, 25, 18] o incluso el modelo de escáner [11]. Existen otros factores no completamente controlables que son también fuente de variabilidad, entre los que se incluyen la variación interna de la señal de MRI y la posición, el movimiento o el ruido fisiológico del paciente [17].

La tercera fuente de variabilidad se centra en los algoritmos usados para el análisis de la sustancia blanca, incluyendo el análisis estadístico de los datos, los estimadores de la difusión y el método de selección de regiones o agrupaciones de fibras. El método específico de análisis de sustancia blanca (volumétrico o basado en tractos) representa otra gran fuente de variabilidad. Un análisis basado en vóxels (VBM) presenta ventajas y desventajas que son motivo de continuo debate [26, 27, 28]. La principal preocupación de estas técnicas es cómo asegurar que las diferencias observadas entre sujetos se deben realmente a diferencias estructurales en la sustancia blanca y no a un alineamiento incorrecto debido al proceso de registrado. Por otro lado, el análisis basado en tractos ha revelado que pueden aparecer inconsistencias en las medidas de un mismo sujeto cuando se emplean distintas técnicas de tractografía [9, 11]. También se ha visto que el método de selección de regiones usado también tiene un impacto sobre los valores de las medidas [29]. Finalmente, incluso asumiendo que el modelo de tensor es correcto, el método seleccionado para la estimación del DT y las suposiciones del modelo de ruido pueden sesgar la estimación del tensor [30]. Las medidas también se ven afectadas por cualquier clase de postprocesado, filtrado o regularización que se aplique sobre los DWI.

En resumen, los estudios realizados los últimos años muestran que hay evidencias muy sólidas de que el potencial uso de DMRI como herramienta diagnóstica está limitado por múltiples factores que, hoy por hoy, impiden que pueda ser utilizado como QIB en el cerebro.

2.3 Variabilidad con los parámetros de adquisición

Entre todas las posibles fuentes de variación que afectan a las medidas de difusión, en este trabajo nos centraremos en la dependencia con los parámetros de adquisición. Tal y como acabamos de referir, la variación en las condiciones de adquisición de los datos produce variaciones en las medidas escalares de difusión. Los parámetros que más afectan a las medidas son los siguientes:

1. El número de repeticiones (NEX): al aumentar el NEX aumenta la relación señal-ruido (SNR), lo que lleva asociando una disminución de la FA [17]. La variación de las medidas con la SNR ha sido estudiada en [23, 22].
2. La sensibilidad de difusión (*b-value*): un incremento en el *b-value* disminuye significativamente los valores de la MD, AD y RD [19, 20, 21, 18]. Se han observado también pequeñas variaciones en la FA [19, 31], especialmente en situaciones de baja SNR [DeS10].
3. Resolución: un tamaño de vóxel menos aumenta el valor de la FA [18, 32, 24].
4. Número de direcciones de gradiente: cuando aumenta el número de direcciones adquiridas se ha observado una disminución significativa en los valores de FA [23, 22, 18, 24] junto con una disminución de su varianza [18, 21].

Esta variabilidad se ha tratado de diferentes formas por distintos autores. Vollmar *et al.* [33] proponen el uso de un factor de escalado global para reducir el coeficiente de variación inter- e intra-ubicación tras una parcelación. De manera similar, en [34, 35] los autores proponen el uso de pesos en los conjuntos de datos adquiridos en diferentes sesiones y/o escáneres basados en su variabilidad. En [36] los autores encontraron que los efectos de la adquisición en distintas máquinas con distinta intensidad de campo pueden minimizarse con una corrección lineal aplicada a regiones específicas. Pohl *et al.* [37] proponen una armonización de las medidas para lidiar con el problema de la variabilidad entre escáneres. Su método se basa en el uso de datos cruzados de humanos y phantoms. Finalmente, una solución alternativa al problema de la armonización de datos multi-fuente es la representación de la señal de difusión en un espacio alternativo como, por ejemplo, en una base de armónicos esféricos (SH). Usando coeficientes de SH, Mirzaalian *et al.* [38, 39] calculan un conjunto de características rotacionalmente invariantes que puede usarse para estimar un mapeo lineal entre la señal adquirida en distintos escáneres.

Como solución, algunas de las metodologías alternativas propuestas para lograr medidas invariantes al proceso de adquisición pueden ser:

1. Propuesta de nuevas medidas de difusión basadas en adquisiciones compatibles con DT.
2. Propuesta de cálculo alternativo de medidas basadas en adquisiciones HARDI.
3. Armonización de datos adquiridos con distintos parámetros usando técnicas de Aprendizaje Automático.

Las dos soluciones anteriores se basan en la obtención de medidas alternativas a las tradicionalmente utilizadas en los estudios clínicos por los expertos médicos. Sin embargo, muchas veces se prefiere el uso de medidas más estandarizadas, como la FA o la MD,

de las que se tiene cierta intuición y en las que se basan la mayor parte de los estudios clínicos. El problema de estas medidas, tal y como se comentó anteriormente, es su variabilidad cuando cambian los parámetros de adquisición, tal y como se ha comentado en múltiples ocasiones a lo largo del documento. Esto limita su uso en bases de datos adquiridas en distintos centros y por distintos grupos.

En el presente trabajo, nos vamos a centrar en la última solución propuesta al problema es, armonizar los datos obtenidos en diferentes entornos: llevar datos con diferentes parámetros de adquisición a un marco de referencia común. Esto presupone la existencia de una función de transferencia entre conjuntos de datos. Para casos sencillos, algunos autores ya han encontrado algunas funciones de transferencia mediante funciones lineales [36] o mapeos lineales en el dominio de los SH [38, 39]. La fuente de variación con la que se va a probar es el número de direcciones de gradiente adquiridos.

Capítulo 3

Teoría

3.1 Resonancia de difusión

3.1.1 Principio físico de la difusión

La difusión es el proceso físico mediante el cual las partículas de materia se mueven de manera aleatoria de un lugar a otro dentro de un medio debido a movimientos térmicos moleculares, denominados movimientos brownianos [40]. Estos movimientos vienen descritos por la ley de Fick (3.1) de 1855, según la cual, cuando existen diferencias locales de soluto en una concentración se origina un flujo irreversible de las moléculas del soluto desde las zonas de mayor concentración a las de menor.

$$J = D \cdot \nabla c \quad (3.1)$$

En la ecuación 3.1, J es el flujo producido por la difusión que atraviesa una superficie, $D \left[\frac{mm^2}{s} \right]$ es el coeficiente de difusión que representa la facilidad con que el soluto se mueve en el medio disolvente y ∇c es el gradiente de concentración.

Einstein demostró que el coeficiente de difusión es proporcional al cuadrado medio de la distancia recorrida por las partículas ($\langle \Delta r^2 \rangle$) dividido entre el tiempo transcurrido (Δt) y el número de dimensiones del medio en el que se produce la difusión (n):

$$D = \frac{\langle \Delta r^2 \rangle}{2n\Delta t} \quad (3.2)$$

En un medio tridimensional ($n = 3$), la ecuación 3.2 queda de la forma:

$$D = \frac{\langle \Delta r^2 \rangle}{6\Delta t} \quad (3.3)$$

Cuando las moléculas no tienen restricciones, la dirección del movimiento es aleatoria y los desplazamientos de las moléculas a lo largo del tiempo son descritos por una distribución gaussiana univariante. En este caso, se trata de difusión isótropa, y la desviación típica es la misma en todas las direcciones. Por el contrario, cuando la movilidad de las moléculas depende de la orientación del medio se trata de difusión anisótropa. En microestructuras como las fibras de la materia blanca el movimiento de las moléculas se ve limitado en ciertas direcciones.

En un medio anisótropo se necesita mayor información para describir el desplazamiento tridimensional de las partículas que en un medio isótropo. Así, mientras que para la difusión isótropa es suficiente con un parámetro D , para describir la difusión anisótropa son necesarios más parámetros independientes, seis si se describe la difusión como una distribución gaussiana multivariante.

3.1.2 Diffusion Weighted Imaging (DWI)

La resonancia de difusión es una variante de la resonancia magnética que aporta información complementaria a las imágenes de resonancia magnética convencionales. Se basa en la detección del movimiento browniano de las moléculas de agua en los tejidos [41].

Dependiendo de la parte del cuerpo humano que se estudie, se puede buscar difusión isótropa o anisótropa. En la materia blanca del cerebro, puesto que la movilidad de las moléculas de agua está restringida por los axones, orientados a lo largo de las fibras, se trata de difusión anisótropa, que será mayor en la dirección de las fibras.

Para medir la difusión se utilizan gradientes de campo magnético para crear imágenes sensibilizadas a la difusión en una determinada dirección, denominadas DWI (*Diffusion Weighted Imaging*). Se emplea la secuencia de imagen de Stejskal y Tanner [42] que aplica un par de pulsos de gradiente bipolares después de la excitación y antes de la adquisición de los datos, de manera que se sensibiliza la imagen a la difusión de los protones de las moléculas de agua. El primero de los gradientes produce un cambio de la fase de magnetización. El segundo, invierte el cambio de fase, de manera que, para las moléculas estacionarias, ambos desfases se cancelan mutuamente. Sin embargo, aquellas moléculas que se han desplazado debido a movimientos brownianos durante el periodo de tiempo entre ambos pulsos experimentan una diferencia de fase aleatoria. Debido a esta diferencia de

fase, la señal de eco de las moléculas que se difunden se atenúa. Por ello, en estas imágenes las zonas con una difusión paralela al gradiente se muestran oscurecidas. A partir de las imágenes DWI, es posible calcular, mediante la ecuación 3.4, un parámetro escalar de difusión, el coeficiente de difusión aparente o ADC (*Apparent Diffusion Coefficient*) [43], que depende de la dirección del gradiente empleado.

$$S = S_0 e^{-bD} \quad (3.4)$$

En esta ecuación, S es la intensidad medida tras aplicar el gradiente de difusión, S_0 es la intensidad de la señal en ausencia del mismo y b es el factor de ponderación de difusión de LeBihan [44], que describe la secuencia de pulsos, la intensidad del gradiente y las constantes físicas. Para pulsos de gradiente rectangulares, el factor b viene descrito por la ecuación 3.5.

$$b = \gamma^2 \delta^2 \left(\Delta - \frac{\delta}{3} \right) |g|^2 \quad (3.5)$$

Donde γ es la relación giromagnética del protón (42 MHz/Tesla), $|g|$ es la fuerza de los pulsos de gradiente, δ es la duración de los pulsos, y Δ es el tiempo entre los pulsos.

En tejidos fibrosos, la difusión del agua depende del ángulo entre el eje de la fibra y el gradiente de campo magnético aplicado. Cuando el gradiente es paralelo a la fibra, la difusión medida será mayor que cuando es perpendicular. Dependiendo de la orientación relativa de los gradientes de difusión y de las fibras en la materia blanca, se pueden medir distintos valores de ADC, pues el valor de ADC obtenido solo mostrará la difusión en la dirección del gradiente aplicado. Para obtener una medida de la difusión que indique también la dirección de la misma se emplean múltiples gradientes, que permiten calcular un coeficiente de difusión no escalar, el tensor de difusión \mathbf{D}) [45].

3.1.3 Diffusion Tensor Imaging (DTI)

DTI es una modalidad de resonancia magnética de difusión que describe la difusión del agua mediante un modelo gaussiano. Permite calcular a partir de la medida de la difusión en distintas direcciones, para cada voxel del volumen, un tensor de segundo orden. Además, mediante esta técnica se pueden reconstruir, de manera no invasiva, las trayectorias de las fibras de los tejidos. La magnitud medida en DTI es la difusividad o coeficiente de difusión, es decir, el parámetro \mathbf{D} de la ecuación 3.1. Sin embargo, debido a la heterogeneidad del tejido causada por las membranas de mielina de los axones y las membranas celulares, las moléculas de agua se difunden a lo largo de las fibras y la difusión varía, por tanto, con la dirección. Es por ello que no sirve con un único parámetro

escalar para describir la difusión, sino que se necesita un tensor que describa la movilidad de las moléculas a lo largo de cada dirección y la correlación entre estas direcciones. Para calcular el tensor de difusión, deben obtenerse imágenes de difusión sensibilizadas con distintas direcciones de gradiente. Para definir el vector se necesitan, al menos, seis imágenes sensibilizadas con distintos gradientes, así como una imagen no sensibilizada a la difusión.

Para caracterizar la difusión gaussiana en medios anisótropos, donde los desplazamientos de las moléculas no son iguales en todas direcciones, se emplean los tensores de difusión. Un tensor de difusión \mathbf{D} 3.6 es una matriz 3x3 simétrica y definida positiva. Por esas características, tiene tres autovectores ortogonales y otros tres autovalores positivos. El mayor autovector indica la dirección principal de difusión. En los tejidos fibrosos anisótropos también indica el eje de la fibra. Los tres autovalores ($\lambda_1, \lambda_2, \lambda_3$, siendo $\lambda_1 \geq \lambda_2 \geq \lambda_3$) indican la difusividad en la dirección de cada autovector.

$$\mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \quad (3.6)$$

Donde $D_{ij} = D_{ji}$ para $j \neq i$.

Para el cálculo de los tensores debe resolverse la ecuación de Stejskal- Tanner para el caso anisótropo (3.7) que describe cómo la intensidad de la señal en cada voxel disminuye en presencia de difusión gaussiana [42]:

$$S_k = S_0 e^{-b \hat{\mathbf{g}}_k^T \mathbf{D} \hat{\mathbf{g}}_k} \quad (3.7)$$

En esta ecuación, S_0 es la intensidad de la señal en ausencia de un gradiente de difusión. S_k es la intensidad medida tras aplicar el k-ésimo gradiente de difusión en la dirección $\hat{\mathbf{g}}_k$. $\hat{\mathbf{g}}_k^T \mathbf{D} \hat{\mathbf{g}}_k$ representa el coeficiente de difusión en la dirección $\hat{\mathbf{g}}_k$. Y b es el factor de ponderación de difusión de LeBihan, relacionado con la intensidad del gradiente aplicado (ecuación[?]).

Puesto que el tensor de difusión \mathbf{D} es una matriz 3x3 simétrica, tiene seis grados de libertad. Por tanto, para estimar el tensor deben realizarse, al menos, siete medidas diferentes, seis tomadas con diferentes direcciones de gradiente, ortogonales entre sí, y otra en ausencia de gradiente. Sin embargo, dado que las medidas obtenidas son bastante ruidosas, suele utilizarse un número bastante mayor de gradientes para conseguir una estimación más robusta. Resolviendo el sistema de ecuaciones resultante para cada vóxel se obtienen los vectores de difusión correspondientes. Dicho sistema se puede resolver de distintas maneras. Aunque la más habitual es mediante mínimos cuadrados, también se

utilizan otras más complejas como mínimos cuadrados ponderados o métodos variacionales, que resuelven ecuaciones de manera iterativa.

La difusión anisótropa puede describirse mediante elipsoides en los que los radios representan la difusión en cada dirección [46]. Los ejes principales de estos elipsoides corresponden con las direcciones del sistema de autovectores de los tensores. Cuanto menos se parezca la forma del elipsoide a una esfera, mayor será la anisotropía. El eje principal del elipsoide corresponde con la dirección de mayor difusión e indica la orientación del eje de la fibra. Hay distintos factores que afectan a la forma de los elipsoides de difusión, entre ellos la densidad de las fibras, el grado de mielinización, el diámetro medio de las fibras y la similitud de las direcciones de las fibras dentro de un vóxel [47].

3.1.4 Medidas escalares

A partir de los tensores pueden calcularse diferentes medidas escalares, que proporcionan distinta información sobre la difusión o la anisotropía. Estas medidas resultan más sencillas de comparar, puesto que, al contrario que los tensores, no son matrices. Además, su visualización es menos compleja. Las principales medidas que se pueden obtener a partir de los tensores de difusión son las siguientes:

Difusividad media (MD, Mean Diffusivity)

Es el valor medio de los autovalores del tensor y representa la cantidad de difusión en cada vóxel [1].

$$MD = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} = \frac{D_{xx} + D_{yy} + D_{zz}}{3} \quad (3.8)$$

Traza (Trace)

Es una medida equivalente a la MD. Se calcula como la suma de los elementos diagonales del tensor, lo que equivale a la suma de los autovalores [46].

$$\text{trace} = D_{xx} + D_{yy} + D_{zz} = \lambda_1 + \lambda_2 + \lambda_3 \quad (3.9)$$

Anisotropía fraccional (FA, Fractional Anisotropy)

Mide la fracción de difusión que es anisótropa, es decir, la diferencia entre el elipsoide que es el tensor y una esfera. Varía entre los valores 0 y 1, y puede obtenerse sin necesidad de calcular explícitamente los autovalores [1].

$$\begin{aligned}
 FA &= \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}}{(\lambda_1 + \lambda_2 + \lambda_3)} \\
 &= \frac{\sqrt{3} |\mathbf{D} - \frac{1}{3} \text{tr } \text{ace}(\mathbf{D})\mathbf{I}|}{\sqrt{2} |\mathbf{D}|}
 \end{aligned} \tag{3.10}$$

Donde \mathbf{I} es el tensor identidad.

Modo (Mode)

Mide la diferencia entre la anisotropía planar y lineal. Es una medida ortogonal a la FA y está referida a la asimetría de los autovalores. Varía entre los valores -1 y 1 [48].

$$\text{mode} = \frac{(-\lambda_1 - \lambda_2 + 2\lambda_3)(2\lambda_1 - \lambda_2 - \lambda_3)(-\lambda_1 + 2\lambda_2 - \lambda_3)}{2(\lambda_1^2 + \lambda_2^2 + \lambda_3^2 - \lambda_1\lambda_2 - \lambda_1\lambda_3 - \lambda_2\lambda_3)^{3/2}} \tag{3.11}$$

Medida lineal (LM, Linear Measure)

Refleja la uniformidad de la dirección de las fibras dentro de un voxel, es decir, describe cómo de cerca está el tensor de ser una línea. Varía entre los valores 0 y 1 [47].

$$LM = \frac{\lambda_1 - \lambda_2}{\lambda_1} \tag{3.12}$$

Difusividad axial (AD, Axial Difusivity)

La difusividad axial, también llamada difusividad paralela, es igual al mayor de los autovalores. Representa la difusividad paralela a las fibras [49].

$$AD = \lambda_1 \quad (3.13)$$

Difusividad radial (RD, Radial Difusivity)

La difusividad radial, también llamada difusividad perpendicular, es igual a la media de los dos autovalores menores. Representa la difusividad perpendicular a las fibras [49].

$$RD = \frac{\lambda_2 + \lambda_3}{2} \quad (3.14)$$

A veces, la difusividad radial y la difusividad axial pueden estar normalizadas entre el valor de la traza o entre $\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$ para que su valor varíe entre 0 y 1.

3.2 Aprendizaje Automático

El Aprendizaje Automático (*Aprendizaje Automático*) es un subcampo de la inteligencia artificial que posibilita que un ordenador pueda aprender a realizar una tarea sin ser explícitamente programado para ello:

«Se dice que un programa de ordenador aprende de la experiencia E con respecto a una tarea T y una medida de rendimiento P, si su rendimiento en T, medido en P, mejora con la experiencia E. *Tom Mitchell, 1997*»

Las aplicaciones típicas del Aprendizaje Automático son:

- Problemas para los que las soluciones existentes requieren mucha sintonización de parámetros o una larga lista de reglas. Usando algoritmos de Aprendizaje Automático se puede a menudo simplificar el código y conseguir un mejor rendimiento.
- Problemas complejos para los que no hay ninguna solución buena usando un enfoque tradicional, que con técnicas de Aprendizaje Automático se resuelven.
- Entornos cambiantes: estas técnicas se pueden adaptar de manera constante a cambios en los datos, donde un programa tradicional fallaría.
- Obtener información sobre problemas complejos y grandes cantidades de datos (minería de datos)

3.2.1 Bases

Existen tantos tipos de sistemas de Aprendizaje Automático que resulta útil clasificarlos en grupos amplios [50]. Estos criterios no son exclusivos, se pueden combinar al gusto a la hora de desarrollar un solución. Según las funcionalidades que se pretendan o el caso particular se usarán unos u otros.

- Si se entrenan o no con supervisión humana: (supervisado, no supervisado ó reforzado).

En el aprendizaje supervisado, los datos de entrenamiento con los que se alimenta el sistema incluye las soluciones deseadas, en forma de etiquetas. Dos tareas que se pueden llevar a cabo con un sistema de este tipo son clasificación (decidir en función de la entrada un grupo determinado) o regresión (asignar un valor dado un conjunto de características que se extraen de los datos). Para entrenar estos sistemas, es necesario contar con muchos ejemplos de datos, de tal manera que los modelos puedan aprender a generalizar con datos desconocidos posteriormente. Este es el caso que se va a dar en este trabajo.

En el aprendizaje supervisado, los datos no están etiquetados. El sistema trata de aprender sin información explícita de las salidas, por lo que, lo que se busca típicamente es estructuración y organización de datos, detección de anomalías, agrupamiento de los datos en categorías o búsqueda de patrones o relaciones entre datos. Una tarea relacionada es la reducción de dimensionalidad, en la que el objetivo es simplificar los datos sin perder demasiada información. Una manera de llevar esto a cabo es combinar distintas características correladas en una.

El aprendizaje reforzado es relativamente distinto a los otros casos. El sistema que aprende recibe el nombre de agente, y puede observar su entorno, elegir de entre un conjunto de acciones y llevar a cabo acciones que a cambio dan recompensas o penalizaciones. Debe aprender por si mismo cuál es la mejor estrategia a seguir para maximizar las recompensas a lo largo del tiempo

- Si pueden aprender de manera incremental durante su uso (entrenamiento por lotes o entrenamiento *online*)

En el aprendizaje por lotes el sistema no puede aprender de manera incremental, tiene que ser entrenado usando todos los datos disponibles. Esto normalmente lleva mucho tiempo y necesita de gran cantidad de recursos computacionales, de tal manera que se hace *offline*. Se entrena un sistema primero y luego se lanza a producción sin que ya aprenda nada más, se limita a aplicar lo aprendido en el entrenamiento. Si se quiere actualizar un sistema por lotes con nuevos datos, debe de reentrenarse y sustituir la nueva versión por la antigua.

El aprendizaje *online* consiste en entrenar al sistema de manera incremental, alimentando con datos de manera secuencial. individualmente o en grupos pequeños. Cada paso del aprendizaje es rápido y barato (computacionalmente hablando), así que el sistema aprende de los nuevos datos según llegan. Este mecanismo es útil

para sistemas en los que los datos llegan como un flujo continuo y se necesita adaptación inmediata a los mismos. También es buena opción para casos en los que los recursos de computación son limitados: los datos, tras pasar por el sistema, se descartan, ya han cumplido su función. Si los datos son demasiado grandes y no caben en la memoria de la máquina todos a la vez, esta solución también es aplicable para entrenar lote a lote mientras se va cargando en memoria. La característica más importante de estos sistemas es el parámetro conocido como tasa de aprendizaje (λ), el cuál especifica como de rápido se debe de adaptar a cambios en los datos. Con una tasa alta, la adaptación es rápida pero el sistema tiende a olvidarse de datos antiguos, y con una baja la reacción es mas lenta pero tiene más memoria y es menos sensible al ruido o a datos no representativos.

- La manera de generalizar del sistema: si funcionan simplemente comparando puntos nuevos con puntos conocidos o detectan patrones en los datos de entrenamiento y construyen un modelo predictivo (aprendizaje basado en instancias o aprendizaje basado en modelos)

El aprendizaje basado en instancias utiliza medidas de similaridad para llegar a sus soluciones. Aprende de los ejemplos que tenga disponibles y trata de generalizar a nuevos datos usando estas métricas aprendidas. Otra manera de generalizar es la creación de un modelo basado en los ejemplos y usarlo para crear predicciones. Usando funciones de coste que evalúan como de acertado es un modelo, se trata de minimizar dicha función para encontrar el más adecuado. El proceso típico para desarrollar un sistema de Aprendizaje Automático es el siguiente:

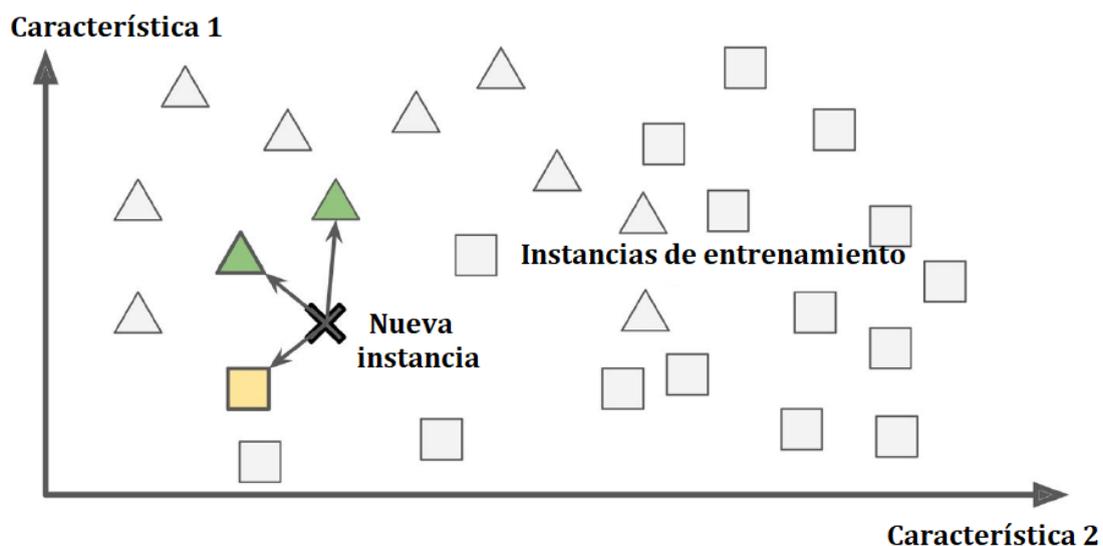


Figura 3.1: Sistema de Aprendizaje automático que generaliza a partir de ejemplos aprendidos.

- Se estudian los datos.
- Se elige un modelo apropiado.

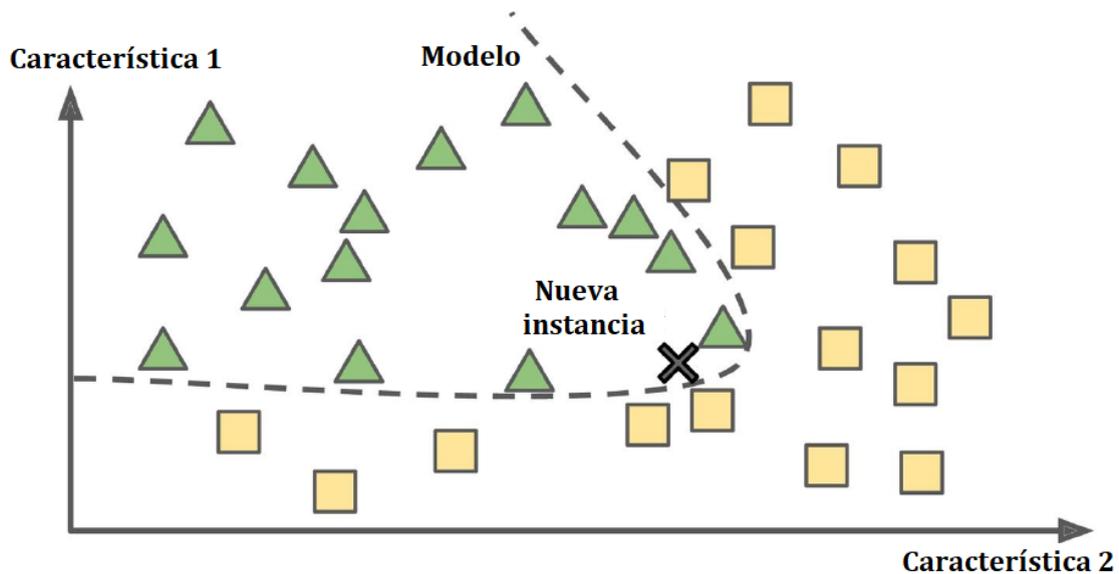


Figura 3.2: Sistema de Aprendizaje automático que genera un modelo para llevar a cada predicciones.

- Se entrena con los datos de entrenamiento
- Se aplica el modelo en datos nuevos (conjunto de test) y se evalúan las nuevas predicciones
- Si los resultados no convencen, se vuelve a empezar, eligiendo otros modelos (otros algoritmos, cambio de parámetros, preprocesado de los datos...)

En su estado actual, los modelos de Aprendizaje Automático tienen una serie de defectos o problemas que puede limitar su utilidad.

- Cantidad de datos insuficiente: incluso para problemas sencillos, se necesitan gran cantidad y variedad de ejemplos, sin estos, no se puede lograr un sistema que generalice bien.
- Datos de poca calidad: La cantidad no es suficiente, es crucial que los datos además sean de calidad. Hay que cuidar los datos con los que se alimenta un sistema, un conjunto de datos no representativo no generalizará correctamente. La preparación de los datos es una tarea clave: eliminar *outliers*, descartar los incorrectos, arreglar errores en los datos o rellenar características que vengan vacías.
- Datos no representativos: Es obvio, pero los datos de entrenamiento tienen que ser de una naturaleza similar a la que se va a enfrentar el sistema en la realidad. Si se entrena con datos no representativos, se puede obtener un modelo muy bueno en ese entorno y luego fracasar en el entorno de producción. Hay que asegurar que tenemos suficientes ejemplos representativos de los casos en los que queremos generalizar.

- Características irrelevantes: El sistema será capaz de aprender siempre y cuando de entre los datos de entrada encuentre información relevante. Cuanto menor sea el porcentaje de información útil, más se complica la tarea de extraer características y el sistema será menos capaz de aprender. Mediante la ingeniería de características, se lleva a cabo tareas de selección, extracción y creación de características para minimizar el problema.
- Sobreajustar o subajustar los datos: Un modelo sobreajusta si funciona muy bien en los datos de entrenamiento, pero no es capaz de generalizar bien. Normalmente este problema viene a raíz de elegir un modelo demasiado complejo para los datos que tenemos, que extrae información de donde no la hay y puede detectar patrones hasta en el ruido. Si se aplica un modelo más complejo que los datos, se puede regularizar el modelo para hacerlo más simple proponiendo restricciones adicionales. Por el contrario, un modelo que subajusta es un modelo demasiado simple para la estructura que subyace en los datos. La realidad es más compleja que la que interpreta el modelo, luego está destinado a fallar en sus predicciones. Las soluciones pasan por usar modelos más complejos o alimentar el modelo con características de más calidad.

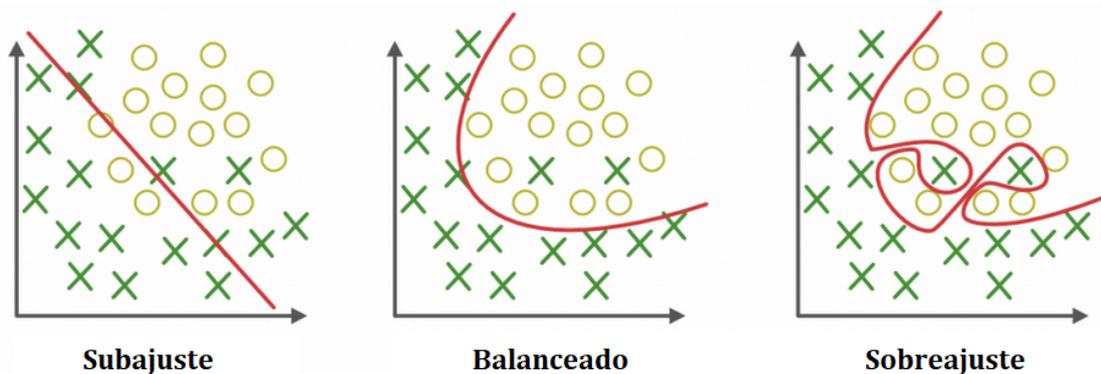


Figura 3.3: Ejemplos gráficos de modelos subajustados o sobreajustado frente a un modelo equilibrado.

A la hora de desarrollar un modelo y saber cómo de bien generaliza la única manera es probarlo en casos nuevos. Pero sin validarlo no podemos llevarlo a un entorno productivo. Por eso, para asegurarse de que un modelo funciona adecuadamente, los datos disponibles se dividen en dos conjuntos: uno de entrenamiento y otro de test. El modelo se entrena usando el conjunto de entrenamiento. El error en casos nuevos se conoce como el error de generalización, y se estima probando dicho modelo en los casos de test. Así podemos saber cómo va a funcionar el modelo con datos que no ha visto antes. Si el error de entrenamiento es bajo, pero el de generalización aumenta mucho, estamos cometiendo sobreajuste en el modelo. Es común usar el 80 % de los datos para entrenar y el 20 % para realizar pruebas. El error de generalización es también la herramienta que nos permite comparar entre distintos modelos o algoritmos o ajustar ciertos parámetros de los mismos.

3.2.2 Algoritmos

En este apartado se van a explicar brevemente los algoritmos de Aprendizaje Automático con los que se va a trabajar en el presente trabajo:

Regresión lineal

Un modelo lineal hace predicciones calculando una suma ponderada de las características de entrada más una constante:

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (3.15)$$

\hat{y} es la predicción, n el número de características, x_i es la característica i y θ_n el parámetro n (incluyendo los pesos y la constante ($n=0$)). Se puede escribir de manera más concisa en su forma vectorial:

$$\hat{y} = h_{\theta}(\mathbf{x}) = \theta^T \cdot \mathbf{x} \quad (3.16)$$

θ es el vector de parámetros y θ^T su traspuesto, \mathbf{x} es el vector de características y h_{θ} es la función hipótesis del modelo usando los parámetros θ . Como medida de rendimiento se utiliza el RMSE (*Root Mean Square Error*). Para entrenar la regresión lineal, es necesario encontrar los valores de θ que minimizan el RMSE. En la práctica es más sencillo minimizar el MSE (*Mean Square Error*) y lleva al mismo resultado. El MSE de una hipótesis h_{θ} se calcula:

$$\text{MSE}(\mathbf{X}h_{\theta}) = \frac{1}{m} \sum_{i=1}^m (\theta^T \cdot \mathbf{x}^{(i)} - y^{(i)})^2 \quad (3.17)$$

Para encontrar el valor θ que minimiza la función de coste, hay una solución cerrada, que es la ecuación normal y nos da el resultado directamente.

$$\hat{\theta} = (\mathbf{X}^T \cdot \mathbf{X}^{-1}) \mathbf{X}^T \mathbf{y} \quad (3.18)$$

\mathbf{y} es el vector de los valores objetivo y $\hat{\theta}$ es el valor de θ que minimiza la función de coste.

Regresión de Ridge

La regresión de Ridge (también llamada regularización de Tikhonov) es un tipo de regresión lineal regularizada. Se añade un término a la función de coste que fuerza al algoritmo a no solo ajustar los datos sino además a mantener los pesos lo más pequeños posibles. El término de regularización sólo se debe añadir a la función de coste durante la etapa de entrenamiento. Posteriormente al entrenamiento, se evalúa el rendimiento sin dicho término.

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^n \theta_i^2 \quad (3.19)$$

El hiperparámetro α controla cuanta regularización se aplica. Si es 0, tenemos el modelo de regresión lineal, y si es muy grande, todos los pesos se acercan a cero y el resultado es una línea recta que pasa por la media de los datos. La constante de la regresión no se regulariza. Si se define \mathbf{w} como el vector de pesos (θ_1 hasta θ_n), entonces el parámetro de regularización es igual a la norma l^2 del vector de pesos. La solución a la regresión de Ridge es:

$$\hat{\theta} = (\mathbf{X}^T \cdot \mathbf{X} + \alpha \mathbf{A})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y} \quad (3.20)$$

K vecinos más próximos

Este algoritmo es un método que se puede aplicar a clasificación o regresión. En ambos casos, las entradas son instancias de entrenamiento del espacio de características, y la salida depende de si hacemos clasificación (la salida se decide por votación de mayoría de los k vecinos más próximos) o regresión (se hace la media de los valores de los vecinos cercanos). Se añade pesos a las contribuciones de los vecinos en función de su cercanía, siendo estos pesos inversamente proporcionales a la distancia. Este algoritmo es un ejemplo de aprendizaje basado en instancias, donde la función se aproxima sólo localmente y la computación se pospone hasta la etapa de clasificación o regresión.

Árboles de decisión

Los árboles de decisión son algoritmos versátiles que pueden realizar tareas tanto de clasificación como de regresión. Son componentes fundamentales de los Random Forest, que está entre los algoritmos más potentes de Aprendizaje Automático. Nos centraremos

en los árboles de regresión, por ser los que aplican en el presente trabajo. Son muy similares a los de clasificación. Se muestra un ejemplo en 3.4. Para realizar una predicción, se recorre el árbol siguiendo las condiciones y se llega a un valor dado. Ese valor es el resultado de calcular el valor medio de todas las instancias de entrenamiento que están asociadas a ese nodo. La profundidad del árbol es el máximo número de pasos que se pueden dar desde el inicio y es un hiperparámetro del sistema.

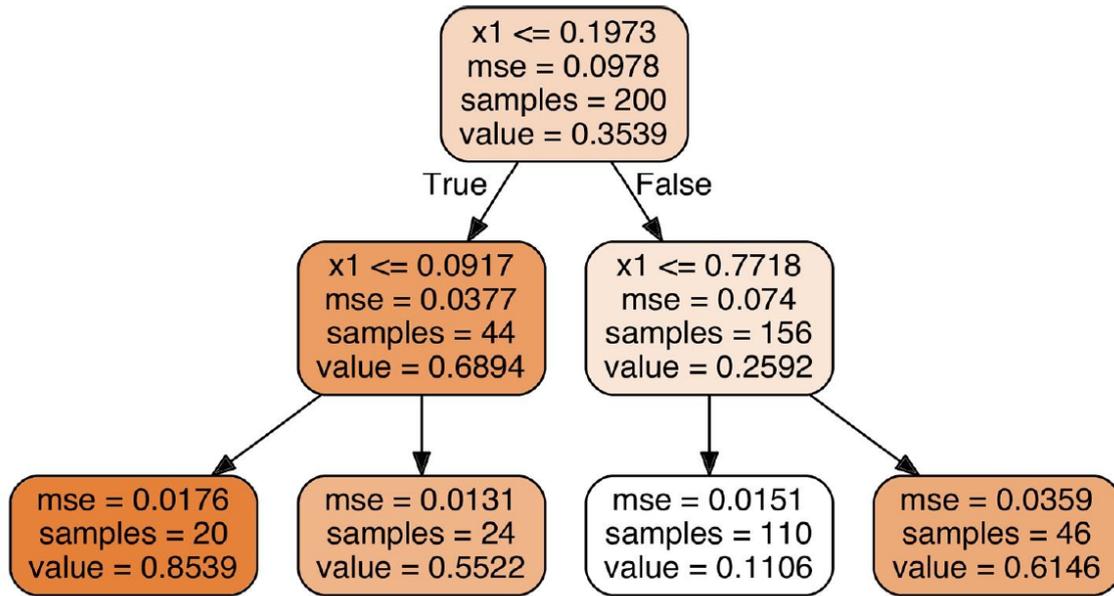


Figura 3.4: Ejemplo de árbol de decisión para regresión.

Para calcular el árbol, se implemente el algoritmo CART (*Classification and Regression Tree*), que intenta dividir el conjunto de entrenamiento de manera que se minimiza el MSE.

$$J(k, t_k) = \frac{m_{\text{left}}}{m} \text{MSE}_{\text{left}} + \frac{m_{\text{right}}}{m} \text{MSE}_{\text{right}} \quad (3.21)$$

$$\begin{cases} \text{MSE}_{\text{node}} = \sum_{i \in \text{node}} (\hat{y}_{\text{node}} - y^{(i)})^2 \\ \hat{y}_{\text{node}} = \frac{1}{m_{\text{node}}} \sum_{i \in \text{node}} y^{(i)} \end{cases} \quad (3.22)$$

Los árboles son propensos al sobreajuste, de manera que siempre se suele aplicar regularización. Algunas de las limitaciones que presentan son una tendencia hacia las fronteras de decisión ortogonales, lo que los hace sensibles a rotaciones en los datos, son sensibles a pequeñas variaciones en los datos.

Random Forest

Random Forest puede limitar estas inestabilidades que presentan los árboles de decisión mediante el promedio de las predicciones sobre muchos árboles distintos. El parámetro principal del algoritmo es el número de árboles. Según crecen los árboles se introduce aleatoriedad adicional, en lugar de buscar la mejor característica, cuando se divide un nodo, busca la mejor característica pero dentro de un subconjunto aleatorio de características. Esto resulta en una mayor diversidad, que intercambia un sesgo mayor por una varianza mejor, lo que generalmente suele proporcionar un modelo mejor.

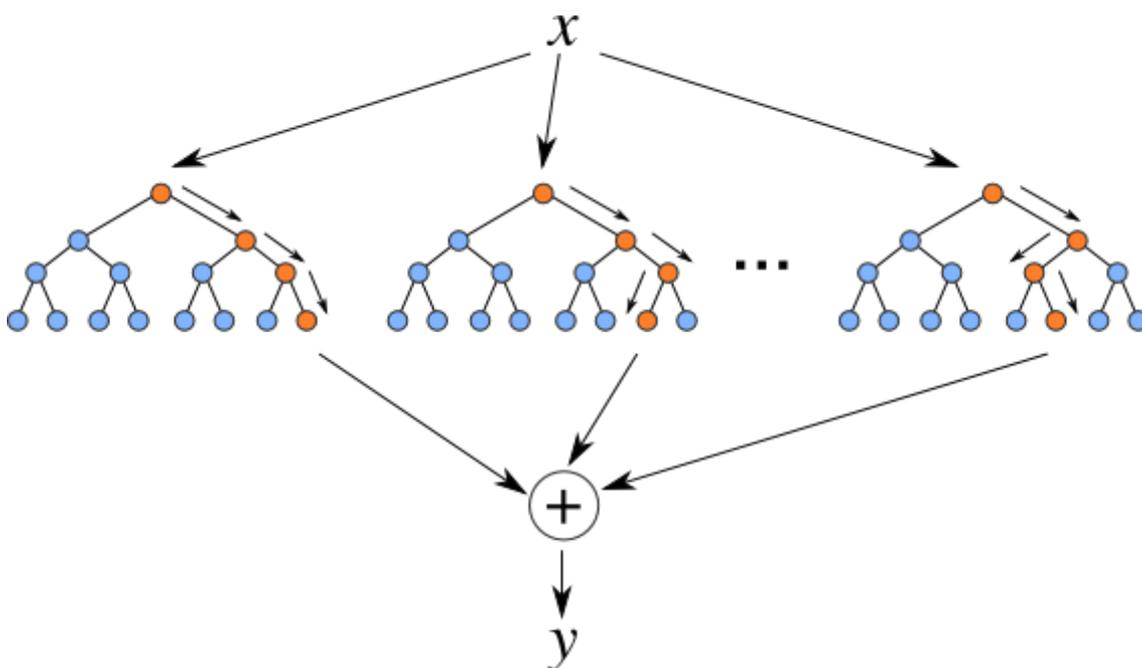


Figura 3.5: Arquitectura de Random Forest para predicción

Si además se usa en cada nodo umbrales aleatorios para cada característica en lugar de buscar los mejor posibles, tenemos un *Extremely Randomized Trees* o *Extra-Trees*, que de nuevo aumenta el sesgo para reducir aún más la varianza. Además, se entrenan más rápido que *Random Forest* ya que la búsqueda de esos umbrales es una de las tareas que más tiempo consume del entrenamiento.

Es difícil predecir si funcionará mejor uno u otro, la mejor manera de averiguarlo es probar ambos y realizar validación cruzada para compararlos.

En un árbol de decisión, las características más importantes es más probable que aparezcan al principio del árbol (raíz), y las menos importantes al final (hojas) o que directamente no aparezcan. Se puede calcular el valor numérico de esta importancia de cada característica, lo cuál resulta muy práctico para realizar selección de características.

3.3 Redes neuronales

Las redes neuronales o ANN (*Artificial Neural Network*) son el núcleo del *Deep Learning*. Son versátiles, potentes y escalables, haciéndolas ideales para atacar problemas grande y altamente complejos de Aprendizaje Automático. Una de las más simples arquitecturas de ANN es el perceptron, compuesta de LTUs (*Linear Threshold Unit*), el cuál sirve para explicar el funcionamiento de la red. Las entradas y salidas son números, y cada conexión de entrada se asigna junto a un peso. Se computa la suma de las entradas y se aplica sobre ella una función escalonada (3.23, 3.24), proporcionando la salida $h_w(x) = \text{step}(w^t \cdot x)$

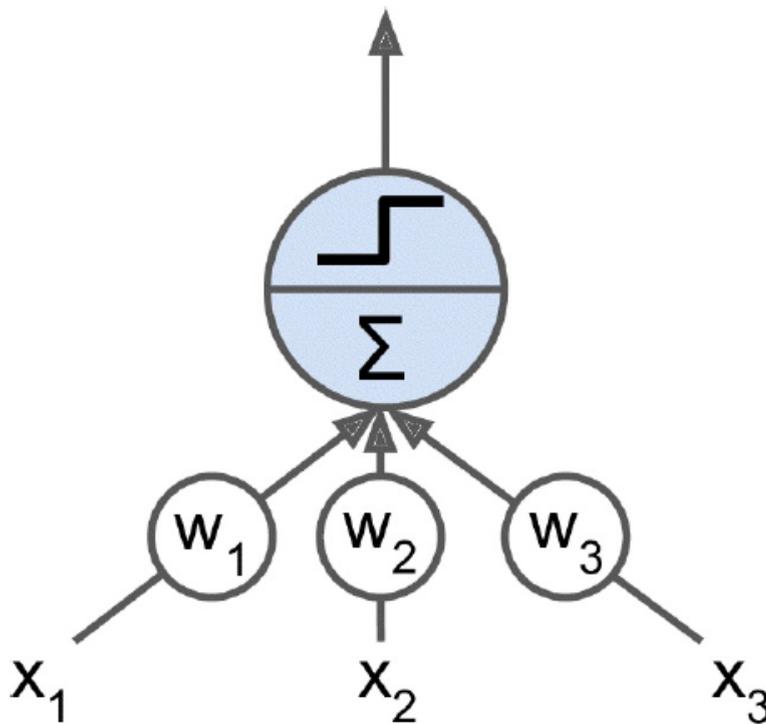


Figura 3.6: Arquitectura de una LTU

$$\text{heaviside}(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases} \quad (3.23)$$

$$\text{sgn}(z) = \begin{cases} -1 & \text{if } z < 0 \\ 0 & \text{if } z = 0 \\ +1 & \text{if } z > 0 \end{cases} \quad (3.24)$$

El perceptron se compone de una capa de LTUs, en el que cada neurona (LTU) está conectada a todas las entradas. Las conexiones se representan usando neuronas de en-

trada (*input*) cuya salida es el mismo valor que tienen en la entrada. Además, se añade una neurona de término constante que siempre proporciona un 1 a su salida. En la imagen 3.7 se puede ver un ejemplo de perceptron. Entrenar el perceptron conlleva encontrar los pesos adecuados para las LTUs.

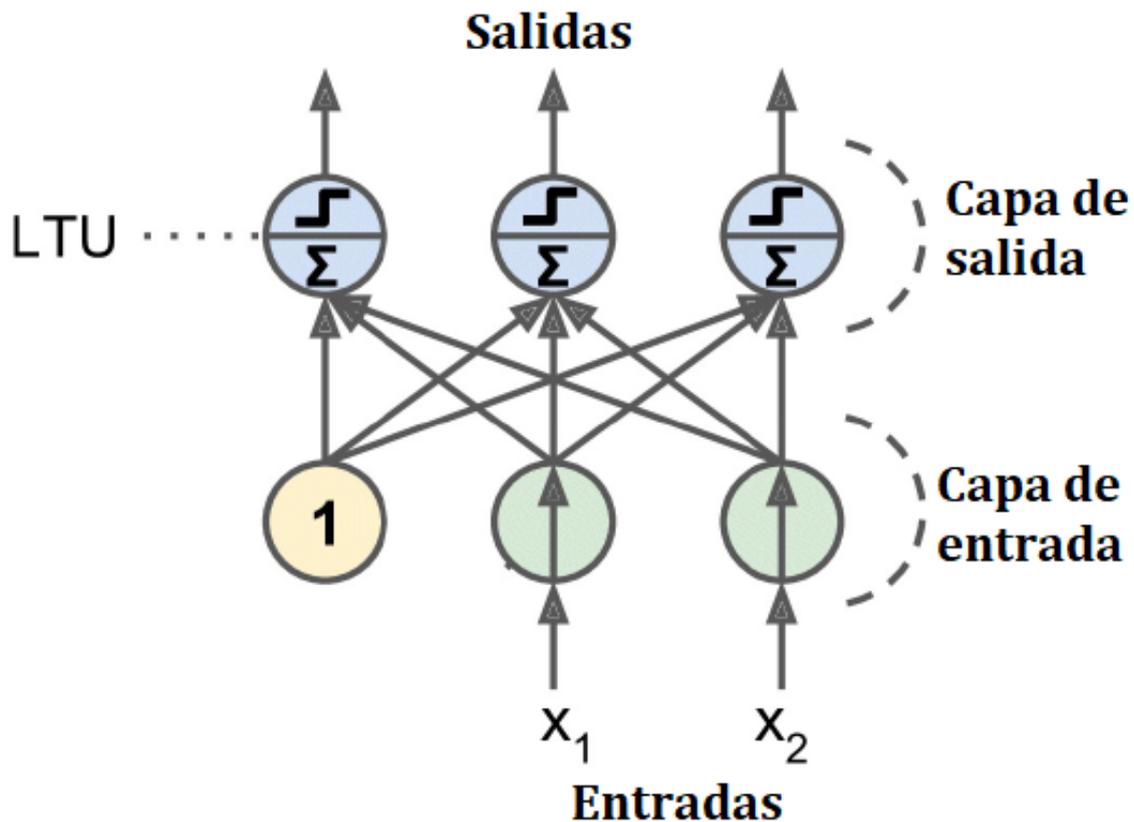


Figura 3.7: perceptron con 2 entradas y 3 salidas

Un MLP ó *Multi-Layer Perceptron* se compone de una capa de entrada, una o más LTUs en el medio, que reciben el nombre de capas ocultas, y una capa de salida de LTU. Cada capa posee una neurona de término constante y está conectada en su totalidad con la siguiente. Si una red neuronal tiene dos o más capas ocultas, recibe el nombre de DNN (*Deep Neural Network*).

Para entrenarlas, se utiliza una técnica llamada *backpropagation* y gradientes descendentes. Para cada instancia de entrenamiento, el algoritmo alimenta la red con los datos y calcula la salida de cada neurona. Se mide entonces el error de la red, y se calcula cuánto ha aportado cada neurona de la capa oculta anterior a dicho error. Recursivamente, se realiza para cada capa oculta hasta llegar a la entrada. Al llegar de nuevo a la entrada, se realiza un paso del algoritmo de gradiente descendente en todos los pesos de la red usando los gradientes del error calculados, lo cual hace variar los pesos, cambiando la predicción para realizar una nueva instancia de entrenamiento.

El gradiente descendente es un algoritmo de optimización usado ampliamente para entrenar variedad de modelos de Aprendizaje Automático. La idea que subyace es la de retocar iterativamente parámetros del sistema para minimizar una función de coste. Si definimos el vector de parámetros como θ , el algoritmo mide la dirección de la función de error con respecto a θ , y modifica los parámetros para seguir el gradiente en su dirección descendente. Cuando el gradiente es 0, llegamos a un mínimo y hemos optimizado la función. Para empezar, es típico que θ se inicialice con valores aleatorios, y el algoritmo mejora paso a paso hasta converger en un mínimo. Un parámetro vital del algoritmo es la tasa de aprendizaje, que determina el tamaño de los pasos, y puede determinar si el algoritmo converge o no y su velocidad de convergencia. Si es alto, convergerá más rápido, pero si lo es demasiado, es posible que los pasos sean demasiado grandes para encontrar el mínimo y acabe divergiendo.

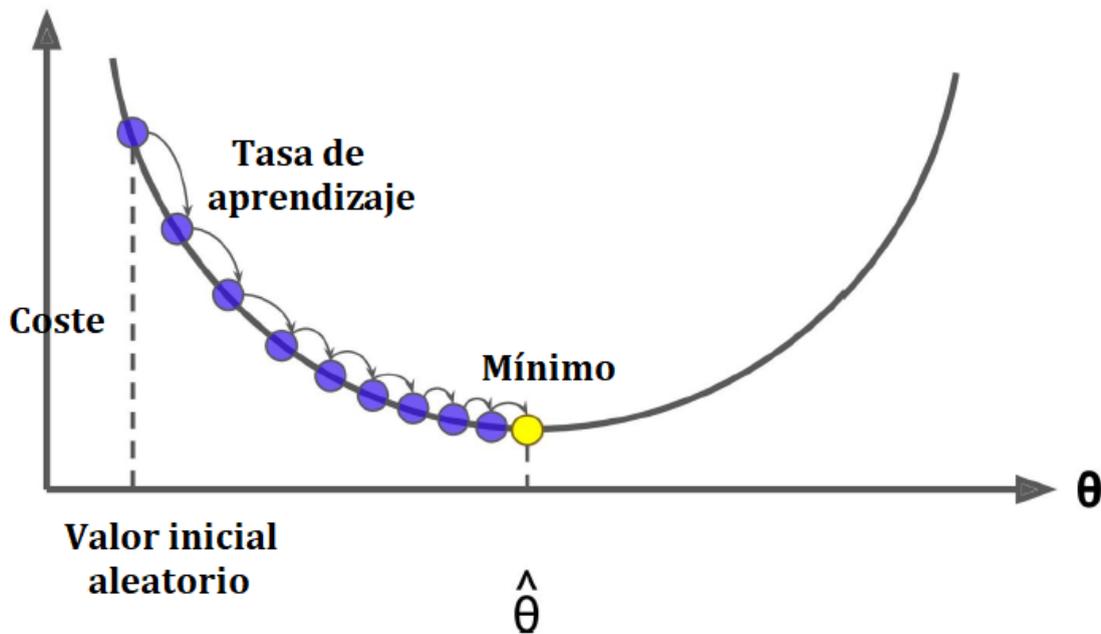


Figura 3.8: Gradiente descendente

Para que este algoritmo funcione, se retoca la arquitectura del MLP, sustituyendo la función escalonada con una en la que haya gradientes con los que trabajar (las propuestas previamente no tenían derivada). Tres funciones que se suelen utilizar son la función logística 3.25., la tangente hiperbólica 3.26. y la ReLU 3.27 (*REctified Linear Unit*). Se muestran en la figura 3.10. La función logística se implementa porque es derivable en todo su dominio mientras que la hiperbólica ofrece la ventaja de ir entre -1 y 1, en lugar de 0 y 1, lo cual tiende a hacer la salida de cada capa más o menos normalizada (centrada alrededor del 0) al inicio del entrenamiento, lo que a menudo permite mejorar la convergencia. La ReLU es continua pero no es diferenciable en $z = 0$, que a priori puede parecer ilógico ya que puede hacer rebotar al gradiente, pero en la práctica funciona muy bien, es sencilla de computar, y el hecho de que no tenga un valor máximo ayuda a reducir algunos

problemas del gradiente descendente.

$$P(t) = \frac{1}{1 + e^{-t}} \quad (3.25)$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.26)$$

$$f(x) = \text{máx}(0, x) \quad (3.27)$$

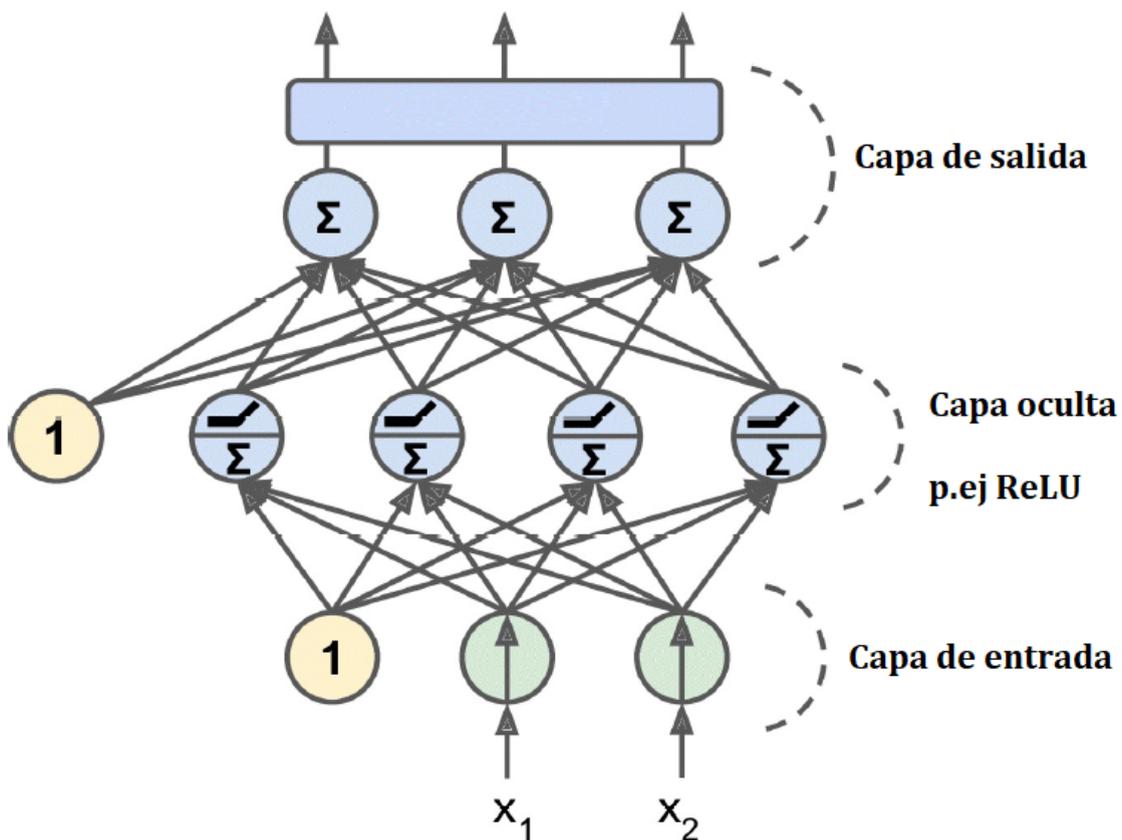


Figura 3.9: MLP de 2 entradas, 1 capa oculta y 3 salidas

Las redes convolucionales ó CNN (*Convolutional Neural Networks*) nacen a partir de las redes neuronales, con la idea de imitar la manera en la que funciona el córtex visual. Dentro del córtex, muchas neuronas tienen un pequeño campo receptivo local, lo que significa que sólo reaccionan ante estímulos en una región limitada del campo visual. Estos campos pueden solaparse, y en conjunto cubren todo el campo de visión. Además, algunas neuronas reaccionan sólo a líneas horizontales, mientras que otras reaccionan a distintas orientaciones. Además, existen neuronas con campos receptivos más grandes que reaccionan a patrones más complejos, lo que lleva a la idea de que estas neuronas de alto

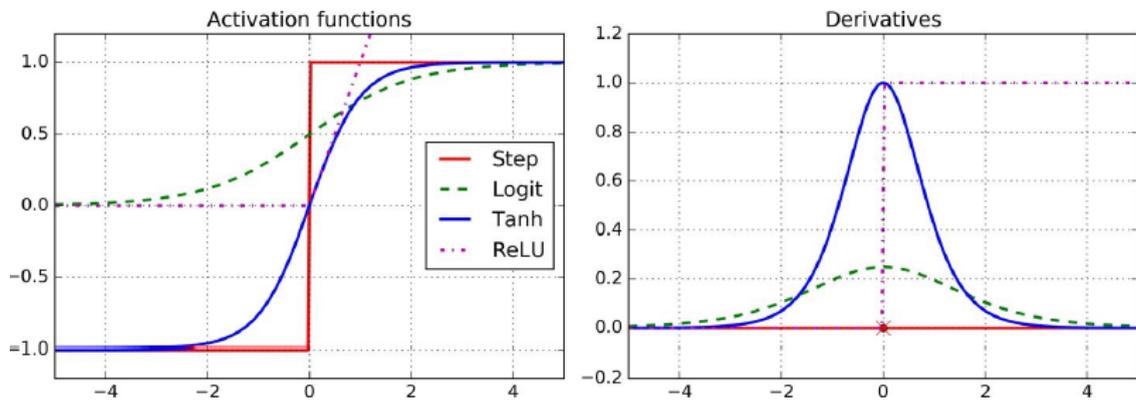


Figura 3.10: Funciones de activación y sus derivadas

nivel se basan en las salidas de sus vecinas de bajo nivel. Imitando esta arquitectura, las CNN son capaces de detectar todo tipo de patrones complejos en el área de imagen.

Además de los bloques ya vistos, cómo las redes conectadas y las funciones de activación, las CNN incorporan dos nuevos bloques, que son las capas convolucionales y las capas de *pooling*.

- En las CNN, las neuronas de las primeras capas no están conectadas a cada píxel de la imagen, si no que cada una se centrará en una pequeña parte de la imagen. Las neuronas de la siguiente capa estarán conectadas sólo a neuronas de la primera que estén localizadas en un pequeño rectángulo de la imagen. Este tipo de arquitectura permite a la red concentrarse inicialmente en características de bajo nivel para ir componiendo gradualmente características de alto nivel (3.11). Los pesos de cada neurona se pueden representar como una imagen del tamaño de su campo receptivo. Estos pesos se conocen cómo filtros o núcleos (*kernels*) de convolución. En función de estos pesos, las neuronas se centrarán en unos u otros aspectos de la imagen, por ejemplo en orientaciones espaciales. De este modo, una capa entera de neuronas y sus distintos filtros, generarán un mapa de características. El entrenamiento de las CNN consiste en encontrar los pesos adecuados para estos filtros y a combinarlos para encontrar patrones complejos en los datos y adaptar la entrada y la salida de la red.
- Las capas de *pooling* tienen el objetivo de submuestrear las imágenes de entrada para reducir la carga computacional, el uso de memoria y el número de parámetros. Además, la reducción de la imagen inicial también hace la red tolerante a rotaciones o giros en la imagen. Estas capas funcionan de manera parecida a las convolucionales, tomando como entrada solo un entorno local de la imagen, pero en lugar de tener pesos, se limita a agregar las entradas, normalmente realizando un máximo (la más típica) o una media de las mismas. Esta capa resulta destructiva, eliminando gran parte de los valores de entrada.

La arquitectura que proponen las CNN consta típicamente de una capa convolucional y una de activación (ReLU), seguidas de una de *pooling*. Este proceso se repite varias veces, en los que la imagen de entrada se vuelve cada vez más pequeña y se recogen mapas de características. Al final, se une con una red neuronal típica y se fuerza un tipo de salida. Habitualmente se utiliza una capa de clasificación que estima las probabilidades de pertenencia a una clase cualquiera llamada *softmax*.

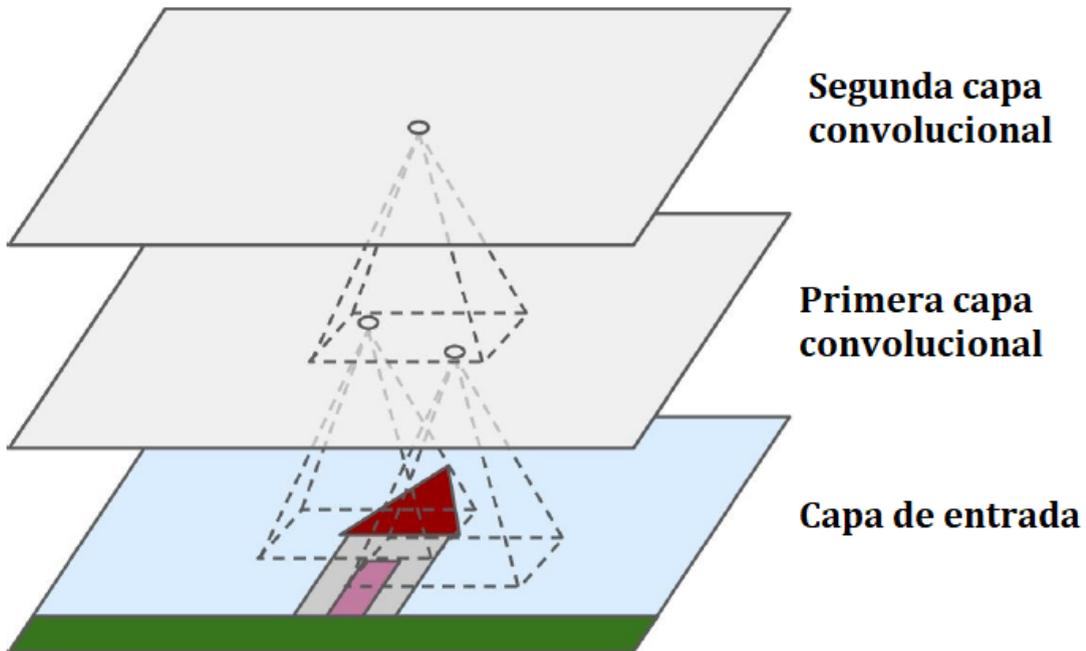


Figura 3.11: Capas de una CNN con campos receptivos locales

Capítulo 4

Armonización usando Aprendizaje Automático

4.1 Planteamiento del problema

Como primera aproximación a la armonización, se plantea realizar un análisis de distintas técnicas y algoritmos típicos de Aprendizaje Automático. La función principal de este experimento es de toma de contacto: queremos familiarizarnos con la naturaleza y con la estructura de los datos y de las imágenes con las que vamos a trabajar. Partimos de las DWI, y se van a analizar que tipos de medidas podemos mapear: DTI (tensores, autovalores..) o medidas tales como FA o MD. A priori nos interesa la FA por dos razones: su uso en estudios clínicos, y su dimensionalidad, que podría simplificar la complejidad de la solución. Los datos con los que se realiza la prueba serán datos pseudo sintéticos (ya que en el momento del desarrollo no se disponía aún de datos reales), por lo que cualquier resultado habría de confirmarse posteriormente en casos reales.

Asimismo, queremos ver cómo se comportan distintos algoritmos y modelos con estos datos, y en caso de obtener resultados positivos, cuáles de ellos resultan más adecuados o mejor orientados al problema. Los algoritmos que se van a probar son los siguientes:

1. Vecino más cercano (K-nn)
2. *Random forest*
3. *Extremely randomized trees*
4. Regresión de Ridge
5. Regresión lineal

4.2 Datos

Se realiza un estudio previo con el *phantom* realista de difusión generado a partir de datos reales propuesto en [51, 52]. La finalidad del uso de este *phantom* es poder comparar resultados con un *golden standard* y poder generar distintos conjuntos de parámetros a voluntad. El volumen original presenta un volumen $256 \times 256 \times 81$ con 51 direcciones de gradiente, $b=1200$ y libre de ruido. Permite generar volúmenes de difusión con diferente número de gradientes, valores de b (cercaos al original) y distintos niveles de SNR.

Asumiremos un modelo de ruido Rician estacionario [53] con un nivel realista ($\sigma = 2$ para un *baseline* con valores entre [0-255]). Para el experimento se mantendrá un solo valor de b y dos conjuntos de gradientes, 51 y 31. Como medida escalar a analizar se considerará la FA (que se sabe que es muy dependiente con el número de gradientes). Se genera una máscara de la sustancia blanca, para utilizar únicamente esos puntos en el entrenamiento de los distintos regresores. Para ilustrar los resultados se mostrará un corte central del volumen (corte 36).

En la figura 4.1 se muestra la FA calculada con 31 gradientes frente a la FA calculada con 51 gradientes. Tal y como se esperaba, hay una caída de los valores de FA cuando aumenta el número de direcciones consideradas. En la figura 4.2 se muestra el error absoluto entre ambas salidas. En algunos casos pasa del 40 %, claramente mucho mayor que las variaciones que habitualmente se buscan en los estudios clínicos. Sin embargo, este error no es lineal: no afecta de la misma manera a todos los valores de FA. Depende de la posición en el cerebro, del nivel de FA, etc. Si representamos en una gráfica la FA de 31 frente a la de 51 gradientes, (figura 4.3), lo que podemos ver es una dispersión de valores, pero no aparece un función de transferencia clara.

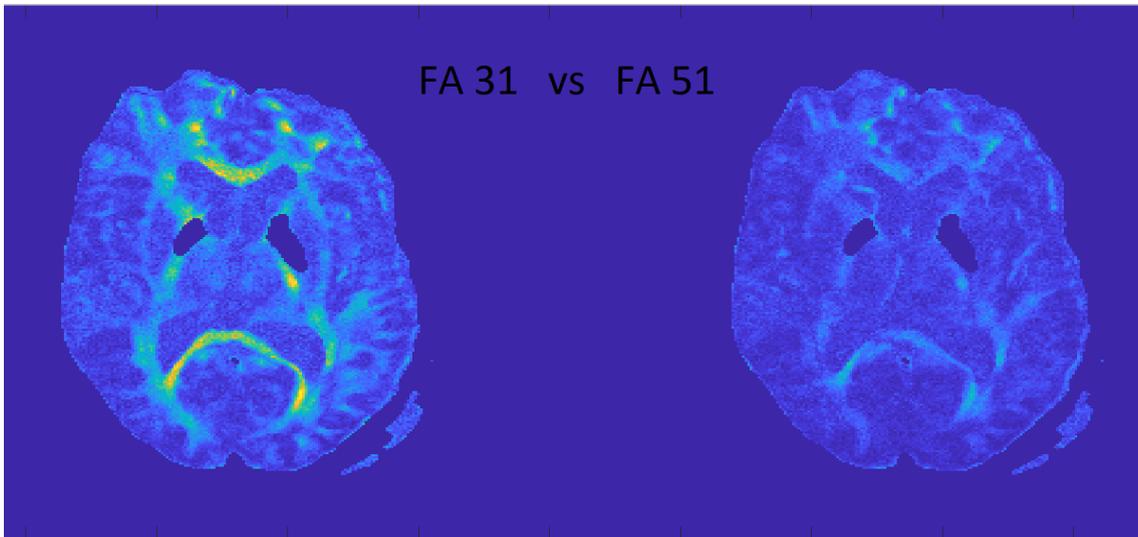


Figura 4.1: Anisotropía fraccional (FA) de un corte central del phantom realista de difusión. Izquierda: FA calculada con 31 gradientes. Derecha: FA calculada con 51 gradientes.

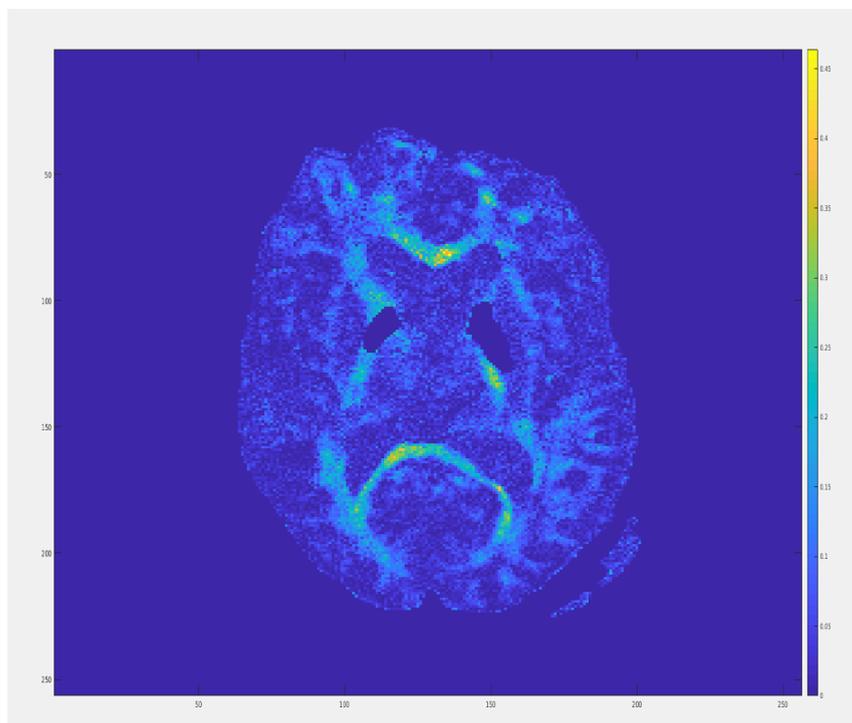


Figura 4.2: Diferencia entre FA de 51 y de 31 gradientes de la figura 4.1

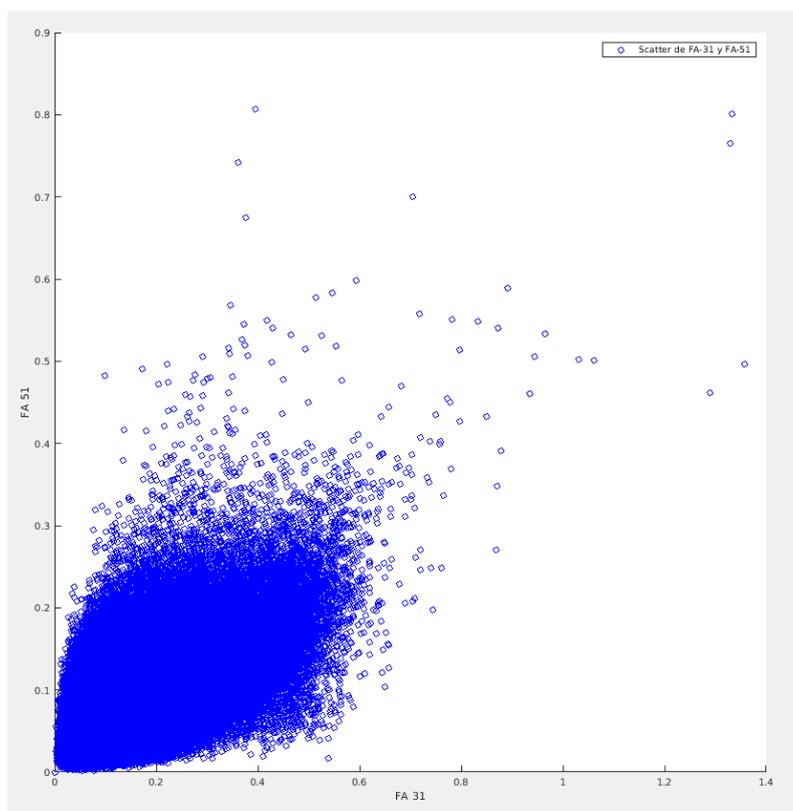


Figura 4.3: Scatter plot de la FA en los vóxeles de la sustancia blanca (phantom) calculada con 31 gradientes frente a 51 gradientes, usando los datos de la figura 4.1

4.3 Procedimiento

El proceso de entrenamiento va a ser el siguiente: se consideran 80 de los 81 cortes disponibles para realizar el entrenamiento de los distintos regresores. Dejaremos fuera el corte 36 que, posteriormente se usará como conjunto de datos de test. Respecto a las entradas (X) y salidas (Y) del sistema, existen las siguientes opciones (en todos los casos se busca relación vóxel a vóxel, sin considerar estructura local. n_1 y n_2 son dos números de gradientes distintos):

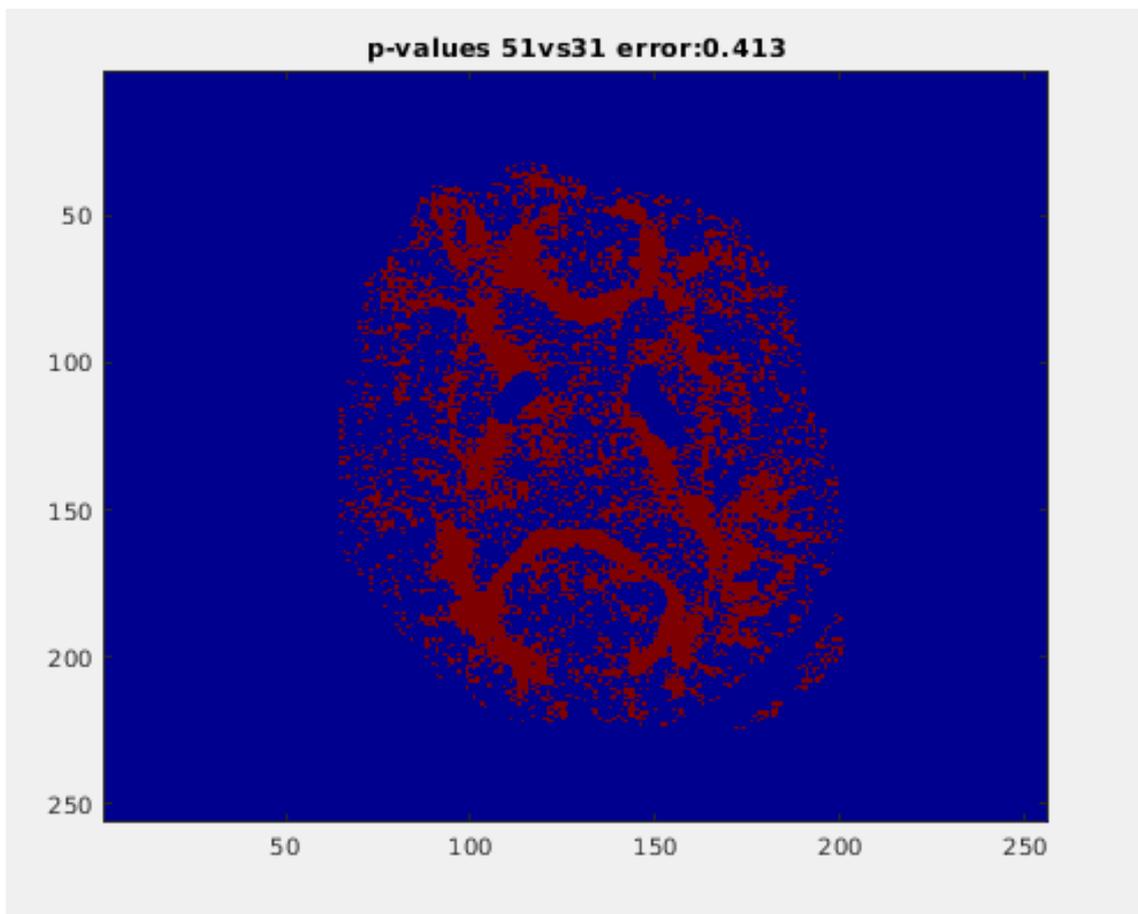


Figura 4.4: Puntos en los que un t-test detecta diferencias significativas entre la FA de 51 y 31 gradientes para $p < 0,01$

1. FA de entrada a FA de salida

$$X = FA_{n1} \quad Y = FA_{n2} \quad 1 \text{ entrada y } 1 \text{ salida}$$

2. Autovalores de entrada a FA de salida

$$X = \text{autovalores}_{n1} \quad Y = FA_{n2} \quad 3 \text{ entradas y } 1 \text{ salida}$$

3. Autovalores de entrada a autovalores de salida

$$X = \text{autovalores}_{n_1} \quad Y = \text{autovalores}_{n_2} \quad 3 \text{ entradas y } 3 \text{ salidas}$$

4. FA/Autovalores con información de vecinos (se añade algo de estructura local)

$$X = \text{FAS}_{n_1} \quad Y = \text{FAS}_{n_2} \quad 9 \text{ entradas y } 1 \text{ salida}$$

$$X = \text{autovalores}_{n_1} \quad Y = \text{FA}_{n_2} \quad 27 \text{ entradas y } 1 \text{ salida}$$

5. Otras posibilidades serían usar los DWIs o los DTIs enteros u otros cálculos a partir de las DTIs: MD, RA, AD, etc.

Para el presente estudio se han considerado solo las 3 primeras opciones. Para validar los resultados se consideran 50 repeticiones del mismo experimento (con distintas realizaciones de ruido). Se armonizan las medidas sobre el corte 36 usando el regresor estimado con los otros 80 cortes y se realiza un T-test con un p valor de $p = 0,01$ punto a punto sobre las 50 realizaciones. De este modo, esperamos generar un mapa de diferencias significativas entre la FA generada por distinto número de gradientes. Como medida de error se considerará el tanto por ciento de los puntos en los que el test detecta diferencias significativas. A modo de ejemplo, en la figura 4.4 se muestra el error inicial: los puntos con diferencias significativas para los datos sin armonizar, considerando la FA de 31 y de 51 gradientes. El 41 % de los puntos de la sustancia blanca presenta diferencias.

4.4 Resultados

En todos los casos se compara el resultado de armonizar 31 con 51 (se armoniza el conjunto de 31 gradientes, generando una salida similar a 51, que llamaremos $\hat{51}$ y de 51 con 31 (se armoniza el conjunto de 51 gradientes, generando una salida similar a 31, que llamaremos $\hat{31}$). En todos los casos ha funcionado mejor el primer caso, por lo que, a modo de ilustración, solo se mostrarán resultados visuales de ese caso.

1. **Entrada: FA - Salida: FA.** Los resultados numéricos están recogidos en la tabla 4.3. En la figura 4.5 se muestra la comparativa entre la FA original (51 gradientes) y las diferentes estimaciones generadas a partir de la señal de 31 gradientes. Los mapas de los p-valores se recogen en la figura 4.6. Los scatter plot entre la señal original y el mapeo están recogidos en la figura 4.7. Las salidas muestran un cierto suavizado que elimina el ruido presente en la imagen.

FA \rightarrow FA

REGRESOR	Error 31 \rightarrow 51	Error 31 \rightarrow $\hat{51}$	Error 31 \rightarrow $\hat{31}$
Knn	0.413	0.128	0.495
Random Forest	0.413	0.154	0.475
X-Tree	0.413	0.164	0.483
Ridge	0.413	0.106	0.540
Lineal	0.413	0.107	0.546

Cuadro 4.1: Error entre los dos conjuntos de datos usando diferentes regresores. Mapeo de FA de entrada en FA de salida.

Eigenvalues \rightarrow FA

REGRESOR	Error 31 \rightarrow 51	Error 31 \rightarrow $\hat{51}$	Error 31 \rightarrow $\hat{31}$
Knn	0.413	0.119	0.298
Random Forest	0.413	0.132	0.362
X-Tree	0.413	0.115	0.298
Ridge	0.413	0.155	0.395
Lineal	0.413	0.124	0.356

Cuadro 4.2: Error entre los dos conjuntos de datos usando diferentes regresores. Mapeo de FA de entrada en FA de salida.

- Entrada: Autovalores - Salida: FA.** Los resultados numéricos están recogidos en la Tabla 5.2. En la figura 4.8 se muestra la comparativa entre la FA original (51 gradientes) y las diferentes estimaciones generadas a partir de la señal de 31 gradientes. Los mapas de los p-valores se recogen en la figura 4.9. Los scatter plot entre la señal original y el mapeo están recogidos en la figura 4.10
- Entrada: Autovalores - Salida: autovalores.** Los resultados numéricos están recogidos en la Tabla 5.2. En la figura 4.11 se muestra la comparativa entre la FA original (51 gradientes) y las diferentes estimaciones generadas a partir de la señal de 31 gradientes. Los mapas de los p-valores se recogen en la figura 4.12. Los scatter plot entre la señal original y el mapeo están recogidos en la figura 4.13

En resumen, los mejores resultados para cada método son los siguientes:

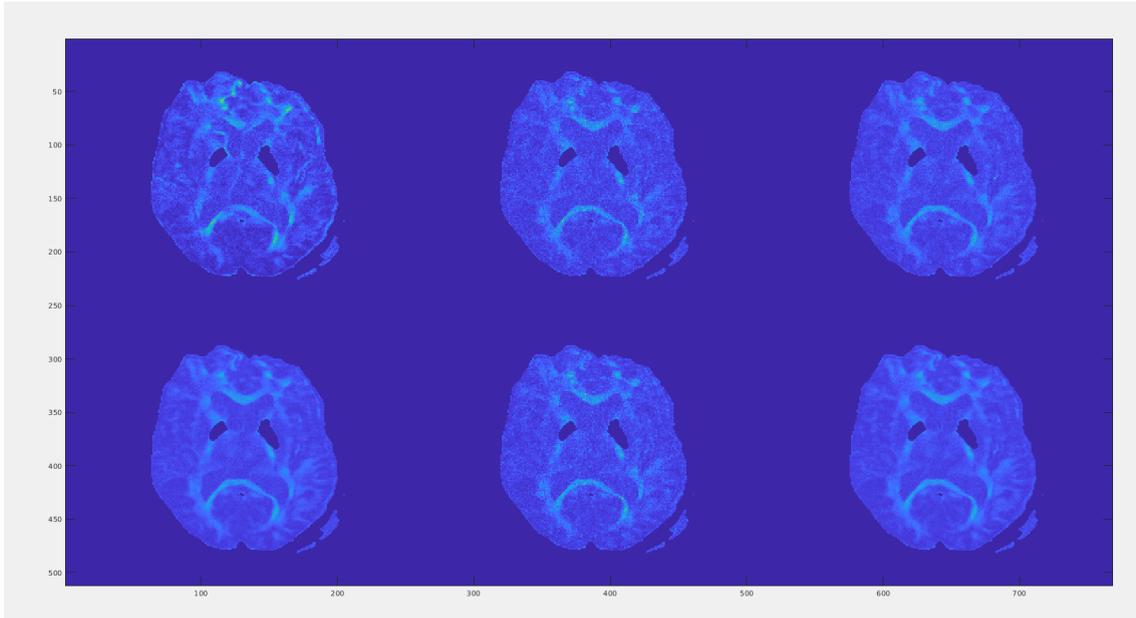


Figura 4.5: Comparación entre la FA original (51 gradientes) y las diferentes estimaciones generadas a partir de la señal de 31 gradientes. De izquierda a derecha y de arriba a abajo: (1) Original; (2) X-tree; (3) K-nn; (4) Lineal; (5) Random Forest; (6) Ridge. (Caso FA \rightarrow FA).

Eigenvalues \rightarrow Eigenvalues

REGRESOR	Error 31 \rightarrow 51	Error 31 \rightarrow $\hat{51}$	Error 31 \rightarrow $\hat{31}$
Knn	0.413	0.123	0.298
Random Forest	0.413	0.172	0.385
X-Tree	0.413	0.166	0.376
Ridge	0.413	0.151	0.372
Lineal	0.413	0.109	0.302

Cuadro 4.3: Error entre los dos conjuntos de datos usando diferentes regresores. Mapeo de FA de entrada en FA de salida.

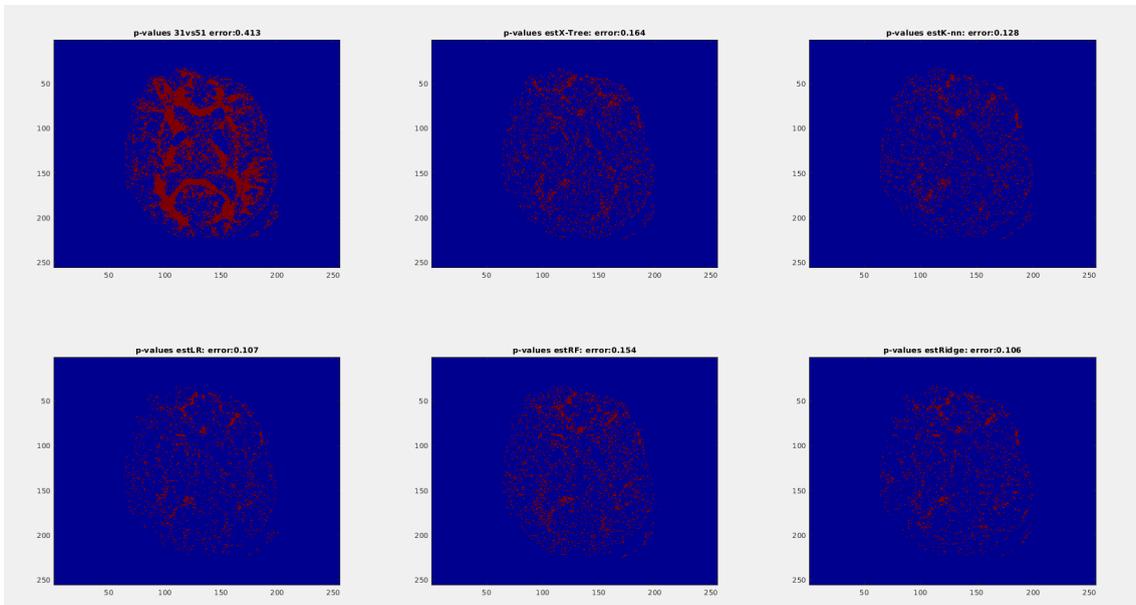


Figura 4.6: Puntos en los que un t-test detecta diferencias significativas entre la FA de 51 y la estimada a partir de 31 gradientes para $p < 0,01$. (Caso FA \rightarrow FA).

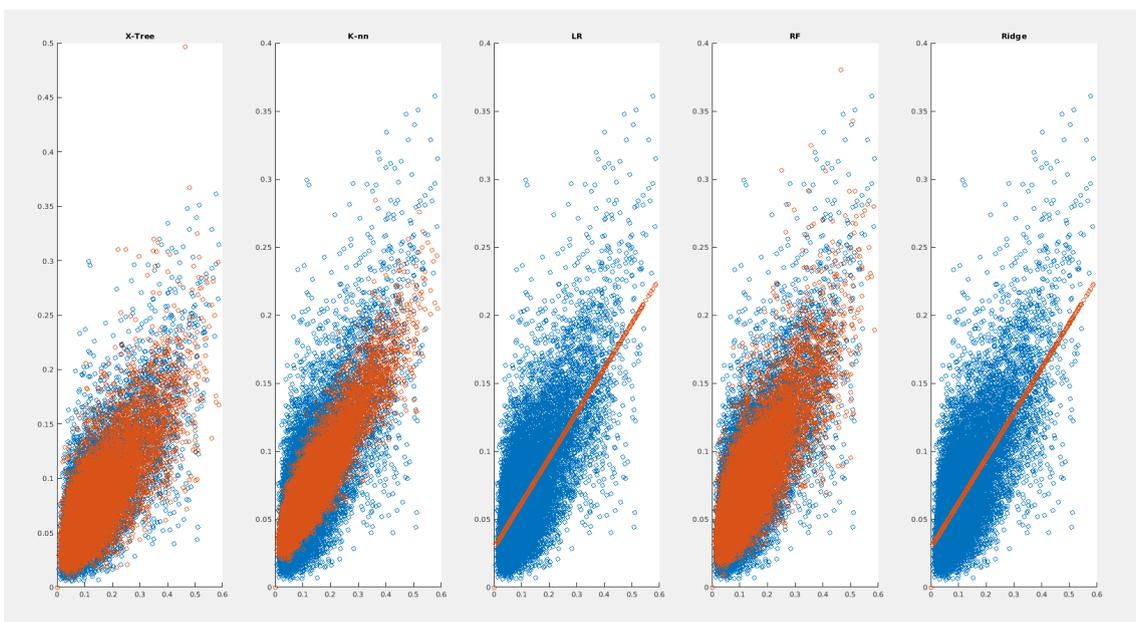


Figura 4.7: Scatter plot de la FA en los vóxeles de la sustancia blanca (phantom) calculada con 31 gradientes frente a 51 gradientes (azul) y 51 frente a 51 estimada (naranja). (Caso FA \rightarrow FA).

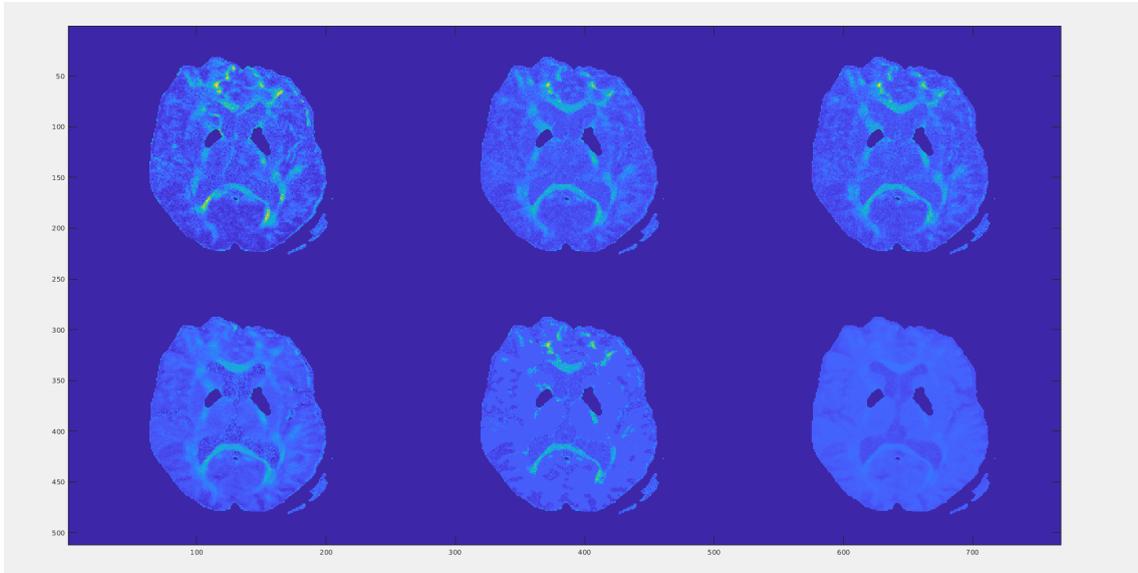


Figura 4.8: Comparación entre la FA original (51 gradientes) y las diferentes estimaciones generadas a partir de la señal de 31 gradientes. De izquierda a derecha y de arriba a abajo: (1) Original; (2) X-tree; (3) K-nn; (4) Lineal; (5) Random Forest; (6) Ridge. (Caso autovalores \rightarrow FA).

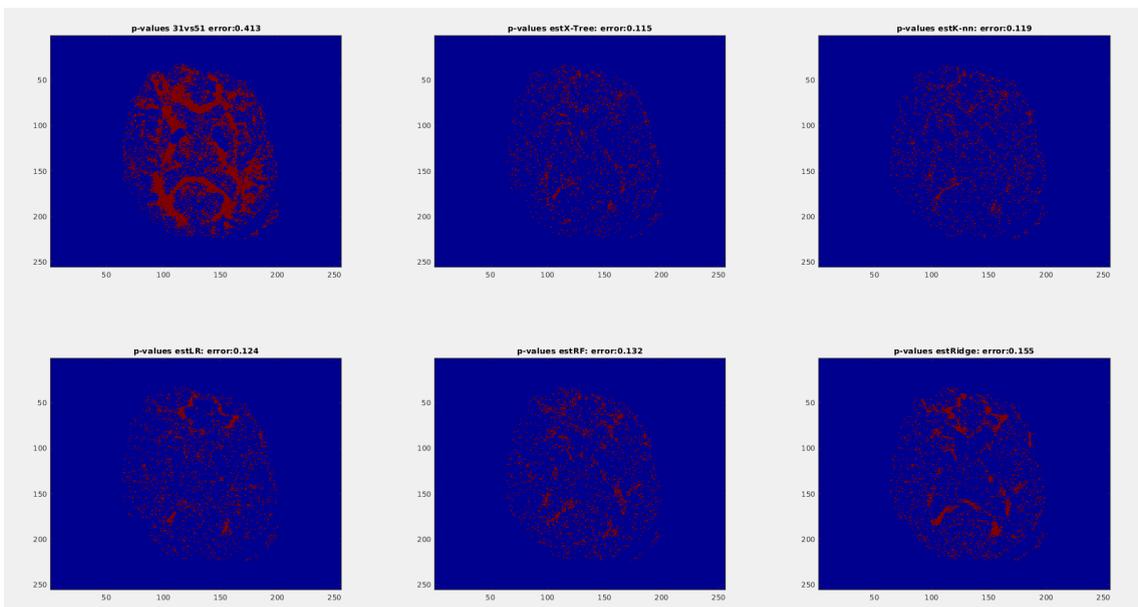


Figura 4.9: Puntos en los que un t-test detecta diferencias significativas entre la FA de 51 y la estimada a partir de 31 gradientes para $p < 0,01$. (Caso autovalores \rightarrow FA).

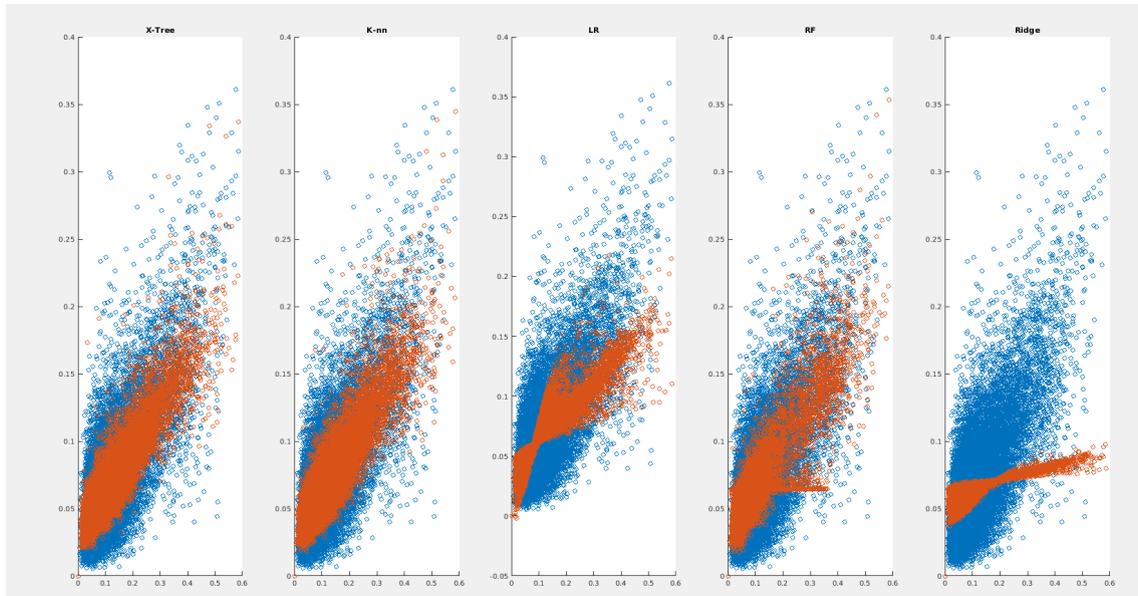


Figura 4.10: Scatter plot de la FA en los vóxeles de la sustancia blanca (phantom) calculada con 31 gradientes frente a 51 gradientes (azul) y 51 frente a 51 estimada (naranja). (Caso autovalores \rightarrow FA).

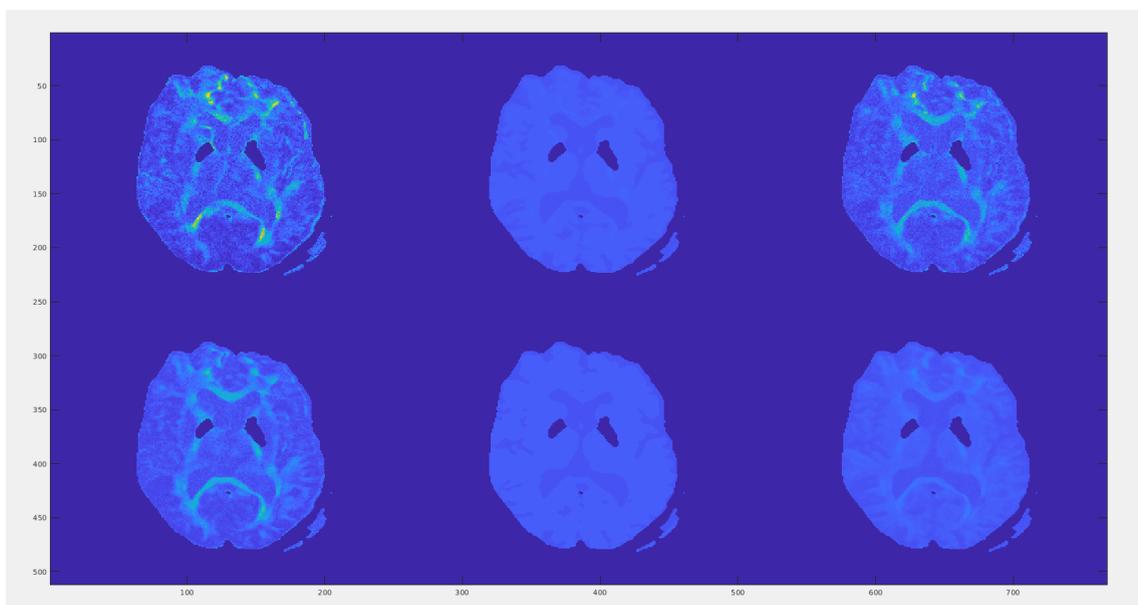


Figura 4.11: Comparación entre la FA original (51 gradientes) y las diferentes estimaciones generadas a partir de la señal de 31 gradientes. De izquierda a derecha y de arriba a abajo: (1) Original; (2) X-tree; (3) K-nn; (4) Lineal; (5) Random Forest; (6) Ridge. (Caso autovalores \rightarrow autovalores).

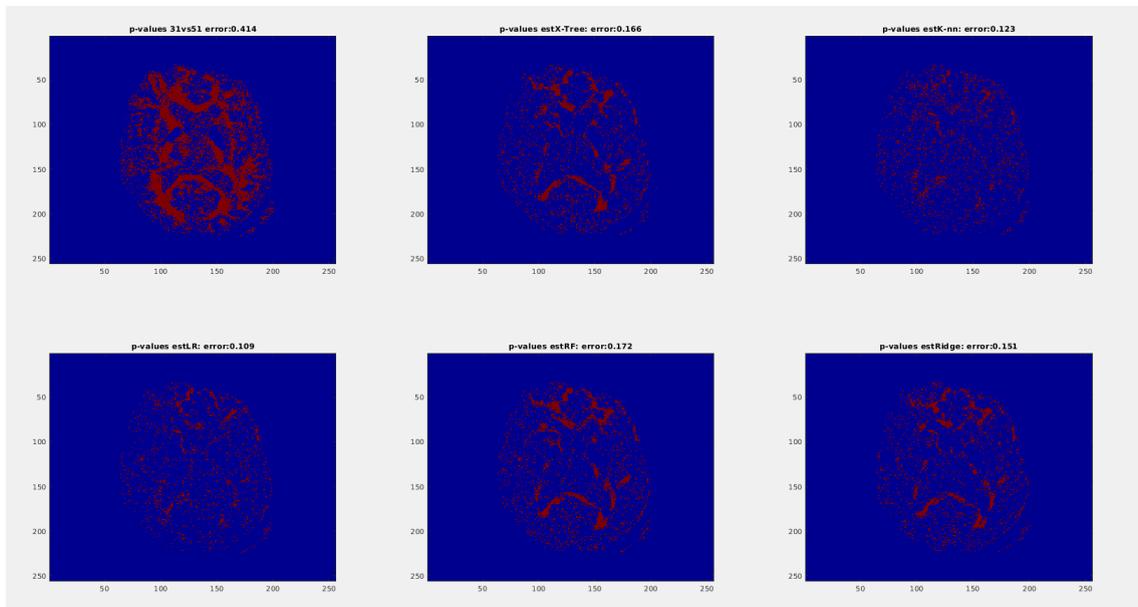


Figura 4.12: Puntos en los que un t-test detecta diferencias significativas entre la FA de 51 y la estimada a partir de 31 gradientes para $p < 0,01$. (Caso autovalores \rightarrow autovalores).

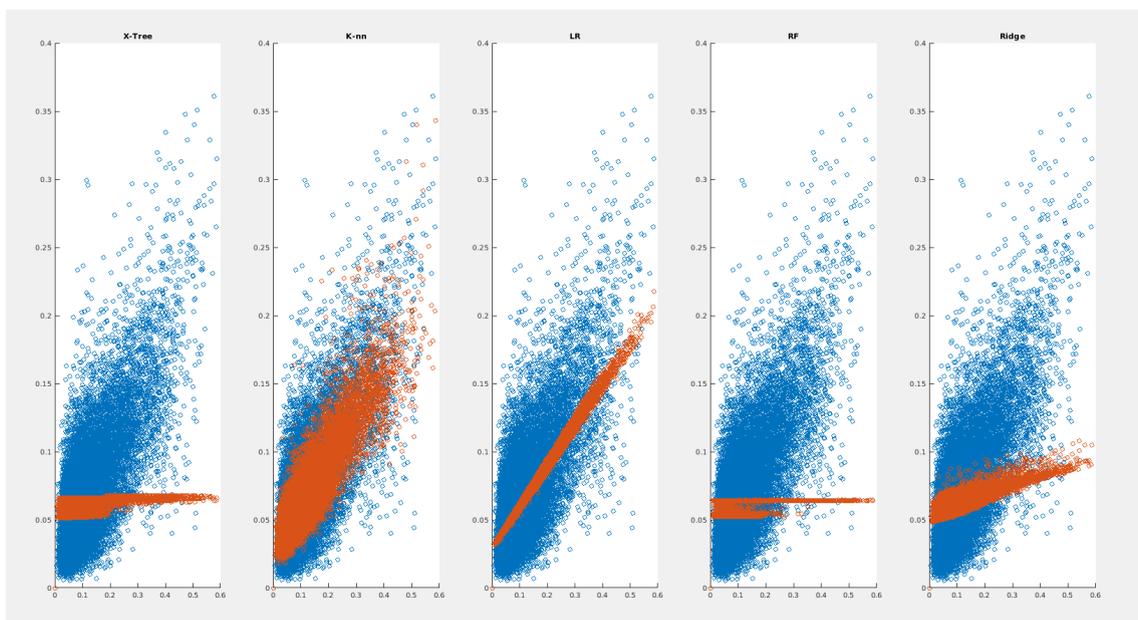


Figura 4.13: Scatter plot de la FA en los vóxeles de la sustancia blanca (phantom) calculada con 31 gradientes frente a 51 gradientes (azul) y 51 frente a 51 estimada (naranja). (Caso autovalores \rightarrow autovalores).

4.5 Conclusiones

Esta primera toma de contacto nos permite obtener alguna aclaración sobre como proseguir. En primer lugar, de los métodos propuestos, el que parece funcionar más razonadamente en todos los casos generando imágenes con una estructurada razonadamente similar es el de realizar mapeos de FA a FA. Las conclusiones que obtenemos es que la FA parece mantener suficiente información para que sea viable realizar la armonización a este nivel. Al fin y al cabo, utilizar autovalores o tratar de mapear FA a partir de autovalores añade un nivel extra de complejidad que puede ser innecesario.

El mayor defecto de este experimento es el planteamiento de la entrada-salida. Tal y cómo trabajamos, la entrada y salida del sistema es vóxel a vóxel, por lo que ocurren dos importantes problemas:

- Los datos no mantienen un tamaño constante. Cada slice dentro de un mismo cerebro es de distinto tamaño, y la misma slice de dos cerebros también lo es. Por lo tanto el mapeo punto a punto se puede estar haciendo en zonas distintas.
- El mapeo vóxel a vóxel está asumiendo que no hay información local y no es así. Al final, un vóxel está rodeado en una estructura 3D por vecinos, y puede ser importante hacerlo notar en la entrada del sistema de alguna manera. Por ejemplo, para estimar un vóxel cualquiera, usar como entradas ese mismo vóxel y todos sus vecinos inmediatos

Capítulo 5

Armonización usando redes neuronales

5.1 Planteamiento del problema

La idea que surge inmediatamente de los resultados previos para dar el siguiente paso es que es necesario definir una estructura local y delimitar los tamaños de entrada y de salida para poder usar estimadores de regresión. Se plantea por tanto bien dar toda la imagen 2D de un slice como entrada al sistema (incluyendo los espacios vacíos), o bien en caso de ser excesivamente grandes, usar pequeños recortes de cada slice constituyendo imágenes más pequeñas como entradas.

Ambos procedimientos nos llevan de manera natural a plantear el uso de redes neuronales convolucionales (CNN). Vamos a limitar la armonización en esta ocasión a la FA como medida, puesto que suponen imágenes de un sólo canal (imagen en escala de grises) y otras aproximaciones, como los tensores, los autovalores o los DWI supondrían como mínimo 3 o más canales, complicando en exceso la red y además se ha de tener en cuenta que la capacidad de computación necesaria para el entrenamiento y despliegue de las CNN es bastante superior a los algoritmos de Aprendizaje Automático vistos en el capítulo anterior.

5.2 CNNs

Las redes CNN están típicamente orientadas a clasificación de imágenes y consisten de capas concatenadas de convoluciones y submuestreo (mediante *pooling* que permiten ir extrayendo características a distintos niveles de profundidad de la imagen, para unirse finalmente a una red neuronal *fully connected* (las neuronas están conectadas todas con

todas) que permite generar una salida a placer. Ya que nosotros queremos que la salida sea otra imagen (regresión), tenemos que realizar algunos cambios sobre la función de activación de la capa de salida. Se cambia la capa de activación de una ReLU a una lineal, y se alimenta para el entrenamiento como salida la imagen esperada como vector 1D.

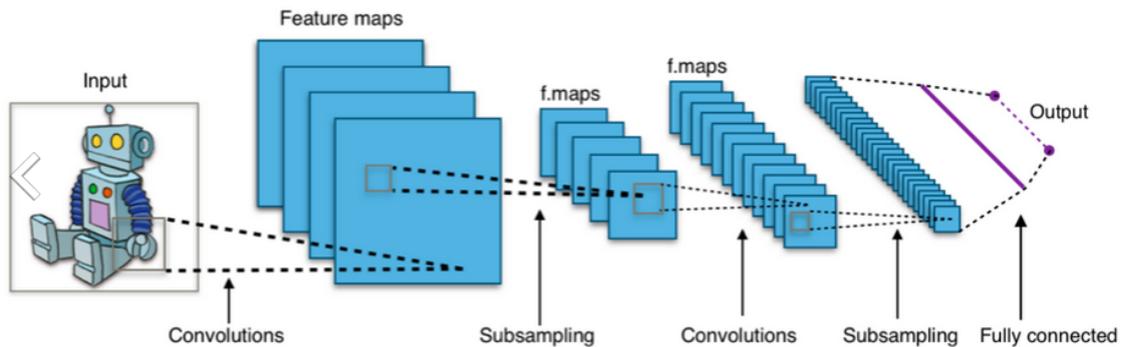


Figura 5.1: Esquema de un modelo típico de CNN

Durante el desarrollo de esta solución surgen dos problemas, las imágenes son relativamente grandes para alimentar la red con ellas enteras, ya que es la capa *fully connected* crece de manera exponencial con el tamaño. Como solución, variamos las entradas de la imagen para alimentar con *patches*, de cada *slice* del cerebro generamos imágenes más pequeñas con *overlap* para entrenar la red. De esta manera la red propuesta se vuelve viable, por cada *patch* de entrada, generamos otro de salida, y para reconstruir la imagen final debemos deshacer ese *overlap* como si de un puzle se tratase. El principal problema es que esto genera problemas de continuidad en la imagen final resultante.

```

In [3]: runfile('/home/export/pfc/dmaramo/DWI/DL/CNN_fas.py', wdir
X_train shape: (88441, 32, 32)
88441 train samples
15608 test samples

```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 30, 30, 32)	320
max_pooling2d_1 (MaxPooling2D)	(None, 15, 15, 32)	0
conv2d_2 (Conv2D)	(None, 13, 13, 32)	9248
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 32)	0
flatten_1 (Flatten)	(None, 1152)	0
dense_1 (Dense)	(None, 1024)	1180672

```

Total params: 1,190,240
Trainable params: 1,190,240
Non-trainable params: 0

```

```

Train on 88441 samples, validate on 15608 samples

```

Figura 5.2: Ejemplo de una red hecha con Keras que toma como entrada *patches* de tamaño 32 x 32 y genera en la salida un vector de 1024 que constituye la imagen de salida

5.3 U-Net

Vistos los problemas que surgen con una red típica CNN, se plantea el uso de distintas arquitecturas. Una de las ideas que surge es utilizar una red convolucional total o FCN (*Fully Convolutional Network*). Este tipo de red está orientada a problemas de segmentación, mucho más complejos que los de clasificación. La diferencia de estas arquitecturas con las CNN es que no hacen uso de esa capa *fully connected*, que resultaba ser en nuestro caso un cuello de botella y nos impedía trabajar con la imagen al completo. En su lugar, la capa de salida se sustituye por otro conjunto de capas convolucionales que realizan deconvoluciones o convoluciones transpuestas, que mediante interpolación, vuelven a generar una imagen del tamaño de la original. Además, aferrándose a la idea original de las redes convolucionales de extraer características a distintos niveles de profundidad de la imagen (información local o espacial), se puede aprovechar estas nuevas capas para adquirir aún más información.

Dentro de la variedad de propuestas de redes FCN que se pueden encontrar, reparamos en la arquitectura U-Net, desarrollada especialmente para la segmentación de imágenes biomédicas en el departamento de Informática de la Universidad de Freiburg [54]. Resulta interesante porque está orientada a imágenes del estilo a las que usamos nosotros. La red consiste de dos partes, una parte inicial en la que entrada se contrae, y un camino expansivo, en el que se recupera a la salida el tamaño original de la imagen. Es esta forma la que da el nombre a la red.

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 256, 256, 1)	0	
conv2d_1 (Conv2D)	(None, 256, 256, 32)	320	input_1[0][0]
conv2d_2 (Conv2D)	(None, 256, 256, 32)	9248	conv2d_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 128, 128, 32)	0	conv2d_2[0][0]
conv2d_3 (Conv2D)	(None, 128, 128, 64)	18496	max_pooling2d_1[0][0]
conv2d_4 (Conv2D)	(None, 128, 128, 64)	36928	conv2d_3[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 64, 64, 64)	0	conv2d_4[0][0]
conv2d_5 (Conv2D)	(None, 64, 64, 128)	73856	max_pooling2d_2[0][0]
conv2d_6 (Conv2D)	(None, 64, 64, 128)	147584	conv2d_5[0][0]
conv2d_transpose_1 (Conv2DTrans	(None, 128, 128, 256)	131328	conv2d_6[0][0]
concatenate_1 (Concatenate)	(None, 128, 128, 320)	0	conv2d_transpose_1[0][0] conv2d_4[0][0]
conv2d_7 (Conv2D)	(None, 128, 128, 64)	184384	concatenate_1[0][0]
conv2d_8 (Conv2D)	(None, 128, 128, 64)	36928	conv2d_7[0][0]
conv2d_transpose_2 (Conv2DTrans	(None, 256, 256, 256)	65792	conv2d_8[0][0]
concatenate_2 (Concatenate)	(None, 256, 256, 288)	0	conv2d_transpose_2[0][0] conv2d_2[0][0]
conv2d_9 (Conv2D)	(None, 256, 256, 32)	82976	concatenate_2[0][0]
conv2d_10 (Conv2D)	(None, 256, 256, 32)	9248	conv2d_9[0][0]
conv2d_11 (Conv2D)	(None, 256, 256, 1)	33	conv2d_10[0][0]
Total params: 797,121			
Trainable params: 797,121			
Non-trainable params: 0			

Figura 5.4: Esquema de la red propuesta en Keras

Un *autoencoder* es un tipo de red neuronal que aprende a copiar su entrada en su salida. Tiene una capa interna que describe un código que representa a una entrada en particular, y está constituida por dos partes principales, un codificador que mapea la entrada al código, y un decodificador que mapea el código a una reconstrucción de la entrada original. Esta copia que puede parecer absurda, se hace aplicando restricciones (extracción de características, capacidad de generalizar de la red), de tal manera que la reconstrucción es solo aproximada, y se priorizan los aspectos más relevantes de la entrada para ser copiados. Las principales utilidades de un *autoencoder* son la reducción de dimensionalidad, el aprendizaje de características y la limpieza del ruido. Esto es lo que está ocurriendo en nuestro caso.

Por tanto, el esquema alternativo que planteamos para las entradas y las salidas del sistema, apoyándonos en la idea de los *autoencoders*, nos lleva a proponer una predicción en dos pasos, como se ilustra en las figuras 5.7 y 5.8.



Figura 5.5: Planteamiento estándar de entradas y salidas

- En un primer paso, se usa la red a modo de autoencoder, proporcionando a la entrada y a la salida la misma FA de un determinado número de gradientes. Esta red memoriza la estructura de las imágenes, y cuando la usamos para realizar predicciones, dando como entrada una FA, nos la da a la salida limpia y filtrada de ruido. Necesitaremos entrenar una de estas redes para cada número de gradientes. Este proceso es compatible para un futuro con otros parámetros que queramos variar, b-value, tamaño de vóxel, etc...
- El segundo paso es el mapeo entre las salidas del primer paso que queramos hacer. Como entrada entrará una predicción y como salida forzaremos a la otra, por tanto, cada red de este tipo a entrenar se definirá con dos parámetros (entrada,salida), por ejemplo (21,61), mapearía FAs de 21 a 61.

Por tanto, y siguiendo este planteamiento, el proceso a seguir dadas las redes entrenadas y nuevas muestras de FA que queramos llevar a un espacio común, sería el que se indica en la figuras 5.9 y 5.10. En este ejemplo, el espacio común sería el de la estimación de 61 gradientes. Si tenemos una imagen de 61 gradientes, la tendremos que pasar por la red del primer paso y ya la tendremos en este espacio común, y si es de 21 gradientes, primero tendremos que limpiarla y luego mapearla a 61 para obtener la predicción.

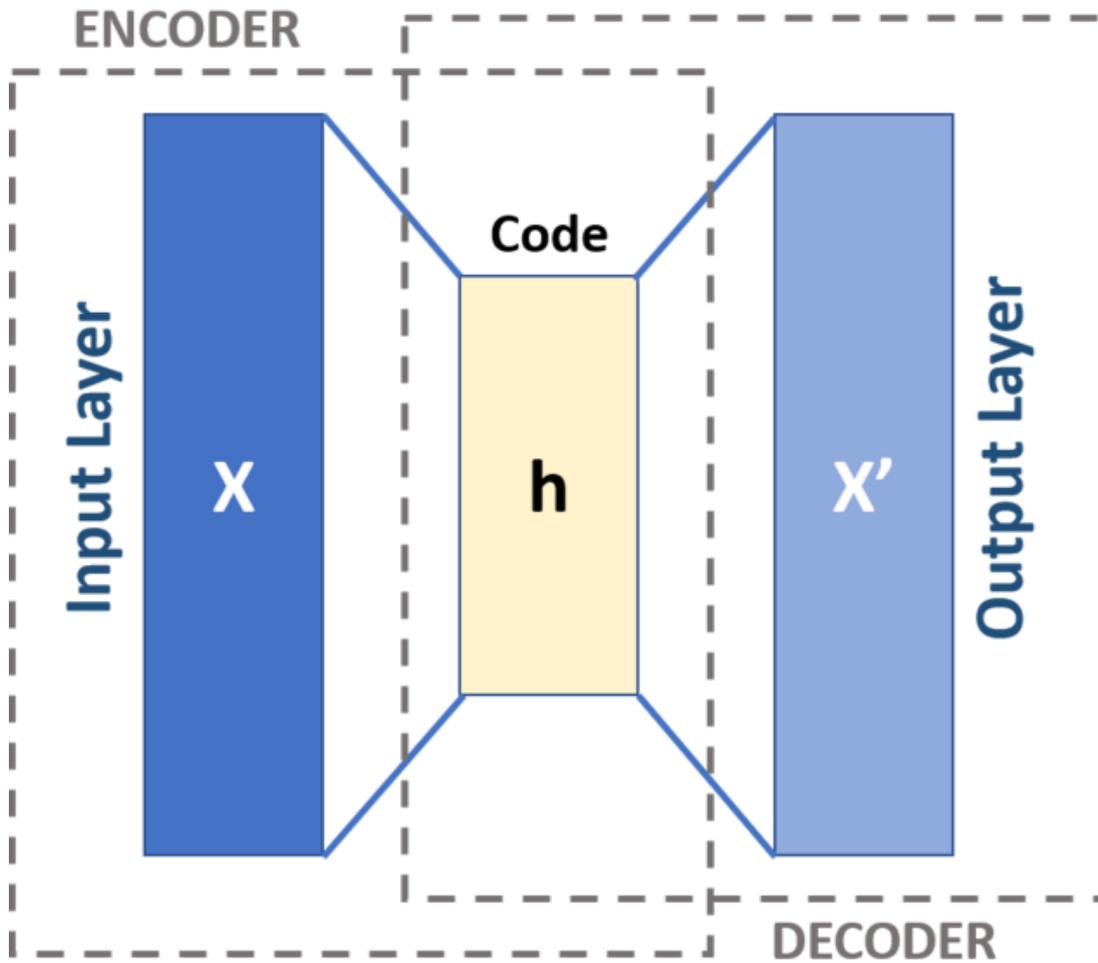


Figura 5.6: Esquema básico de un *autoencoder*

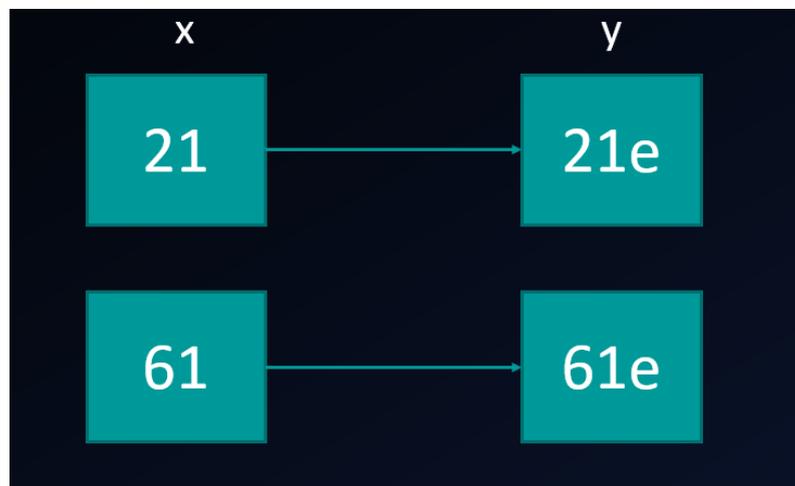


Figura 5.7: Primer paso: entradas, salidas. Filtrado de las FA.



Figura 5.8: Segundo paso: entradas, salidas. Mapeo de las FA.

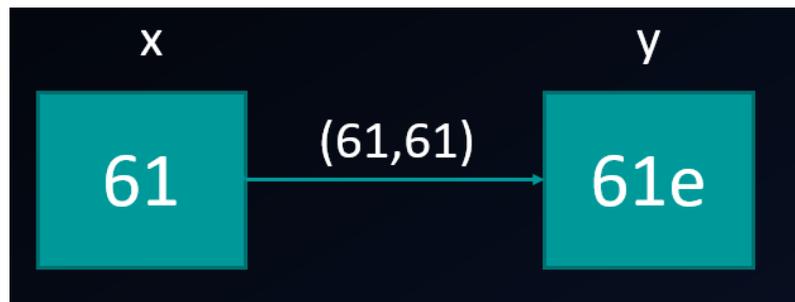


Figura 5.9: Llevar al espacio común una FA de 61 gradientes

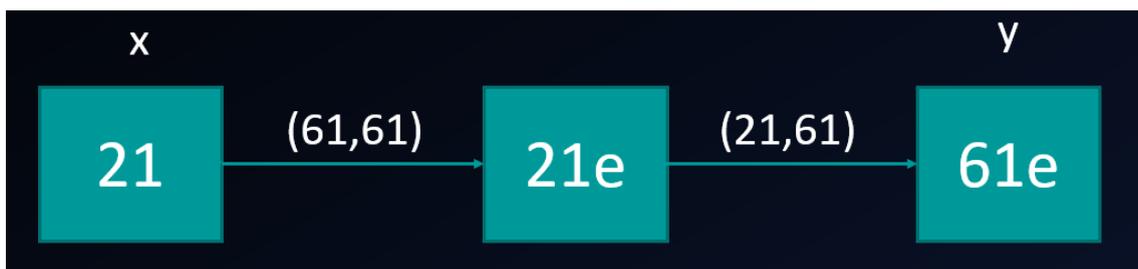


Figura 5.10: Llevar al espacio común una FA de 21 gradientes

5.4 Datos

En este caso, se plantea el uso de un *dataset* real para realizar los experimentos. Para experimentos previos, no estaba disponible. De cualquier manera, se consideraba correcto como primera aproximación al problema el trabajo con datos pseudo sintéticos, que conservan una estructura muy parecida a los datos reales.

El dataset disponible consiste en un conjunto de datos de MRI que han sido obtenidos de personas de distintos sexos y edades. Los datos consisten de:

- Set de entrenamiento: 30 DWIs, consistiendo de 10 controles sanos (HC), 10 paciente con esquizofrenia (SZ) y 10 pacientes con casos de cefaleas
- Sets de test: 32 DWIs de controles sanos y 32 DWIs de pacientes con esquizofrenia

32 pacientes con esquizofrenia (SZ) y de 32 controles sanos (HC) . Estas adquisiciones han sido tomadas usando la máquina *Philips Achieva 3T* en el centro de MRI de la Universidad de Valladolid, incluyendo las imágenes T1 y las DWI. Para las imágenes T1 anatómicas, los parámetros de adquisición usados fueron los siguientes: Secuencia TFE, tamaño de matriz de $256 \times 256, 1 \times 1 \times 1 \text{ mm}^3$ de resolución espacial y un total de 160 cortes cubriendo el cerebro en su totalidad. En cuanto a las DWIs, los parámetros de adquisición son: 61 direcciones de gradiente, un volumen de baseline, un *b-value* de 1000 s/mm^2 , tamaño de vóxel de $2 \times 2 \times 2 \text{ mm}^3$, tamaño de matriz de 128×128 y 34 cortes del volumen total del cerebro. La elección de las direcciones de los 61 gradientes está diseñada de tal manera que se puede submuestrear en otro conjunto de 40 gradientes válidos o de 21. Esto resulta de gran interés para el estudio que estamos realizando, puesto que precisamente este es el parámetro principal que se cubre en este trabajo. Más información sobre los detalles clínicos de los participantes, así como los criterios de inclusión o exclusión, se pueden encontrar en [55]

5.5 Preprocesado de los datos

Partiendo de los DWIs previamente mencionados, necesitamos llegar a las FAs con las que vamos a trabajar. Los cálculos se van a realizar usando los programas Matlab, MRtrix [56] y FSL [57, 58]. Los DWIs se encuentran en formato NIfTI (Neuroimaging Informatics Technology Initiative), un estándar médico para la representación de imágenes 3D como un array de *slices* o imágenes 2D. Junto a cada DWI, disponemos de un fichero *.bval* y otro *.bvec*, que nos indican respectivamente los *b-values* y las direcciones de los gradientes. A continuación se listan los pasos que damos:

- **Reordenar los ficheros *.bval* y *.bvec*:** DWIConvert es el programa que nos proporciona los DWIs en formato *.nii* y estos dos ficheros con información de la captura de los DWIs. El formato de los b-values es de un valor por línea, y para usarlos en FSL necesitamos que estén todos en la misma línea. Para las direcciones de gradiente, vienen ordenados por columnas, y necesitamos que estén ordenados por filas. Se realiza por tanto un script en Matlab que corrige estos ficheros para cada DWI a procesar.
- **Submuestreo de los DWIs:** Como se explicaba previamente, los DWIs están tomados con un conjunto de 61 direcciones de gradiente, tales que resultan submuestreables. Si cogemos los primeros 21 gradientes, tendremos un DWI de 21 gradientes, y si usamos los 40 posteriores, generaremos el DWI de 40 gradientes. El volumen de baseline es común para los 3 casos. Los nuevos *.nii* se generan mediante la utilidad *flsroi* y *fslmerge* de FSL
- **Creación de las máscaras:** Para cada DWI y cada número de gradientes, generamos la máscara del cerebro usando la utilidad *dwi2mask* de MRtrix
- **Cálculo de tensores y valores asociados:** A partir del *.nii*, de la máscara, de los *b-values* y las direcciones de gradiente, calculamos con la utilidad *dtifit* los tensores, autovalores de FSL y las FA de cada DWI.
- **Preprocesado y data augmentation:** Leemos los *.nii* con Matlab. Comprobamos que las imágenes son correctas y se detectan dos DWIs incorrectos (los tamaños no son correctos), uno del set de entrenamiento y otro del set de test, que se eliminan. Para el resto, se realizan dos pasos de preprocesado. Se cambia el orden de las dimensiones de [128 x 128 x 66] a [66 x 128 x 128] porque las utilidades de Python que vamos a usar interpretan los *slices* como la primera dimensión de la matriz. Además, para tener un mayor número de datos para entrenar, y que la red pueda generalizar sobre las posiciones de las imágenes, de cada DWI se generan 4 distintos rotando el original 90, 180 y 270 grados respectivamente. Los datos de entrenamiento se guardan todos en un mismo fichero *.mat* y los de test cada uno en fichero *.mat* separado, y estructurando según sean controles o pacientes.

5.6 Procedimiento

Así pues, se procede al proceso de entrenamiento de las redes mencionadas. La estrategia que se sigue es la siguiente: se cargan todos los slices los datos provenientes de los 30 sujetos (FAs con *data augmentation*) para 21 y 61 gradientes, que es el mapeo principal que vamos a realizar, ya que a priori es donde debe haber más diferencias, y se dividen de forma aleatoria en dos conjuntos, conteniendo cada uno de ellos la mitad de los datos. La primera mitad servirá para entrenar las redes del primer paso (*data augmentation*) y el resto para las del segundo paso (mapeo).

Las redes están hechas en Python 3, usando Tensorflow y Keras. De todos los datos que dediquemos al entrenamiento de la red, el 15 % se usarán para validación del entrenamiento. Puesto que los datos son todos de la misma naturaleza y oscilan entre los mismos valores (valores de FA entre 0 y 1), no se aplica normalizado previo a los datos antes de entrar a la red. Se entrena en la versión GPU de Tensorflow para acelerar el proceso, y se realizan 5 *epochs* de un tamaño de *batch* de 20.

Una vez las redes finalizan su entrenamiento, se guardan los pesos y el modelo de la red en ficheros con formato *.h5*, que se pueden recuperar en cualquier momento para realizar predicciones sobre datos nuevos. Usando estos modelos se crean las predicciones para los 31 controles sanos (el DWI que quitamos del conjunto de test pertenecía a este grupo) y para los 32 pacientes del conjunto de test, con la idea de validar estas predicciones y ver los resultados.

A continuación, se recogen gráficas de las imágenes generadas y comparaciones entre ellas y las originales, así como un cálculo de los *p-values* (como el que se realizó en el primer experimento) en los casos de un paciente de esquizofrenia y de un control sano para la slice número 30 (se ha preparado un script que permite ver todo un cerebro también, pero no es práctico de plasmar en papel):

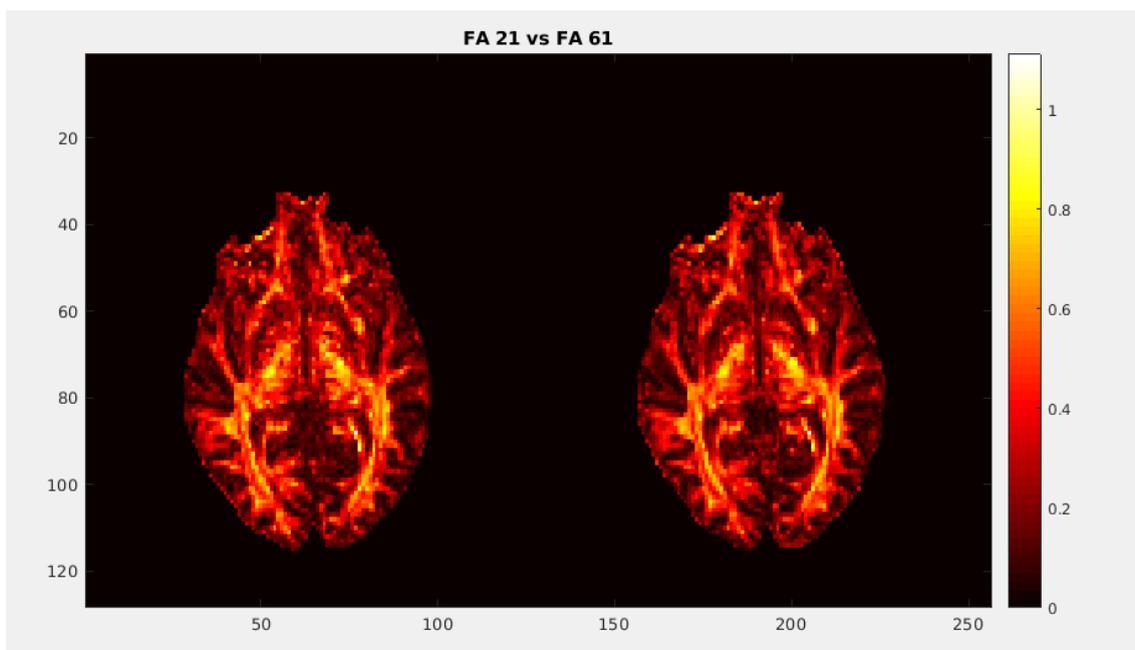


Figura 5.11: Comparativa entre las FA originales para un paciente de esquizofrenia

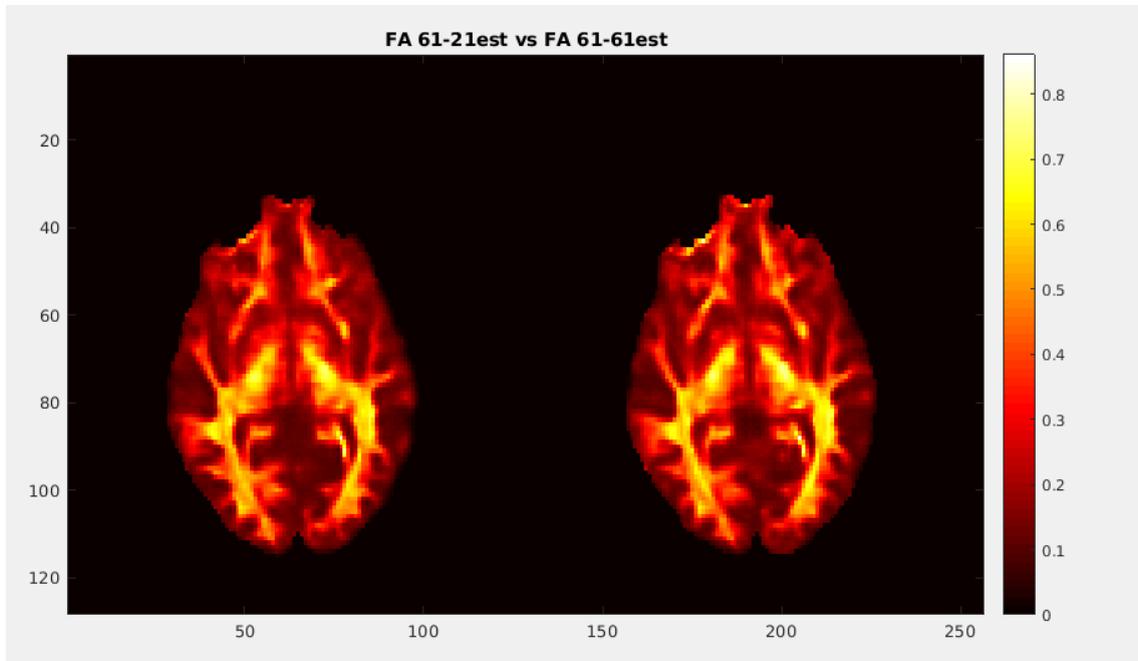


Figura 5.12: Comparativa entre las FA generadas por la red para un paciente de esquizofrenia

	Error FA originales	Error FA estimadas	Cambio (%)
SZ Slice nº 30	0.1310	0.0465	64.5 %
HC Slice nº 30	0.0805	0.0317	60.6 %

Cuadro 5.1: Comparativa entre el error en las FA originales y en las generadas en un slice en los casos de un paciente y de un control sano

	Error medio FA orig.	Error medio FA est.	Cambio (%)
SZ Todo el volumen	0.1563	0.0886	43.3 %
HC Todo el volumen	0.0659	0.0252	61.7 %

Cuadro 5.2: Comparativa entre el error medio en las FA originales y en las generadas en todo el cerebro en los casos de un paciente y de un control sano

	Mediana del error FA orig.	Mediana del error FA est.	Cambio
SZ Todo el volumen	0.1226	0.0514	58.1 %
HC Todo el volumen	0.0654	0.0266	59.3 %

Cuadro 5.3: Comparativa entre la mediana del error en las FA originales y en las generadas en todo el cerebro en los casos de un paciente y de un control sano

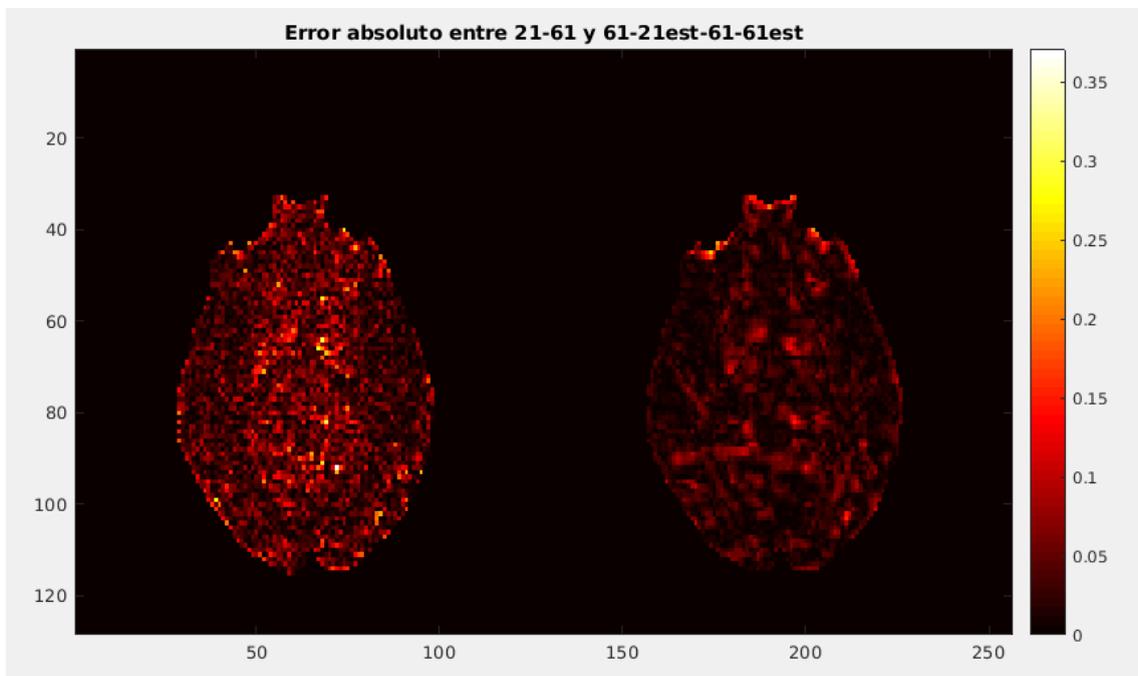


Figura 5.13: Diferencia entre las FA de 21 y de 61 para las FA originales y las generadas por la red para un paciente de esquizofrenia

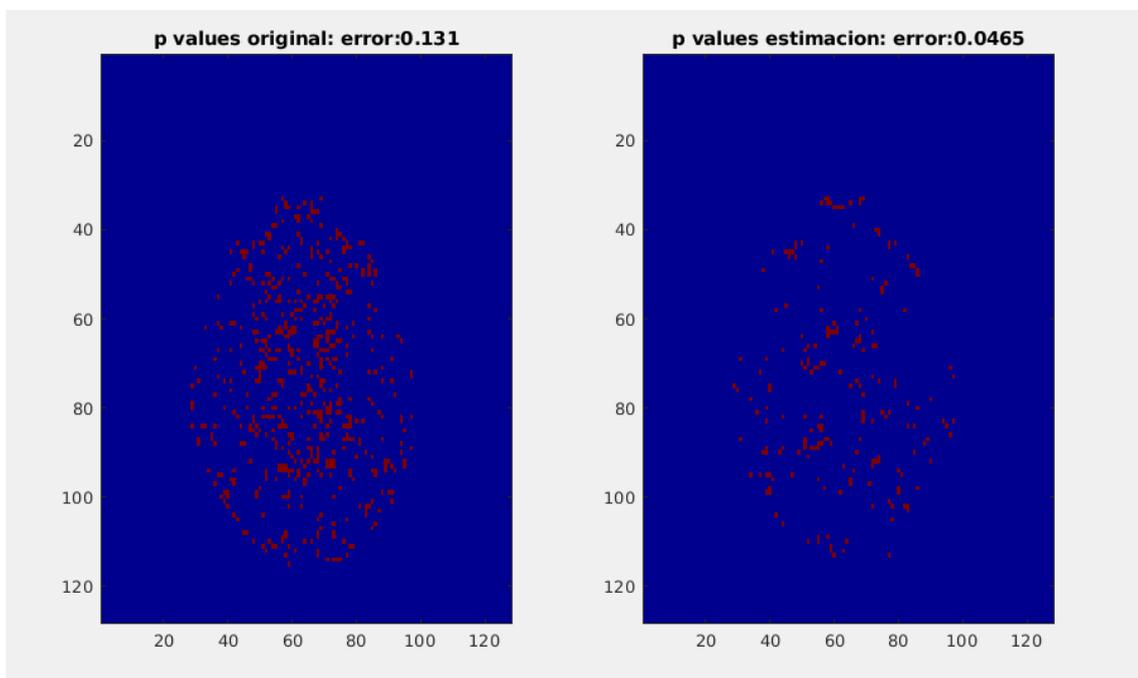


Figura 5.14: Puntos en los que un t-test detecta diferencias significativas entre las FA originales y las generadas por la red para $p < 0,01$ para un paciente de esquizofrenia

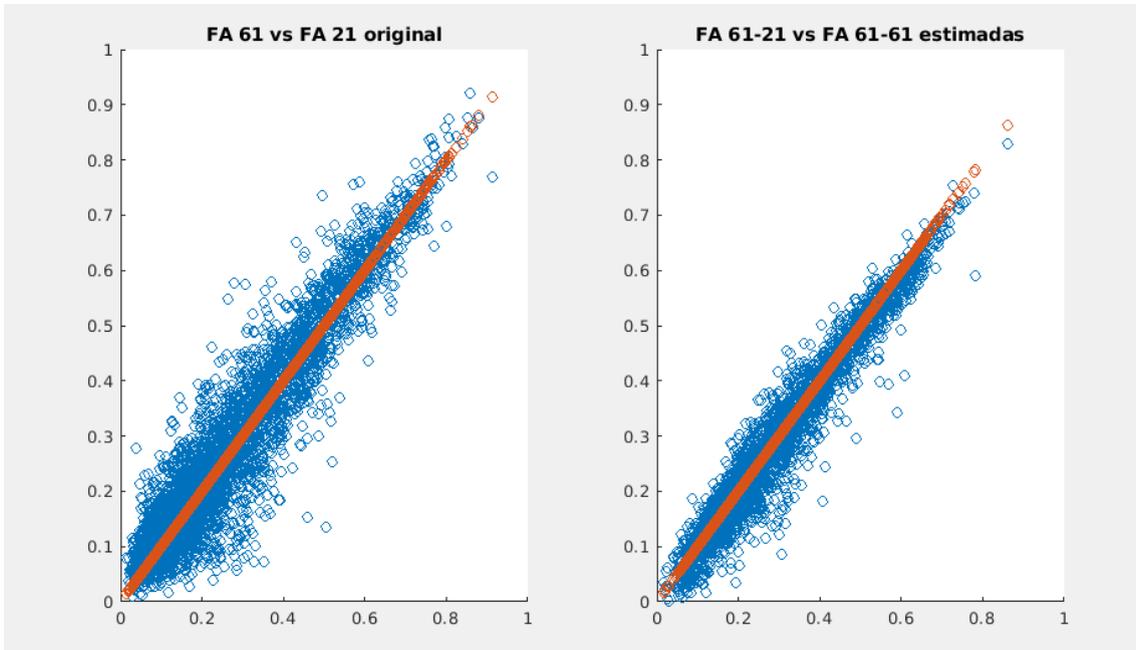


Figura 5.15: Scatter plot de la FA en todos los vóxeles de un slice para las FA originales y las generadas por la red para un paciente de esquizofrenia. (Azul: 21 gradientes, Naranja: 61 gradientes).

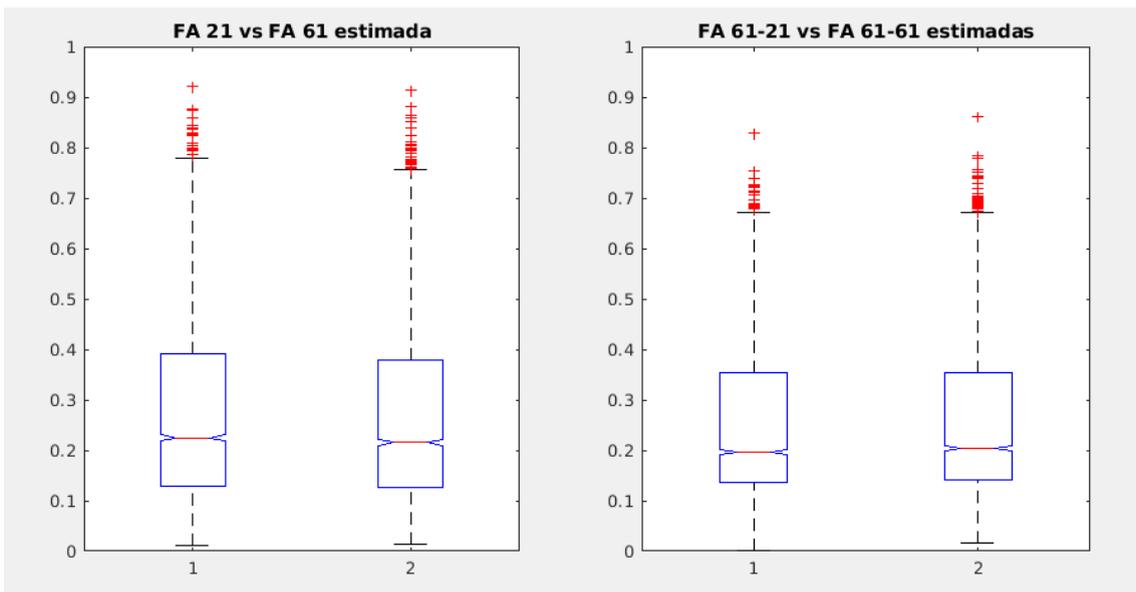


Figura 5.16: Box plot de la FA en todos los vóxeles de un slice para las FA originales y las generadas por la red para un paciente de esquizofrenia.

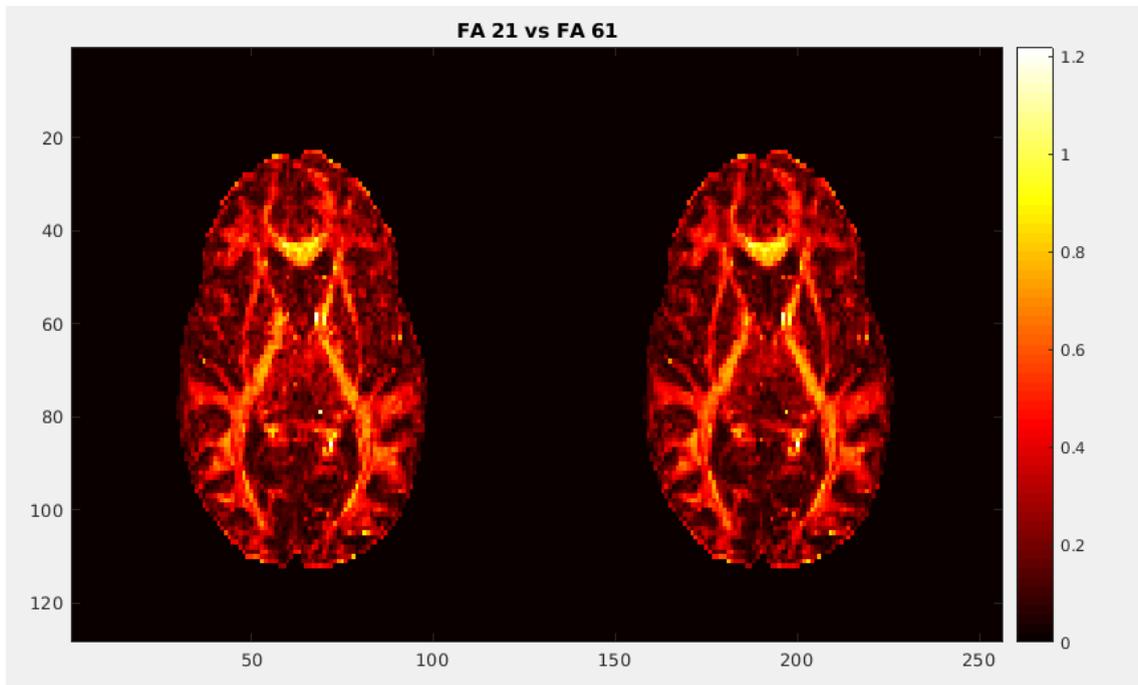


Figura 5.17: Comparativa entre las FA originales para un control sano

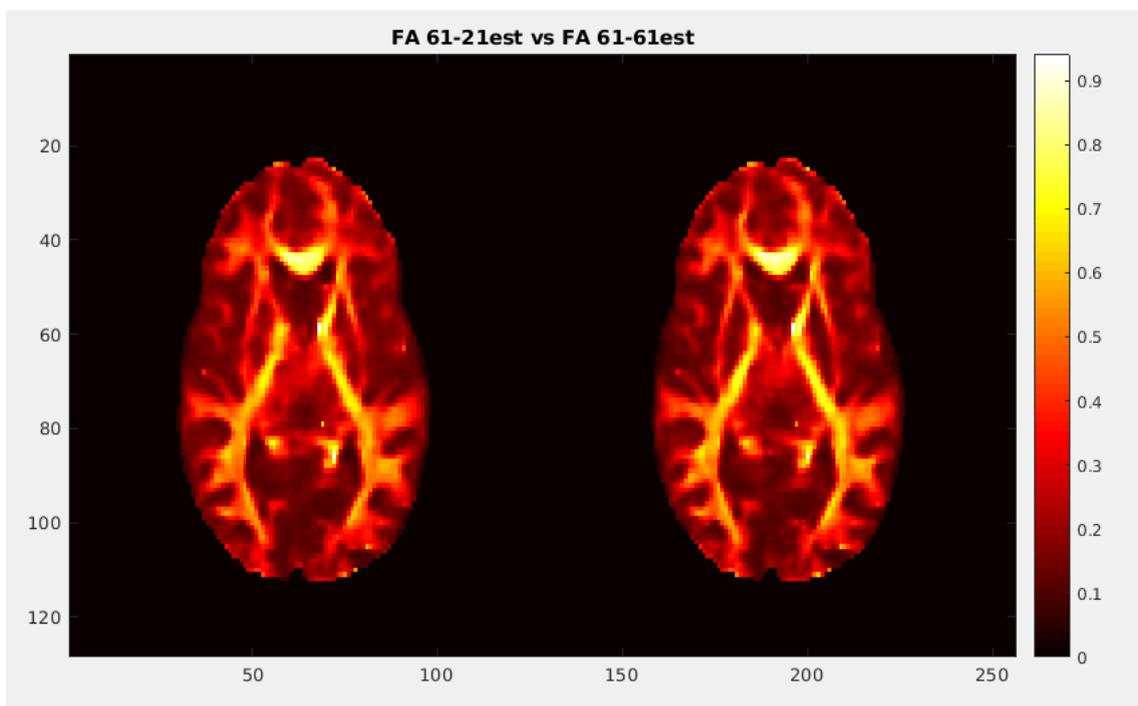


Figura 5.18: Comparativa entre las FA generadas por la red para un control sano

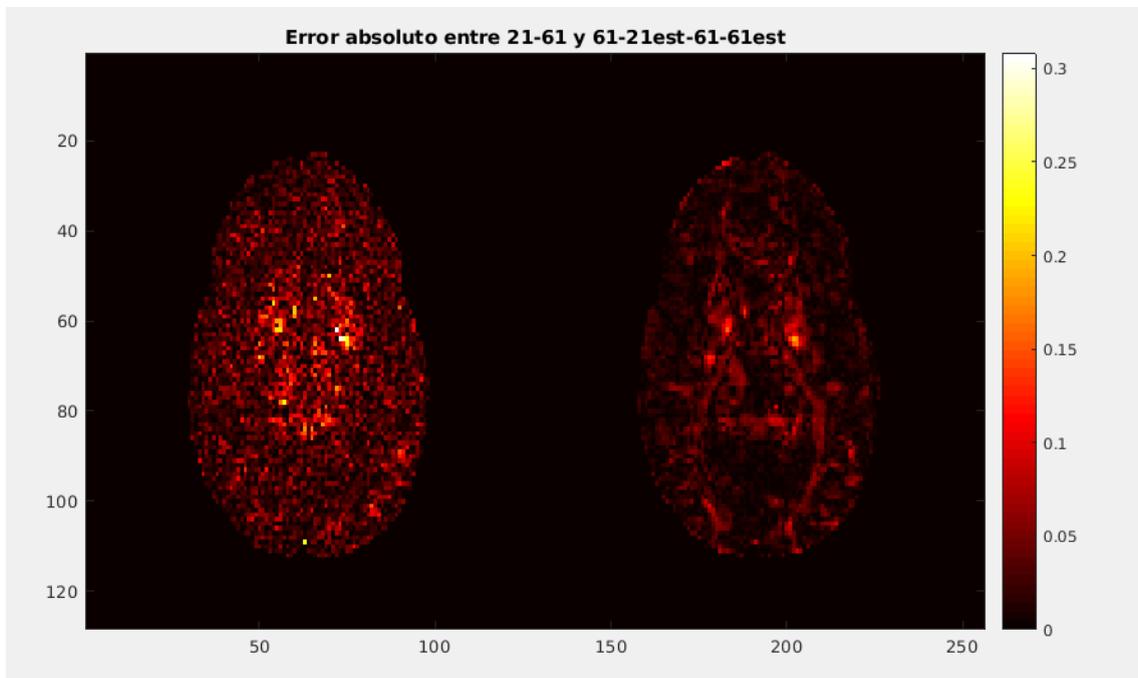


Figura 5.19: Diferencia entre las FA de 21 y de 61 para las FA originales y las generadas por la red para un control sano

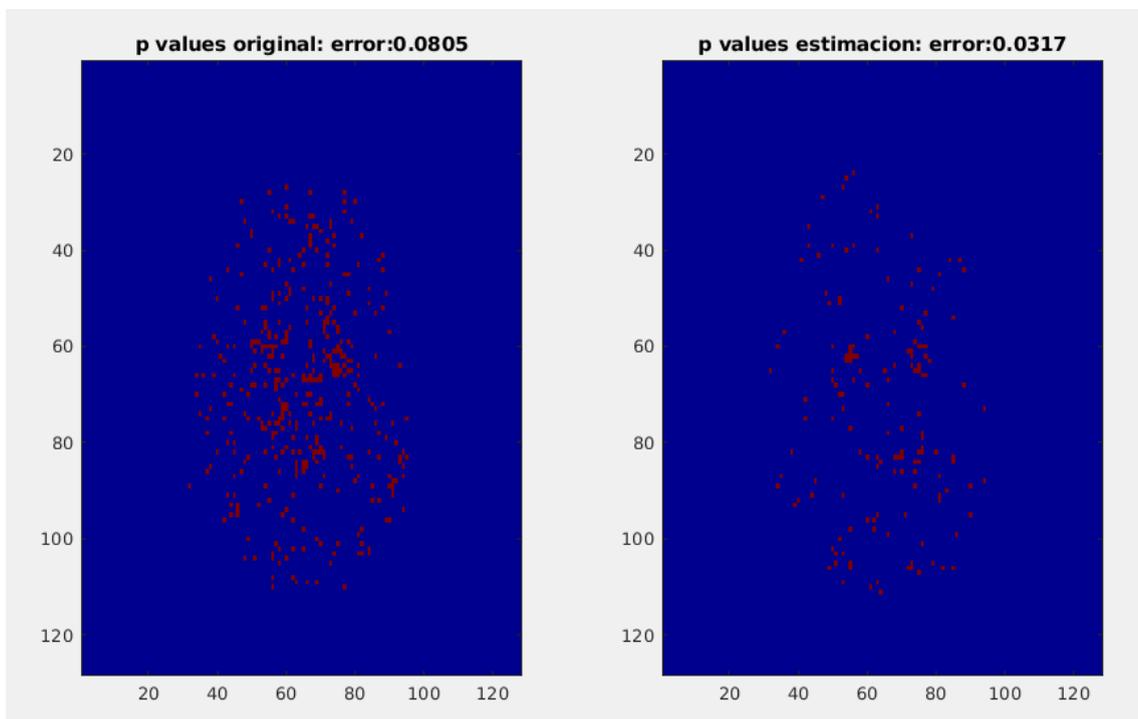


Figura 5.20: Puntos en los que un t-test detecta diferencias significativas entre las FA originales y las generadas por la red para $p < 0,01$ para un control sano

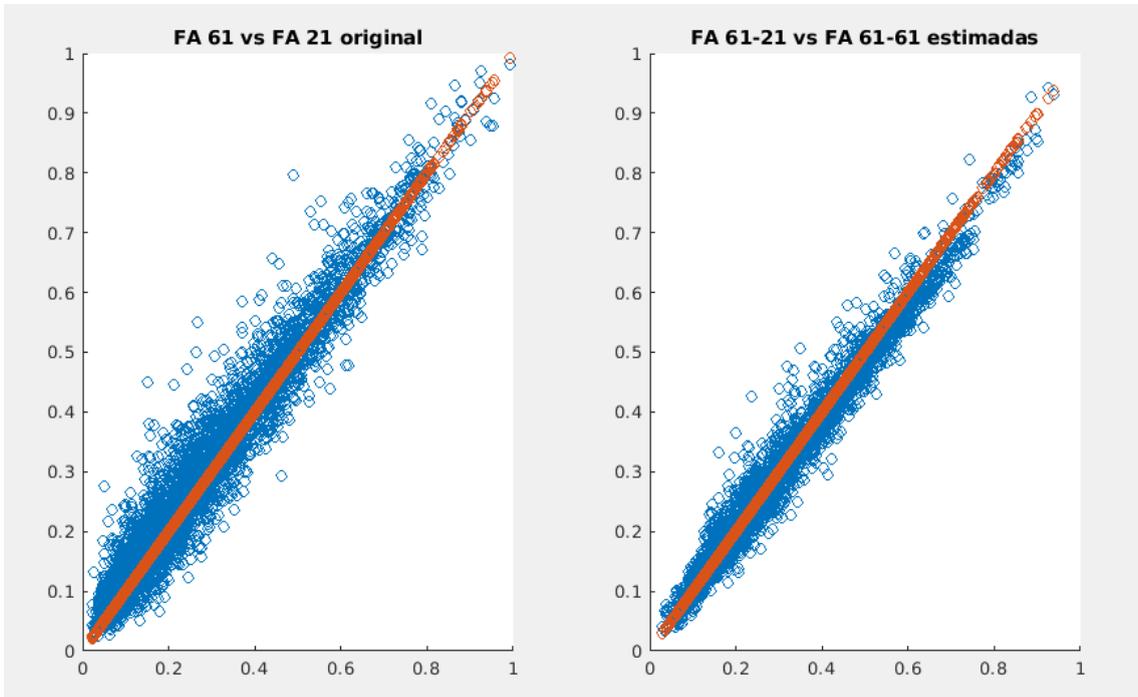


Figura 5.21: Scatter plot de la FA en todos los vóxeles de un slice para las FA originales y las generadas por la red para un control sano. (Azul: 21 gradientes, Naranja: 61 gradientes).

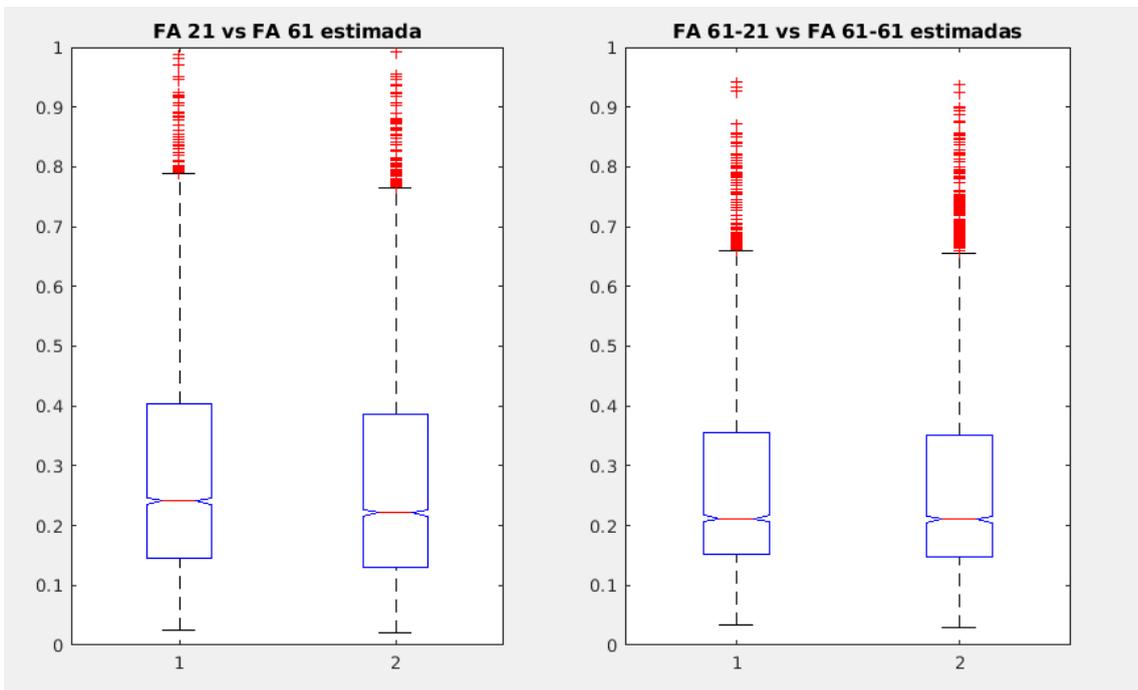


Figura 5.22: Box plot de la FA en todos los vóxeles de un slice para las FA originales y las generadas por la red para un control sano.

5.7 TBSS

Como podemos comprobar, el error entre las imágenes se reduce de manera importante. Pero para validar adecuadamente los resultados, la manera correcta que tenemos de comprobar que estamos mejorando es utilizar estas imágenes en un estudio clínico, para ver si se reducen los errores o no tienen ningún efecto. Para ello, usaremos la utilidad TBSS (Tract-Based Spatial Statistics) [28, 59] de FSL para comparar dos grupos de sujetos (controles y pacientes) en las FA originales y en las generadas por la red.

TBSS es un proceso que permite alinear la FA de varios sujetos de una manera en la que se puedan sacar conclusiones válidas de un análisis vóxel a vóxel posterior. Los pasos que se sigue para ejecutar TBSS son los siguientes:

- **Preprocesado:** Se ha de colocar las imágenes del estudio en un nuevo directorio. La nomenclatura es del estilo `CON_*z "PAT_*` para distinguir entre los grupos. El script de preprocesado retoca levemente las imágenes de la FA y pone a cero los últimos slices para eliminar probables outliers del ajuste. Tras este paso, se puede ver una preview de las imágenes que se van a tomar en el estudio en la figura 5.23, gracias a la cual se pueden encontrar problemas obvios en alguna de las imágenes.

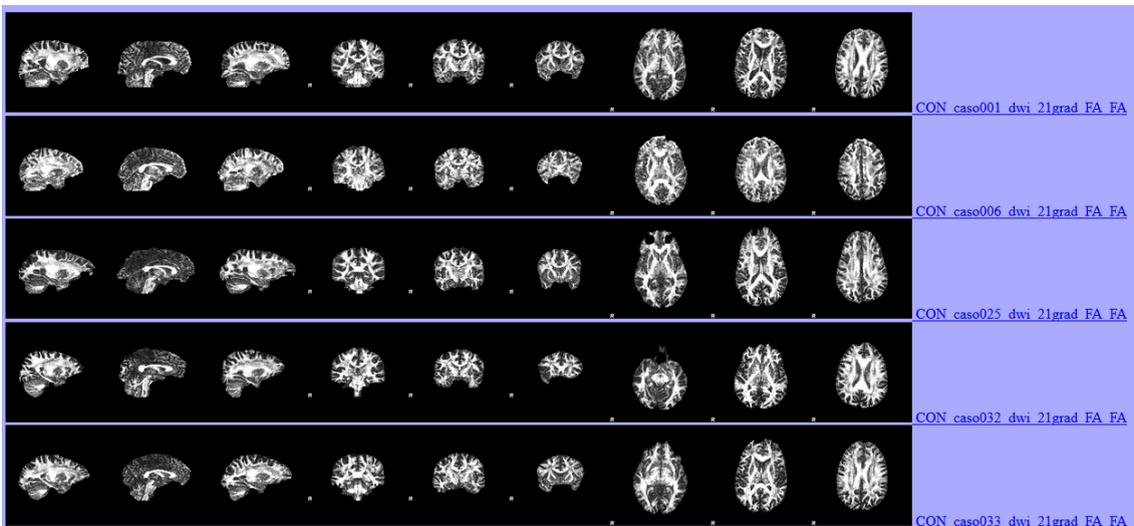


Figura 5.23: Resumen de las imágenes de un estudio con TBSS

- **Registrado:** En este paso, se lleva a cabo el registrado no lineal, alineando todas las FA en un espacio estándar de tamaño 1 x 1 x 1 mm. La imagen objetivo a usar se puede elegir de entre tres opciones:
 - Una imagen predefinida. Se recomienda la FMRIB58_FA [60], que suele dar buenos resultados y solo conlleva un registrado por sujeto. También podemos suministrar nosotros nuestra propia imagen.

- Elegida automáticamente para ser el sujeto más "típico" del estudio. Se alinea cada FA al resto de FAs, y se identifica la más representativa, usándose como imagen objetivo. Esta imagen se alinea en el espacio estándar MNI152 [60]. El resto de imágenes se llevan al espacio MNI152 combinando una transformación no lineal a la FA objetivo y la transformada de ese resultado a la del objetivo en el espacio estándar. Se recomienda esta opción para estudios específicos, por ejemplo, si todos los sujetos del estudio son niños (caso en el cual la imagen FMRIB58_FA, de un adulto, es inapropiada).

El primer método lleva aproximadamente un tiempo de procesado de 10 minutos \times N (donde N es el número de sujetos), mientras que el segundo, puesto que hay que comparar todos con todos lleva en torno a 5 minutos \times N \times N. Para este estudio, elegimos el primer método puesto que todos los sujetos son adultos.

- **Postregistrado:** Aquí se aplican las transformadas no lineales calculadas en la etapa anterior. Si se ha ejecutado el paso anterior con la opción de buscar el sujeto más típico del estudio como el objetivo, este paso se realiza aquí. El resultado del postregistrado es una versión en el espacio estándar MNI152 de la FA de cada sujeto, que se unen para formar una imagen 4D. Se calcula la FA media de todos los sujetos y con esta se alimenta al programa que calcula la esqueletización de la FA. Como alternativa, se puede utilizar directamente la FA media y el esqueleto de FMRIB58_FA, aunque no se recomienda. Un ejemplo de esqueleto de la FA se puede ver en la figura 5.24

Al final de la ejecución, se determina si 0.2 es o no es un valor adecuado para el esqueleto de la FA (la FA 0.2 o mayor típicamente ocurre en la sustancia blanca). Dicho valor lo usamos en el siguiente paso para visualizar el esqueleto. Podemos comprobar variando este umbral en la visualización si el resultado es correcto. Si lo bajamos de 0.2, se debería extender hacía los extremos, lugares donde existe mucha más variabilidad entre sujetos y donde el registrado no lineal no es capaz de realizar un alineado muy bueno.

- **Proyectado de los sujetos en el esqueleto:** Se toma el esqueleto y se aplica el umbral previamente mencionado para generar una máscara que define los grupos de vóxels que se usan. Finalmente, se realiza la proyección de los sujetos sobre el esqueleto de la FA, resultando en una imagen 4D que contiene la FA esqueletizada.
- **Cálculo de estadísticas:** La imagen resultante se utiliza como base de la comparación en el estudio estadístico, el cual nos dice que vóxels del esqueleto son significativamente diferentes entre dos grupos de sujetos. Para calcular las estadísticas se utiliza un t-test y la herramienta *randomise* [61] con la opción TFCE (*Threshold-Free Cluster Enhancement*). En el test diseñado el color rojo (TSTAT1) indica que el valor de FA de controles es mayor que el de pacientes, y el color azul (TSTAT2) que es menor.

El objetivo del estudio que se propone para verificar las predicciones es:

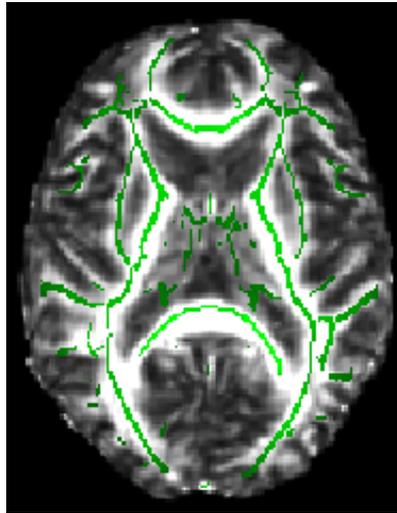


Figura 5.24: Esqueleto de la FA generado por TBSS

- Verificar que entre las FAs originales de controles y pacientes hay diferencias. Verificar que esas diferencias se mantienen en las predicciones.
- Verificar que entre controles de 21 gradientes y controles de 61 gradientes hay diferencias. Lo mismo para los pacientes.
- Que al comparar las predicciones de 21 gradientes frente a las de 61 gradientes, la diferencia disminuye o desaparece.

Los experimentos y sus resultados, se recogen en la tabla 5.4. Durante el cálculo del esqueleto se encontraron problemas con algunos sujetos, que estropeaban el cálculo del esqueleto. Las FAs que causaban los problemas se detectaron y se eliminaron del estudio, de ahí el número reducido de sujetos en el estudio. TSTAT1 (color rojo) indica que grupo 1 > grupo 2 y TSTAT2 (color azul) que grupo 1 < grupo 2. Se acompaña con una imagen del experimento en FSLView en los casos en los que se encuentren diferencias. Se añaden a continuación explicaciones detalladas sobre cada uno de los experimentos:

- **Prueba 1:** Hay diferencias significativas. Están localizadas en la zona central del cerebro (zona del tálamo).
- **Prueba 2:** Esas diferencias desaparecen en al comparar las predicciones.
- **Prueba 3:** No aparecen diferencias al comparar al 95 %. Al 85 % si que aparecen algunas diferencias, y en el mismo sentido y misma región que en la prueba 1.
- **Prueba 4:** De nuevo, sin diferencias en las predicciones.
- **Prueba 5:** Diferencias enormes, incluso por encima del 99 %, por todas partes. Esto viene a demostrar qué si hiciéramos un estudio controles vs pacientes, usando datos

Número de prueba	Grupo 1	Grupo 2	Nºsujetos grupo 1	Nºsujetos grupo 2
1	FA ORIGINAL CONTROLES 21	FA ORIGINAL CONTROLES 61	28	28
2	FA PREDICCIÓN CONTROLES 21	FA PREDICCIÓN CONTROLES 61	28	28
3	FA ORIGINAL PACIENTES 21	FA ORIGINAL PACIENTES 61	23	23
4	FA PREDICCIÓN PACIENTES 21	FA PREDICCIÓN PACIENTES 61	23	23
5	FA ORIGINAL CONTROLES 61	FA ORIGINAL PACIENTES 61	28	23
6	FA PREDICCIÓN CONTROLES 61	FA PREDICCIÓN PACIENTES 61	28	23
7	FA ORIGINAL CONTROLES 21	FA ORIGINAL PACIENTES 21	28	23
8	FA PREDICCIÓN CONTROLES 21	FA PREDICCIÓN PACIENTES 21	28	23
9	FA ORIGINAL CONTROLES 21	FA ORIGINAL PACIENTES 61	28	23
10	FA PREDICCIÓN CONTROLES 21	FA PREDICCIÓN PACIENTES 61	28	23
11	FA ORIGINAL CONTROLES 61	FA ORIGINAL PACIENTES 21	28	23
12	FA PREDICCIÓN CONTROLES 61	FA PREDICCIÓN PACIENTES 21	28	23

Cuadro 5.4: Lista de pruebas realizadas y los distintos grupos que las componen

comparables (61 gradientes), saldría lo esperado. Las diferencias son en TSTAT1, que indica controles > pacientes.

- **Prueba 6:** Enormes diferencias aquí también. Lo ideal sería que se mantuviesen las mismas diferencias que en la prueba 5. Debido al efecto de la predicción del suavizado y filtrado de los datos puede ser que encontremos más o menos diferencias. Habría que cuantificar los resultados de estas dos pruebas para concretar más los resultados.
- **Prueba 7:** Misma prueba que la 5, pero con 21 gradientes. Diferencias por todas partes, pero algo menos que en la 5. Lo esperable sería encontrar más diferencias con 61 gradientes que con 21, por lo que se contabilizará el número de vóxels que superan el umbral en cada caso.
- **Prueba 8:** De nuevo, y similarmente a la pareja de pruebas 5 y 6, muchas diferencias, y hay que compara el número de diferencias con la prueba 7 para ver si tenemos más o menos que con los datos originales.
- **Prueba 9:** Se empiezan a mezclar por igual tipos de sujeto y número de gradientes. Las diferencias se disparan a niveles absurdos, sobreestimando las diferencias que realmente hay entre los dos grupos.
- **Prueba 10:** Lo ideal sería unos resultados similares a las pruebas 5 o 7. Aparecen diferencias, pero parece haber menos. De nuevo será necesario cuantificarlos.
- **Prueba 11:** El efecto al intercambiar los grupos de 21 y 61 gradientes, debería ser el resultado inverso, en lugar de sobreestimar diferencias, se subestiman. Todas las diferencias desaparecen.
- **Prueba 12:** Vuelven a aparecer muchas diferencias. Se corrige el problema, pero no sabemos si estamos corrigiendo en exceso.

Finalmente, cuantificamos el número de vóxels con diferencias entre los grupos en las distintas pruebas, y calculamos el % de vóxels que presentan diferencias, así como el valor medio de los p-values en los vóxels del esqueleto. Estos valores se muestran en la tabla 5.5

Las conclusiones que extraemos de la tabla 5.5 son las siguientes:

- Al comparar controles y pacientes al mismo número de gradientes, se encuentran más diferencias en las predicciones que en los datos originales. Puede ser efecto del filtrado que produce el autoencoder y puede ser un efecto positivo.
- Cuando se comparan controles y pacientes, si el número de gradientes es mayor, se espera encontrar más diferencias habitualmente. Esto se cumple para las FA originales y para las de la predicción.

Número de prueba	% vóxeles diferentes a 0.95	% vóxeles diferentes a 0.85	Valor medio p values
1	2.04 %	14.69 %	0.570
2	0.00 %	0.00 %	0.100
3	0.00 %	1.82 %	0.623
4	0.00 %	0.00 %	0.038
5	52.99 %	66.86 %	0.831
6	59.73 %	74.88 %	0.858
7	48.45 %	60.18 %	0.800
8	50.23 %	70.27 %	0.819
9	76.62 %	82.95 %	0.901
10	17.42 %	57.24 %	0.752
11	0.00 %	9.04 %	0.605
12	70.89 %	80.33 %	0.893

Cuadro 5.5: Número de vóxeles en los que se encuentran diferencias en las distintas pruebas

- En los casos en los que comparamos controles contra pacientes, a distinto número de gradientes, en las predicciones se contrarrestan los efectos que ocurren con los datos originales:
 - En la prueba 9 se sobreestiman las diferencias, y en la 10 se corrige, pero en exceso
 - En la prueba 11 se subestiman las diferencias, y en la 11 se corrige, pero en exceso.

Si el funcionamiento fuese idóneo, en las predicciones deberíamos encontrar unos errores parecidos en las pruebas 9 y 11 a los de las pruebas 5 y 7.



Figura 5.25: Prueba 1: FA Original controles 21 vs FA Original controles 61.

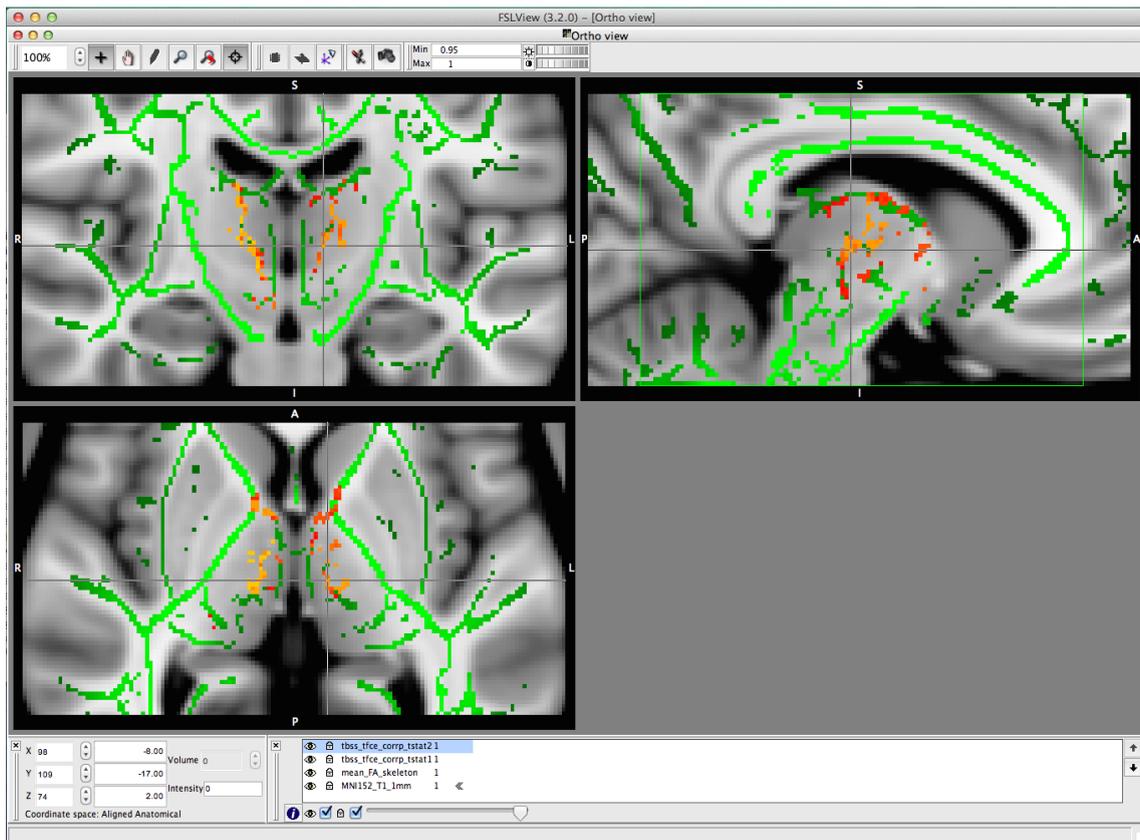


Figura 5.26: Zoom de la prueba 1.

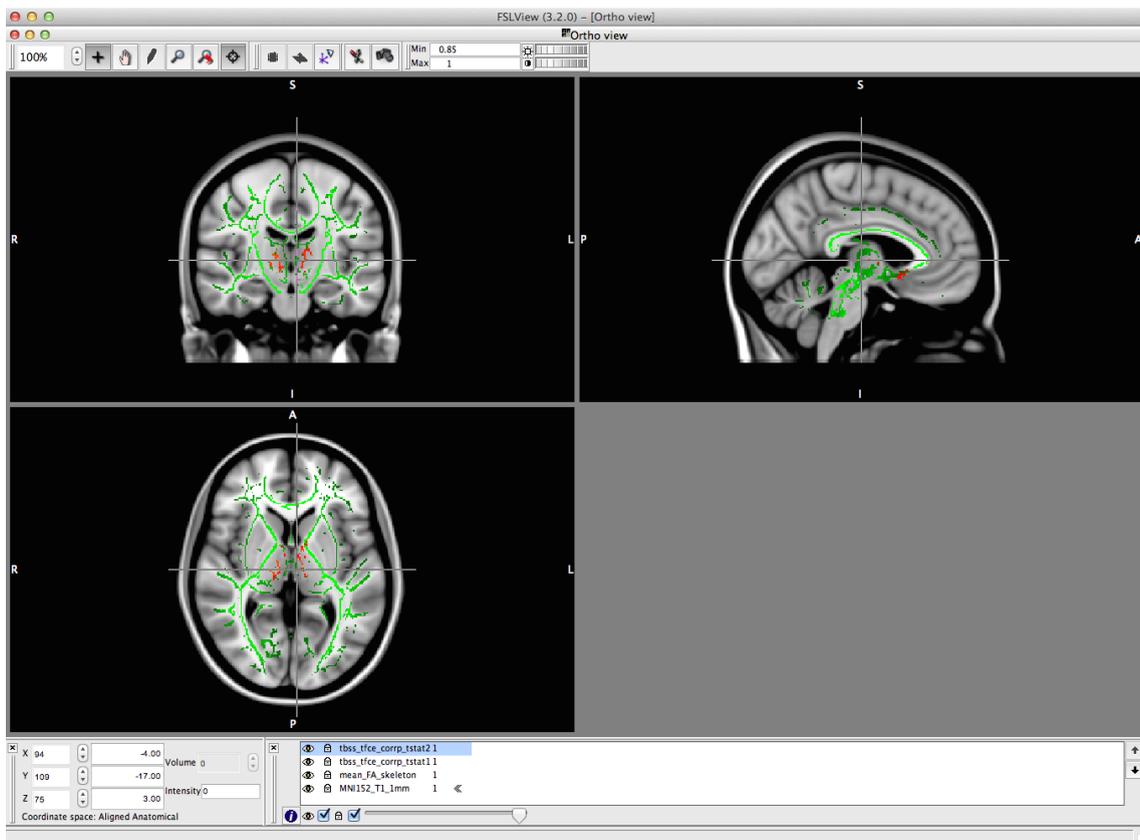


Figura 5.27: Prueba 3: FA Original pacientes 21 vs FA Original pacientes 61.

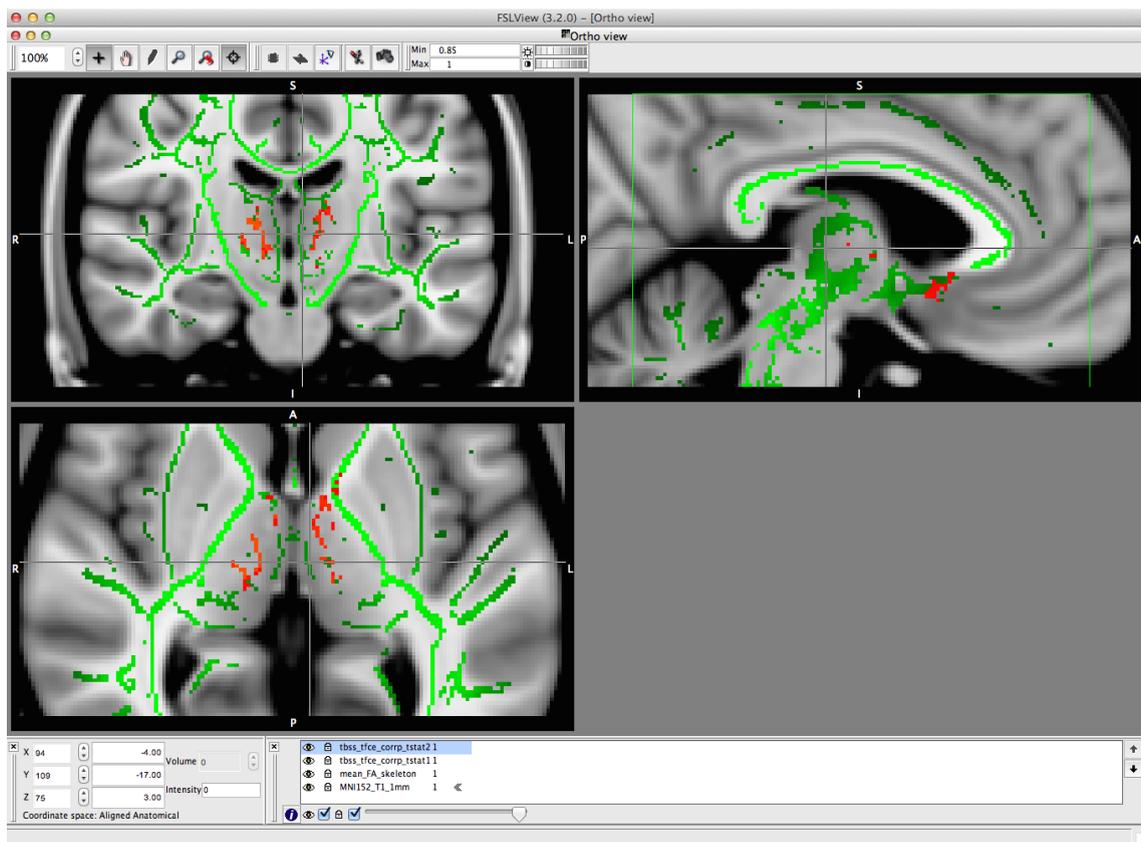


Figura 5.28: Zoom de la prueba 3.

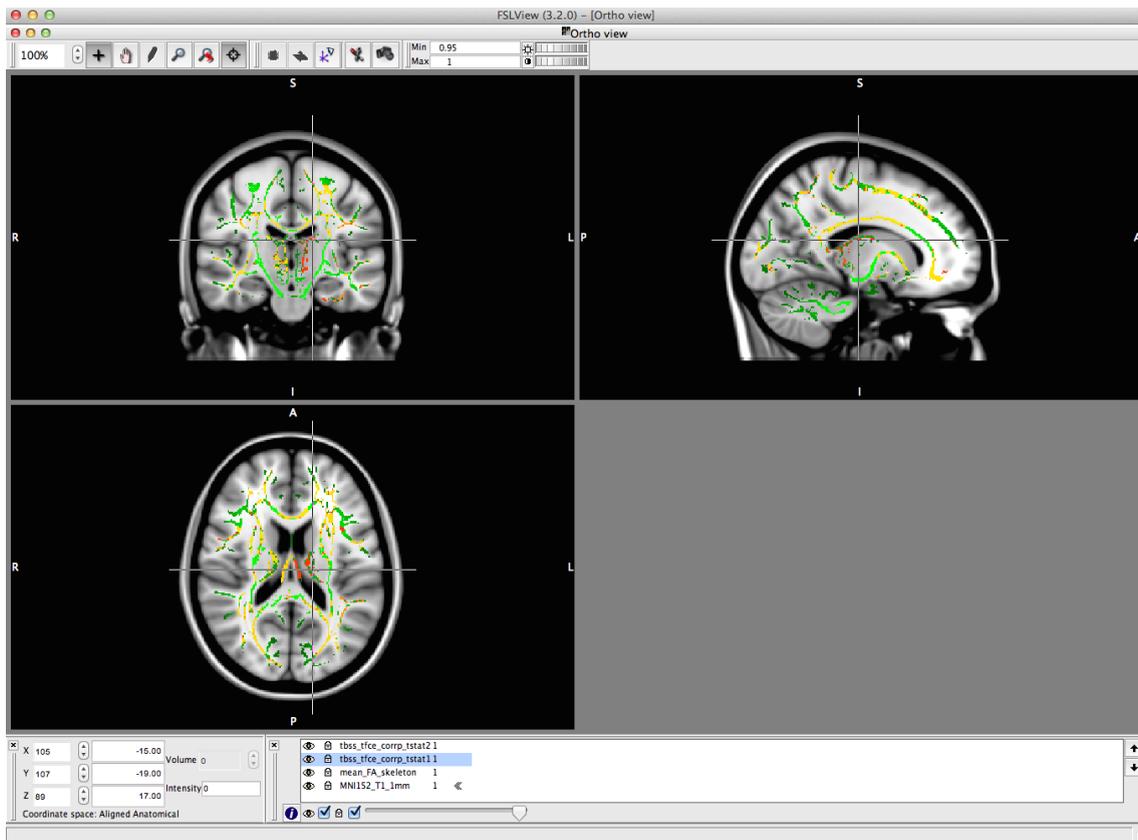


Figura 5.29: Prueba 5: FA Original controles 61 vs FA Original pacientes 61.



Figura 5.30: Prueba 6: FA Predicción pacientes 61 vs FA Predicción pacientes 61.

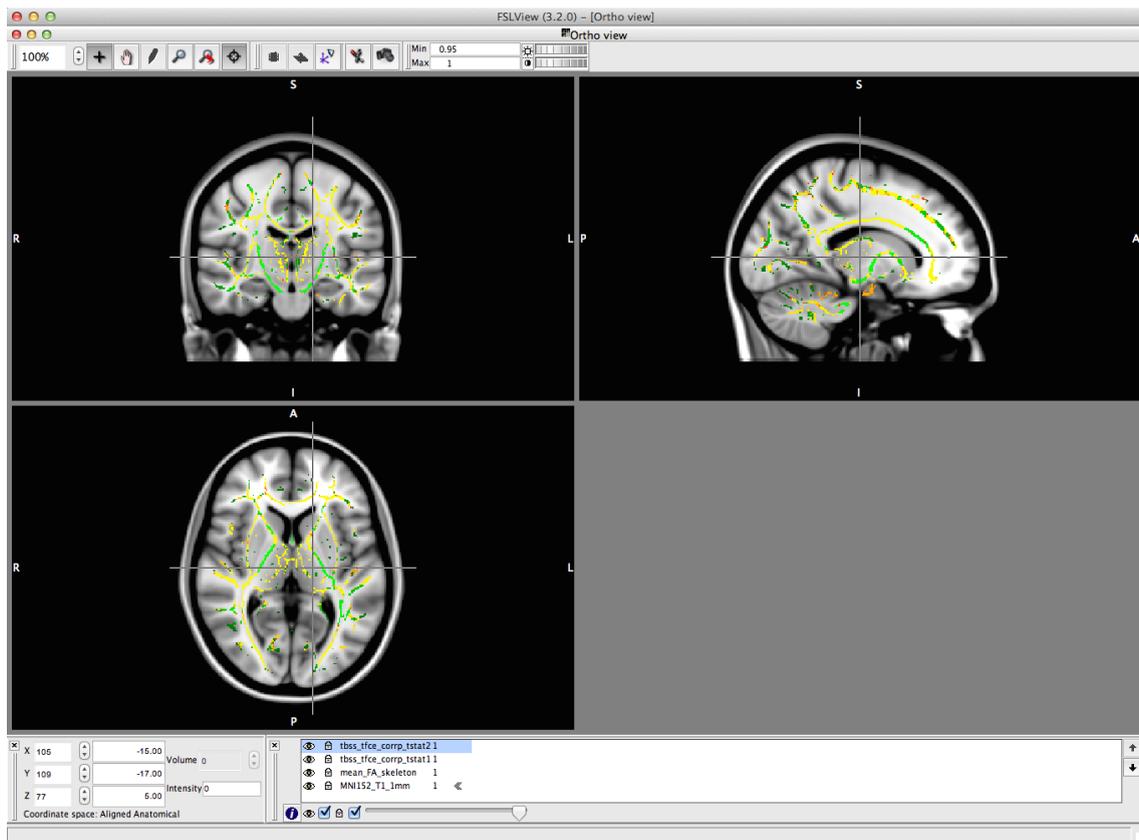


Figura 5.31: Prueba 7: FA Original controles 21 vs FA Original pacientes 21.

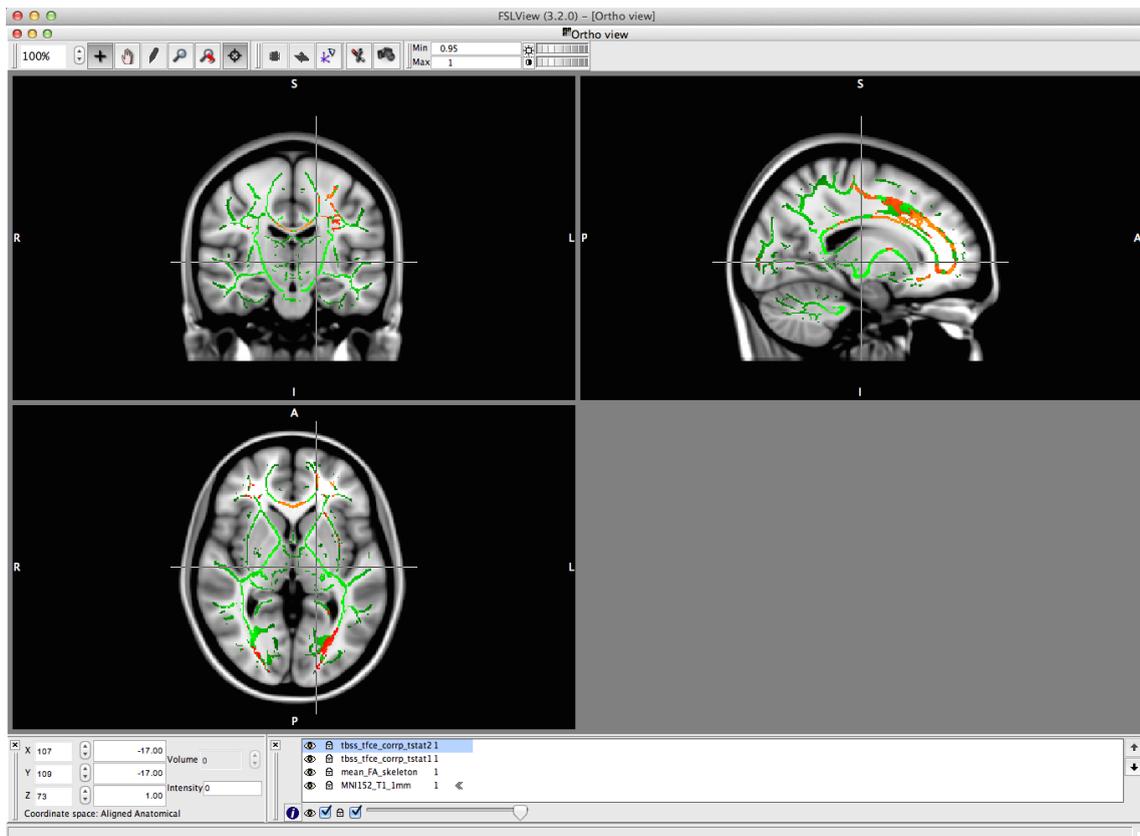


Figura 5.32: Prueba 8: FA Predicción controles 21 vs FA Predicción pacientes 21.

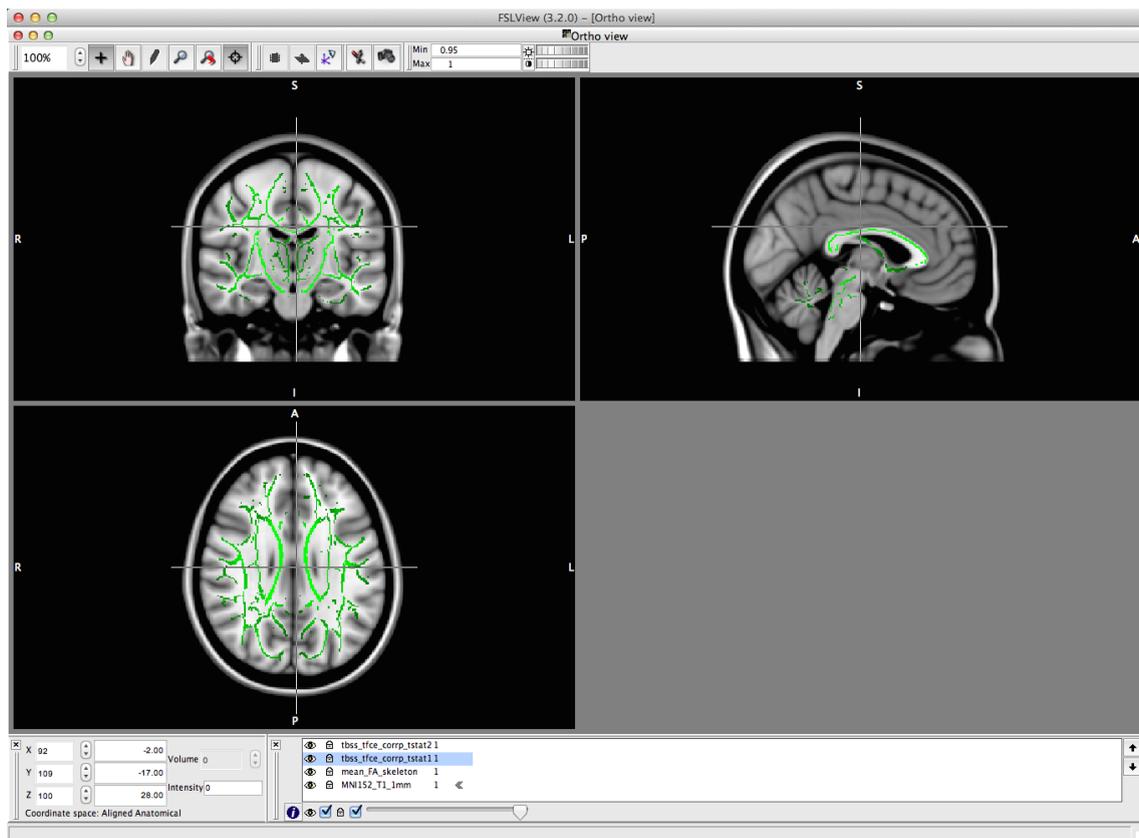


Figura 5.33: Prueba 9: FA Original controles 21 vs FA Original pacientes 61.

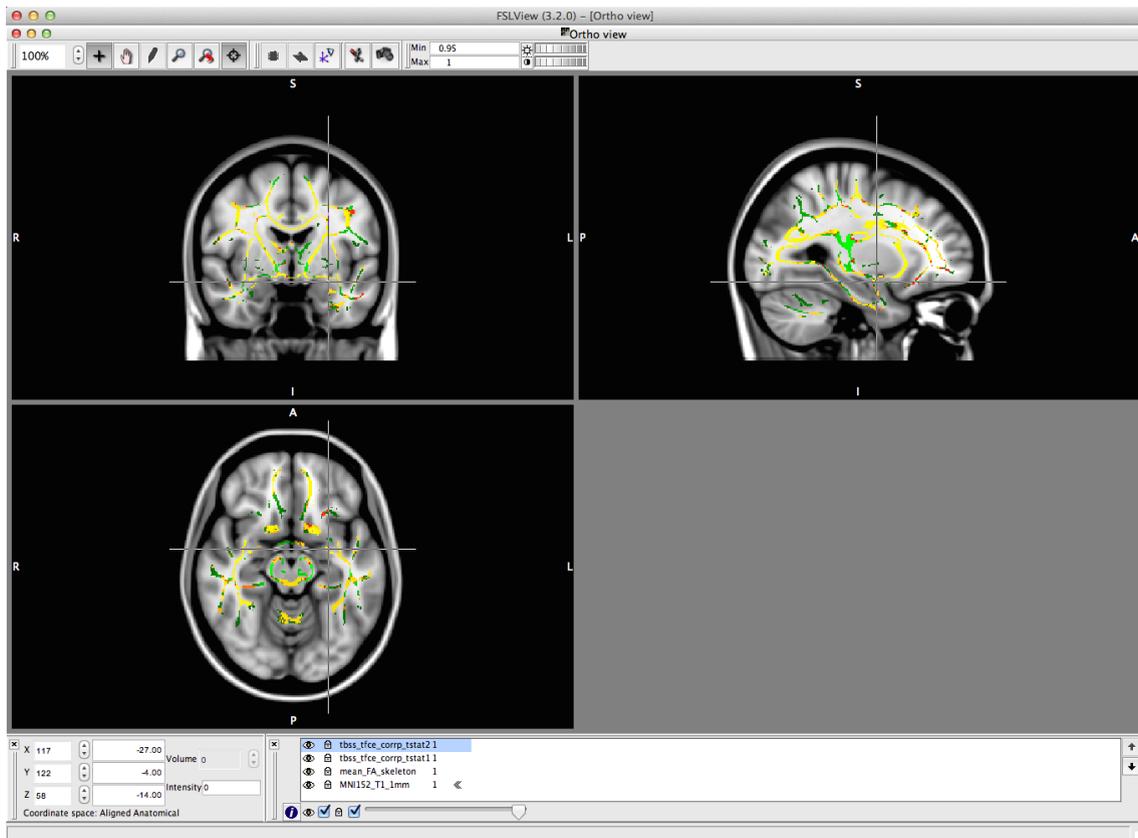


Figura 5.34: Prueba 10: FA Predicción controles 21 vs FA Predicción pacientes 61.

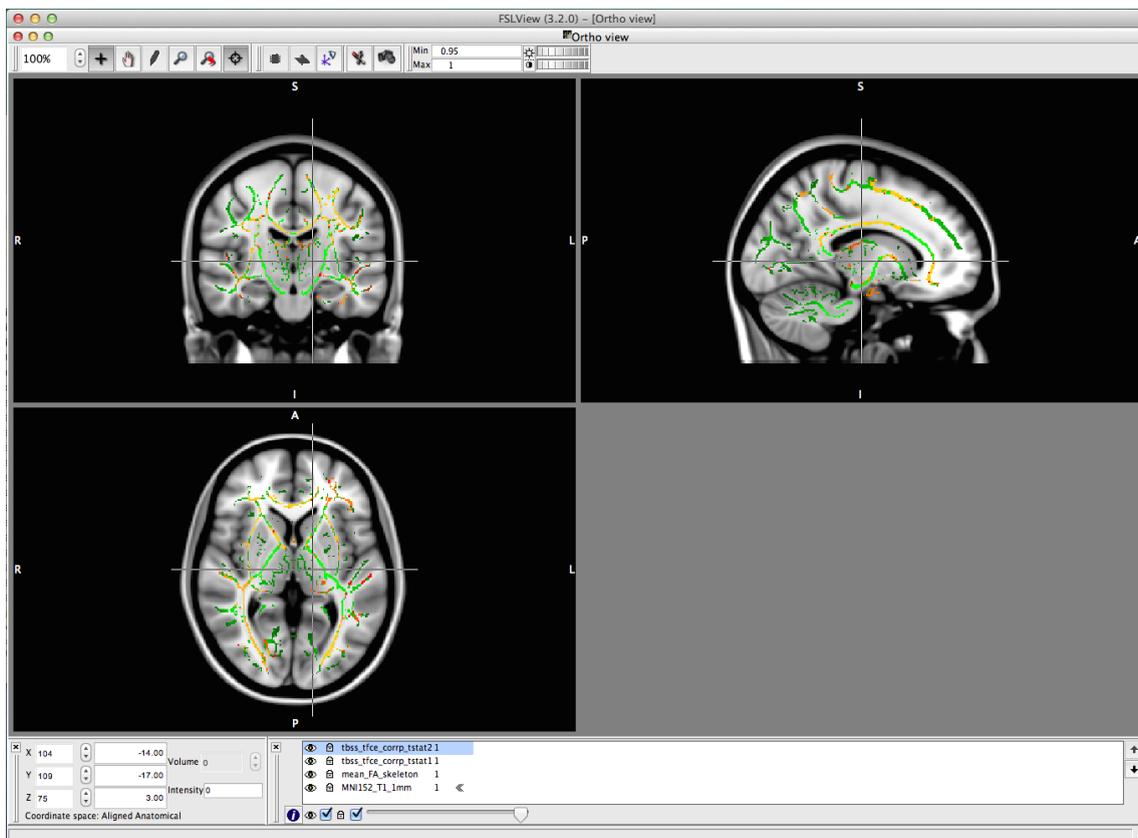


Figura 5.35: Prueba 11: FA Original controles 61 vs FA Original pacientes 21.

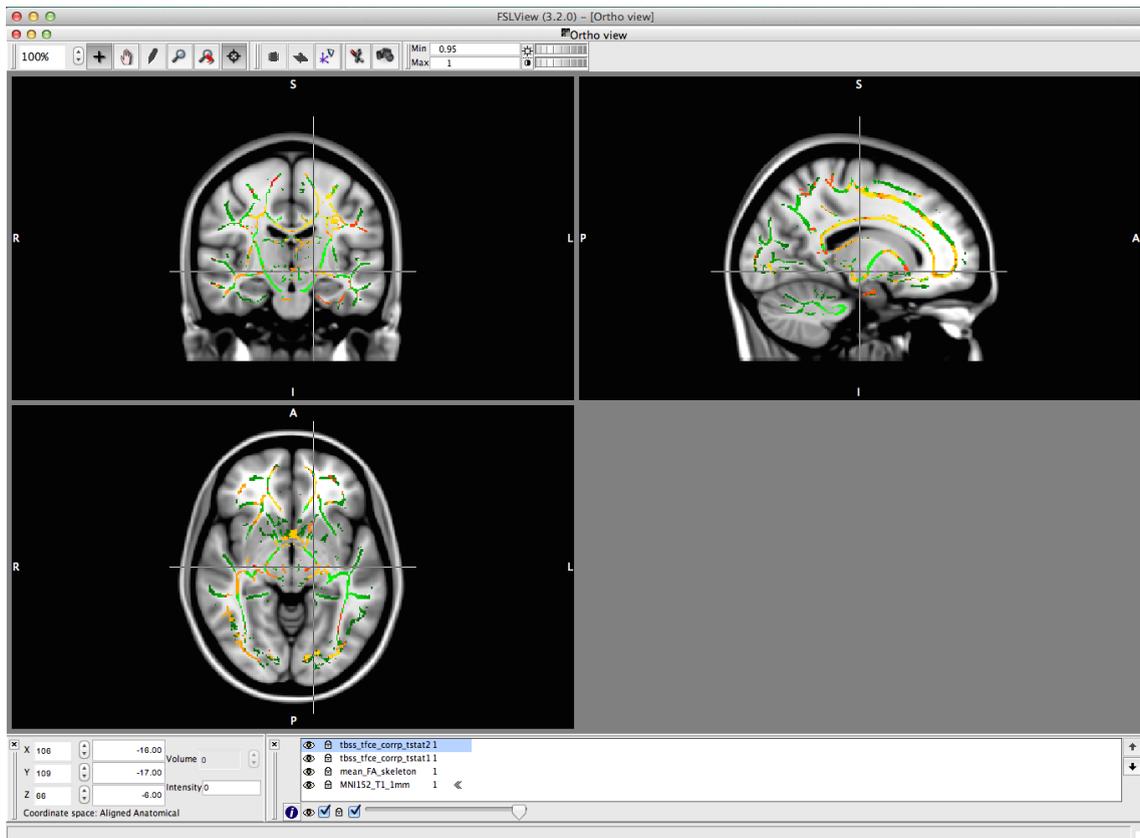


Figura 5.36: Prueba 12: FA Predicción controles 61 vs FA Predicción pacientes 21.

Capítulo 6

Conclusiones y líneas futuras de trabajo

6.1 Conclusiones

A lo largo de este proyecto se han estudiado y propuesto diversas soluciones para la armonización de medidas escalares de DWI. Durante todo el trabajo, hemos puesto el foco en el número de gradientes como parámetro variable, puesto que es uno de los que causa mayores diferencias. En el primer experimento realizado, hemos aplicado algoritmos usando como origen de datos las medidas del phantom sintético, y hemos llegado a dos conclusiones interesantes que nos han servido para saber que pasos a dar a continuación:

- A priori no es necesario partir de toda la información de los DWI y con una medida escalar como es la FA podemos tener información suficiente en si misma para realizar la armonización.
- El mapeo vóxel a vóxel resulta insuficiente, y hay que añadir información local de alguna manera para completar los datos.

La unión de estos dos puntos nos lleva a plantear el uso de imágenes (cada slice del cerebro) como entrada y como salida del sistema: tenemos toda la información local de una dimensión, y como usamos la FA y sólo hay un canal de información la estructura de la red, la complejidad es manejable y se puede entrenar y aplicar el modelo en la práctica.

Al tener imágenes como entrada, proponemos cambiar el modelo y usar redes convolucionales en lugar de los algoritmos típicos de Aprendizaje Automático. Se propone una arquitectura para el mapeo que consiste en una red FCN basada en la U-Net. Este tipo de red se desarrolló para tareas de segmentación, así que realizamos modificaciones en las capas de salida para obtener imágenes completas a la salida y generar predicciones.

Se han generado las redes y se han entrenado con datos reales tomados de controles sanos y de pacientes de esquizofrenia y de cefaleas. Las predicciones generadas por esta red realizan un mapeo no lineal que filtra el ruido de las imágenes y las suaviza. Se ha realizado un T-test para comparar las imágenes originales y las predicciones de la red y se han obtenido mejoras en el error. Con estos resultados, planteamos un estudio clínico usando TBSS para probar la viabilidad de las predicciones. Se comparan controles sanos con pacientes para las imágenes originales y las predicciones generadas por la red y se extraen los siguientes resultados:

- En casos de la misma naturaleza (pacientes o controles), donde había diferencias a distinto número de gradientes en las imágenes originales, estas desaparecen en las predicciones.
- Al comparar casos de distinta naturaleza, se mantienen de forma aproximada las diferencias, e incluso aumentan un poco, lo cuál puede ser positivo ya que hace más fácil diferenciar un grupo de otro.
- En los casos en los que se sobreestiman o subestiman las diferencias, se corrigen, aunque parece que en exceso.

En definitiva, en el presente trabajo se ha desarrollado con éxito un modelo entrenado en GPU, que permite realizar un mapeo sobre FAs de distinto número de gradientes a un espacio común de manera que resulten comparables. Dicho modelo se puede lanzar en cualquier sistema que tenga instalado Python (y las librerías necesarias) y genera las predicciones sin consumir excesivo tiempo ni recursos.

6.2 Líneas futuras

A partir de este trabajo, se proponen de manera natural una serie de líneas desde las que seguir trabajando en la materia:

- Pensamos que los desarrollos realizados se podrían aplicar con éxito a otro tipo de parámetros como el *b-value* o el tamaño de vóxel, quizá con ciertos retoques en la red para cada caso particular.
- Si se consigue solventar las diferencias entre distintos parámetros, el siguiente paso sería mezclarlos todos y lograr un sistema que mapee una imagen de cualquier naturaleza, incluso por modelos de máquinas de resonancia.
- También se propone el desarrollo de redes similares para otras medidas, como la MD, los autovalores o incluso los DWI. Estas dos últimas constituirían unas redes

más extensas para las que se necesitaría una mayor potencia de cálculo, dada la multidimensionalidad de los datos, pero puede que aporten información adicional que se está perdiendo en medidas como la FA.

- Asimismo, la forma de encarar el problema y el desarrollo de las redes FCN se puede aplicar con una metodología similar en otras áreas del procesado de imagen médica tales como la segmentación o el registrado.
- Ya que estamos usando autoencoders, se puede estudiar la información que se tiene en las capas más profundas de la red, porque las representaciones de los datos a ese nivel que pueden ser útiles para extraer información o directamente podrían llegar a constituir nuevos tipos de medidas aplicables en estudios clínicos.
- La red propuesta se puede seguir retocando y mejorando. A lo largo del desarrollo del trabajo ha sufrido cambios que la han ido adecuando al problema. Hay multitud de parámetros que se pueden ajustar en busca de un mejor rendimiento.
- Para los resultados obtenidos en TBSS, sería idóneo trabajar en conjunto con un médico experto en la materia que pudiera aportar mejores conclusiones sobre los estudios clínicos y que conozca en profundidad los datos y pueda valorar mejor si las predicciones son correctas o dónde se pueden mejorar.

Bibliografía

- [1] PJ Basser and C Pierpaoli. Microstructural features measured using diffusion tensor imaging. *J Magn Reson B*, 111(3):209–219, 1996.
- [2] M Kubicki, H Park, Carl-Fredrik Westin, et al. DTI and MTR abnormalities in schizophrenia: analysis of white matter integrity. *Neuroimage*, 26(4):1109–1118, 2005.
- [3] PC Sundgren, Q Dong, D Gomez-Hassan, SK Mukherji, P Maly, and R Welsh. Diffusion tensor imaging of the brain: review of clinical applications. *Neuroradiology*, 46(5):339–350, 2004.
- [4] Marco Rovaris and Massimo Filippi. Diffusion tensor MRI in multiple sclerosis. *J Neuroimaging*, 17(s1):27S–30S, 2007.
- [5] Andrew J Buckler, Linda Bresolin, N Reed Dunnick, and Daniel C Sullivan. A collaborative enterprise for multi-stakeholder participation in the advancement of quantitative imaging. *Radiology*, 258(3):906–914, 2011.
- [6] John J Smith, A Gregory Sorensen, and James H Thrall. Biomarkers in imaging: Realizing radiology’s future. *Radiology*, 227(3):633–638, 2003.
- [7] Antonio Tristán-Vega, Santiago Aja-Fernández, and Carl-Fredrik Westin. Least squares for diffusion tensor estimation revisited: Propagation of uncertainty with Rician and non-Rician signals. *Neuroimage*, 59(4):4032–4043, 2012.
- [8] Santiago Aja-Fernández, Marc Niethammer, Marek Kubicki, Martha E Shenton, and Carl-Fredrik Westin. Restoration of DWI data using a Rician LMMSE estimator. *IEEE Trans Med Imaging*, 27(10):1389–1403, 2008.
- [9] Fatima Tensaouti, Ihssan Lahlou, Perrine Clarisse, Jean Albert Lotterie, and Isabelle Berry. Quantitative and reproducibility study of four tractography algorithms used in clinical routine. *J Magn Reson Imaging*, 34(1):165–172, 2011.
- [10] MJ Vaessen, PAM Hofman, HN Tijssen, AP Aldenkamp, Jacobus FA Jansen, and Walter H Backes. The effect and reproducibility of different clinical DTI gradient sets on small world brain connectivity measures. *Neuroimage*, 51(3):1106–1116, 2010.

- [11] Laura E Danielian, Nobue K Iwata, David M Thomasson, and Mary Kay Floeter. Reliability of fiber tracking measurements in diffusion tensor imaging for longitudinal study. *Neuroimage*, 49(2):1572–1580, 2010.
- [12] Setsu Wakana, Arvind Caprihan, Martina M Panzenboeck, et al. Reproducibility of quantitative tractography methods applied to cerebral white matter. *Neuroimage*, 36(3):630–644, 2007.
- [13] Haz-Edine Assemlal, David Tschumperlé, Luc Brun, and Kaleem Siddiqi. Recent advances in diffusion mri modeling: Angular and radial reconstruction. *Medical image analysis*, 15(4):369–396, 2011.
- [14] David S Tuch. Q-ball imaging. *Magnetic resonance in medicine*, 52(6):1358–1372, 2004.
- [15] Antonio Tristan-Vega, Carl-Fredrik Westin, and Santiago Aja-Fernandez. A new methodology for the estimation of fiber populations in the white matter of the brain with the Funk-Radon transform. *NeuroImage*, 49(2):1301–1315, 2010.
- [16] Antonio Tristán-Vega, Carl-Fredrik Westin, and Santiago Aja-Fernández. Estimation of fiber orientation probability density functions in high angular resolution diffusion imaging. *NeuroImage*, 47(2):638–650, 2009.
- [17] JA. Farrell, BA. Landman, CK. Jones, SA. Smith, JL. Prince, PCM. van Zijl, and S. Mori. Effects of SNR on the accuracy and reproducibility of DTI-derived fractional anisotropy, mean diffusivity and principal eigenvector measurements at 1.5T. *J Magn Reson Imag*, 26:756–767, 2007.
- [18] Gonzalo Barrio-Arranz, Rodrigo de Luis-García, Antonio Tristán-Vega, Marcos Martín-Fernández, and Santiago Aja-Fernández. Impact of MR acquisition parameters on DTI scalar indexes: a tractography based approach. *PloS one*, 10(10):e0137905, 2015.
- [19] ES. Hui, MM. Cheung, KC. Chan, and EX. Wu. B-value dependence of DTI quantitation and sensitivity in detecting neural tissue changes. *Neuroimage*, 49:2366–2374, 2010.
- [20] C. Metzler-Baddeley, MJ. O’Sullivan, S. Bells, O. Pasternak, and DK. Jones. How and how not to correct for CSF-contamination in diffusion MRI. *Neuroimage*, 59:1394–1403, 2012.
- [21] DK. Jones. The effect of gradient sampling schemes on measures derived from diffusion tensor MRI: a Monte Carlo study. *Neuroimage*, 51:807–815, 2004.
- [22] JY. Wang, H. Abdi, K. Bakhadirov, R. Diaz-Arrastia, and MD. Devous. A comprehensive reliability assessment of quantitative diffusion tensor tractography. *Neuroimage*, 60(2):1127–1138, 2012.

- [23] BA. Landman, JA. Farrell, and CK. Jones. Effects of diffusion weighting schemes on the reproducibility of DTI-derived fractional anisotropy, mean diffusivity, and principal eigenvector measurements at 1.5T. *Neuroimage*, 36:1123–1138, 2007.
- [24] X. Santarelli, G. Garbin, M. Ukmar, and R. Longo. Dependence of the fractional anisotropy in cervical spine from the number of diffusion gradients, repeated acquisition and voxel size. *J Magn Reson Imag*, 28:70–76, 2010.
- [25] Derek K Jones, Thomas R Knösche, and Robert Turner. White matter integrity, fiber count, and other fallacies: the do's and don'ts of diffusion mri. *Neuroimage*, 73:239–254, 2013.
- [26] Christos Davatzikos. Why voxel-based morphometric analysis should be used with great caution when characterizing group differences. *Neuroimage*, 23(1):17–20, 2004.
- [27] Eric D Melonakos, Martha E Shenton, Yogesh Rathi, Douglas P Terry, Sylvain Bouix, and Marek Kubicki. Voxel-based morphometry (vbm) studies in schizophrenia - can white matter changes be reliably detected with vbm? *Psychiatry Research: Neuroimaging*, 193(2):65–70, 2011.
- [28] S.M. Smith, M. Jenkinson, H. Johansen-Berg, et al. Tract-based spatial statistics: Voxelwise analysis of multi-subject diffusion data. *Neuroimage*, 31:1487–1505, 2006.
- [29] Rodrigo de Luis-García, Carl-Fredrik Westin, and Carlos Alberola-López. Geometrical constraints for robust tractography selection. *Neuroimage*, 81:26–48, 2013.
- [30] Antonio Tristán-Vega, Santiago Aja-Fernández, and Carl-Fredrik Westin. Least squares for diffusion tensor estimation revisited: Propagation of uncertainty with Rician and non-Rician signals. *Neuroimage*, 59(4):4032 – 4043, 2012.
- [31] S. Bisdas, D. Bohning, N. Besenski, JS. Nicholas, and Z. Rumboldt. Reproducibility, interrater agreement, and age-related changes of fractional anisotropy measures at 3T in healthy subjects: effect of the applied b-value. *Am J Neuroradiol*, 29(6):1128–1133, 2008.
- [32] S. Fujiwara, M. Sasaki, Y. Kanbara, T. Inoue, R. Hirooka, and A. Ogawa. Feasibility of 1.6-mm isotropic voxel diffusion tensor tractography in depicting limbic fibers. *Neuroradiology*, 50:131–136, 2008.
- [33] Christian Vollmar, Jonathan O’Muircheartaigh, Gareth J Barker, Mark R Symms, Pamela Thompson, Veena Kumari, John S Duncan, Mark P Richardson, and Matthias J Koepp. Identical, but not the same: intra-site and inter-site reproducibility of fractional anisotropy measures on two 3.0 T scanners. *Neuroimage*, 51(4):1384–1394, 2010.

- [34] Tong Zhu, Rui Hu, Xing Qiu, Michael Taylor, Yuen Tso, Constantin Yiannoutsos, Bradford Navia, Susumu Mori, Sven Ekholm, Giovanni Schifitto, et al. Quantification of accuracy and precision of multi-center DTI measurements: a diffusion phantom and human brain study. *Neuroimage*, 56(3):1398–1411, 2011.
- [35] Tong Zhu, Xiaoxu Liu, Patrick R Connelly, and Jianhui Zhong. An optimized wild bootstrap method for evaluation of measurement uncertainties of DTI-derived parameters in human brain. *Neuroimage*, 40(3):1144–1156, 2008.
- [36] Vijay K Venkatraman, Christopher E Gonzalez, Bennett Landman, Joshua Goh, David A Reiter, Yang An, and Susan M Resnick. Region of interest correction factors improve reliability of diffusion imaging measures within and across scanners and field strengths. *Neuroimage*, 119:406–416, 2015.
- [37] Kilian M Pohl, Edith V Sullivan, Torsten Rohlfing, et al. Harmonizing DTI measurements across scanners to examine the development of white matter microstructure in 803 adolescents of the NCANDA study. *Neuroimage*, 130:194–213, 2016.
- [38] H Mirzaalian, L Ning, P Savadjiev, et al. Inter-site and inter-scanner diffusion MRI data harmonization. *Neuroimage*, 135:311–323, 2016.
- [39] Hengameh Mirzaalian, Lipeng Ning, Peter Savadjiev, et al. Multi-site harmonization of diffusion MRI data in a registration framework. *Brain Imaging Behav*, 2017.
- [40] Robert Brown. Xxvii. a brief account of microscopical observations made in the months of june, july and august 1827, on the particles contained in the pollen of plants; and on the general existence of active molecules in organic and inorganic bodies. *The Philosophical Magazine*, 4(21):161–173, 1828.
- [41] Denis Le Bihan, Jean-François Mangin, Cyril Poupon, Chris A Clark, Sabina Papapata, Nicolas Molko, and Hughes Chabriat. Diffusion tensor imaging: concepts and applications. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 13(4):534–546, 2001.
- [42] E.-O. Stejskal and J.-E. Tanner. Spin diffusion measurements: Spin echoes in the presence of a time-dependent field gradient. *J Chem Phys*, 42:288–292, 1965.
- [43] Denis Le Bihan, Eric Breton, Denis Lallemand, Philippe Grenier, Emmanuel Cabanis, and Maurice Laval-Jeantet. Mr imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders. *Radiology*, 161(2):401–407, 1986.
- [44] D Bihan Le. Molecular diffusion nuclear magnetic resonance imaging. *Magnetic resonance quarterly*, 7(1):1–30, 1991.
- [45] Peter J Basser, James Mattiello, and Denis LeBihan. Mr diffusion tensor spectroscopy and imaging. *Biophysical journal*, 66(1):259–267, 1994.

- [46] Peter J Basser, James Mattiello, and Denis LeBihan. Estimation of the effective self-diffusion tensor from the nmr spin echo. *Journal of Magnetic Resonance, Series B*, 103(3):247–254, 1994.
- [47] C-F Westin, Stephan E Maier, Hatsuho Mamata, Arya Nabavi, Ferenc A Jolesz, and Ron Kikinis. Processing and visualization for diffusion tensor mri. *Medical image analysis*, 6(2):93–108, 2002.
- [48] Daniel B Ennis and Gordon Kindlmann. Orthogonal tensor invariants and the analysis of diffusion tensor magnetic resonance images. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 55(1):136–146, 2006.
- [49] Lauren J O’Donnell and Carl-Fredrik Westin. An introduction to diffusion tensor image analysis. *Neurosurgery Clinics*, 22(2):185–196, 2011.
- [50] Aurélien Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. .o’Reilly Media, Inc.”, 2017.
- [51] Antonio Tristán-Vega and Santiago Aja-Fernández. Design and construction of a realistic DWI phantom for filtering performance assessment. In *Med Image Comput Comput Assist Interv*, volume 5761, pages 951–958, 2009.
- [52] Antonio Tristán-Vega and Santiago Aja-Fernández. DWI filtering using joint information for DTI and HARDI. *Med Imag Anal*, 14(2):205–218, 2010.
- [53] Santiago Aja-Fernández and Gonzalo Vegas-Sánchez-Ferrero. *Statistical Analysis of Noise in MRI*. Springer, 2016.
- [54] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [55] Vicente Molina, Alba Lubeiro, Oscar Soto, Margarita Rodriguez, Aldara Álvarez, Rebeca Hernández, and Rodrigo de Luis-García. Alterations in prefrontal connectivity in schizophrenia assessed using diffusion magnetic resonance imaging. *Prog Neuropsychopharmacol Biol Psychiatry*, 76:107–115, 2017.
- [56] J-Donald Tournier, Fernando Calamante, and Alan Connelly. Mrtrix: diffusion tractography in crossing fiber regions. *International journal of imaging systems and technology*, 22(1):53–66, 2012.
- [57] Jenkinson M, Beckmann C.F, Behrens T.E, Woolrich M.W, and Smith S.M. Fsl. *NeuroImage*, 62:782–90, 2012.
- [58] Woolrich M.W, Jbabd S, Patenaude B, Chappell M, Makni S, Behrens T, Beckmann C, Jenkinson M., and Smith S.M. Bayesian analysis of neuroimaging data in fsl. *NeuroImage*, 45:173–86, 2009.

- [59] Tsss user guide. <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/TBSS/UserGuide>. Accessed: 2019-08-25.
- [60] Atlases included with fsl. <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases>. Accessed: 2019-08-25.
- [61] Randomise user guide. <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Randomise/UserGuide>. Accessed: 2019-08-25.