Universidad de Buenos Aires

Facultad de Ciencias Exactas y Naturales

Departamento de Computación

Segmentación automática multi-tarea de imágenes médicas utilizando redes neuronales profundas

Tesis de Licenciatura en Ciencias de la Computación

Nicolás Roulet

Director: Dr. Enzo Ferrante

Codirector: Dr. Diego Fernández Slezak

Buenos Aires, 2018

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
N. Roulet, D. Fernandez Slezak & E. Ferrante; "Segmentación automática multi-tarea de imágenes médicas utilizando redes neuronales profundas"
Facultad de Ciencias Exactas y Naturales - Universidad de Buenos Aires, 2018.

Índice

1. Introducción **9**

1.1. Objetivo 13

1.2. Estructura del trabajo 13

2. Segmentación de imágenes médicas **14**

2.1. Imágenes por Resonancia Magnética 14

2.2. Primeros algoritmos de Segmentación 15

2.3. Redes Neuronales Artificiales 16

2.4. Redes Neuronales Convolucionales 19

2.5. Arquitecturas de segmentación de imágenes médicas 21

2.6. Convolución sobre imágenes médicas tridimensionales 24

2.7. Métricas de evaluación 24

3. Modelos de segmentación propuestos **26**

3.1. Formalización del problema 26

3.2. Modelos preliminares 26

3.2.1. Modelo Naive 26

3.2.2. Modelo Base 27

3.3. Funciones de costo 28

3.3.1. Entropía cruzada 28

3.3.2. Entropía Cruzada Selectiva 29

3.3.3. Dice 30

3.3.4. Dice Selectivo 31

3.4. Estrategias de muestreo 31

3.5. Preprocesamiento de datos 33

3.6. Multiplicación de datos 34

4. Infraestructura de entrenamiento y evaluación **35**

5. Escenarios multi-tarea **37**

5.1. Estructuras anatómicas 37

5.2. Hiperintensidades de materia blanca 38

5.3. Tumores 39

6. Resultados **41**

6.1. Escenarios mono-tarea 43

6.1.1. Segmentación de estructuras anatómicas 44

6.1.2. Segmentación de lesiones WMH 46

6.1.3. Segmentación de tumores 48

6.2. Estructuras anatómicas y lesiones de WMH 50

6.3. Estructuras anatómicas y tumores 56

7. Conclusiones **60**

Resumen

Las distintas técnicas de imágenes médicas constituyen una parte fundamental de la investigación y el diagnóstico médico. En la actualidad, la aplicación de métodos computacionales para enriquecer la información provista por estos estudios presenta posibilidades muy variadas para la asistencia a profesionales en distintos ámbitos médicos. En particular, la identificación de estructuras anatómicas y patológicas en imágenes cerebrales es de gran utilidad para elaborar diagnósticos y estudiar la evolución temporal de enfermedades o lesiones.

El presente trabajo estudia el problema de la segmentación automática de imágenes médicas para identificar tanto estructuras anatómicas (los diferentes tejidos cerebrales como materia blanca, gris, etc.) como patológicas (como tumores o distintos tipos de lesiones) simultáneamente, utilizando Redes Neuronales Convolucionales. El objetivo es entrenar un único modelo capaz de producir segmentaciones de ambas estructuras de interés, a partir de dos conjuntos de datos de referencia, cada uno etiquetado en función de sólo una de las tareas. En esta tesis de licenciatura se discutirán las particularidades de la segmentación multi-tarea con bases de datos disjuntas y sus diferencias con el clásico problema de segmentación multi-clase, se propondrán nuevas estrategias para abordar el problema en base a *funciones de costo selectivas* y se estudiará su desempeño en el contexto de la segmentación de neuroimágenes, evaluando los resultados sobre datasets con etiquetas combinadas de referencia.

Los resultados obtenidos sugieren que las nuevas *funciones de costo selectivas* propuestas presentan un mejor rendimiento que las existentes en la literatura, y abren la puerta al desarrollo de nuevos métodos de segmentación multi-tarea en el contexto de las neuroimágenes.

Abstract

The different medical imaging techniques are a fundamental part of medical research and diagnosis. Currently, the application of computational methods to enrich the information provided by these studies presents a wide range of possibilities for assistance to professionals in different medical fields. In particular, the identification of anatomical and pathological structures in brain images is very useful to make diagnoses and study the temporal evolution of diseases or injuries.

This work studies the problem of automatic segmentation of medical images to identify both anatomical structures (different brain tissues such as white matter, gray matter, etc.) and pathological (such as tumors or different types of injuries) simultaneously, using Convolutional Neural Networks. The objective is to train a single model capable of producing segmentations of both structures of interest, from two sets of reference data, each one labeled according to only one of the tasks. In this thesis, the details of multi-task segmentation with disjunct databases and their differences with the classical problem of multi-class segmentation will be discussed, new strategies will be proposed to approach the problem based on *selective loss functions* and their performance in the context of neuroimaging segmentation will be studied, evaluating the results on datasets with combined reference labels.

The results obtained suggest that the proposed new *selective loss functions* present a better performance than those existing in the literature, and open the door to the development of new multi-task segmentation methods in the context of neuroimaging.

Agradecimientos

Quisiera agradecer a la gran cantidad de personas que me ayudaron en este proceso, sin las cuales no me hubiera sido posible llegar hasta acá.

En primer lugar, a Diego y Enzo, mis directores, por plantearme un problema tan interesante y guiarme en el largo camino de buscarle una solución. Por estar disponibles y responder mis preguntas, por motivarme cuando nada parecía funcionar. Por estar abiertos a charlar mis ideas más descabelladas.

A mis jurados, por tomarse el tiempo de leer y corregir este trabajo.

A mis amigos y compañeros de la facu, que desde un principio hicieron de esta carrera una experiencia genial. Por los TPs infinitos, por las cursadas, los metegoles, los partidos de fútbol y los viajes juntos.

A mis papás, por estar siempre, por venir a visitar, por esperarme cada vacación con los brazos abiertos. Por leer y releer mi tesis, atentos a todos los detalles. A Javi, por la convivencia y el compañerismo durante gran parte de la carrera. Al resto de mi familia, por las reuniones y las alegrías.

A mis amigos de la infancia, por las juntadas en las que nadie hable de grafos. Por el cable a tierra.

A la UBA, por darme acceso a una educación de primera. Por darme oportunidades. Por hacerme conocer tanta gente. Por cambiar mi forma de pensar.

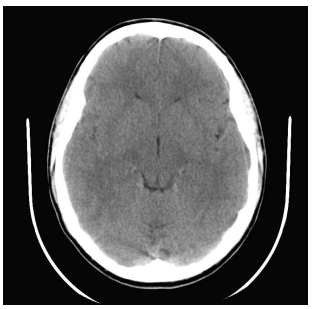
A Maru, por acompañarme todo este tiempo. Por los planes pasados y los planes futuros.

sinc(*i*) Research Institute for Signals, Systems and Computational Intelligence (sinc.unl.edu.ar)
N. Roulet, D. Fernandez Slezak & E. Ferrante; "Segmentación automática multi-tarea de imágenes médicas utilizando redes neuronales profundas"
Facultad de Ciencias Exactas y Naturales - Universidad de Buenos Aires, 2018.

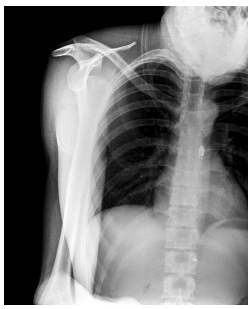
1. Introducción

Las imágenes médicas son una herramienta fundamental en la medicina moderna que utiliza un conjunto variado de técnicas para generar representaciones visuales del interior de un cuerpo de forma rápida, precisa y generalmente no invasiva. Son de gran utilidad para el estudio del funcionamiento del cuerpo y el reconocimiento de patologías, permitiendo la detección temprana y el monitoreo de la evolución de enfermedades y lesiones. Los tipos de imágenes médicas más utilizadas son las radiografías, resonancias magnéticas nucleares, tomografías computadas y ecografías, entre otras. La interpretación de cada tipo de imagen es una tarea compleja que requiere un grado de especialización muy alto por parte de los profesionales involucrados.

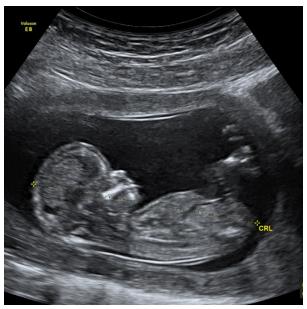
Por este motivo, el análisis de imágenes médicas mediante métodos computacionales es un área de investigación muy activa que brinda la posibilidad de automatizar un proceso que de ser realizado manualmente, además de requerir personal capacitado, es lento y presenta el riesgo del error humano en circunstancias en las que éste puede tener consecuencias graves. En muchos casos, dichos métodos computacionales alcanzan una precisión y efectividad mayores que las de un profesional en su ámbito [1].



(a) Tomografía computada cerebral.



(b) Radiografía de hombro.



(c) Ecografía fetal.

Figura 1: Ejemplos de imágenes médicas.

Los análisis que se hacen a partir de estas imágenes son muy variados, desde determinar el género de un bebé a partir de una ecografía fetal a estimar el volumen de un órgano o medir actividad cerebral frente a distintos estímulos. Dentro de las tareas de asistencia al diagnóstico que resulta provechoso automatizar, la *segmentación de imágenes* consiste en identificar regiones dentro de una imagen que compartan una misma característica. La elección de dicha característica determina una *tarea*. En el ámbito médico esta tarea puede ser, por ejemplo, identificar distintos tipos de tejidos, órganos o patologías. Este trabajo se enfoca en la segmentación de imágenes cerebrales de resonancia magnética (MRI, por sus siglas en inglés).

En los últimos años, con la evolución de técnicas de aprendizaje automático como las redes neuronales profundas y con la disponibilidad creciente de capacidad de cómputo paralelo mediante unidades de procesamiento gráfico (GPUs), este campo de investigación también ha alcanzado resultados similares al desempeño humano para distintas tareas de segmentación. Algunos ejemplos de arquitecturas de redes neuronales que han obtenido resultados notables en este contexto son UNet [2], FCN [3] y DeepMedic [4]. Estas técnicas son algoritmos de aprendizaje supervisado: utilizan datos ya etiquetados en función de la tarea específica que

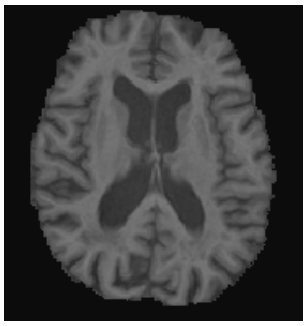
se quiere segmentar. Estos datos son creados por profesionales que deben etiquetar manualmente imágenes completas. Su obtención es, consecuentemente, muy costosa.

La segmentación multi-tarea consiste en identificar regiones en una imagen en función de más de una tarea. Por ejemplo, identificar tanto tejidos anatómicos como tumores en la misma imagen cerebral. Notar que esto difiere del clásico problema de segmentación de imágenes médicas multi-clase, donde el objetivo también es segmentar diferentes estructuras o patologías, pero todas ellas se encuentran presentes en las imágenes de una única base de datos. En este trabajo, utilizaremos el concepto de segmentación multi-tarea para referirnos a subconjuntos de anotaciones provistos en distintas bases de datos. En el ámbito de las imágenes médicas cerebrales, por ejemplo, es normal encontrar bases de datos públicas con etiquetas de estructuras anatómicas (como los diversos tejidos cerebrales) o patológicas (como tumores o lesiones cerebrales), pero no abundan aquellas en que ambas anotaciones hayan sido realizadas sobre las mismas imágenes (ver Figura 2). En ocasiones, los especialistas médicos desean visualizar ambas estructuras en el mismo paciente (por ejemplo, para visualizar el efecto que un tumor cerebral produce en las estructuras anatómicas aledañas). Sin embargo, no resulta trivial obtener las segmentaciones conjuntas sin contar con datos etiquetados simultáneamente para ambas tareas. El foco central de este trabajo será la construcción de dichos modelos, a partir de bases de datos con segmentaciones disjuntas.

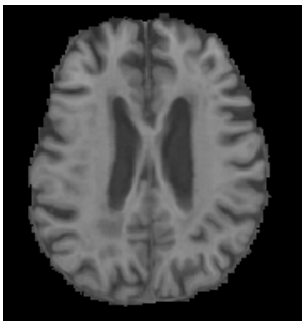
El problema reside en que entrenar un modelo directamente y de forma indistinta con imágenes de varios datasets de tareas diferentes suele reducir la calidad de los resultados obtenidos con respecto a modelos entrenados para tareas independientes, ya que lo que en una imagen se identifica con una etiqueta, puede estar presente en otra imagen pero etiquetado de otra forma, lo que "confunde" al modelo. Uno de los resultados que se observan al hacer esto es que en lugar de generar etiquetas combinadas, el modelo entrenado intenta utilizar diferencias típicas entre ambos conjuntos de datos para reconocer el origen de las imágenes y segmentarlas únicamente en función de la tarea correspondiente, como se puede ver en la Figura 3.

Estrategias para abordar variantes de este problema son exploradas en algunos trabajos recientes. El trabajo [5] estudia técnicas de muestreo de partes de imágenes en tres planos ortogonales para entrenar una misma red en reconocimiento de estructuras anatómicas en cerebro, pecho y corazón, obteniendo resultados similares en cada tarea a los obtenidos con una red entrenada exclusivamente para dicha tarea. El trabajo [6] diseña una función de costo y propone adaptaciones en la arquitectura de una red neuronal para el entrenamiento de la misma sobre datos con distintos conjuntos de etiquetas. El trabajo [7] estudia los resultados de distintas funciones de costo en el entrenamiento en escenarios con gran desbalance de etiquetas, que es un fenómeno común al combinar distintas tareas. El trabajo [8] estudia el concepto de aprendizaje continuo, o «aprender sin olvidarse», que consiste en entrenar un modelo para una tarea y luego entrenarla para otra, proponiendo técnicas para que el modelo mantenga la capacidad de realizar la primer tarea mientras aprende la segunda. En esta tesis de licenciatura, se adoptará un enfoque que comparte similitudes con los trabajos [5] y [6], en el sentido de propondremos nuevas funciones de costo pensadas exclusivamente para el entrenamiento de modelos a partir de datos disjuntos.

Existe una problemática paralela que puede tener influencia sobre los resultados obtenidos, conocida

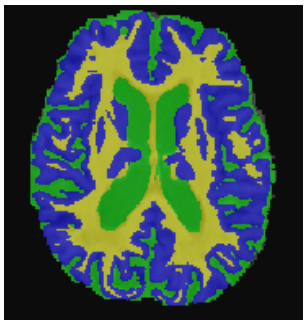


(a.1) Resonancia magnética correspondiente a un paciente sano

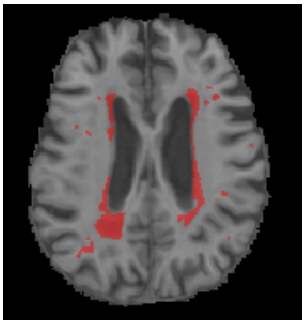


(b.1) Resonancia magnética correspondiente a un paciente con lesiones de materia blanca.

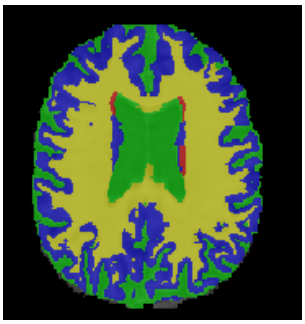
(c.1) Resonancia magnética correspondiente a un paciente con lesiones de materia blanca.



(a.2) Etiquetas anatómicas

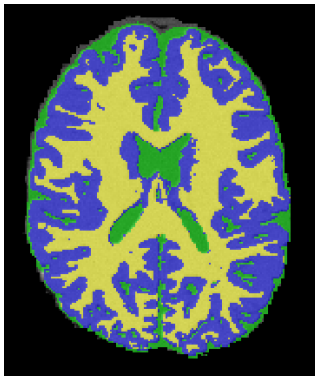


(b.2) Etiquetas de lesión

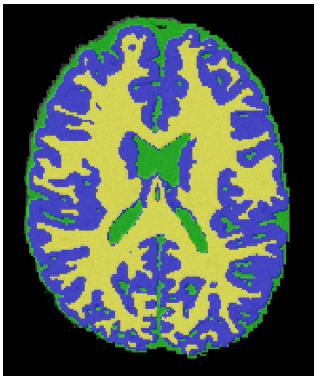


(c.2) Etiquetas combinadas

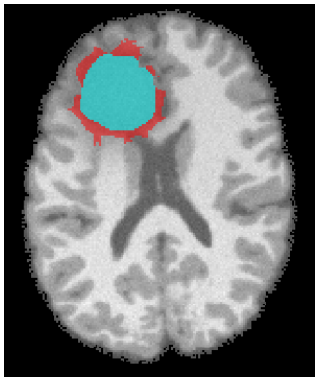
Figura 2: Ejemplo de segmentación multi-tarea. **a)** Etiquetado de estructuras anatómicas de líquido cefalorraquideo (verde), materia gris (azul) y materia blanca (amarillo). **b)** Etiquetado de lesiones de hiperintensidades de materia blanca (rojo). **c)** Etiquetado combinado (multi-tarea). En esta tesis de licenciatura, se estudia y proponen diversas estrategias para entrenar un modelo basado en RNC capaz de producir el etiquetado combinado **c** a partir de dos conjuntos disjuntos de anotaciones, donde uno contiene sólo etiquetas anatómicas (**a**) y otro sólo etiquetas patológicas (**b**).



(a.1) Imagen con etiquetas anatómicas.



(a.2) Predicción del modelo.



(b.1) Imagen con etiquetas de tumores.

(b.2) Predicción del modelo.

Figura 3: Predicciones de un mismo modelo entrenado directamente sobre imágenes de dos tareas distintas: segmentación de estructuras anatómicas y segmentación de tumores, al ser aplicado sobre imágenes de validación de ambos conjuntos de datos. Notar que en el caso **b.2**, pese a que las estructuras anatómicas están presentes (materia blanca, gris y líquido cefaloraquídeo), el modelo las identifica como background. Esto se debe a que, durante la fase de entrenamiento, el modelo fue 'confundido' respecto al significado de la etiqueta background, ya que dichas áreas aparecen con etiquetas anatómicas en la base de datos **(a)**, pero como background en la base de datos **(b)**. El modelo aprendió entonces que si la distribución de las intensidades de gris de la imagen proviene de una con tumores, entonces a esa imagen no le asignará estructuras anatómicas.

como problema *multi-dominio*. Este problema surge al aplicar un modelo entrenado sobre un conjunto particular de imágenes a otro conjunto de imágenes que fueron obtenidas de una fuente distinta. En el caso de las imágenes médicas, estas pueden haber sido obtenidas con equipamiento diferente, utilizando otros parámetros y haber recibido un post-procesamiento distinto. Estas diferencias en la naturaleza de las imágenes pueden deteriorar significativamente el desempeño del modelo. Utilizar imágenes de entrenamiento de distintos dominios es útil para generar modelos más robustos y menos susceptibles a este problema, dado que la variabilidad de los datos permite una mejor capacidad de generalización. También es posible adaptar los modelos para atacar explícitamente esta problemática [6]. Si bien el problema multi-dominio está fuera del alcance de esta tesis (dado que el foco estará puesto en la problemática multi-tarea), en la Sección 6 se incluye un pequeño estudio para caracterizar el impacto que las diferencias de dominio (en término de diferencias en las distribuciones de las intesidades de gris de las imágenes) pueden acarrear en los resultados de los modelos entrenados.

1.1. Objetivo

El objetivo de esta tesis de licenciatura es diseñar e implementar un modelo de software para la segmentación automática multi-tarea de imágenes médicas tridimensionales de resonancia magnética cerebral. Para este fin, se considerarán modelos de aprendizaje supervisado mediante redes neuronales profundas, con adaptaciones particulares para favorecer el caso multi-tarea. El desempeño de estos modelos se evaluará con nuevos datasets con etiquetas combinadas.

El foco estará puesto en problemas con dos tareas. Los escenarios multi-tarea a considerar serán segmentación anatómica de tejidos, combinada con algún tipo de segmentación de patologías. Las patologías consideradas son tumores e hiperintensidades de materia blanca (WMH).

1.2. Estructura del trabajo

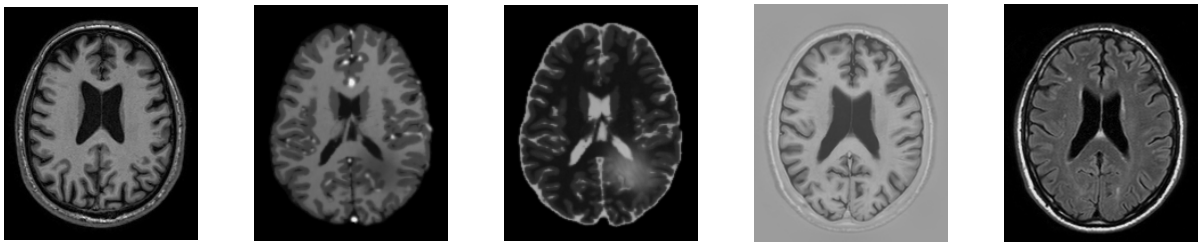
En la Sección 2 se desarrollará la base teórica necesaria para abordar el problema, explicando desde los conceptos generales sobre imágenes de resonancia magnética hasta las técnicas modernas de segmentación de imágenes médicas basadas en Redes Neuronales Profundas. A continuación, en la Sección 3 se plantearán los varios esquemas de segmentación propuestos y se explicará cómo se aplican a escenarios multi-tarea. En la Sección 4 se explica la infraestructura desarrollada para implementar las soluciones propuestas. La Sección 5 detalla los escenarios multi-tarea que serán considerados durante la experimentación. La Sección 6 reporta los resultados de la experimentación incluyendo un análisis detallado de los datos utilizados y el desempeño de cada modelo propuesto. La sección 7 resume los puntos más relevantes del trabajo, plantea una conclusión general y posibles trabajos futuros.

2. Segmentación de imágenes médicas

Desde un punto de vista formal, la segmentación de imágenes consiste en asignar a cada pixel de una imagen (o voxel[1] en el caso de imágenes tridimensionales) una etiqueta de un conjunto predeterminado en base a un criterio. En función del tipo de imagen utilizado, la naturaleza de este problema varía.

2.1. Imágenes por Resonancia Magnética

La técnica de Imágenes por Resonancia Magnética es un método no invasivo para observar las estructuras internas de un cuerpo. A diferencia de otras técnicas como las Radiografías o las Tomografías Computadas, no utiliza radiación ionizante sino que se basa en aplicar pulsos magnéticos para alinear los espines de determinados tipos de átomos y medir los campos magnéticos generados cuando estos átomos vuelven a estabilizarse. Esta técnica permite generar imágenes volumétricas (tridimensionales) capturando múltiples planos paralelos del mismo objeto.



(a) T1 **(b)** T1 Gadolinio **(c)** T2 **(d)** IR **(e)** FLAIR

Figura 4: Modalidades de Imágenes de resonancia magnética utilizadas en este trabajo

Existen varias modalidades de MRI, en función de la frecuencia del campo magnético utilizado, el retraso entre la aplicación del campo magnético y la medición del campo magnético generado por los átomos, la utilización de agentes de contraste y otras variantes. Las modalidades presentes en este trabajo son: T1, T1 con contraste de Gadolinio, T2, IR y FLAIR. T1 se caracteriza por utilizar un Tiempo de Repetición (TR) entre pulsos magnéticos muy breve y un Tiempo de Eco (TE) para medir la respuesta también muy breve. T2 utiliza TRs y TEs más largos, y FLAIR aún más. IR utiliza pulsos normales y pulsos invertidos alternadamente. T1 con Gadolinio utiliza el Gadolinio como elemento de contraste paramagnético. Cada variante genera imágenes en las que se resaltan distintos tejidos y sustancias como las grasas, los agentes de contraste o el agua, entre otros, como se puede ver en la Figura 4. Las imágenes mostradas en la figura se denominan *cortes axiales*, muestran un plano horizontal del volumen tridimensional representado por la imagen completa. Esta será la visualización utilizada a lo largo de todo el trabajo para mostrar imágenes volumétricas, escogiendo planos con información relevante. Otros cortes posibles son el *corte sagital* (separando el cerebro en una sección izquierda y una derecha) y el *corte coronal* que separa el cerebro en una sección delantera y una trasera.

[1] Un *voxel* (de *volumetric pixel*) es el equivalente de un píxel para un volumen tridimensional.

2.2. Primeros algoritmos de Segmentación

Los primeros algoritmos de segmentación de imágenes médicas se basaban en la utilización de técnicas relativamente simples de procesamiento de imágenes. Se describen a continuación algunas de ellas [9]:

- **Umbralado:** es una técnica de segmentación binaria que consiste en determinar un umbral de brillo tal que los pixeles que excedan ese brillo son considerados de una clase y los que estén por debajo, de la otra. Esto depende fuertemente de que la estructura que se quiere segmentar tenga un nivel de brillo distinto que el resto de la imagen y es susceptible a ruido. Se suele combinar con otros métodos para obtener resultados más confiables.

- **Crecimiento de regiones:** consiste en generar regiones a partir de puntos particulares (semillas) de la imagen, extendiéndolos a pixeles vecinos que sean similares, bajo alguna métrica de similitud. La elección de semillas es un aspecto importante de este proceso.

- **Detección de bordes:** esta técnica utiliza el gradiente de las intensidades de los pixeles para identificar cambios bruscos en la imagen, que se utilizan para reconocer bordes de estructuras y así separar secciones de la imagen. Este método también es muy suceptible a ruido, por lo que es usual suavizar las imágenes antes de aplicarlo.

- **Watershed:** es un método también basado en regiones, que interpreta la imagen como un modelo topográfico, donde la intensidad de cada pixel representa altura. Se simula una inundación del espacio desde los puntos más bajos, de manera que se forman lagunas en las zonas de menor intensidad. Al continuar la inundación, las fronteras donde eventualmente se unen las lagunas representan las divisiones entre regiones. Este método es útil para separar objetos similares, por ejemplo para contar la cantidad de células presentes en una imagen.

- **Contornos activos:** también llamada *snake*, es un modelo deformable que ajusta una curva (o superficie, en imágenes tridimensionales) mediante la minimización iterativa de una función de energía. Esta función tiene una componente de influencia externa basada en la información de la imagen y otra componente de influencia interna que depende de las propiedades del modelo mismo, como la curvatura, que permiten garantizar suavidad y otras características deseables del contorno detectado. Este enfoque requiere un contorno inicial, que puede ser indicado manualmente o detectado mediante algún otro método de los aquí listados.

- **Level-set:** es otro tipo de modelo deformable, que plantea la división entre regiones como la superficie de nivel de una función φ. El modelo evoluciona la función φ para ajustar la superficie de nivel en función del gradiente de φ y la intensidad de los pixeles de la imagen.

Con el apogeo de las técnicas de aprendizaje automático, modelos basados en extracción de características y aplicación de algoritmos de clasificación comenzaron a ser más populares. Estos métodos luego fueron reemplazados por redes neuronales profundas, que demuestran mayor flexibilidad y mejores resultados, y actualmente constituyen el estado del arte en problemas de segmentación de imágenes médicas.

A continuación se expondrá una breve introducción a las Redes Neuronales Artificiales (RNA), como una herramienta de aprendizaje supervisado orientado a problemas de clasificación. Se denomina **problemas de clasificación** a la familia de problemas que consisten en asignar una categoría a cada muestra de un conjunto de observaciones. Esto es, aprender una función $f : \mathbb{R}^n \to \mathcal{L}$, donde $\mathcal{L} = \{l_1, \dots l_k\}$ es un conjunto de etiquetas posibles. Luego se extenderá el modelo a problemas de segmentación.

2.3. Redes Neuronales Artificiales

A efectos de este trabajo, la forma más conveniente de interpretar una Red Neuronal Artificial es como un modelo matemático que codifica una función, con la particularidad de que el modelo tiene la capacidad de aprender aproximaciones de una función particular en base a valuaciones de la misma.

Perceptrones Simples y Perceptrones Multiclase

Las redes neuronales más simples son los Perceptrones. Un **Perceptrón Simple** es una función

$$f : \mathbb{R}^n \to \mathbb{R}, \quad f(x; w, b) = h(\langle w, x \rangle + b) \tag{1}$$

donde $w \in \mathbb{R}^n$, $b \in \mathbb{R}$ son parámetros del modelo[2] y $h : \mathbb{R} \to \mathbb{R}$ es una función de activación no lineal. Un Perceptrón Simple tiene dos capas: una capa de entrada de dimensión n y una capa de salida de dimensión 1. La función f conforma la transición entre ambas capas.

En problemas de clasificación binarios (sólo dos clases 0 y 1), una función de activación típica es *Heaviside*:

$$hs(z) = \begin{cases} 0 & \text{si } z < 0 \\ 1 & \text{si } z \geq 0. \end{cases} \tag{2}$$

Es decir que si $\langle w, x \rangle + b < 0$, el clasificador asigna a x la clase 0, en caso contrario, la clase 1. En problemas de clasificación con más de dos clases, el Perceptrón Simple se puede extender a un **Perceptrón Multiclase**. A diferencia de un Perceptrón Simple, un Perceptrón Multiclase con m clases codifica una función $f : \mathbb{R}^n \to \mathbb{R}^m$, $f(x; W, b) = h(Wx + b)$. Ahora W es una matriz de $\mathbb{R}^{m \times n}$, b un vector en $\mathbb{R}^m$, y h una función no lineal $h : \mathbb{R}^m \to \mathbb{R}^m$. La salida del perceptrón multiclase es un vector de m dimensiones que representa los puntajes asignados a cada clase.

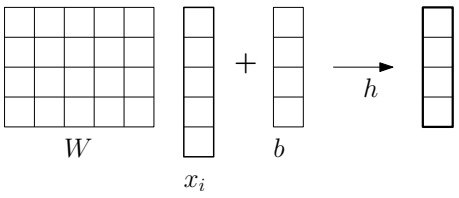


Figura 5: Ejemplo de un perceptrón multiclase.

[2]La notación $f(x; w, b)$ significa: "$f(x)$, dados los parámetros w y b"

Una función de activación muy utilizada es *softmax*:

$$softmax(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{m} e^{z_k}}. \tag{3}$$

Softmax interpreta el vector de entrada $\{z_j\}$ como los logaritmos de las probabilidades no normalizadas de pertenencia a cada clase l_j, retornando un vector de probabilidades entre 0 y 1, cuya suma da 1. Para obtener la predicción final, se toma la clase con mayor probabilidad:

$$\underset{j=1...m}{\operatorname{argmax}} \quad softmax(z)_j \tag{4}$$

Existen otras funciones no lineales que pueden ser aplicadas a la salida de un perceptrón y cuyo resultado no necesariamente es interpretable como un vector de probabilidades, pero por lo general representan algún tipo de puntaje por clase. Una propiedad conveniente tanto de la función lineal $Wx + b$ como de $softmax$ es que ambas son derivables. Se verá a continuación la utilidad de esta observación.

Los parámetros que determinan la función representada por un Perceptrón Multiclase son los coeficientes de la matriz W y del vector b. Dado un conjunto de K valuaciones puntuales $\{(x_i, y_i)\}_{1 \leq i \leq K}$, se busca optimizar estos parámetros para minimizar una función de error L, que dados y_i y $f(x_i; W, b)$ computa un coeficiente de error entre ellos. La función de error más comúnmente usada es la entropía cruzada, sobre la que se entrará en detalle en la sección 3.3.1. Existen múltiples algoritmos para optimizar los parámetros del modelo. Uno de los más utilizados es el Descenso por el Gradiente, que consiste en actualizar iterativamente los coeficientes mediante las derivadas de la función de error en función de los propios coeficientes, hasta que estos converjan:

$$W'_{kl} = W_{kl} - \lambda \frac{1}{K} \sum_{i=1}^{K} \frac{\partial L}{\partial W_{kl}}(y_i, f(x_i; W, b)) \tag{5}$$

$$b'_k = b_k - \lambda \frac{1}{K} \sum_{i=1}^{K} \frac{\partial L}{\partial b_k}(y_i, f(x_i; W, b)). \tag{6}$$

λ es un hiperparámetro del modelo denominado *learning rate*, que determina la velocidad de aprendizaje. El Descenso por el Gradiente utiliza el promedio de las derivadas de los errores sobre todas las muestras disponibles. Una variante típica es el Descenso por el Gradiente Estocástico, que consiste en promediar las derivadas de un subconjunto aleatorio de las muestras, reduciendo la cantidad de cómputos necesarios en cada iteración.

Existen muchas variantes al algoritmo de Descenso por el Gradiente Estocástico. Una de las más utilizadas en la actualidad es el optimizador Adam [10]. A grandes rasgos, este algoritmo utiliza un *learning rate* particular para cada parámetro de la red y actualiza ese valor en cada iteración de aprendizaje en función de las medias móviles de los gradientes. Esto resta importancia en modo considerable a la elección de *learning rate* inicial.

Perceptrones Multicapa

La capacidad expresiva de los Perceptrones Simples y los Perceptrones Multiclase es muy limitada, dado que sólo pueden representar patrones linealmente separables [11]. Los **Perceptrones Multicapa**[3] son una extensión del modelo anterior que permite aproximar cualquier función continua [12]. La idea consiste en componer Perceptrones Simples para generar modelos más complejos. Por ejemplo, un perceptrón de tres capas tiene la siguiente estructura:

$$f(x; W_1, b_1, W_2, b_2) = h_2(W_2 \cdot h_1(W_1 \cdot x + b_1) + b_2) \tag{7}$$

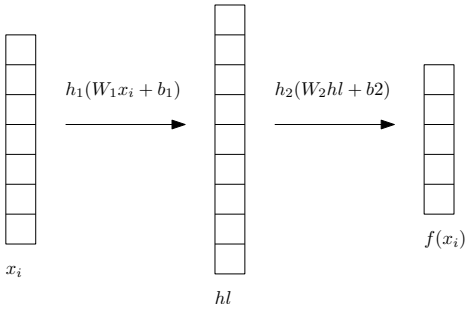


Figura 6: Ejemplo de un perceptrón de tres capas.

Se denota Θ al conjunto de parámetros de una red, con lo que la función representada por el modelo se puede denotar $f(x; \Theta)$.

Las capas intermedias (*hidden layers*) no suelen usar softmax como función de activación, dado que la interpretación como probabilidades sólo es util en el resultado de la última capa. Una de las más utilizadas es ReLU (por *Rectified Linear Unit*): $ReLU(z)_j = max(z_j, 0)$. Tiene varias ventajas con respecto a otras funciones populares como la función sigmoidea y la tangente hiperbólica, como ser muy eficiente computacionalmente, fácilmente derivable (excepto en el 0) y ser menos sensible al problema de desvanecimiento del gradiente[13]. En la práctica, es muy utilizada porque empíricamente muestra una convergencia más rápida que otras funciones.

Notar que las dimensiones de las capas intermedias no dependen de la dimensión de entrada y la de salida, con lo que es un hiperparámetro que se puede ajustar para obtener mejores resultados. El diseño de las capas de una red, tanto la cantidad y las dimensiones de las capas intermedias como las funciones de activación utilizadas forman parte de la *arquitectura* de la red. Las capas de un perceptrón se denominan *capas densas*, en contraposición con las capas convolucionales que serán presentadas a continuación.

El entrenamiento de un Perceptrón Multicapa también se lleva a cabo a través del método de Descenso por el Gradiente, o alguna de sus variantes. El problema radica en que computar el gradiente de los parámetros de las capas intermedias en función del error en la capa de salida no es trivial. Esto se realiza mediante el algoritmo de backpropagation[14], que propaga el gradiente del error desde la última capa hacia atrás. Los detalles de este algoritmo escapan al objetivo de este trabajo, por lo que no serán incluidos.

[3] El Perceptron Multicapa es el modelo tradicional de Redes Neuronales.

Perceptrones para clasificación de imágenes

Si bien la entrada de un perceptrón es un vector unidimensional y una imagen es representada por una matriz, la clasificación de imágenes es posible convirtiendo la imagen en un vector, concatenando por ejemplo las filas de la misma. Esta estrategia, sin embargo, pierde información posicional sobre el entorno de cada píxel, que es fundamental en el reconocimiento de estructuras en una imagen. Adicionalmente, la arquitectura de la red depende del tamaño de la entrada (las dimensiones de la imagen), con lo que no se puede utilizar la misma red para clasificar imágenes de distintos tamaños. Por último, para imágenes de gran tamaño la cantidad de parámetros del modelo se vuelve rápidamente inmanejable.

Un modelo que mitiga estos problemas son las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés).

2.4. Redes Neuronales Convolucionales

Las CNNs son similares a las Redes Neuronales tradicionales: se estructuran en capas, aprenden funciones ajustando un conjunto de parámetros y son evaluadas efectuando productos internos y aplicando funciones no lineales intercaladamente. El entrenamiento se basa en minimizar una función de error y la salida representa puntajes asignados a cada clase.

Lo que distingue a las CNNs es que están diseñadas específicamente para el procesamiento de imágenes, aprovechando la distribución espacial de la entrada. Los elementos de una capa de una CNN están distribuidos en tres dimensiones: alto, ancho y profundidad. Su valor no depende de todos los valores de la capa anterior como en una red tradicional sino de una pequeña parte de ella, denominada campo receptivo. Una capa codifica entonces una función $g : \mathbb{R}^{w_1 \times h_1 \times d_1} \to \mathbb{R}^{w_2 \times h_2 \times d_2}$.

Los dos tipos principales de capas de una CNN son:

- **Convolución:** son las capas donde se lleva a cabo el grueso del procesamiento y del aprendizaje. Los parámetros de una capa de convolución se agrupan en filtros. Cada filtro abarca un pequeño entorno en altura y anchura del volumen de entrada (típicamente 3×3, 5×5 o 7×7) denominado *tamaño del filtro* o *tamaño del kernel*, pero toda la profundidad. El tamaño del kernel es un hiperparámetro de la capa. Cada filtro T es un tensor de rango 3 (matriz tridimensional) de coeficientes $\{t_{ijk}\}$. Aplicar un filtro sobre un sector $\{l_{ijk}\}$ del volumen de entrada de las mismas dimensiones consiste en multiplicar elemento a elemento los coeficientes del filtro con los valores de la capa y sumar todo:

$$T(l) = \sum_{ijk} l_{ijk} t_{ijk}. \tag{8}$$

  La cantidad de filtros de una capa es igual a la profundidad del volumen de salida. Se denominan "canales" a cada tajada de profundidad fija de un volumen. Cada canal del volumen de salida está dado por la aplicación de un mismo filtro a todo el volumen de entrada.

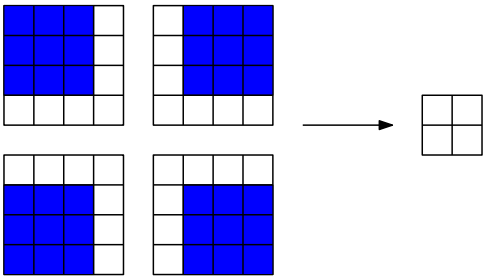


Figura 7: Convolución sin padding. El área azul representa dónde se aplica el filtro en cada paso. Reduce las dimensiones de la capa de salida.

Otro hiperparámetro de una capa convolucional es el *paso* (*stride*, en inglés), que determina la separación entre dos aplicaciones distintas de un mismo filtro. El paso típico es de 1, con lo que el filtro se aplica sobre cada sector consecutivo de la capa de entrada. Notar que aplicar un filtro de 3×3 a una capa de $m \times n$ (en anchura y altura) con paso 1 resulta en una salida de $(m-2 \times n-2)$, como se ve en la figura 7. Para mantener el alto y ancho de la capa anterior, se puede utilizar otro hiperparámetro, el *padding*, que determina cuántas filas y columnas de ceros se agregan a los bordes de la imagen para compensar, como se ve en la figura 8. Un volumen de entrada de ancho w con un tamaño de filtro k, un paso de s y un padding de p produce un volumen de salida de $\lfloor (w - f + 2p)/s \rfloor + 1$. La misma fórmula vale para el alto del volumen.

Comparada con una capa densa, una capa convolucional presenta la ventaja de que la cantidad de parámetros de la capa no depende del tamaño del volumen de entrada, sólo de su profundidad. Por este motivo, una misma capa puede ser aplicada sobre volúmenes de diferente ancho y alto, produciendo volúmenes de salida de distintas dimensiones según la fórmula ya mencionada. El hecho de que un mismo filtro se aplique a toda una capa también reduce drásticamente la cantidad de parámetros necesarios.

- **Agregación:** las capas de agregación sintetizan la información de una capa para reducir su dimensionalidad, reduciendo también el costo computacional de evaluar y entrenar la red. El ejemplo típico de capa de agregación es *max-pooling*. Consiste en particionar la capa de entrada en una grilla de celdas de $k \times k \times 1$, típicamente con $k = 2$ en las dimensiones de alto y ancho, y computar el valor máximo en cada celda. Esto reduce una capa de dimensión $w \times h \times d$ a una de $\frac{w}{k} \times \frac{h}{k} \times d$. Otras variantes de capas de agregación incluyen computar el promedio de cada celda, o alguna norma vectorial[4], como la norma $l_2 = ||\cdot||_2$ (max-pooling es un caso particular de esto último, utilizando la norma $||\cdot||_\infty$).

Típicamente las redes convolucionales alternan grupos de capas de convolución con capas de agregación. Se denomina *campo receptivo* de una capa a la porción del volumen de la red que influye en el valor de un elemento particular de la capa. Cuanto más profunda sea la capa, mayor es su campo receptivo. Se puede observar mediante técnicas de visualización de los coeficientes de una red que los filtros de las primeras capas convolucionales detectan patrones visuales como líneas en distintos ángulos, en un entorno pequeño. Las

[4]Se nota $||\cdot||_k$ a la función $||(x_1 \dots x_n)||_k = \sqrt[k]{\sum_{i=1}^{n} x_i^k}$. La norma $||\cdot||_\infty$ se define como $\lim_{k\to\infty} ||(x_1 \dots x_n)||_k = max(x_1 \dots x_n)$

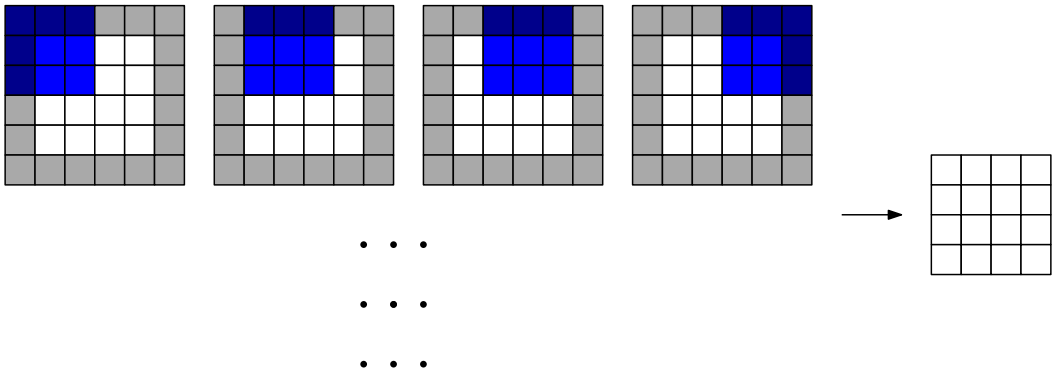


Figura 8: Convolución con padding. Por simplicidad, sólo se muestra la primer fila de convolución. Mantiene las dimensiones de la capa de salida.

capas más profundas detectan patrones cada vez más complejos y de mayor escala, como diversas figuras geométricas u objetos particulares. Por este motivo se suelen aumentar la cantidad de filtros por capa a medida que se profundiza la red, ya que la cantidad de patrones posibles de detectar aumenta.

Hasta este punto, las redes fueron presentadas como herramientas de clasificación. La siguiente sección detalla estrategias para utilizar los mismos conceptos para problemas de segmentación.

2.5. Arquitecturas de segmentación de imágenes médicas

Como se discutió en la Introducción, la segmentación de imágenes es el proceso de asignar una etiqueta a cada pixel de una imagen, en función de algún criterio. Para aplicar una red convolucional a un problema de segmentación, las dimensiones de la salida deben ser iguales (o suficientemente parecidas) a las del volumen de entrada. Este requerimiento entra en conflicto con la utilización de capas de agregación, que reducen la dimensionalidad. Esto motiva la introducción de un nuevo tipo de capa:

Convolución transpuesta[5]: es una capa de Convolución, donde cambia la interpretación del hiperparámetro *stride*. Mientras que en una Convolución con stride s se aplica cada filtro sobre parches separados por s casillas, una Convolución Traspuesta con stride s impone una separación entre las casillas de entrada de s, rellenando las posiciones intermedias con ceros, como se muestra en la figura 9. El efecto de esto es que una Convolución Transpuesta con stride de 2 duplique el ancho y alto del volumen de entrada, convirtiéndolo en una buena forma de compensar un max-pooling de 2×2.

Usando principalmente los tres tipos de capas convolucionales vistos (Convolución, Max-pooling y Convolución Transpuesta), algunas arquitecturas modernas mantienen una estructura con un camino de contracción con sucesivas capas de convolución y max-pooling, con el objetivo de sintetizar contexto de alto nivel de la imagen, seguido de un camino de expansión con sucesivas capas de convolución y convolución transpuesta para recuperar información posicional de bajo nivel. En este trabajo se utiliza una versión de

[5]La convolución transpuesta también es denominada Deconvolución. Este trabajo se atiene al nombre Convolución Transpuesta porque técnicamente esta capa realiza también una convolución, con lo que el nombre Deconvolución puede resultar engañoso.

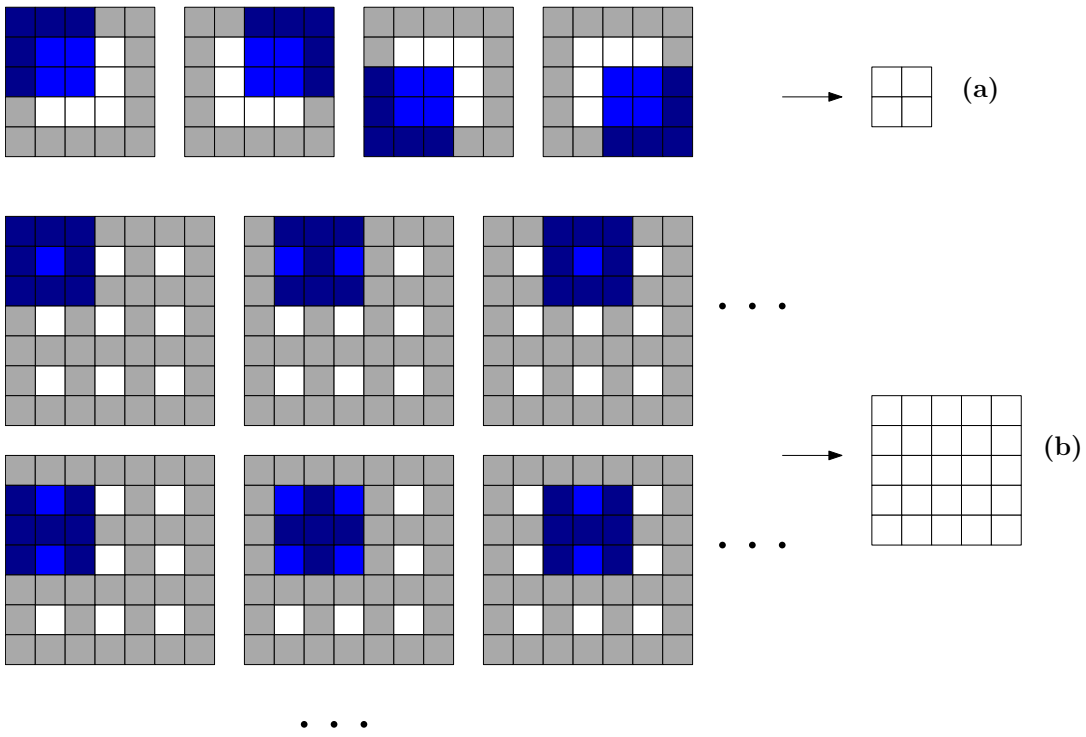


Figura 9: Convolución vs. Convolución Transpuesta con los mismos hiperparámetros: kernel = 3×3, padding = 1, stride = 2. **(a)** Convolución, stride = 2 produce saltos de a dos casillas. **(b)** Convolución Transpuesta, stride = 2 produce separación entre las casillas de entrada, aumentando la dimensión de la salida. Filas y columnas finales omitidas por simplicidad.

la arquitectura U-net[2], que es un ejemplo de este tipo de modelos. Esta arquitectura utiliza bloques de dos capas convolucionales con filtros de 3×3 y activación ReLU. La versión implementada, a diferencia de la del trabajo original, utiliza un padding de 1 para mantener las dimensiones de los volúmenes. Cada uno de estos bloques en el camino de contracción está unido al siguiente mediante una capa de max-pooling de 2×2, con lo que los bloques sucesivos tienen la mitad de alto y ancho, pero se duplican la cantidad de filtros. El camino de contracción tiene cuatro bloques. El camino de expansión tiene también cuatro bloques, conectados por convoluciones transpuestas. Cada bloque tiene la mitad de filtros que el anterior. El modelo original presentado por [2] concatena a la entrada de la primer capa de convolución de cada bloque la salida de la última capa de convolución del bloque correspondiente en el camino de contracción. Esto combina información de alto nivel del camino de expansión con información localizada del camino de contracción. Una modificación utilizada en este trabajo es reemplazar la concatenación por la suma elemento a elemento del volumen del camino de expansión con el volumen del de contracción. Esto reduce la cantidad de parámetros de la red sin impactar significativamente en su desempeño [15]. Al final del camino de expansión se agrega una capa convolucional con n filtros de 1×1, donde n es la cantidad de etiquetas de segmentación posibles. Notar que una capa convolucional de 1×1 con una entrada de profundidad m y n filtros es equivalente a aplicar una capa densa con entrada de tamaño m y salida de tamaño n a cada elemento, sobre todos los canales. Así se obtiene un vector de n posiciones para cada pixel de la imagen de entrada. Sobre la salida de esta última capa se utiliza la función de activación softmax, que produce un vector de probabilidades de cada etiqueta para cada pixel.

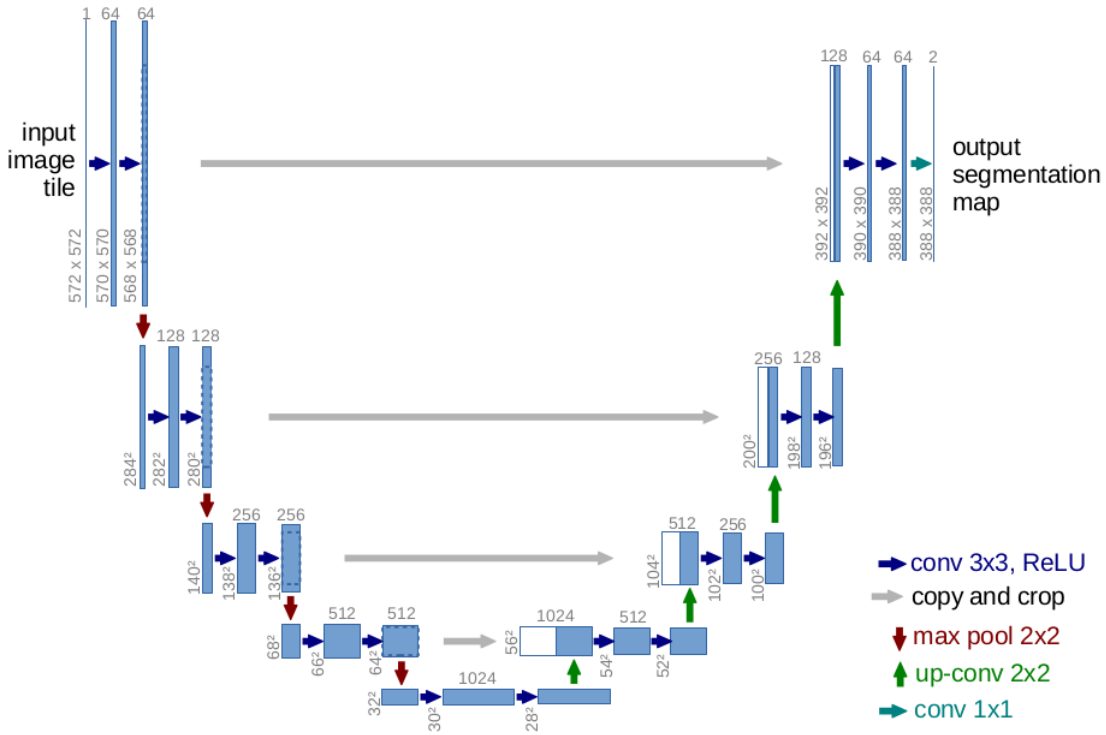


Figura 10: Esquema de arquitectura U-net. Cada bloque azul representa un volumen de atributos. La cantidad de canales (profundidad) está anotada en la parte superior de los bloques. El alto y ancho está indicado en la parte inferior izquierda de cada bloque. Los bloques blancos representan atributos copiados. Las flechas indican distintas operaciones. Figura tomada del trabajo original [2].

Redes totalmente convolucionales

El hecho de utilizar convoluciones de 1×1 en lugar de capas densas al final de la arquitectura, es un detalle fundamental para construir redes *totalmente convolucionales* (es decir, que no posean capas densas). La principal diferencia entre una red que posee capas densas y una red completamente convolucional radica en que la primera sólo puede recibir imágenes de un tamaño predeterminado (la capa densa hace que la arquitectura sea 'fija'). Sin embargo, las redes totalmente convolucionales pueden ser entrenadas con imágenes de un tamaño, y testeadas en imágenes de tamaño diferente (ver [16] para una descripción detallada sobre redes totalmente convolucionales para segmentación de imágenes). Esto abre la posibilidad utilizar parches (en lugar de imágenes completas) durante la etapa de entrenamiento, para entrenar modelos que luego son evaluados en imágenes completas. Volveremos sobre este punto en la sección 3.4, donde se brindarán más detalles sobre los métodos de muestreo de parches utilizados en este trabajo.

Batch Normalization

Un fenómeno frecuente en el entrenamiento de redes neuronales es que a medida que cada capa modifica sus parámetros, afecta la distribución probabilística de la entrada de la capa siguiente. Este efecto ralentiza considerablemente el entrenamiento de las redes, particularmente si los valores iniciales de los parámetros

no son favorables y el *learning rate* es muy alto.

Batch Normalization [17] es una técnica utilizada para mantener las activaciones de las capas intermedias homogéneas. Consiste en aplicar normalización a media cero y varianza unitaria a la salida de una capa. El efecto de esta normalización es que el modelo sea más robusto frente a la elección de hiperparámetros (*learning rate*, valores iniciales de los coeficientes, etc.) y acelera el proceso de aprendizaje. Se suele aplicar Batch Normalization luego de cada capa de Convolución.

Dropout

Otro problema típico en redes neuronales con gran número de parámetros es que su alto poder expresivo les permite sobreajustar fácilmente los datos de entrenamiento. Dropout [18] es un tipo de capa que durante el entrenamiento descarta aleatoriamente un porcentaje de los valores de un volumen intermedio para reducir la dependencia entre coeficientes. Resulta efectiva para reducir el sobreajuste del modelo. En este trabajo se utiliza un Dropout con 30 % de descarte en el bloque de mayor profundidad.

2.6. Convolución sobre imágenes médicas tridimensionales

Los modelos presentados reciben como entrada una imagen bidimensional de uno o más canales (típicamente con imágenes a color se utiliza un canal para cada una de las componentes RGB). Como se mencionó anteriormente, las imágenes médicas de MRI son volumétricas, con lo que las técnicas ya desarrolladas no se pueden aplicar directamente a este problema. Sin embargo, todos los conceptos se pueden extender a entradas tridimensionales. Una imagen se codifica con tres coordenadas espaciales y una coordenada que indica el canal. La misma representación se utiliza para las capas intermedias. Los filtros entonces tienen un kernel de tres dimensiones para cada canal de entrada. Las capas de max-pooling calculan el máximo en un entorno tridimensional.

Si varias modalidades de imágenes están disponibles (T1, T1 con Gadolinio, T2, FLAIR), se puede utilizar cada modalidad como un canal de entrada. Hacer esto, sin embargo, impone la restricción de que sólo se puede entrenar y evaluar la red con imágenes de las que se posean todas las modalidades incluidas en la arquitectura. El desarrollo de estrategias para permitir modalidades ausentes es un campo de investigación activo, como la generación de modalidades sintéticas [19] o adaptaciones en la arquitectura de la red [20]. Buscar este tipo de flexibilidad no es el objetivo de este trabajo con lo que se trabajará con la restricción de utilizar siempre el mismo conjunto de modalidades para una misma red.

2.7. Métricas de evaluación

Durante el transcurso de este capítulo se detallaron formas de generar predicciones en problemas de segmentación. Las estructuras de RNA expuestas anteriormente permiten proponer una enorme variedad de arquitecturas de redes profundas para una gran cantidad de problemas. Surge así la necesidad de cuantificar

el desempeño de los distintos modelos ya entrenados, evaluándolo sobre un conjunto de datos etiquetados nuevo para poder compararlos. Para eso se definen métricas que comparan el etiquetado dado por cierto (*ground truth*) con la predicción generada por el modelo.

La métrica más simple y ampliamente utilizada es la **accuracy**, que se define como la cantidad de pixeles etiquetados correctamente sobre la cantidad total de pixeles:

$$accuracy = \frac{\#pixeles\_correctos}{\#pixeles\_totales}. \tag{9}$$

La principal desventaja de esta métrica es que en presencia de clases desbalanceadas (donde la cantidad de pixeles de cada clase es muy distinta), un desempeño muy pobre en una clase poco frecuente puede pasar completamente desapercibido. En imágenes médicas, es habitual que las clases estén fuertemente desbalanceadas, dado que buena parte de las etiquetas son *background* y, particularmente en patologías, la cantidad de pixeles con etiquetas patológicas es muy pequeña.

Por esta razón se suelen utilizar otras métricas. Una de las más populares en segmentación de imágenes médicas es el **Coeficiente de Dice**. Para definir esta métrica primero es necesario hablar de los distintos tipos de aciertos y errores que se pueden cometer en el etiquetado.

En un problema de segmentación o clasificación binario (con sólo dos etiquetas), a una etiqueta se la considera positiva y a la otra negativa. Así, los aciertos se denominan True Positive (**TP**) si son sobre elementos positivos y True Negative (**TN**) en caso contrario, mientras que las predicciones negativas sobre elementos positivos se denominan False Negative (**FN**) y las predicciones positivas sobre elementos negativos se denominan False Positive (**FP**). En base a estos términos, el Coeficiente de Dice para un problema binario se define como

$$dice = \frac{2TP}{2TP + FN + FP}. \tag{10}$$

Una observación importante es que si el ground truth consta exclusivamente de etiquetas negativas y el modelo las etiqueta correctamente, entonces $TP = FP = FN = 0$, con lo que Dice no está bien definido. Dependiendo del contexto, puede ser conveniente que este coeficiente esté definido para cualquier valor, haciendo que tome el valor cero cuando no lo está.

Para escenarios multiclase, se puede calcular el Coeficiente de Dice para una clase i contra el resto, tomando como caso positivo la pertenencia a la clase i y como caso negativo la no pertenencia a dicha clase. Esto resulta en un coeficiente por cada clase. Una forma típica de condensar esta información en un solo valor es promediar los coeficientes de cada clase, aunque esto pierde información que puede resultar interesante. El coeficiente de Dice de la etiqueta background no se suele incluir en este promedio.

3. Modelos de segmentación propuestos

Los modelos de RNA presentados son una herramienta poderosa para aprender tareas de segmentación de imágenes. Sin embargo, su aplicación directa en el contexto descripto en la introducción de este trabajo (contexto multi-tarea donde las bases de datos poseen anotaciones disjuntas, ver Figura 3) resulta en una baja performance. A continuación, se proponen diversas alternativas desarrolladas en esta tesis, para mejorar el rendimiento de dichos modelos en el contexto multi-tarea.

3.1. Formalización del problema

Se cuenta con un conjunto de K datasets $\{D_k\}$, $1 \leq k \leq K$. Cada dataset $D_k = (x_i^k, y_i^k)$ está compuesto por pares, donde x_i^k es una imagen e y_i^k una segmentación que asigna a cada pixel de x_i^k una etiqueta $e \in E_k$. E_k representa el conjunto de etiquetas del dataset D_k. En este trabajo, asumiremos conjuntos de etiquetas disjuntos a excepción de la etiqueta *background*, presente en todos los E_k.

El problema entonces consiste en, a partir del conjunto $\{D_k\}$, entrenar un modelo que dada una nueva imagen $\hat{x}$, pueda etiquetarla con las etiquetas $\hat{E} = \bigcup_{k=1}^{K} E_k$.

Esta es la versión general del problema. En la sección 1.1 se restringió el alcance a problemas de segmentación de dos tareas. En este contexto, se proponen modelos que utilicen dos datasets de entrenamiento, uno con etiquetas anatómicas y otro con etiquetas de algún tipo de patología y tengan la capacidad de generar predicciones combinadas sobre imágenes nuevas.

3.2. Modelos preliminares

A continuación se proponen dos formas simples de abordar la segmentación multi-tarea que servirán como puntos de comparación para los modelos más complejos.

3.2.1. Modelo Naive

Un primer enfoque ingenuo es entrenar un modelo de la misma forma que se entrena para una sola tarea, pero utilizando imágenes de ambos datasets. Para esto se utilizará la arquitectura U-Net con entropía cruzada como función de costo (ver Sección 3.3.1). Como se mencionó en la introducción, esto conlleva el problema de que, por ejemplo, una estructura anatómica en el dataset patológico se indicará con una etiqueta distinta que en el dataset anatómico, con lo que el modelo tomará como error a una predicción correcta, o viceversa.

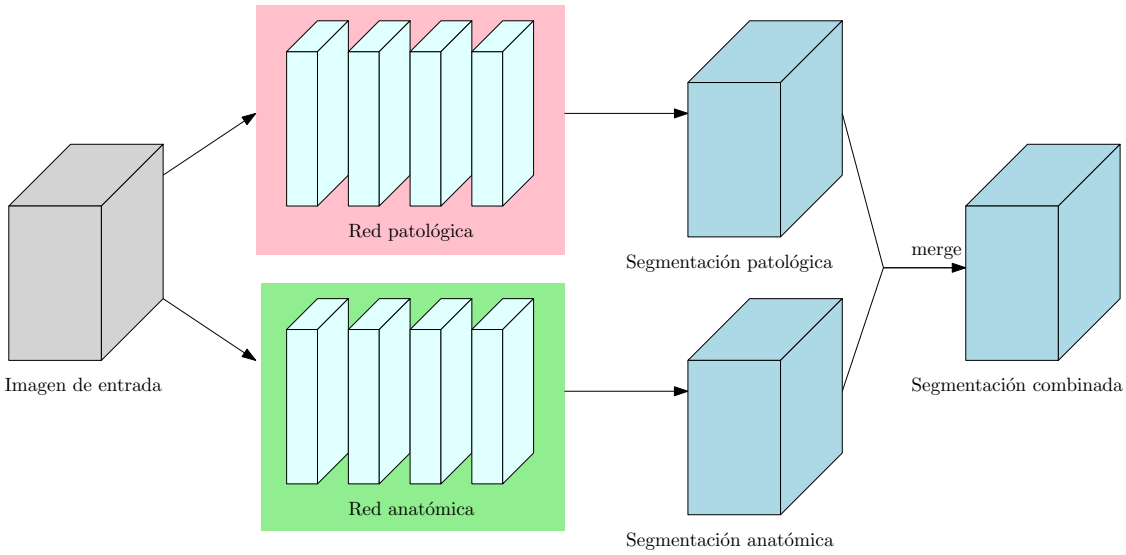


Figura 11: Modelo base para segmentación anatómica y patológica.

3.2.2. Modelo Base

En contraposición al Modelo Naive, se propone un modelo simple que promete resultados buenos basado en una suposición relativamente débil: que las etiquetas de un dataset prevalecen sobre las etiquetas del otro. En el caso de etiquetas anatómicas y etiquetas patológicas, por ejemplo, las segundas tienen mayor prioridad que las primeras. Esto se debe a que ante la presencia de una zona patológica, por más que ésta se presente sobre un tejido anatómico particular, se etiquetará como patología.

Asumiendo esto, el modelo propuesto consiste en entrenar una red para cada tarea y luego, para evaluar una nueva imagen, combinar las predicciones resultantes en una donde prevalezcan las etiquetas de mayor prioridad. Regresando al ejemplo de segmentación anatómica y patológica, esto se traduce en "pegar" la segmentación patológica encima de la anatómica. Un ejemplo de esta arquitectura puede verse en la Figura 11.

Si se cumple la suposición, se pueden esperar resultados relativamente buenos del Modelo Base, dado que el entrenamiento de cada tarea no se ve afectado por el entrenamiento de la otra, como sí sucede en el Modelo Naive.

La evidente desventaja es que son necesarias dos redes distintas. Esto implica más tiempo de entrenamiento, el doble de parámetros y, sobre todo, el doble de tiempo de evaluación, dado que es necesario evaluar la imagen de entrada sobre ambas redes.

Existe una desventaja algo más sutil de este modelo en comparación con el anterior, que se desprende también del hecho de que se utilicen dos redes distintas y está relacionada con la problemática multi-dominio: cada una de las dos redes es entrenada con imágenes provenientes de un solo dataset. Por lo general, las imágenes de un mismo dataset son obtenidas con el mismo equipamiento y reciben el mismo procesamiento, con lo que se pueden considerar del mismo dominio. Por esta razón, al validar el Modelo Base sobre imágenes procedentes de un tercer dominio es esperable que las redes tengan una menor capacidad de

generalización que el Modelo Naive, donde la única red es entrenada con imágenes de dos datasets distintos, y consecuentemente de dos dominios también distintos.

Teniendo en cuenta estas observaciones, el foco de las siguientes secciones estará en proponer un modelo que se base en el entrenamiento de una única red para segmentar ambas tareas, de modo que tenga la eficiencia del Modelo Naive, pero con un desempeño similar al Modelo Base.

Algunas alternativas posibles para conseguir este objetivo son:

- Modificaciones en la arquitectura: así como el Modelo Base plantea una arquitectura que combina dos redes, se puede idear una arquitectura de red especialmente diseñada para reducir el efecto multi-tarea. Los autores de [6] utilizan este enfoque para abordar el problema de multi-dominio. La desventaja de este tipo de técnicas es que el resultado es fuertemente dependiente de la elección de arquitectura y es difícil de trasladar a otras arquitecturas. Teniendo en cuenta que el diseño de arquitecturas de redes neuronales es un campo muy activo, atenerse a un diseño particular tiene el riesgo de que los resultados se vuelvan obsoletos en poco tiempo.

- Técnicas de muestreo: la estrategia de selección de las imágenes de entrenamiento y la extracción de parches de entrenamiento de las mismas impacta en el resultado final de la red. Esto abre la posibilidad de utilizar estrategias de muestreo que sean favorables para el entrenamiento multi-tarea.

- Funciones de costo: en la sección 2.3 se mencionó la función de entropía cruzada como ejemplo de función de costo para el entrenamiento de redes neuronales. Una posibilidad es diseñar una función de costo específicamente para entrenamiento multi-tarea. Esta opción, al igual que las técnicas de muestreo, tiene la ventaja de ser fácilmente trasladable de una arquitectura de red a otra.

3.3. Funciones de costo

3.3.1. Entropía cruzada

Como ya se mencionó, la función de costo más frecuentemente utilizada es la entropía cruzada. Esta función computa un coeficiente de diferencia entre dos vectores a y b (donde cada uno representa una distribución de probabilidad), y se define como:

$$h(a,b) = -\sum_{j=1}^{n} a_j \cdot log(b_j). \tag{11}$$

Para evaluar el error generado por la predicción de un voxel específico x_i de etiqueta y_i con C clases posibles, se computa la entropía cruzada entre el y_i-ésimo vector canónico n-dimensional $e^{(y_i)}$ y la predicción

del modelo. La entropía cruzada con respecto a un vector canónico puede simplificarse como:

$$\begin{aligned} h(e^{(y_i)}, f(x_i;\Theta)) &= -\sum_{j=1}^{C} e_j^{(y_i)} \cdot log(f(x_i;\Theta)_j) \\ &= -\sum_{j=1}^{C} \mathbb{1}_{y_i=j} \cdot log(f(x_i;\Theta)_j) \\ &= -log(f(x_i;\Theta)_{y_i}). \end{aligned} \tag{12}$$

Intuitivamente, para minimizar $-log(f(x_i;\Theta)_{y_i})$, la probabilidad de la clase correcta $f(x_i;\Theta)_{y_i}$ debe ser lo más cercana a 1 posible, luego la probabilidad de las clases incorrectas $\sum_{j=1,j\neq y_i}^{n} f(x_i;\Theta)_j = 1 - f(x_i;\Theta)_{y_i}$ será lo más cercana a 0 posible. La entropía cruzada H sobre una imagen con m voxeles $\{x_i\}_{1\leq i\leq m}$ y etiquetas $\{y_i\}_{1\leq i\leq m}$ se define como:

$$H = -\sum_{i=1}^{m} log(f(x_i;\Theta)_{y_i}). \tag{13}$$

3.3.2. Entropía Cruzada Selectiva

La entropía cruzada es mínima cuando la segmentación producida por el modelo es exactamente igual al ground truth. En escenarios multi-tarea, esto genera un problema ya que las etiquetas background de una tarea pueden pertenecer a zonas relevantes en otra tarea pero el modelo será penalizado si las reconoce correctamente. Se puede ilustrar el problema con el siguiente ejemplo: se desea entrenar un modelo para el reconocimiento de tejidos anatómicos y un tipo particular de patología. Se lo entrena con imágenes escogidas aleatoriamente entre ambos datasets. El mayor impedimento para que el modelo produzca predicciones combinadas es que ante una imagen con etiquetado patológico, las predicciones anatómicas del modelo serán consideradas como errores. La situación inversa, en cambio, no será un problema en este caso porque las imágenes con etiquetado anatómico provienen de pacientes sanos, con lo que no presentan patologías. Por este motivo, cualquier predicción patológica del modelo será correctamente considerada un error.

Con esto en mente, sería deseable una función de costo que, a diferencia de la entropía cruzada, no espere una segmentación igual al ground truth sino que sólo espere que las secciones relevantes de la segmentación generada lo sean. Para esto, se puede reinterpretar la etiqueta background de las segmentaciones patológicas como "cualquier etiqueta no perteneciente a esta tarea", y plantear una función de costo que la interprete como tal. Una forma de implementar esta idea es cambiar la etiqueta background de las imágenes con etiquetas patológicas por una etiqueta distintiva (-1, por ejemplo) y adaptar la función de entropía cruzada. Este trabajo propone la Entropía Cruzada Selectiva para un voxel i (HS_i):

$$HS_i = \begin{cases} -log(f(x_i;\Theta)_{y_i}) & \text{si } y_i \neq -1 \\ -log(\sum_{c=1}^{C} \mathbb{1}_{(\forall y_k) y_k \neq c} f(x_i;\Theta)_{y_i}) & \text{si } y_i = -1. \end{cases} \tag{14}$$

Si el voxel i tiene una etiqueta distinta de -1, se aplica la fórmula tradicional de entropía cruzada. Si tiene la etiqueta -1, se calcula $-log(s)$, donde $s = \sum_{c=1}^{C} \mathbb{1}_{(\forall y_k) y_k \neq c} f(x_i;\Theta)_{y_i}$ es la suma de los scores $f(x_i;\Theta)_{y_i}$

para las clases que no están presentes en la imagen[6]. Esto es equivalente a unificar todas las etiquetas no presentes en la imagen en una única etiqueta a la que se le asigna la suma de los puntajes de las clases unificadas y computar la entropía cruzada tradicional sobre el conjunto de etiquetas resultante.

Al buscar un conjunto de parámetros $\Theta^\star$ que minimice HS_i, cuando la etiqueta es distinta de -1 se busca minimizar $-log(f(x_i;\Theta)_{y_i})$, lo que maximiza el score de la clase correcta. Cuando la etiqueta es -1, se busca minimizar $-log(s)$, maximizando así la suma s de los puntajes asignados a las clases no presentes en la imagen. Esto muestra que la etiqueta -1 representa efectivamente cualquier etiqueta no presente en la imagen.

Para computar la Entropía Cruzada Selectiva sobre la imagen completa, simplemente se promedia el costo de cada voxel:

$$HS = \frac{1}{m}\sum_{i=1}^{m} HS_i. \tag{15}$$

3.3.3. Dice

El coeficiente de Dice puede adaptarse para ser utilizado como función de costo. La definición de Dice dada en la ecuación 10 fue:

$$dice = \frac{2TP}{2TP + FN + FP}.$$

La definición de TP, FN y FP está dada en base a las etiquetas de ground truth y las etiquetas extraídas de los vectores de probabilidad del modelo, con lo que parte de la información de los propios vectores se pierde. Se desea una función diferenciable basada en el Coeficiente de Dice que pueda ser utilizada como función de costo para el entrenamiento. Una posibilidad es definir TP_d, FP_d y FN_d[7] para la clase c de la siguiente forma:

- TP_d: suma de las probabilidades asignadas a la clase c para todos los elementos de la clase c:

$$TP_d = \sum_{i=1}^{m} \mathbb{1}_{y_i=c} \cdot f(x_i;\Theta)_c. \tag{16}$$

- FP_d: suma de las probabilidades asignadas a la clase c para todos los elementos no pertenecientes a la clase c:

$$FP_d = \sum_{i=1}^{m} \mathbb{1}_{y_i\neq c} \cdot f(x_i;\Theta)_c. \tag{17}$$

- FN_d: suma de las probabilidades asignadas a las clases distintas de c para los elementos de la clase c:

$$FN_d = \sum_{i=1}^{m} \mathbb{1}_{y_i=c} \cdot (1 - f(x_i;\Theta)_c). \tag{18}$$

[6] $\mathbb{1}_q$ representa la función indicadora, que vale 1 si la condición q es cierta y 0 en caso contrario.

[7] Aquí d representa "difuso", en referencia a la *lógica difusa*.

Notar que si se reemplazan los vectores de probabilidades por los canónicos de la clase con mayor puntaje, estas definiciones son equivalentes a las anteriores.

Utilizando estos conceptos, se puede definir un Coeficiente de Dice Difuso para la clase c:

$$CDD_c = \frac{2TP_d}{2TP_d + FN_d + FP_d} = \frac{2\sum_{i=1}^{m} \mathbb{1}_{y_i=c} \cdot f(x_i;\Theta)_c}{\sum_{i=1}^{m} f(x_i;\Theta)_c + \sum_{i=1}^{m} \mathbb{1}_{y_i=c}} \tag{19}$$

Es muy conveniente que CDD_c esté bien definido en todo el dominio, con lo que se suele sumar un valor ϵ pequeño al denominador para que no valga 0. Con este coeficiente, se puede definir la función de costo de Dice en base al promedio p de los coeficientes difusos para las clases que no son background. Este promedio está entre 0 y 1 y crece a medida que mejora el desempeño del modelo. Para utilizarlo como función de costo, es deseable que sea decreciente. Se puede utilizar $1 - p$ para lograr esto:

$$D = 1 - \frac{1}{C}\sum_{c=1}^{C} CDD_c \tag{20}$$

Esta función es diferenciable y decrece a medida que mejora la performance del modelo, lo que la hace una posible función de costo por ser minimizable mediante cualquier algoritmo basado en gradientes. Si bien D se puede utilizar para el entrenamiento de una red, los autores de [15] muestran para dos tipos distintos de lesiones que el desempeño de una red entrenada con esta función es ligeramente peor que el de una entrenada con entropía cruzada, incluso utilizando el coeficiente de dice como métrica de evaluación. Una ventaja que tiene, sin embargo, es que es bastante robusta a desbalances entre las clases.

3.3.4. Dice Selectivo

Este trabajo propone un modificación simple a la función de costo de Dice para hacerla más propicia al caso multi-tarea. En lugar de promediar el coeficiente de Dice sobre todas las clases, se lo hará sobre las clases presentes en el ground truth de la imagen que se está evaluando. Se llamará Dice Selectivo a esta función:

$$DS = 1 - \frac{\sum_{c=1}^{C} \mathbb{1}_{(\exists y_i) y_i=c} CDD_c}{\sum_{c=1}^{C} \mathbb{1}_{(\exists y_i) y_i=c}} \tag{21}$$

La motivación detrás de esta función de costo es que al entrenar sobre una imagen etiquetada en función de una tarea, se ignorarán las predicciones sobre etiquetas que pertenezcan a otras tareas, con lo que no se penalizará equivocadamente a la red.

3.4. Estrategias de muestreo

La forma de elegir las imágenes de entrenamiento tiene un impacto significativo en la convergencia de la red. Por ejemplo, tomar imágenes aleatoriamente y entrenar la red con una imagen entera por vez no es

conveniente por dos razones:

- La utilización de una única imagen resulta en una pobre estimación del gradiente de la función de costo con respecto a los parámetros de la red. Como se discutió en la sección 2.3, la correcta aplicación del método del Descenso por el Gradiente Estocástico utiliza varias imágenes por vez, en grupos denominados *batches*, para obtener una mejor estimación del gradiente sin necesidad de evaluar todo el dataset.

- Una imagen de MRI cerebral esta compuesta por un volumen del orden de $200 \times 200 \times 200$ voxeles. El proceso de evaluar un volumen completo con una arquitectura de red como la propuesta requiere más memoria que la disponible en GPUs modernas.

Es decir que por un lado, es conveniente utilizar varias imágenes a la vez y por otro, evaluar una sola ya es impracticable. Esta situación contradictoria se suele solucionar realizando muestreo por parches, es decir, formando batches compuestos por pequeñas porciones de varias imágenes. Tal como fue discutido en la Sección 2.5, el hecho de que nuestra arquitectura sea totalmente convolucional, hace posible entrenar la misma con parches de imagen y luego evaluarla en imágenes completas sin mayores esfuerzos. El tamaño y la cantidad de parches por batch se determina en función de la capacidad de la GPU utilizada.

La generación de parches a partir de imágenes también es un proceso determinante en el entrenamiento. Se suele procurar que las porciones tomadas contengan información relevante. El procedimiento utilizado en este trabajo para extraer parches de imágenes de un conjunto de datasets de entrenamiento es el siguiente:

1. Elegir un dataset, equiprobablemente.
2. Elegir una imagen dentro de ese dataset, equiprobablemente.
3. Elegir una etiqueta relevante (no background ni -1, si hubiera) entre las presentes en la imagen, equiprobablemente.
4. Entre todos los voxeles de la imagen que pertenezcan a la etiqueta elegida, tomar uno cualquiera, equiprobablemente.
5. Extraer un parche de la imagen, que contenga ese voxel.

Esto fuerza la presencia de información relevante en cada parche, con una distribución relativamente homogénea. La decisión de omitir la etiqueta background del muestreo se basa en que es la que genera mayor conflicto en el entrenamiento multi-tarea. Este tipo de muestreo mostró ser particularmente efectivo en tareas con gran desbalance entre clases, principalmente en lesiones pequeñas como Hiperintensidades de Materia Blanca, donde un muestreo más naive no permite que la red aprenda a reconocerlas. El hecho de no muestrear deliberadamente background no significa que la red nunca sea expuesta a porciones de imágenes pertenecientes a esta etiqueta. Principalmente al muestrear lesiones, dado que el tamaño de los parches es típicamente mayor que el tamaño de las lesiones, la red verá sectores de background.

Otra razón para no muestrear parches a partir de la etiqueta background es que los coeficientes de Dice no tienen ningún efecto sobre parches donde la única etiqueta presente es background: no hay True Positives, con lo que la función evalúa a cero y su derivada también.

Tampoco se muestrean parches a partir de la etiqueta -1 dado que la Entropía Cruzada Selectiva (el único escenario donde se utiliza esta etiqueta) tampoco tiene ningún efecto sobre parches que sólo contengan esa etiqueta.

3.5. Preprocesamiento de datos

Como en cualquier problema de aprendizaje automático, la preparación previa de los datos es un componente fundamental del proceso. Se detallarán algunos de los procesos aplicados a las imágenes crudas para mejorar el desempeño.

Resampleo

Una característica notoria de las redes convolucionales es que son sensibles a la escala y orientación de los objetos en la imagen de entrada. En la medida de lo posible, es conveniente que todas las imágenes de entrenamiento y evaluación tengan la misma escala. Cada dataset utiliza su propia escala para almacenar las imágenes, con lo que es necesario un proceso de estandarización. Para esto, se redimensionaron todas las imágenes para que cada voxel referencie un cubo de $1mm \times 1mm \times 1mm$ en el cerebro representado. Además es necesario que todas las imágenes tengan la misma orientación, con lo que en ocasiones puede hacer falta rotarlas.

Al resamplear un volumen, existen distintas técnicas de interpolación para aproximar los valores de los voxeles del volumen resultante. Es necesario resamplear tanto las imágenes de resonancia como las etiquetas de segmentación. Para el resampleo de las imágenes en sí, se obtienen resultados satisfactorios utilizando interpolación lineal. Para el resampleo de las etiquetas de segmentación, la interpolación lineal no es una buena opción porque produce valores en un espacio continuo, mientras que las etiquetas tienen valores discretos. Una opción posible es tomar el valor la etiqueta del voxel original más cercano (estrategia *nearest neighbour*). Esta técnica produce buenos resultados cuando no es necesario rotar las imágenes. Al aplicar tanto escalado como rotaciones, *nearest neighbour* produce *artifacts* bastante evidentes en las imágenes. Para estos casos se obtienen mejores resultados con una interpolación gaussiana sobre cada etiqueta.

Región de interés

Para facilitar el proceso de segmentación de una imagen médica, es usual identificar la Región de Interés (RoI, por *Region of Interest*) de la imagen y anular todos los voxeles que se encuentren fuera de ésta. En el caso de imágenes de MRI cerebrales, la RoI es la región de la imagen perteneciente al cerebro, descartando así el cráneo, cuello y entornos que no sean relevantes para la segmentación. El proceso de identificación de esta

región de interés se llama *brain stripping*. Se utilizará la herramienta ROBEX (RObust Brain EXtraction) [21] para identificar la sección correspondiente al cerebro.

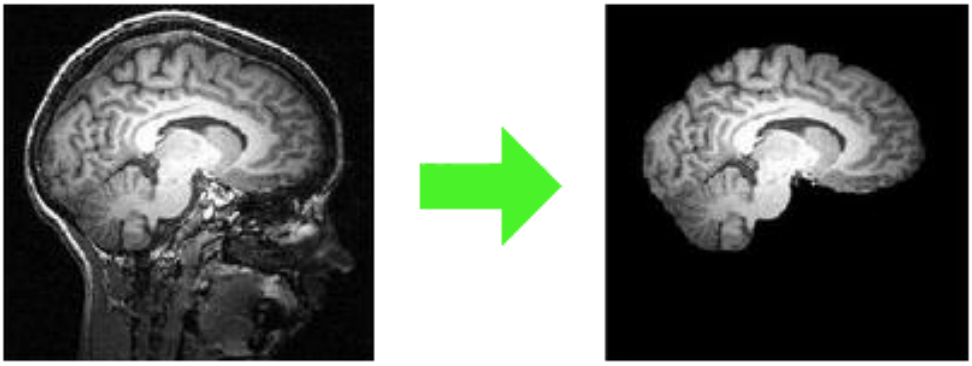


Figura 12: Ejemplo de *brain stripping* sobre un corte sagital. Imagen tomada del sitio de ROBEX.

Normalización

Otra práctica común en cualquier proceso de segmentación o clasificación de imágenes es la normalización por z-score de las imágenes. Consiste en sustraer la media μ de la intensidad de los voxeles y dividir por su desvío estándar σ:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{22}$$

El proceso de normalización se realiza solamente dentro de la RoI. Luego se fijan los valores fuera de la RoI a un valor fijo inferior a los valores presentes en la RoI normalizada, para reducir las posibilidades de confundir al modelo. En este trabajo se utiliza el valor -4.

3.6. Multiplicación de datos

Debido al costo humano de la segmentación manual de imágenes, los datasets de imágenes médicas utilizados en este trabajo no superan las 100 imágenes. El entrenamiento con datasets reducidos es muy propenso a producir *sobreajuste*, dado que el modelo no tiene tanta necesidad de generalizar. Para compensar la reducida cantidad de imágenes, se realizará un proceso de multiplicación de datos sobre los parches:

- Aprovechar la simetría derecha-izquierda del cerebro para invertir aleatoriamente la mitad de los parches en este sentido.
- Añadir ruido gaussiano por pixel, con media cero y varianza baja.

Si bien existen técnicas más avanzadas de multiplicación de datos, éstas acarrean un mayor costo computacional y no son el foco de este trabajo.

4. Infraestructura de entrenamiento y evaluación

La implementación de la infraestructura y el modelo de red neuronal se realizó íntegramente en Python, utilizando la librería de aprendizaje profundo Keras [22], con TensorFlow [23] como backend. Esto permite ejecutar con facilidad la red tanto en GPU como en CPU. Para el manejo de imágenes médicas se utilizaron las librerías NiBabel [24], NiLearn [25] y SimpleITK [26].

El mayor costo computacional asociado al entrenamiento de un modelo de red neuronal es indudablemente la evaluación de la propia red sobre cada batch de entrenamiento para computar los gradientes con respecto a la función de error. Sin embargo, el costo de preprocesamiento del dataset, la carga de imágenes de disco y la generación de batches es también significativo. Por este motivo se tomaron las siguientes decisiones de diseño:

- El preprocesamiento del dataset, incluyendo el resampleo de imágenes, brain stripping y normalización se realiza antes del entrenamiento, y las imágenes preprocesadas se almacenan en disco en forma de numpy arrays sin compresión para minimizar el tiempo de acceso a ellas.
- Los accesos a disco durante el aprendizaje se realizan en un thread separado que carga imágenes y las almacena en memoria en una cola compartida, para no interferir con el entrenamiento de la red. Dado que cada dataset completo ocupa un espacio en memoria del orden de varios GigaBytes, es impráctico mantenerlos completos en memoria durante el entrenamiento. Por esto se utiliza un pool con un subconjunto de P imágenes de las cuales se toman parches. Cada T parches, se reemplaza la imagen que lleva más tiempo en el pool por una imagen nueva. El reemplazo gradual del pool tiene como objetivo dispersar en el tiempo los accesos a disco.
- Realizar la extracción de parches y multiplicación de datos en otro thread separado no mostró mejoras en la velocidad de ejecución, por lo que se decidió realizar en el thread principal.
- El proceso de generación de batches se realiza íntegramente en CPU, mientras que el entrenamiento de la red se realiza en GPU.
- El procesamiento de imágenes de evaluación se realiza sobre el volumen completo, en CPU. Si bien esto es un orden de magnitud más lento, la memoria de una GPU actual no es suficiente para evaluar un volumen completo (con la arquitectura utilizada, una imagen completa requiere de unos 15 o 20 GB de memoria, mientras que una GPU de última generación alcanza los 12 GB). En CPU, en cambio, es factible disponer de tal cantidad de memoria.

Para el entrenamiento de las redes se utilizó una computadora con un procesador Intel i7 de séptima generación con 8 núcleos, una GPU NVIDIA M1200 de 4GB y un disco de estado sólido para acelerar el acceso a las imágenes.

Los datasets de entrenamiento (MRBrainS13, MRBrainS17, IBSR, BrainWeb y BraTS12) fueron particionados en un 80 % de imágenes de entrenamiento y un 20 % de validación, mientras que los datasets de

evaluación (MRBrainS18 y TumorSim) fueron utilizados íntegramente para validar. El entrenamiento de los modelos se realizó en épocas de 200 batches. Se utilizaron batches de 7 parches de $32 \times 32 \times 32$ voxels para aprovechar al máximo la capacidad de la GPU.

Se computaron métricas sobre 10 parches de $128 \times 128 \times 128$[8] tomados de la partición de validación de los datasets de entrenamiento, al finalizar cada época. Se entrenó cada modelo hasta que la función de costo sobre los datos de validación se estabilizó, manteniendo registro del conjunto de parámetros que había generado el costo mínimo, para ser utilizado en el modelo final.

Notar que si una función de costo no es buena para evaluar la calidad de una predicción en un contexto multi-tarea, no sólo generará dificultades para optimizar el desempeño del modelo sino también para determinar cuál fue el conjunto de parámetros que produjo los mejores resultados.

[8]Se tomaron parches de este tamaño en vez de imágenes completas para validar porque es el máximo volumen que permitía ser evaluado en GPU.

5. Escenarios multi-tarea

Para evaluar la efectividad de los distintos modelos propuestos se contemplaron dos escenarios multi-tarea posibles:

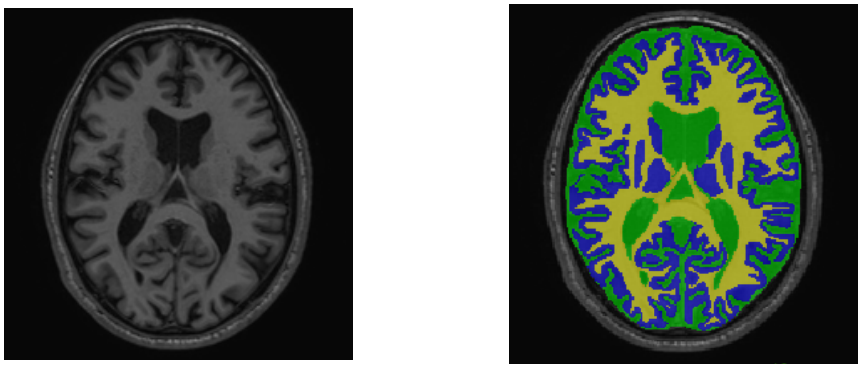
- Estructuras anatómicas e Hiperintensidades de Materia Blanca.
- Estructuras anatómicas y Tumores.

La restricción principal para la elección de estos escenarios fue la disponibilidad de datasets con la segmentación combinada de ambas tareas, necesarios para poder evaluar los modelos entrenados.

5.1. Estructuras anatómicas

Existen tres estructuras anatómicas principales que se suelen segmentar en imágenes cerebrales:

- Líquido Cefalorraquídeo (**CSF**, por *Cerebrospinal Fluid*): es un fluido que embebe al cerebro y cumple funciones de protección, regulación de la presión, transporte de nutrientes y sustancias, entre otras cosas.
- Materia blanca (**WM**, por *White Matter*): es la parte del cerebro que contiene principalmente fibras nerviosas o axones. El color blanco está dado por la abundante presencia de grasas en la mielina que recubre los axones. Las grasas son resaltadas por las resonancias de modalidad T1, por lo que suelen identificarse con zonas de mayor brillo en imágenes médicas de este tipo.
- Materia Gris (**GM**, por *Grey Matter*): en contraposición con la materia blanca, contiene principalmente los núcleos neuronales.



(a) Resonancia magnética cerebral modalidad T1.

(b) Etiquetas anatómicas: CSF (verde), GM (azul) y WM (amarillo).

Figura 13: Ejemplo de segmentación anatómica tomado del dataset MRBrainS13.

Datasets anatómicos

Se utilizaron tres datasets con segmentaciones de estructuras anatómicas:

- **IBSR (v2)**[27]: dataset del Internet Brain Segmentation Repository. Consta de 18 imágenes en modalidad T1 con sus respectivas segmentaciones anatómicas en CSF, GM y WM.
- **MRBrainS13**[28]: dataset del MRBrainS13 challenge (Magnetic Resonance Brain Segmentation 2013), una competencia organizada por la Medical Image Computing and Computer Assisted Intervention Society (MICCAI Society). El dataset consiste de 5 imágenes en modalidades T1, IR y FLAIR con sus correspondientes segmentaciones anatómicas en CSF, GM y WM.
- **BrainWeb**[29]: dataset sintético de 20 imágenes en modalidad T1 segmentado con etiquetas anatómicas CSF, GM, WM, grasa, músculo, piel, cráneo, vasos sanguíneos, tejido conectivo, duramadre y médula. Los vasos sanguíneos se unificaron con CSF y el resto de las etiquetas no deseadas, con background. De estas imágenes se utilizaron solo 15 (ver TumorSim a continuación).

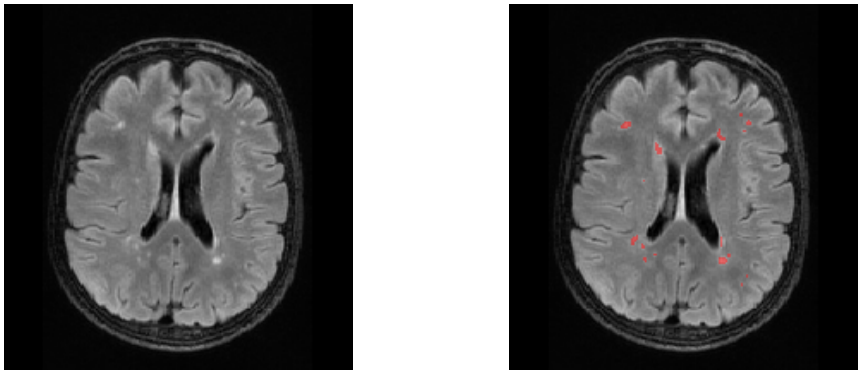
Puede llamar la atención la reducida cantidad de imágenes de cada dataset. En otras tareas de aprendizaje basado en imágenes el tamaño de los datasets utilizados suele ser varios órdenes de magnitud mayor. Este problema se ve compensado por dos factores:

- Al ser imágenes volumétricas, la cantidad de parches distintos que se pueden extraer de cada imagen es alta. Esto hace que aunque la cantidad de imágenes sea poca, la cantidad de datos con los que se alimenta a la red es más de la que parece.
- Particularmente en tareas anatómicas, la variabilidad de los datos y de los resultados esperados es baja en comparación con otras tareas, tanto en imágenes médicas como en otros tipos de imágenes. Esto es porque los cerebros sanos son relativamente similares entre sí, al igual que las segmentaciones anatómicas correspondientes. Esto hace que aún con un número reducido de imágenes de entrenamiento el modelo sea expuesto a muestras suficientemente representativas. En los escenarios de segmentación patológica que se describen a continuación, en cambio, es deseable que la cantidad de datos sea mayor dado que las formas en las que se pueden presentar las estructuras a reconocer es más variada.

5.2. Hiperintensidades de materia blanca

Las Hiperintensidades de Materia Blanca (**WMH**, por *White Matter Hyperintensities*) son lesiones de Materia Blanca producto de la desmielinización de los axones. Aparecen con la edad y también se pueden observar en algunos desórdenes neurológicos y enfermedades psiquiátricas. Se llaman Hiperintensidades por verse particularmente brillantes en imágenes de resonancia magnética de tipo T2 y FLAIR.

Las lesiones de WMH tienen la particularidad de ser muy pequeñas, como se puede observar en la Figura 14. Esto dificulta en gran medida el aprendizaje y puede traer problemas al combinarla con etiquetas que abarcan mucho espacio, como las anatómicas.



(a) Resonancia modalidad FLAIR. **(b)** Segmentación de WMH (rojo).

Figura 14: Ejemplo de segmentación de WMH tomado del dataset MRBrainS17.

Dataset WMH

- **MRBrainS17**[30]: dataset de una competencia organizada por MICCAI en 2017, enfocada en segmentación de WMH. El dataset consta de 60 imágenes en modalidades T1 y FLAIR, con sus correspondientes segmentaciones de WMH y otras patologías. Para este trabajo, las otras patologías fueron unificadas con la etiqueta background.

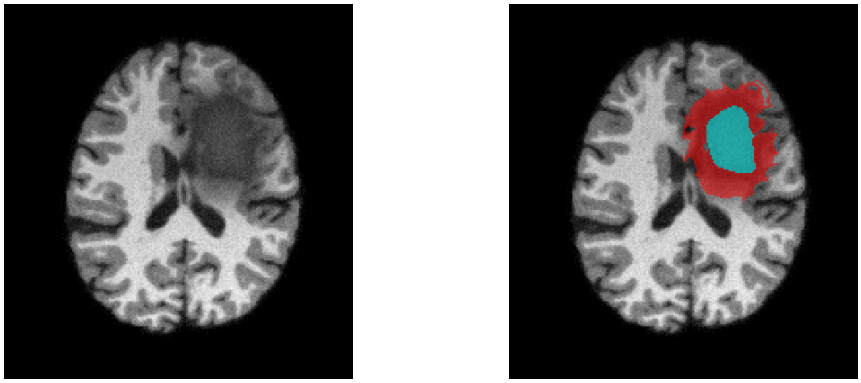
Dataset combinado de estructuras anatómicas y WMH

- **MRBrainS18**[31]: otra competencia de MICCAI de 2018, enfocada en la segmentación de tejidos anatómicos y otras estructuras cerebrales, incluyendo WMH. El dataset consta de 7 imágenes en modalidades T1, IR y FLAIR. Para cada imagen se tiene la segmentación en materia gris cortical, ganglios basales, WM, WMH, CSF extracerebral, ventrículos, cerebelo, tallo cerebral, infarto y otras. Para obtener la segmentación en las tres estructuras anatómicas requeridas y WMH, se unificaron la materia gris cortical y los ganglios basales en GM y el CSF extracerebral y los ventrículos en CSF, siguiendo las indicaciones de MICCAI. Las etiquetas restantes se ignoraron en la evaluación.

5.3. Tumores

Un tumor es una multiplicación anormal de una célula o un conjunto de células. Existen muchas variedades de tumores benignos y malignos. Los tumores contemplados en este trabajo son los Gliomas: tumores cerebrales producto de la multiplicación de células gliales, que son células auxiliares del cerebro que, a diferencia de las neuronas, no forman contactos sinápticos. Son visibles en varias modalidades de MRI, incluyendo T1, T2 y T1 con contraste de Gadolinio. Usualmente, en torno a un tumor cerebral se produce una acumulación de líquidos denominada edema.

Como se puede ver en la Figura 15, los tumores son considerablemente más grandes que las lesiones de WMH, con lo que el desbalance de etiquetas en este caso será menor.



(a) Resonancia magnética cerebral modalidad T1.

(b) Segmentación de Edema (rojo) y Tumor (verde).

Figura 15: Ejemplo de segmentación de Tumores tomado del dataset BraTS12.

Dataset de tumores

- **BraTS12**[32]: dataset sintético de la competencia MICCAI BraTS 2012 (Brain Tumor Segmentation), consta de 50 imágenes en modalidades T1, T1 con contraste de Gadolinio, T2 y FLAIR, con sus correspondientes segmentaciones en Tumores y Edema.

Dataset combinado de estructuras anatómicas y tumores

- **TumorSim**: para obtener un dataset combinado de etiquetas anatómicas y tumorales, se utilizó el simulador de tumores TumorSim[33] para generar imágenes sintéticas con tumores, basadas en los mapas de probabilidades de las etiquetas anatómicas correspondientes a 5 imágenes de BrainWeb (de las 20 imágenes inicialmente provistas por BrainWeb, 15 se utilizaron para entrenamiento, y 5 fueron conservadas para ser luego utilizadas en conjunto con TumorSim, en la generación de imágenes combinadas con etiquetas anatómicas y patológicas). TumorSim simula el crecimiento de un tumor a partir de una semilla dada, en varias iteraciones. El efecto de crecimiento tumoral está sujeto a un conjunto de parámetros de configuración. A partir de eso, genera imágenes en modalidades T1, T1 con contraste de Gadolinio, T2 y FLAIR con su respectiva segmentación de estructuras anatómicas, tumor y edema. Notar que el simulador no utiliza directamente las imágenes sino los mapas de probabilidades, con lo que no hay garantía de que el proceso de síntesis sea similar al utilizado por BrainWeb. Utilizando 4 semillas distintas para cada imagen, se generaron 20 imágenes. Cabe mencionar que el dataset **BraTS12** también fue generado utilizando este simulador.

No fue posible acceder a un dataset real con etiquetas combinadas de tumores y estructuras anatómicas, con lo que el uso de TumorSim para generarlo fue la única opción. Considerando esto, se decidió utilizar exclusivamente datasets sintéticos en la experimentación de Tumores porque se espera que las diferencias de dominio entre ellos sean menores que entre un dataset sintético y uno real.

6. Resultados

Este trabajo utiliza siete conjuntos de datos distintos, como se describió en la sección anterior: MRBrainS13, MRBrainS17, MRBrainS18, IBSR, BrainWeb, BraTS12 y TumorSim. Cada uno de ellos fue obtenido con una metodología y equipamiento médico particular, o generado de forma sintética. Algunos fueron recopilados por la misma organización y preprocesados de forma similar, otros no. Un tema recurrente de discusión en durante las secciones anteriores es el efecto multi-dominio. Si las características de las imágenes de los distintos datasets difieren mucho, es probable que los modelos entrenados sobre uno de los datasets no den buenos resultados al evaluarlos sobre otros datasets.

Los datasets fueron normalizados por z-score (ver Sección 3.5) para reducir las diferencias entre imágenes. Surge la pregunta de si luego de este proceso siguen existiendo diferencias intrínsecas entre las imágenes de distintos datasets, o si los distintos dominios en efecto se homogeneizaron.

Una forma de cuantificar la diferencia entre las características de dos imágenes es comparar los histogramas de intensidades de los voxeles de cada una. Una particularidad de este enfoque es que el histograma descarta la información posicional. Esto resulta conveniente ya que interesa más la naturaleza de la imagen en sí que discrepancias espaciales. Esto es, estudiar qué intensidades de pixels son más frecuentes en cada dataset permite un análisis de alto nivel de las diferencias existentes.

La **Divergencia de Jensen-Shannon**[34] es una métrica que se utiliza para medir la similitud entre dos distribuciones de probabilidad P y Q. Se define como:

$$JS(P,Q) = \frac{H(P,M) + H(M,Q)}{2} \tag{23}$$

Donde $M = (P+Q)/2$ y H es la función de entropía.

Para cuantificar la diferencia de dominio adentro de un mismo dataset y entre datasets, se propuso la siguiente métrica Δ_{JS}, basada en la divergencia de Jensen-Shannon. La diferencia dentro de un mismo dataset $D = \{x_1 \dots x_n\}$ es el promedio de las divergencias de Jensen-Shannon entre pares de imágenes distintas de dicho dataset[9]:

$$\Delta_{JS}(D) = \frac{1}{n(n-1)} \sum_{i=1,j=1,i\neq j}^{n} JS(hist(x_i), hist(x_j)) \tag{24}$$

donde $hist(x)$ es el histograma de intensidades de voxeles de x. La diferencia entre dos datasets distintos $D_1 = \{x_1^1 \dots x_n^1\}$ y $D_2 = \{x_1^2 \dots x_m^2\}$ es el promedio de las distancias entre cada imagen de D_1 y cada imagen de D_2:

$$\Delta_{JS}(D_1, D_2) = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} JS(hist(x_i^1), hist(x_j^2)) \tag{25}$$

Se computaron las divergencias Δ_{JS} entre todos los pares de datasets, únicamente sobre la modalidad T1 de cada imagen por ser la única modalidad presente en todos los datasets. Los resultados se pueden ver en la Figura 16. Se puede observar que las distancias entre los datasets de MRBrainS son comparativamente

[9]Existe una generalización de la divergencia de Jensen-Shannon para más de dos distribuciones que podría haber sido aplicada en este caso, pero se decidió utilizar esta métrica por parecerse más a la utilizada entre datasets distintos.

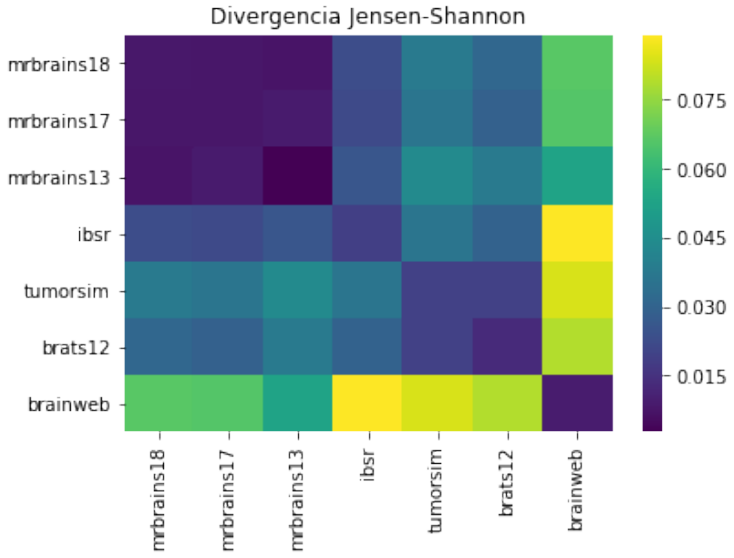


Figura 16: Divergencia ΔJS entre los datasets utilizados en la experimentación: MRBrainS18, MRBrainS17, MRBrainS13, IBSR, TumorSim, BraTS12 y BrainWeb.

pequeñas. Esto se explica porque provienen de la misma organización y probablemente fueron capturados utilizando equipamiento y configuración de parámetros similares.

Lo mismo sucede entre los datasets sintéticos TumorSim y BraTS12, que fueron generados con el mismo simulador, si bien presumiblemente utilizando distintos parámetros de configuración. Estos dos resultados suman confianza a la verosimilitud de la métrica Δ_{JS}.

Por otro lado, es llamativa la diferencia entre el dataset BrainWeb y el resto, lo que contradice la hipótesis de que los datasets sintéticos serían más similares entre sí. Es evidente que el proceso de síntesis de imágenes utilizado por BrainWeb es distinto al utilizado por TumorSim. Una diferencia tan marcada puede presentar un problema para el entrenamiento y la evaluación de los modelos. Sin embargo, también puede presentar una ventaja para los modelos que utilizan una sola red, dado que la red es expuesta a imágenes de dos dominios distintos, potencialmente permitiendo mayor generalización.

La diferencia entre los datasets de MRBrainS y los de TumorSim y BraTS12 es mayor que las diferencias dentro de cada uno de estos grupos. Esto puede indicar una diferencia de dominio mayor, aunque puede verse afectada también por el hecho de que las imágenes del segundo grupo contienen tumores. Éstos son claramente visibles y ocupan un espacio considerable en las imágenes, con lo que impactan en cierta medida en los histogramas de intensidades, pudiendo verse reflejados en las métricas Δ_{JS}.

Un aspecto más que puede resultar relevante es que la autosimilitud de MRBrainS13 es mayor que la del resto de los datasets. El valor de la divergencia en este caso da entre 4 y 8 veces menos que para el resto. Esto, combinado con el hecho de que MRBrainS13 tiene tan sólo 5 imágenes (y sólo 4 de entrenamiento) presenta el riesgo de que la capacidad de generalización de una red entrenada exclusivamente sobre este dataset sea muy pobre.

6.1. Escenarios mono-tarea

Antes de estudiar el desempeño de los modelos propuestos en escenarios multi-tarea, cabe estudiar cómo se comportan en escenarios de una sola tarea. Sin buenos resultados para una tarea tiene poco sentido esperar buenos resultados para más de una. Se mostrará el desempeño de las distintas funciones de costo propuestas: entropía cruzada, dice y dice selectivo; sobre un dataset de cada tarea: IBSR para segmentación anatómica (Sección 5.1), MRBrainS17 para WMH (sección 5.2) y BraTS12 para tumores (Sección 5.3). No se tuvo en cuenta la función de entropía cruzada selectiva, dado que con una sola tarea su definición es idéntica a la entropía cruzada tradicional.

El primer paso en el proceso de evaluación y comparación de los modelos propuestos es determinar valores iniciales para los hiperparámetros de cada uno. Se estudió la convergencia de las distintas redes en función de los valores de cada hiperparámetro. Se presenta a continuación una visión general de los parámetros utilizados.

- Optimizador: como se discutió en la Sección 2.3, el optimizador más comunmente utilizado en la actualidad y que suele producir los mejores resultados es Adam. Una ventaja adicional de Adam es que es relativamente robusto respecto de la elección inicial de sus hiperparámetros. Se utilizó la configuración sugerida por el artículo original [10], que también es la configuración por defecto de las implementaciones de Adam en gran parte de las bibliotecas de Aprendizaje Profundo como TensorFlow, Keras, Lasagne, Torch o Caffe.

- Inicialización de los pesos de la red: se utilizó una inicialización aleatoria de los parámetros de cada capa tomados de una distribución normal truncada con media cero y desvío estándar $\sigma = \sqrt{\frac{2}{\#inputs}}$, donde $\#inputs$ es la dimensión de entrada de la capa correspondiente. Esta inicialización garantiza una varianza inicial homogénea de las activaciones de las capas intermedias durante el entrenamiento utilizando ReLU como función de activación[35]. De todas formas, el uso de Batch Normalization luego de cada Convolución reduce el impacto de una mala inicialización, con lo que esta elección no se consideró tan relevante.

- Dropout: mediante reiteradas pruebas se determinó que un valor razonable de tasa de descarte para este problema es 30 % en el último bloque del camino de contracción de la arquitectura U-Net utilizada. Aumentar este porcentaje o agregar Dropout en otros bloques convolucionales sólo degradó el desempeño del modelo.

Una vez determinado el conjunto de hiperparámetros que produce resultados más satisfactorios, se procedió a realizar un estudio detallado del desempeño de cada modelo propuesto, para cada uno de los casos enunciados.

6.1.1. Segmentación de estructuras anatómicas

Para medir la performance de los distintos modelos propuestos sobre el reconocimiento de estructuras anatómicas, se entrenó cada uno de ellos sobre el 60 % de las imágenes del dataset IBSR designadas para el entrenamiento. Cada voxel de cada imagen tiene una de 4 etiquetas posibles: background y las tres estructuras anatómicas presentadas. Sólo la modalidad T1 está disponible en este dataset. Cada modelo se evaluó sobre el 40 % restante de las imágenes, comparando la predicción generada con la segmentación de referencia mediante el Coeficiente de Dice tomando como elementos positivos cada una de las etiquetas anatómicas. Se utilizó una partición 60 %-40 % en lugar de 80 %-20 % para aumentar la cantidad de imágenes de validación a 6 y así tener resultados más representativos. De todas formas, no se observaron diferencias notables con respecto a utilizar 80 %-20 %. Se presentan las métricas obtenidas para cada etiqueta sobre las imágenes de validación en la Figura 17, en forma de diagramas de caja. Las cajas cubren el rango intercuartil de los datos (el 50 % central), las barras de error muestran el resto de la distribución. Las líneas horizontales representan la mediana. Se ilustra con la segmentación generada para una imagen de ejemplo en la Figura 18.

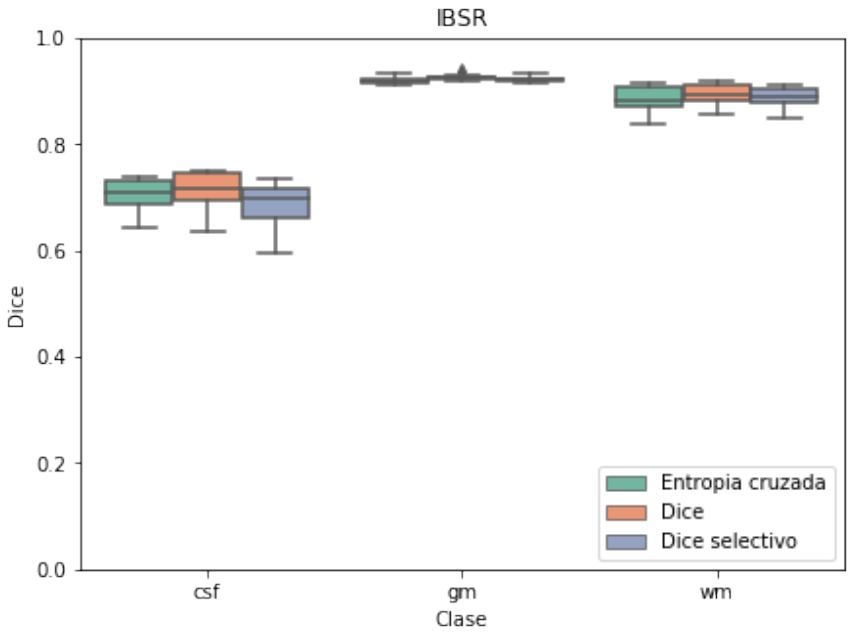
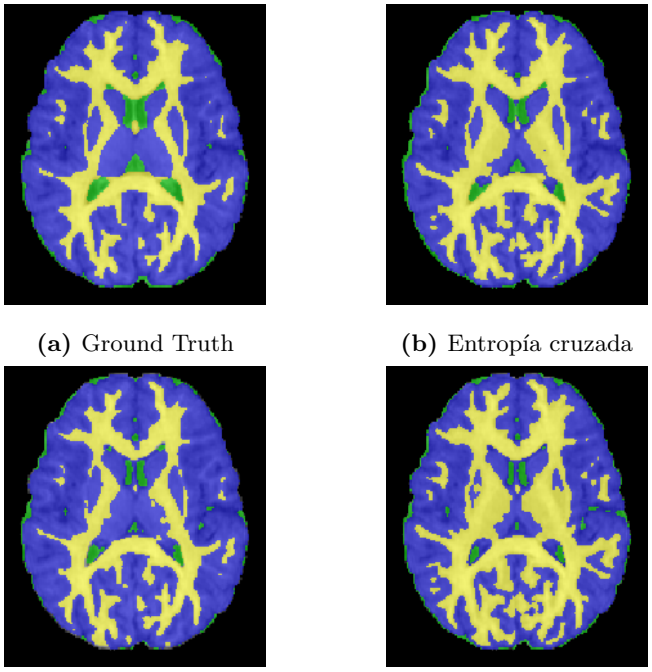


Figura 17: Performance por etiqueta para cada función de costo propuesta, para el escenario mono-tarea de segmentación de estructuras anatómicas. Modelos entrenados y validados sobre las particiones de entrenamiento y validación de IBSR, respectivamente.



(a) Ground Truth **(b)** Entropía cruzada

(c) Dice **(d)** Dice selectivo

Figura 18: Ground Truth y predicciones de los modelos propuestos para segmentación de estructuras anatómicas.

En líneas generales, se puede ver un desempeño muy similar en todos los modelos. Se observa una marcada diferencia entre la calidad de las segmentaciones de CSF con respecto a WM y GM. Esto se debe, principalmente, a que las estructuras presentes en la etiqueta CSF son muy afinadas y segmentarlas resulta una tarea mucho más compleja que segmentar estructuras mayores como WM o GM. Los resultados obtenidos en el escenario monomodal son consistentes con los reportados en la literatura[36].

Para estudiar la significancia estadística de las diferencias observadas entre los modelos, se utilizó el **Test de Wilcoxon**. Este es un test no paramétrico para muestras pareadas que evalúa la hipótesis nula de que las diferencias entre pares tienen una distribución simétrica en torno a 0, contra la hipótesis alternativa de que estas diferencias no sean simétricas en torno a 0. Esto permite comparar el desempeño de dos modelos distintos evaluados sobre un mismo conjunto de imágenes con sus respectivas segmentaciones de referencia $\{(x_i, y_i)\}_{1 \leq i \leq N}$, tomando como muestras pareadas las métricas obtenidas sobre cada etiqueta de cada imagen, para un modelo y el otro: $\left\{\left\langle dice_\ell(pred_1(x_i), y_i),\ \ dice_\ell(pred_2(x_i), y_i)\right\rangle \mid \forall \ell \in \mathcal{L}, \forall i \in \{1, \ldots, n\}\right\}$ donde $dice_\ell$ es el coeficiente de Dice sobre la etiqueta ℓ, $pred_1(x)$ y $pred_2(x)$ son las predicciones de un modelo y otro sobre la imagen x y $\mathcal{L}$ es el conjunto de etiquetas considerado. Se tomó como nivel de significancia estadística un p-valor menor que 0.05, con Corrección de Bonferroni con el fin de reducir el problema de comparaciones de hipótesis múltiples.

Según el test de Wilcoxon, en este escenario Dice presenta diferencias significativas con los otros dos modelos, entre los que no se pudo determinar significancia estadística. Se grafican las medias generales por modelo, entre todas las etiquetas de todas las imágenes, en la Figura 19. Se indican las diferencias no significativas.

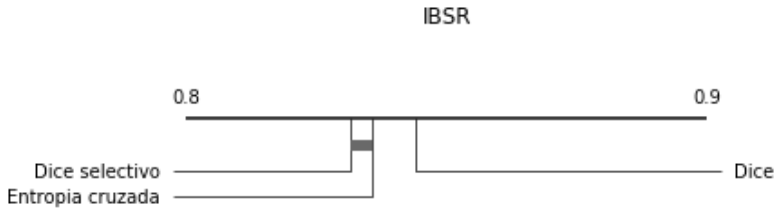


Figura 19: Medias generales del coeficiente de Dice sobre las predicciones de cada modelo para IBSR. Las lineas gruesas representan las diferencias no significativas entre pares de modelos.

6.1.2. Segmentación de lesiones WMH

En base a la sección anterior, se puede considerar que los modelos son efectivos identificando diferentes estructuras anatómicas. Se observó mayor dificultad para segmentar estructuras pequeñas (como la clase CSF). Surge entonces el interés en evaluar estos modelos sobre lesiones de WMH, donde la tarea consiste en identificar una única etiqueta muy poco frecuente.

Se entrenaron los distintos modelos propuestos en la tarea de segmentación de lesiones de WMH sobre el dataset MRBrainS17, se separó el dataset en 80 % sobre el cual se entrenaron los modelos y 20 % de evaluación. Se computó el Coeficiente de Dice de la etiqueta WMH entre las predicciones de los modelos sobre las imágenes de evaluación y la segmentación de referencia. Las modalidades de MRI utilizadas en este escenario son T1 y FLAIR.

Se muestran los resultados en la Figura 20. La primer diferencia evidente con respecto al escenario anterior es la notable varianza de las métricas. Esto quiere decir que el desempeño de los modelos es poco consistente entre imágenes. Esto puede deberse a que reconocen correctamente algunos tipos de lesiones de WMH (en zonas o tamaños particulares), pero fallan en otras.

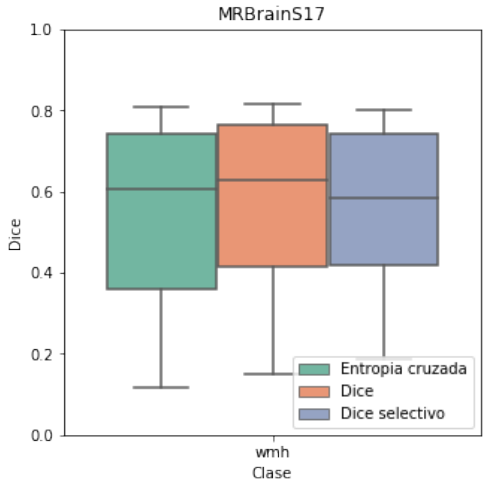
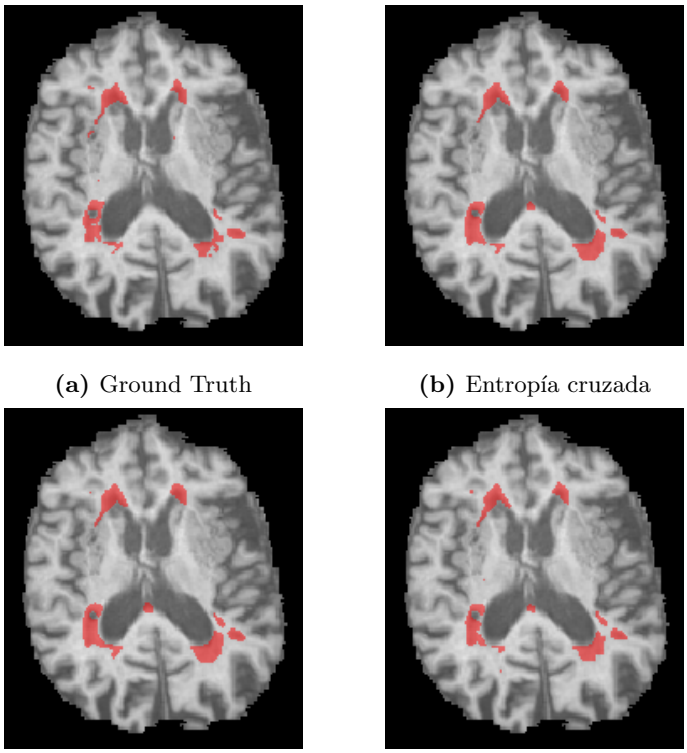


Figura 20: Performance por etiqueta para cada función de costo propuesta, para el escenario mono-tarea de segmentación de WMH. Modelos entrenados y validados sobre las particiones de entrenamiento y validación de MRBrainS17, respectivamente.

En la Figura 21 se ven las segmentaciones generadas por los modelos para una imagen particular, junto con el ground truth de referencia. Se puede ver que mas allá de mínimas discrepancias, el resultado de todos los modelos es muy similar. Para lograr estos resultados demostró ser crucial la estrategia de muestreo utilizada, principalmente la extracción de parches de zonas relevantes. Estrategias más ingenuas produjeron resultados parciales e insatisfactorios.



(a) Ground Truth **(b)** Entropía cruzada

(c) Dice **(d)** Dice selectivo

Figura 21: Ground Truth y predicciones de los modelos propuestos para segmentación de WMH.

Si bien la Figura 20 muestra considerable solapamiento entre las distribuciones de los resultados de cada modelo, el test de Wilcoxon indicó una diferencia significativa entre entropía cruzada y Dice. Se muestran a continuación las medias de cada modelo.

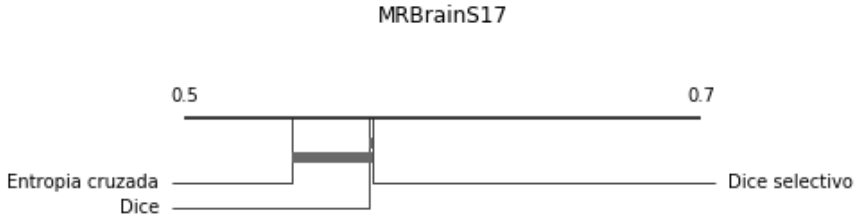


Figura 22: Medias del coeficiente de Dice sobre las predicciones de cada modelo para MRBrainS17. Las líneas gruesas indican diferencias no significativas entre modelos, según el test pareado de Wilcoxon.

6.1.3. Segmentación de tumores

El último escenario mono-tarea a considerar es la segmentación de tumores. En este caso las etiquetas presentes son edema y tumor. Sus estructuras son más grandes que las de lesiones de WMH, con lo que se espera una mayor facilidad para reconocerlas.

Se muestra a continuación el desempeño de los distintos modelos para segmentación de tumores, utilizando el dataset BraTS12 con la misma metodología que en los casos anteriores, particionando en 80 % de entrenamiento y 20 % de evaluación. Las modalidades de MRI disponibles son T1, T1 con contraste de Gadolinio, T2 y FLAIR. En la figura 23 se ve el desempeño general sobre la partición de evaluación. Si bien los valores medios son elevados, hay mucha variabilidad, con lo que el desempeño no es uniforme.

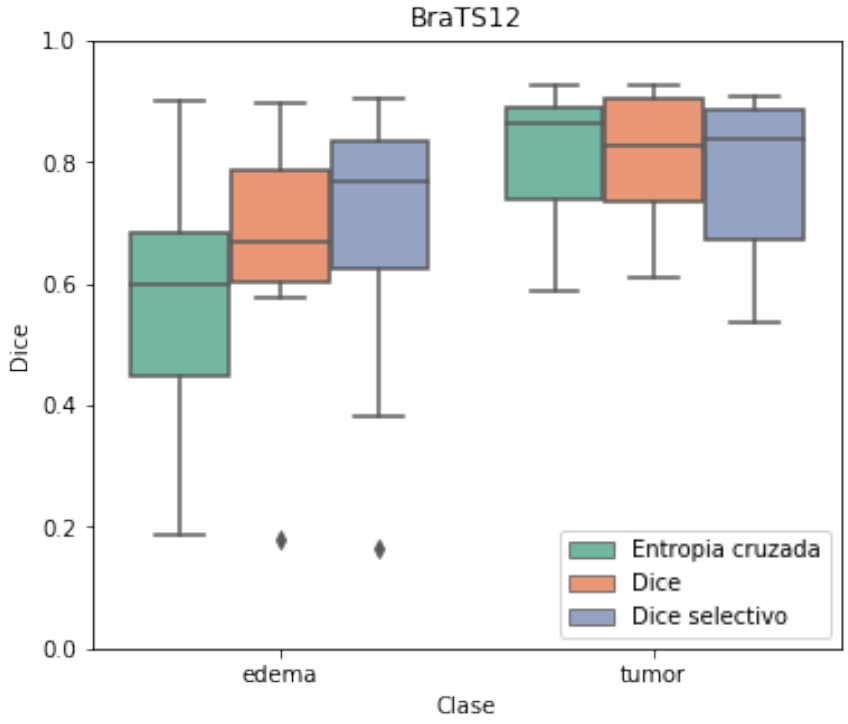


Figura 23: Performance por etiqueta para cada función de costo propuesta, para el escenario mono-tarea de segmentación de tumores. Modelos entrenados y validados sobre las particiones de entrenamiento y validación de BraTS12, respectivamente.

Las diferencias en este caso según el test de Wilcoxon no resultan ser significativas. Se muestran a continuación las medias generales de cada modelo sobre todas las etiquetas de todas las imágenes.

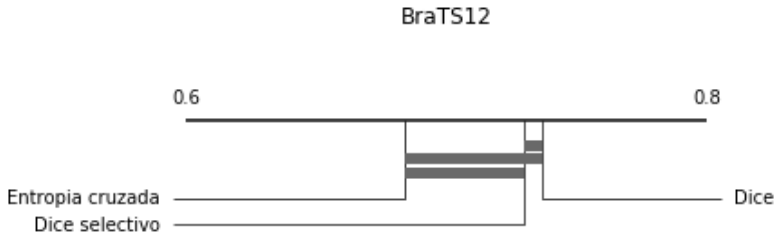
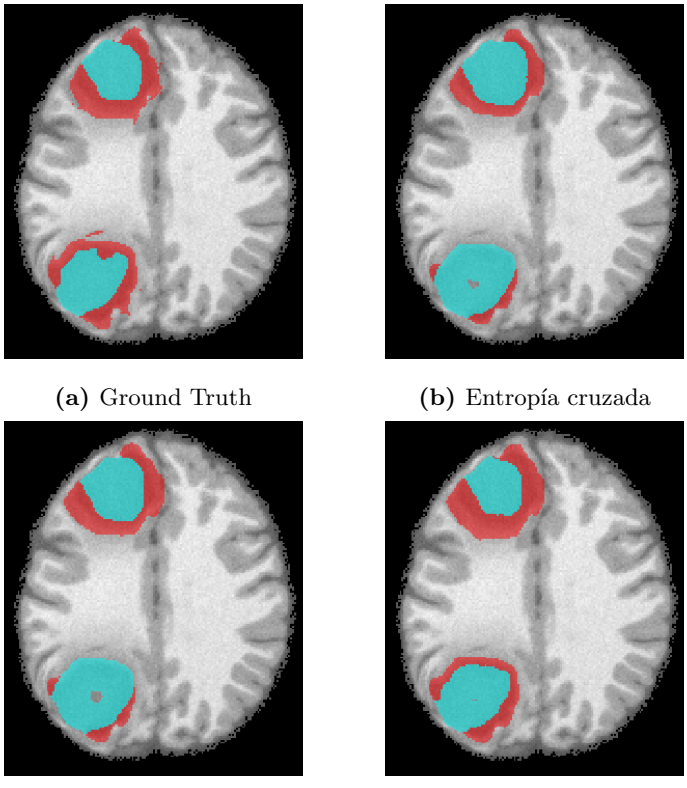


Figura 24: Medias generales del coeficiente de Dice sobre las predicciones de cada modelo para BraTS12. No se observaron diferencias significativas.

La Figura 25 muestra un ejemplo de segmentación generada por los distintos modelos para una de las imágenes de validación. Aquí se puede apreciar la variabilidad ya mencionada, dado que si bien todas las predicciones en líneas generales reconocen las mismas estructuras, hay diferencias visibles en los detalles de cada una.



(a) Ground Truth **(b)** Entropía cruzada

(c) Dice **(d)** Dice selectivo

Figura 25: Ground Truth y predicciones de los modelos propuestos para segmentación de tumores, sobre una imagen de BraTS12.

Estos resultados muestran que todos los modelos son capaces de aprender todas las tareas planteadas de forma relativamente satisfactoria. En la siguente sección se estudiará la capacidad de los modelos de aprender las mismas tareas en forma combinada.

6.2. Estructuras anatómicas y lesiones de WMH

El primer escenario multi-tarea propuesto en la Sección 5 es la segmentación conjunta de estructuras anatómicas y lesiones de Hiperintensidades de Materia Blanca. Una posible dificultad es el gran desbalance entre las etiquetas anatómicas y las etiquetas de lesión. Se midió el desempeño de las distintas funciones de pérdida para los siguientes casos:

- Modelos entrenados con MRBrainS13 y MRBrainS17, validando sobre MRBrainS18.

- Modelos entrenados con IBSR y MRBrainS17, validando sobre MRBrainS18.

El motivo para plantear dos experimentos distintos es analizar el impacto de la diferencia de dominios y de los tamaños de los datasets. Ya se mencionó que MRBrainS13 es un dataset con sólo 4 imágenes de entrenamiento y mucha similitud entre las imágenes, lo que permite poco grado de generalización. Sin embargo, también es más similar al dataset de lesiones y al de validación, con lo que quizás la falta de generalización en este caso no tiene un impacto tan grave.

Los modelos propuestos a comparar en cada caso son el Modelo Base (ver Sección 3.2.2), que también se denominará MultiUNet, por componerse de varias arquitecturas U-Net, y modelos U-Net entrenados utilizando como función de costo la Entropía Cruzada (el Modelo Naive, ver Sección 3.2.1), la Entropía Cruzada Selectiva (ver Sección 3.3.2), Dice (ver Sección 3.3.3) y Dice Selectivo (ver Sección 3.3.4).

MRBrains13 y MRBrainS17

Se entrenó cada modelo muestreando parches indistintamente de las particiones de entrenamiento de MRBrainS13 y MRBrainS17, tomando el conjunto de parámetros que minimizara la función de pérdida utilizada sobre las particiones de validación. Se evaluaron los modelos entrenados, sobre la totalidad del dataset MRBrainS18 y se computó el coeficiente de Dice entre las predicciones para cada imagen y la segmentación de referencia correspondiente. Se utilizaron las modalidades de MRI en común entre las imágenes de estos tres datasets: T1 y FLAIR.

En la figura 26 se puede ver el desempeño comparativo de cada modelo en la clasificación de cada etiqueta. Se reporta también el desempeño de un modelo mono-tarea entrenado con entropía cruzada sólo sobre el dataset de entrenamiento correspondiente y validado sobre la porción de validación del mismo, con los datos provenientes de las figuras de la Sección 6.1. Dado que MRBrainS13 posee una sola imagen de validación, el valor de Dice mono-tarea reportado para las etiquetas anatómicas puede ser utilizado como referencia pero no se debería considerar una métrica muy precisa por falta de muestras.

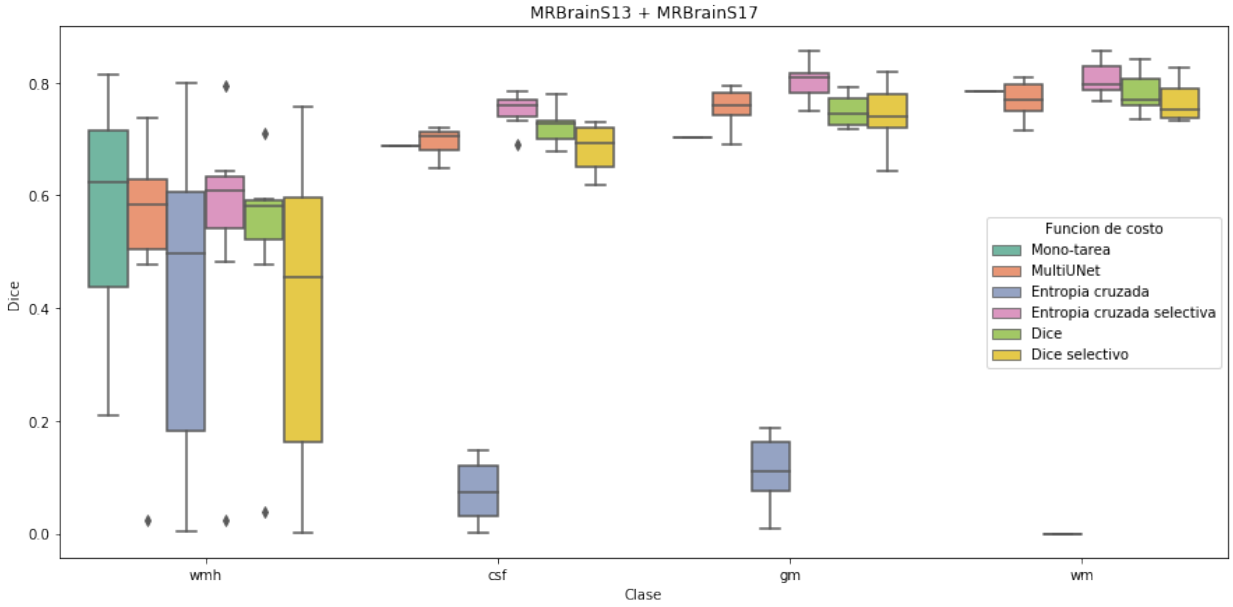


Figura 26: Performance por etiqueta para cada modelo propuesto, entrenados sobre MRBrainS13 y MRBrainS17, validado sobre MRBrainS18. La performance mono-tarea se agrega como punto de comparación, y está validada sobre la partición de validación del dataset usado para entrenar el modelo correspondiente.

Se puede observar que el desempeño de los distintos modelos fue relativamente similar, a excepción del modelo entrenado con una sola red utilizando entropía cruzada (el Modelo Naive), que presenta una muy baja performance en las etiquetas anatómicas. Si bien se esperaba una falta de capacidad de aprendizaje de este modelo, es notable la abismal diferencia con los otros modelos. Sorprende también la precisión del modelo entrenado con Dice. Dice selectivo presenta un buen desempeño en las tareas anatómicas pero una alta varianza en el reconocimiento de WMH.

Se computó la significancia estadística entre los distintos modelos con el test de Wilcoxon, sin considerar el caso mono-tarea dado que es evaluado sobre otro conjunto de imágenes, con lo que no se pueden tomar muestras pareadas para la comparación. Las diferencias dieron todas significativas, a excepción de la diferencia de MultiUNet con Dice. En la figura 27 se muestran las medias generales por modelo como referencia. Nuevamente, la Entropía cruzada selectiva produjo la media más alta, con diferencias significativas contra todos los modelos.

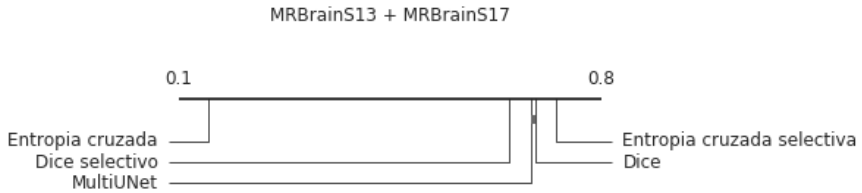
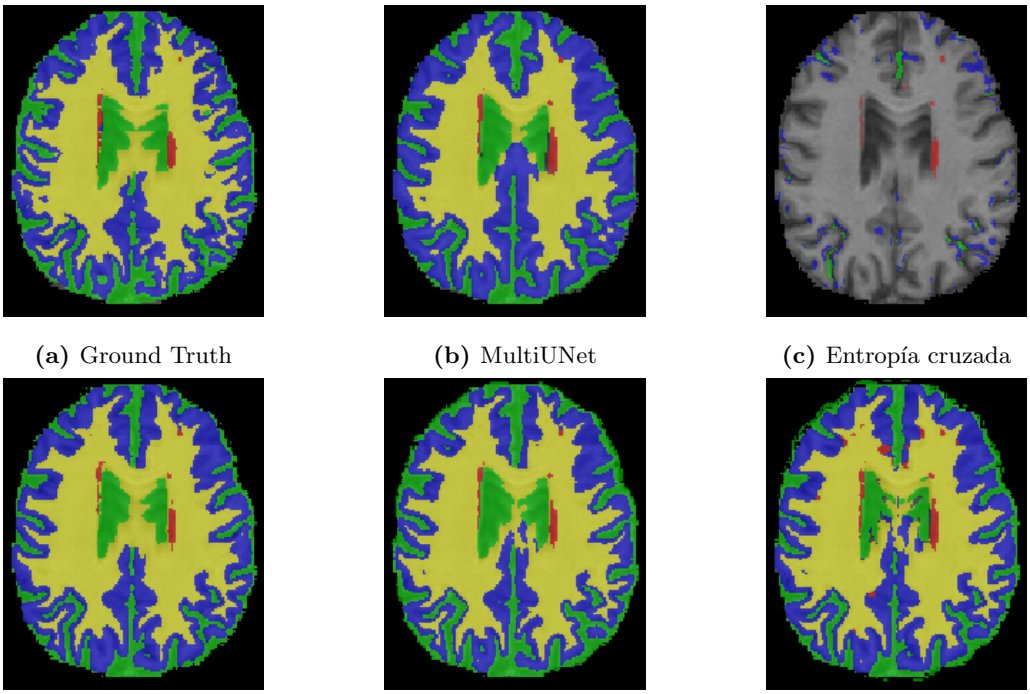


Figura 27: Medias generales del coeficiente de Dice sobre las predicciones de cada modelo entrenados con MRBrainS13 y MRBrainS17 y evaluados sobre MRBrainS18. Todas las diferencias son significativas a excepción de MultiUNet con Dice.

Para un análisis más detallado de las predicciones generadas, en la Figura 28 se presenta un corte axial de una imagen de MRBrainS18 con las etiquetas de la segmentación generada por los distintos modelos, incluyendo el ground truth de referencia.



(a) Ground Truth **(b)** MultiUNet **(c)** Entropía cruzada

(d) Entropía cruzada selectiva **(e)** Dice **(f)** Dice selectivo

Figura 28: Ground Truth y predicciones de los modelos propuestos para segmentación multi-tarea de estructuras anatómicas y WMH, entrenado con MRBrains13 y MRBrainS17. Evaluado sobre una imagen de MRBrainS18.

Esta figura deja aún más en evidencia las falencias del Modelo Naive. Prácticamente no aplica etiquetas anatómicas en la segmentación combinada. Se estudiará más en detalle este fenómeno en el caso IBSR + MRBrainS17 a continuación.

Por otro lado, se ve que Dice selectivo reconoce correctamente las lesiones de WMH presentes, pero también encuentra lesiones donde no las hay. Al igual que en los escenarios mono-tarea, se vuelve a observar que la Entropía cruzada selectiva obtiene resultados significativamente mejores que el resto.

IBSR y MRBrainS17

Se vio en el experimento anterior que las funciones de costo propuestas se desempeñaron bien en un entorno con muy poca diferencia de dominio. A continuación, se utilizó la misma metodología con el fin de reproducir los resultados utilizando IBSR y MRBrainS17 como datasets de entrenamiento.

Las imágenes del dataset IBSR no contienen la modalidad FLAIR, con lo que la única modalidad de MRI en común entre los datasets utilizados es T1. Como se dijo en la sección 5.2, las lesiones de WMH se denominan hiperintensidades por resaltar en imágenes de modalidad T2 y FLAIR. En imágenes T1, su reconocimiento es más difícil, lo que puede traer problemas en este escenario para los modelos de una

sola red. Para el entrenamiento de MultiUNet, se puede entrenar la red de segmentación anatómica sobre la modalidad T1 y la de segmentación de lesiones sobre T1 y FLAIR, dado que ambas están disponibles tanto en el dataset de entrenamiento MRBrainS17 como en el de validación, MRBrainS18. Esto evidencia una diferencia importante entre ambos enfoques. Si las modalidades disponibles para cada tarea difieren considerablemente, utilizar redes distintas para cada tarea permite utilizar más información.

La Figura 29 muestra el desempeño de cada modelo propuesto en este escenario.

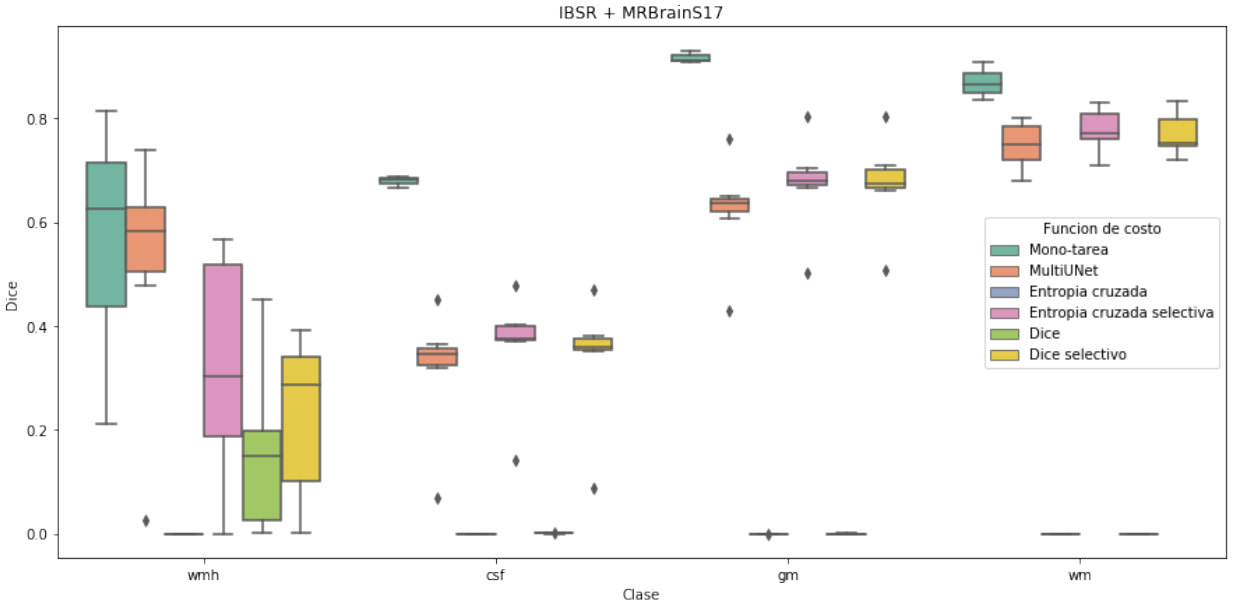


Figura 29: Performance por etiqueta para cada modelo propuesto, entrenados sobre IBSR y MRBrainS17, validado sobre MRBrainS18. La performance mono-tarea se agrega como punto de comparación, y está validada sobre la partición de validación del dataset usado para entrenar el modelo correspondiente.

Las diferencias observadas son significativas a excepción de Dice con Entropía cruzada, Dice selectivo con MultiUNet y Entropía cruzada selectiva con MultiUNet. La Figura 30 muestra las métricas promedio obtenidas por cada modelo. La Entropía cruzada selectiva tuvo un desempeño significativamente mejor que el resto de los modelos a excepción de MultiUNet, donde no se pudo determinar significancia estadística.

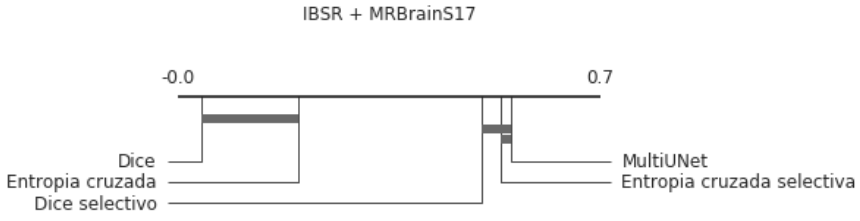
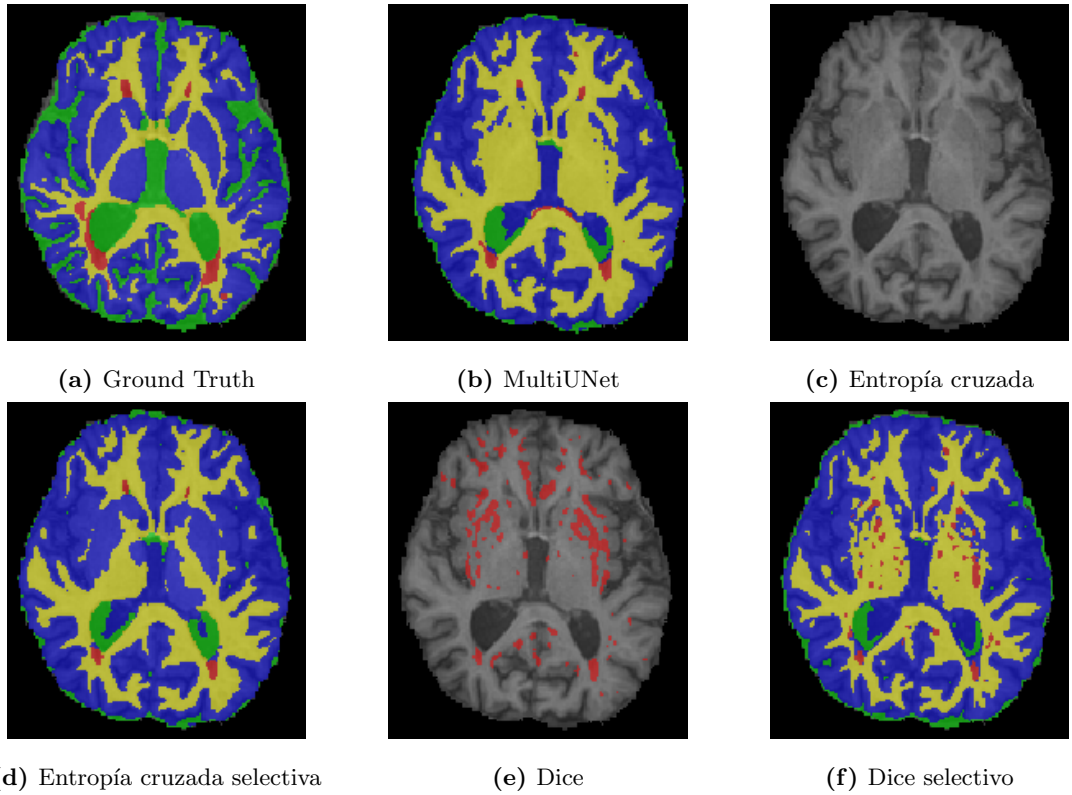


Figura 30: Medias generales del coeficiente de Dice sobre las predicciones de cada modelo entrenados con IBSR y MRBrainS17 y evaluados sobre MRBrainS18. Todas las diferencias son significativas excepto Dice con Entropía cruzada, Dice selectivo con MultiUNet y Entropía cruzada selectiva con MultiUNet.

Ejemplos de segmentaciones concretas de cada modelo en una imagen de MRBrainS18 se muestran en

la Figura 31.



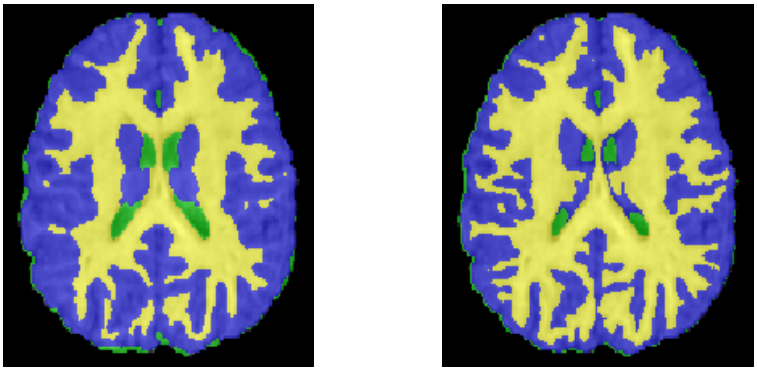
(a) Ground Truth **(b)** MultiUNet **(c)** Entropía cruzada

(d) Entropía cruzada selectiva **(e)** Dice **(f)** Dice selectivo

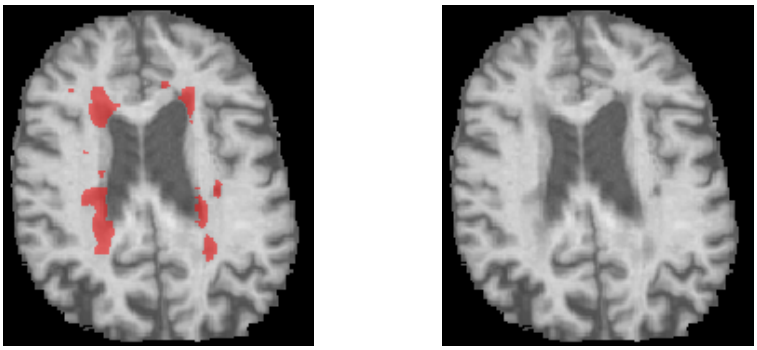
Figura 31: Ground Truth y predicciones de los modelos propuestos para segmentación multi-tarea de estructuras anatómicas y WMH, entrenado con MRBrains13 y MRBrainS17. Evaluado sobre una imagen de MRBrainS18.

En comparación con el desempeño de los modelos entrenados con MRBrains13 y MRBrainS17, las predicciones de los entrenados con IBSR y MRBrainS17 dan considerablemente peores resultados, particularmente el modelo entrenado utilizando la función de costo Dice. Un factor que puede influir en este resultado puede ser la ausencia de la modalidad FLAIR utilizada en el escenario anterior. Adicionalmente, basándose en el análisis de las distancias Δ_{JS} entre datasets, se puede especular que el deterioro en la performance se debe a un efecto multi-dominio, dado que MRBrains13, MRBrainS17 y MRBrainS18 son muy similares entre sí, mientras que la diferencia con IBSR es mayor. Para respaldar esta teoría, se realizaron dos análisis.

En primer lugar, se estudiaron las segmentaciones producidas por el Modelo Naive con IBSR y MRBrainS17, que es el que presenta peores resultados, sobre imágenes de validación de IBSR, MRBrainS17. Ejemplos representativos de estos dos casos se pueden ver en la figura 32.



(a.1) Ground truth de IBSR **(a.2)** Predicción de IBSR



(b.1) Ground truth de MRBrainS17 **(b.2)** Predicción de MRBrainS17

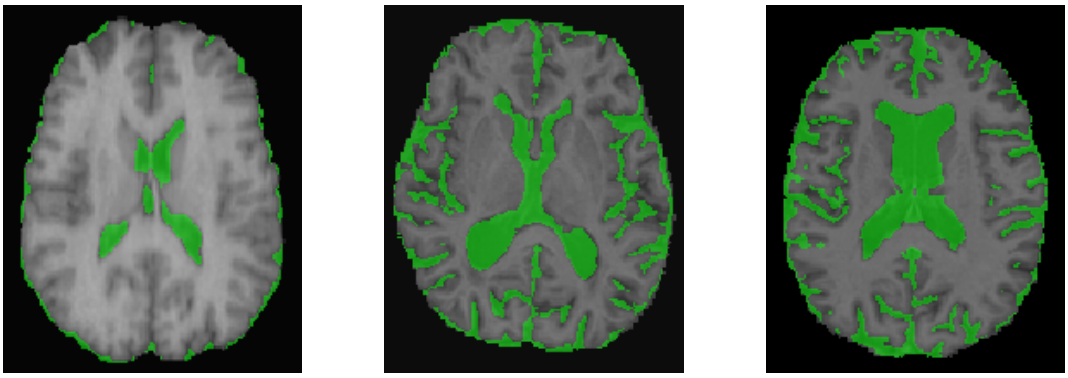
Figura 32: Predicciones del Modelo Naive entrenado sobre IBSR y MRBrainS17. **a)** Ground truth y predicción sobre una imagen de IBSR. **b)** Ground truth y predicción sobre una imagen de MRBrainS17.

Es notorio que la segmentación producida para IBSR contiene un etiquetado anatómico muy preciso, mientras que la segmentación de MRBrainS17 generada por el mismo modelo no contiene ninguna etiqueta (no solo en el corte mostrado sino en todo el volumen). Sobre imágenes de MRBrainS18, se reproduce este fenómeno. El modelo asigna background a todos los voxeles de cada imagen del dataset. Una explicación posible es que el modelo aprende a reconocer el origen de la imagen e intenta generar una segmentación acorde, aunque falla en la tarea de segmentación de WMH. Esto sucede porque como ya se dijo, la entropía cruzada espera una segmentación exactamente igual al ground truth. La única forma de que el modelo pueda cumplir eso sobre los datos de entrenamiento es reconociendo el origen de las imágenes y aplicando sólo las etiquetas del dataset correspondiente. Volviendo al análisis de distancias entre datasets, cuanto mayor sea la diferencia entre ambos datasets de entrenamiento, más fácil será para el modelo reconocer el origen de cada imagen.

Este fenómeno sucede de forma similar en el modelo entrenado con Dice: sobre imágenes de validación de IBSR, genera una segmentación anatómica bastante acertada, mientras que en imágenes de MRBrainS17 intenta agregar etiquetas de WMH de forma muy imprecisa. No sucede lo mismo en los modelos entrenados sobre MRBrainS13 y MRBrainS17: aquí las predicciones para imágenes de validación de ambos datasets generadas por Dice aplican etiquetas combinadas. Esto evidentemente sucede porque las imágenes de ambos datasets son más difíciles de distinguir. Es curioso que en este último caso, la imposibilidad de distinguir las imágenes hace que el modelo entrenado con Dice termine generando una segmentación combinada muy

precisa.

El segundo análisis consiste en comparar los etiquetados de referencia de cada dataset. Se puede apreciar en la Figura 29 que el desempeño general sobre la etiqueta CSF es particularmente bajo y particularmente parejo entre los modelos que parecen haberlo aprendido. Para identificar el problema se compararon las segmentaciones de ground truth de CSF de los distintos datasets. Tres ejemplos representativos se pueden ver en la Figura 33. Es notoria la diferencia en el etiquetado. Ambos MRBrainS presentan etiquetados similares, mientras que el etiquetado de CSF de IBSR podría decirse que es considerablemente más conservador.



(a) IBSR **(b)** MRBrainS13 **(c)** MRBrainS18

Figura 33: Etiquetas de ground truth de CSF para imágenes de **a)** IBSR, **b)** MRBrains13 y **c)** MRBrainS18 respectivamente, cortes axiales a similares alturas.

Esto deja en evidencia otro conflicto multi-dominio, que no había sido tenido en cuenta: el etiquetado de ground truth, al ser realizado manualmente por expertos, puede diferir notablemente entre una persona y otra. De esta forma, un modelo que aprenda perfectamente a segmentar imágenes a partir de un dataset puede tener un mal desempeño al ser validado sobre otro dataset pese a aplicar correctamente lo aprendido, simplemente por el hecho de que los criterios utilizados para generar las segmentaciones de referencia en ambos son distintos.

6.3. Estructuras anatómicas y tumores

El último escenario multi-tarea propuesto es la segmentación conjunta de estructuras anatómicas y tumores cerebrales. En este contexto se evaluaron los distintos modelos propuestos utilizando BrainWeb y BraTS12 como datasets de entrenamiento, utilizando la misma metodología que en los experimentos anteriores. Los modelos fueron validados sobre la totalidad del dataset TumorSim. La única modalidad en común entre estos datasets es T1. Los resultados obtenidos se ven en la Figura 34.

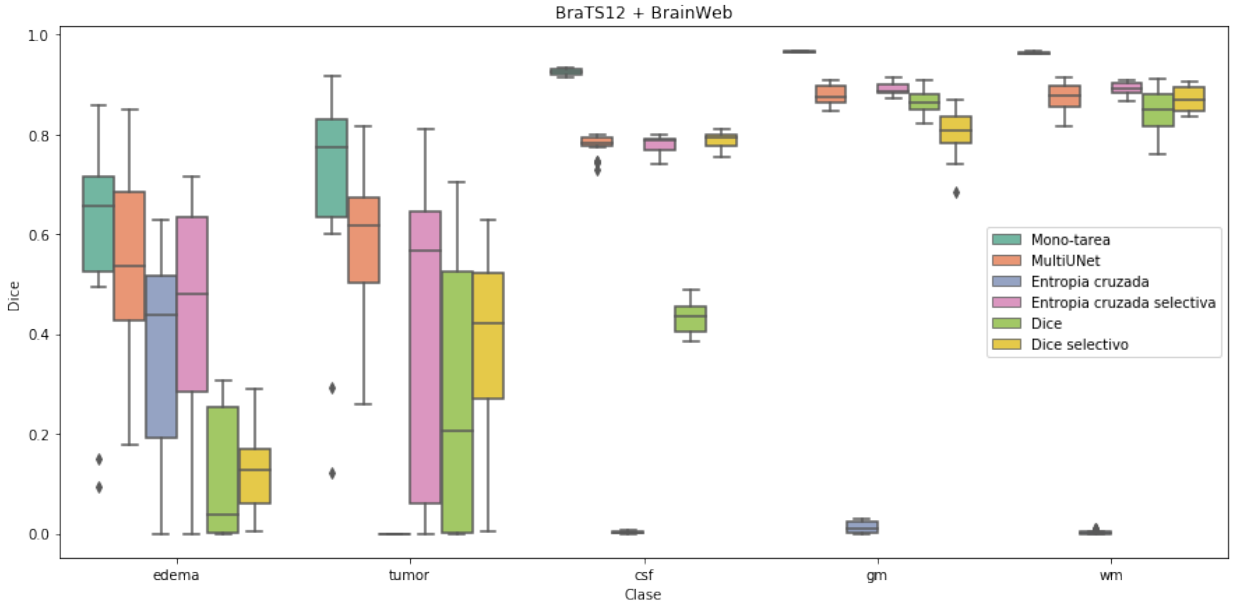


Figura 34: Performance por etiqueta para cada modelo propuesto, entrenados sobre BraTS12 y BrainWeb, validado sobre TumorSim. La performance Mono-tarea se agrega como punto de comparación, y está validada sobre la partición de validación del dataset usado para entrenar el modelo correspondiente.

Las diferencias observadas son todas significativas, a excepción de la diferencia entre Entropía cruzada selectiva y MultiUNet. Los promedios generales se muestran en la Figura 35.

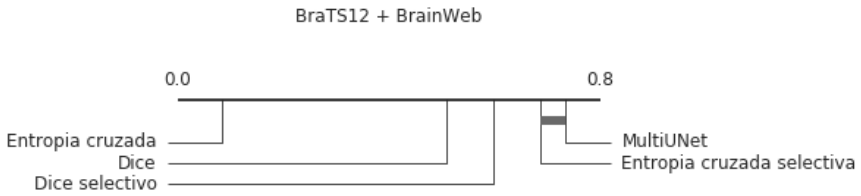
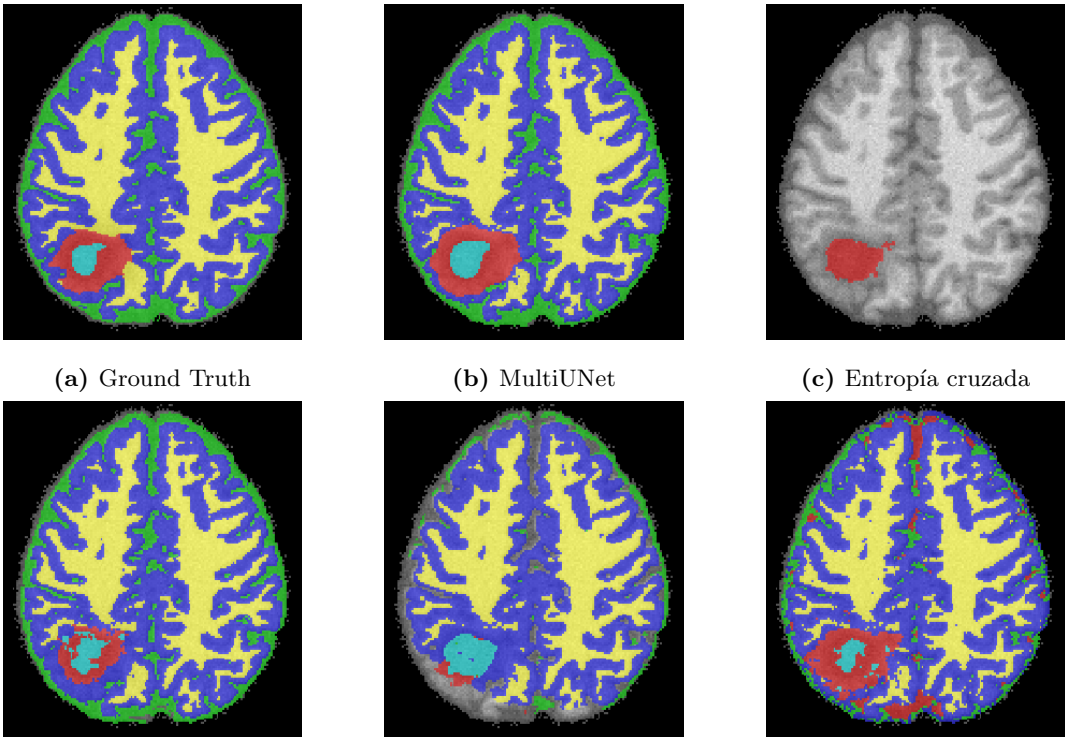


Figura 35: Medias generales del coeficiente de Dice sobre las predicciones de cada modelo entrenados con BraTS12 y BrainWeb y evaluados sobre TumorSim. Todas las diferencias son significativas excepto MultiUNet con Entropía cruzada selectiva.

Nuevamente la Entropía cruzada selectiva supera significativamente a los otros modelos propuestos, a excepción de MultiUNet, con el que no presenta diferencias significativas. La entropía cruzada tradicional, al igual que en los casos anteriores, es incapaz de aprender la mayor parte de las etiquetas. Ambos coeficientes de Dice tienen un desempeño particularmente pobre sobre la etiqueta edema, pero relativamente bueno sobre las etiquetas anatómicas. La Figura 36 muestra un ejemplo de segmentación producida por cada modelo, comparado con la referencia de ground truth para una imagen de TumorSim.



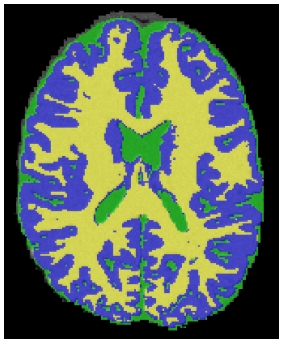
(a) Ground Truth **(b)** MultiUNet **(c)** Entropía cruzada

(d) Entropía cruzada selectiva **(e)** Dice **(f)** Dice selectivo

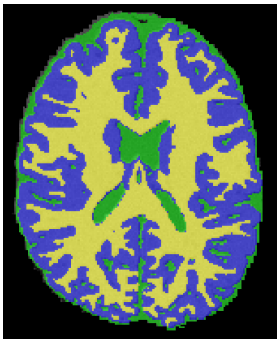
Figura 36: Ground Truth y predicciones de los modelos propuestos para segmentación multi-tarea de estructuras anatómicas y WMH, entrenado con MRBrains13 y MRBrainS17. Evaluado sobre una imagen de MRBrainS18.

Aquí se evidencia la falla de Dice en reconocer edemas y CSF, así como las dificultades de Dice Selectivo sobre la etiqueta edema.

Se puede ver también el ya característico resultado de la entropía cruzada, que al menos logra reconocer la ubicación de la patología, aunque no genera un etiquetado correcto, mientras que ignora completamente las etiquetas anatómicas. La Figura 37 repite el análisis de la Figura 32 para este escenario. Se puede ver que sobre imágenes de validación de cada uno de los datasets de entrenamiento, genera etiquetados que utilizan sólo las etiquetas del dataset correspondiente, aunque la segmentación de tumores sobre BraTS12 no es buena. Evidentemente vuelve a reconocer el origen de cada imagen y aplica sólo las etiquetas de ese dataset.



(a.1) Ground truth de BrainWeb



(a.2) Predicción de BrainWeb

(b.1) Ground truth de BraTS12

(b.2) Predicción de BraTS12

Figura 37: Predicciones del Modelo Naive entrenado sobre BraTS12 y BrainWeb. **a)** Ground truth y predicción sobre una imagen de BrainWeb. **b)** Ground truth y predicción sobre una imagen de BraTS12.

Surge aquí la pregunta de por qué tanto en el caso de segmentación de estructuras anatómicas y WMH como en el de estructuras anatómicas y tumores, el modelo entrenado con entropía cruzada genera principalmente etiquetas patológicas. Considerando el análisis de distancias entre datasets, una explicación posible es que en ambos casos el dataset de entrenamiento de lesiones es más similar al dataset de validación que el dataset de entrenamiento anatómico (MRBrainS18 se parece más a MRBrainS17 que a IBSR y TumorSim se parece más a BraTS12 que a BrainWeb).

7. Conclusiones

Durante el transcurso de este trabajo se estudiaron los problemas que presenta el entrenamiento de modelos de aprendizaje profundo para la segmentación multi-tarea de imágenes médicas cerebrales y se propusieron varios enfoques para obtener buenos resultados a partir de datos con etiquetas disjuntas.

Los modelos propuestos fueron entrenados en distintos escenarios multi-tarea y evaluados sobre datos con etiquetas combinadas. Se utilizaron como puntos de comparación el Modelo Naive y el Modelo Base, con el objetivo de alcanzar la eficiencia del primero y la eficacia del segundo. Se estudió el desempeño comparativo de cada modelo sobre cada etiqueta en los distintos casos.

Un aspecto que resultó determinante en el correcto entrenamiento de las redes fue la estrategia de muestreo de imágenes utilizada, basada en tomar siempre información relevante, de forma balanceada.

Además de las técnicas de muestreo, se exploraron distintas funciones de costo existentes y se propusieron nuevas, especialmente diseñadas para el entrenamiento multi-tarea sin etiquetas combinadas. El principio básico de las funciones propuestas es no buscar la igualdad entre las predicciones generadas por el modelo y las etiquetas de ground truth sino comparar únicamente la información de interés aportada por ambos etiquetados.

Una ventaja de basar el enfoque en la función de costo y las técnicas de muestreo a utilizar es que tiene un bajo acoplamiento con el resto de la arquitectura, con lo que es fácil trasladar los resultados obtenidos a otros diseños de redes neuronales. Un área de interés para trabajo futuro es experimentar con otras arquitecturas de red, particularmente con variantes más reducidas de U-Net, donde la cantidad de parámetros sea menor, reduciendo el costo computacional y la memoria requerida para evaluar volúmenes completos. Algunas pruebas preliminares sugieren que se puede disminuir sustancialmente el tamaño del modelo sin deteriorar significativamente los resultados. Esto es particularmente útil si se desea aplicar estas técnicas como herramientas de asistencia al diagnóstico en un entorno médico real con rápida disponibilidad sin requerir el acceso a gran capacidad de cómputo.

En cuanto al desempeño de los distintos modelos, el Modelo Base produjo buenos resultados de forma consistente, lo que incrementó la confianza en poder utilizarlo como punto de comparación frente a los otros modelos. Las redes entrenadas con entropía cruzada selectiva mostraron generar predicciones de calidad similar al Modelo Base en los distintos escenarios y significativamente mejores al resto de los modelos propuestos. El impacto de la utilización de imágenes y segmentaciones provenientes de dominios distintos para el entrenamiento y la evaluación mostró tener un efecto negativo durante la experimentación más influyente de lo que se esperaba. En vistas a este problema, se planteó un método para cuantificar la diferencia de dominio entre conjuntos de imágenes. Esto permitió observar, por ejemplo, que los modelos basados en Dice y Dice selectivo produjeron resultados buenos en escenarios con dominios similares pero considerablemente peores en dominios con mayor diferencia. La entropía cruzada selectiva se vio menos afectada por este factor. De todas formas, se pudo comprobar que todos los modelos propuestos tuvieron un mejor desempeño mejor que el Modelo Naive.

Resultó interesante observar el tipo de errores que se generaron en los modelos que tuvieron mal desempeño, para dejar en evidencia las dificultades que acarrea el entrenamiento multi-tarea. Para estos modelos, si bien los resultados no fueron buenos para generar segmentaciones con etiquetas combinadas, esto no quiere decir que el entrenamiento haya fallado sino que las funciones de pérdida utilizadas en esos casos penalizaban a modelos que generasen etiquetas combinadas. A fin de cuentas, cada modelo intenta aprender lo que su función de pérdida le exige.

Una alternativa no explorada por este trabajo es la de aplicar técnicas deliberadamente diseñadas para reducir el efecto multi-dominio dentro de los modelos, tales como las descriptas en [6]. Considerando el deterioro en los resultados producido por este efecto, parece ser un punto de foco importante para seguir refinando la calidad de los modelos.

Otro análisis que excedió el objetivo del trabajo, pero que permitiría estudiar con mayor profundidad la robustez de los distintos modelos para el aprendizaje multi-tarea es generalizar el problema a una cantidad arbitraria de tareas. Por ejemplo, ¿puede un mismo modelo aprender a reconocer estructuras anatómicas, lesiones de WMH y tumores? ¿Mejora el desempeño si se utilizan datos de dominios distintos para una misma tarea? ¿Qué sucede si las etiquetas de distintas tareas se solapan?

Referencias

[1] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[4] Konstantinos Kamnitsas, Enzo Ferrante, Sarah Parisot, Christian Ledig, Aditya V Nori, Antonio Criminisi, Daniel Rueckert, and Ben Glocker. Deepmedic for brain tumor segmentation. In *International Workshop on Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, pages 138–149. Springer, 2016.

[5] Pim Moeskops, Jelmer M Wolterink, Bas HM van der Velden, Kenneth GA Gilhuijs, Tim Leiner, Max A Viergever, and Ivana Išgum. Deep learning for multi-task medical image segmentation in multiple modalities. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 478–486. Springer, 2016.

[6] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Natalia Neverova, Alain Trémeau, and Christian Wolf. Multi-task, multi-domain learning: Application to semantic segmentation and pose regression. *Neurocomputing*, 251:68 – 80, 2017.

[7] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 240–248. Springer, 2017.

[8] Chaitanya Baweja, Ben Glocker, and Konstantinos Kamnitsas. Towards continual learning in medical imaging. *Medical Imaging meets NIPS Workshop*, 2018.

[9] Geoff Dougherty. *Digital image processing for medical applications*. Cambridge University Press, 2009.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 2017.

[12] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[13] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[14] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[15] R Guerrero, C Qin, O Oktay, C Bowles, L Chen, R Joules, R Wolz, MC Valdés-Hernández, DA Dickie, J Wardlaw, et al. White matter hyperintensity and stroke lesion segmentation and differentiation using convolutional neural networks. *NeuroImage: Clinical*, 17:918–934, 2018.

[16] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[19] Benjamin Bischke, Patrick Helber, Florian König, Damian Borth, and Andreas Dengel. Overcoming missing and incomplete modalities with generative adversarial networks for building footprint segmentation. *arXiv preprint arXiv:1808.03195*, 2018.

[20] Mohammad Havaei, Nicolas Guizard, Nicolas Chapados, and Yoshua Bengio. Hemis: Hetero-modal image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 469–477. Springer, 2016.

[21] Juan Eugenio Iglesias, Cheng-Yi Liu, Paul M Thompson, and Zhuowen Tu. Robust brain extraction across datasets and comparison with publicly available methods. *IEEE transactions on medical imaging*, 30(9):1617–1634, 2011.

[22] François Chollet et al. Keras. `https://keras.io`, 2015.

[23] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[24] Matthew Brett, Michael Hanke, Chris Markiewicz, Marc-Alexandre Côté, Paul McCarthy, Chris Cheng, Yaroslav Halchenko, Satrajit Ghosh, Demian Wassermann, Stephan Gerhard, and et al. nipy/nibabel: 2.3.1, Oct 2018.

[25] NiLearn. NiLearn: machine learning for neuroimaging in Python. `https://nilearn.github.io/`. [Online].

[26] Bradley Christopher Lowekamp, David T Chen, Luis Ibáñez, and Daniel Blezek. The design of simpleitk. *Frontiers in neuroinformatics*, 7:45, 2013.

[27] Torsten Rohlfing. Image similarity and tissue overlaps as surrogates for image registration accuracy: widely used but unreliable. *IEEE transactions on medical imaging*, 31(2):153–163, 2012.

[28] Adriënne M Mendrik, Koen L Vincken, Hugo J Kuijf, Marcel Breeuwer, Willem H Bouvy, Jeroen De Bresser, Amir Alansary, Marleen De Bruijne, Aaron Carass, Ayman El-Baz, et al. Mrbrains challenge: online evaluation framework for brain image segmentation in 3t mri scans. *Computational intelligence and neuroscience*, 2015:1, 2015.

[29] Chris A Cocosco, Vasken Kollokian, Remi K-S Kwan, G Bruce Pike, and Alan C Evans. Brainweb: Online interface to a 3d mri simulated brain database. In *NeuroImage*. Citeseer, 1997.

[30] MICCAI. WMH Segmentation Challenge. `http://wmh.isi.uu.nl/`, 2017. [Online].

[31] MICCAI. MRBrainS18. `http://mrbrains18.isi.uu.nl/`, 2018. [Online].

[32] MICCAI. MICCAI 2012 Challenge on Multimodal Brain Tumor Segmentation. `http://www.imm.dtu.dk/projects/BRATS2012`, 2012. [Online].

[33] Marcel Prastawa, Elizabeth Bullitt, and Guido Gerig. Simulation of brain tumors in mr images for evaluation of segmentation efficacy. *Medical image analysis*, 13(2):297–311, 2009.

[34] Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[36] Sergi Valverde, Arnau Oliver, Mariano Cabezas, Eloy Roura, and Xavier Lladó. Comparison of 10 brain tissue segmentation methods using revisited ibsr annotations. *Journal of Magnetic Resonance Imaging*, 41(1):93–101, 2015.