

# Segmentación de Tumores Cerebrales con CNNs

Iván Penedo

Máster de Investigación en Inteligencia Artificial

Universidad Intenacional Menéndez Pelayo

Madrid, España

100012493@alumnos.uimp.es

**Abstract**—Este trabajo se centra en la aplicación de técnicas de Aprendizaje Profundo para segmentar tumores cerebrales de una manera precisa, utilizando un conjunto de datos de resonancias magnéticas (MRI) que incluye secuencias T1, T2, T1ce y FLAIR, acompañadas de sus correspondientes máscaras de segmentación binaria. Además, se sugieren y contrastan dos arquitecturas de redes neuronales convolucionales (CNNs) desarrolladas de manera *ad-hoc*. El primer modelo establece una base sobre la que posteriormente el segundo modelo mostraría un incremento significativo a través de la aplicación de estrategias sofisticadas. Concretamente, se utilizan métodos como el *transfer learning*, la regularización a través del uso de *dropout*, y la ampliación del conjunto de datos para evitar contra el sobreajuste de los modelos. Asimismo, el proceso de entrenamiento se mejora a través del algoritmo Adam, el método de *early stopping* para prevenir el sobreentrenamiento, y unas tasas de aprendizaje variables, donde se utilizan tasas de aprendizaje más altas en las fases iniciales que se disminuyen en las finales. En última instancia, la meta es superar un *baseline* de 0.60 en el coeficiente de similitud Dice, valorando la habilidad de los modelos para generar segmentaciones exactas y localizar los tumores en las MRI del conjunto de datos. Este proyecto evidencia la implementación práctica de los conocimientos obtenidos sobre Aprendizaje Profundo para dar solución a un problema de segmentación de imágenes médicas.

**Index Terms**—Deep Learning, segmentación, MRI, CNNs

## I. INTRODUCCIÓN

En las últimas décadas, varias técnicas de Aprendizaje Profundo se han utilizado en el ámbito sanitario. Una de las tareas más comunes es el uso de secuencias de imágenes por resonancias magnéticas MRI para detectar tumores cerebrales en los pacientes.

### A. Conjunto de datos

Estas imágenes son de las más utilizadas para tareas de detección y segmentación, donde dadas estas secuencias, se pretende obtener una máscara de la región en la que se localiza el tumor. Además de esto, la maquinaria con la que se realizan las resonancias, utilizando diferentes configuraciones y parámetros, permiten obtener diferentes tipos de imágenes, siendo algunas de ellas útiles para capturar la anatomía del cerebro como las “T1”, apreciar roturas de la barrera hemoencefálica “T1ce”, ayudar a identificar tejidos inflamados “T2” y para visualizar edemas y áreas específicas del tumor “FLAIR”, como se observa en la figura 1.

Con ayuda de expertos radiólogos, además de estas secuencias, se han utilizado imágenes con la máscara en forma

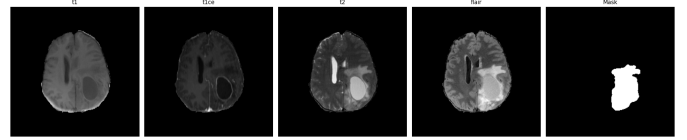


Fig. 1. Secuencias y máscara dada de una de las instancias del conjunto de datos.

de segmentación binaria de los tumores de cada una de las secuencias.

### B. Objetivos

Mediante el uso de estos datos, se han realizado y entrenado modelos de redes convolucionales CNN para dadas las imágenes de las resonancias, se puede obtener la segmentación binaria correspondiente a la región tumoral. Para ello, se han utilizado dos subconjuntos ya divididos donde el primero de ellos se utiliza para entrenar el modelo y el otro para validar la calidad de este. Esta se estimará utilizando la métrica del coeficiente de similitud *dice*, la cual es comúnmente utilizada para cuantificar la similitud de dos conjuntos.

$$dice = \frac{2|A \cap B|}{|A| + |B|}$$

En este caso, se aplica sobre las segmentaciones dadas por los radiólogos *A* y por las predichas por los modelos entrenados *B* y el objetivo será superar un *baseline* cuyo *dice* = 0.60.

## II. MÉTODOS

Para poder aportar solución a este problema, se han diseñado dos modelos de redes convolucionales. En la primera de ellas se proporciona una red más simple con el objetivo de observar cómo se comporta una red con nivel de complejidad bajo enfrentándose una tarea de segmentación de máscaras binarias. Por otra parte, se ha diseñado una red *ad-hoc* basada en ideas de un modelo del estado del arte para observar la eficiencia de técnicas avanzadas de Aprendizaje Profundo.

Para ambos modelos, se ha utilizado en la fase de entrenamiento el conjunto correspondiente normalizado aplicándole técnicas de *data augmentation* para evitar el sobreajuste de los modelos. Por ello, se han añadido nuevas MRI mediante la rotación y volteo de las imágenes originales, obteniendo así un conjunto de datos de entrenamiento más amplio y diverso.

La función de pérdida utilizada para ambos entrenamientos ha sido la inversa de la métrica *dice*, de manera que se priorice la maximización de dicha métrica durante el entrenamiento, en lugar de minimizar el valor de la función de entropía cruzada binaria o `binary_crossentropy`.

Asimismo, debido a los recursos de los que se disponen, se ha realizado una división del lotes de tamaño 16 sobre los conjuntos de entrenamiento y validación con el fin de optimizar el uso de la memoria, así como la eficiencia del modelo durante el entrenamiento. También se han normalizado los valores de las imágenes para mejorar la estabilidad de los entrenamientos.

#### A. CCN simple

Para el primer modelo creado (CNN1), se ha creado una primera CNN simple para poder comprobar la calidad de las soluciones generadas con una red que sea rápida en la segmentación de imágenes. Por lo tanto, se ha definido una arquitectura personalizada ad-hoc en lugar de usar modelos preentrenados, los cuales además de requerir una gran cantidad de recursos computacionales, no suelen utilizar 4 canales de entrada como es el caso actual con las secuencias del MRI.

Con el fin de prevenir el sobreajuste del modelo, se han aplicado *dropout* tras las capas de *pooling* para reducir la coadaptación de las neuronas, cuya configuración se ha dejado parametrizada con la finalidad de realizar un estudio de su eficacia. Este modelo se ha entrenado aplicando *early stopping* para detener el entrenamiento si no mejora la métrica en validación y evitar así el sobreentrenamiento de este, mientras que el algoritmo de optimización Adam mejora la adaptabilidad y el buen rendimiento en tareas de visión por computador.

#### B. CNN profunda

En cuanto al segundo modelo (CNN2), se ha diseñado una arquitectura más profunda y sofisticada, la cual se encuentra inspirada en una arquitectura tipo U-Net comúnmente utilizada para tareas de segmentación de imágenes, la cual implementa un enfoque codificador-decodificador con múltiples niveles de convoluciones, *max-pooling* y *skip connections*. Esta organización permite extraer características especiales y complejas en la etapa de codificación y luego recuperar la resolución espacial en la etapa de decodificación, combinando información de bajo y alto nivel a través de conexiones directas entre capas simétricas del codificador y decodificador. También se ha dejado parametrizadas las capas de *dropout* para analizar su aportación en cada bloque tanto del codificador como del decodificador, reduciendo la dependencia entre neuronas durante el entrenamiento.

Desde el punto de vista del entrenamiento, se ha aplicado de nuevo *early stopping* para detener el entrenamiento si la pérdida deja de mejorar, el mismo algoritmo de optimización Adam que destaca por su capacidad de adaptación y buena convergencia en tareas complejas, y el mismo entrenamiento en lotes de tamaño 16. Asimismo, se ha añadido un entrenamiento por fases, donde la primera de ellas cuenta con un *learning rate* superior de  $1 \times 10^{-4}$  para poder realizar

un entrenamiento más rápido de la red, para posteriormente congelar las primeras capas de esta y se ajusta un nuevo *learning rate* menor de  $1 \times 10^{-6}$  para refinar la parte superior del modelo sin alterar lo aprendido en la base, simulando la técnica de *fine-tuning* sin utilizar un modelo preentrenado.

### III. RESULTADOS

En cuanto al primero de los modelos, aun siendo un diseño más simple, se han conseguido buenos resultados. En la figura 2 se puede observar que casi todas las configuraciones probadas con diferentes valores de *dropout* han superado el baseline de 0.6. La única configuración que no ha conseguido superarlo ha sido el que cuenta con un *dropout* de 0.2.

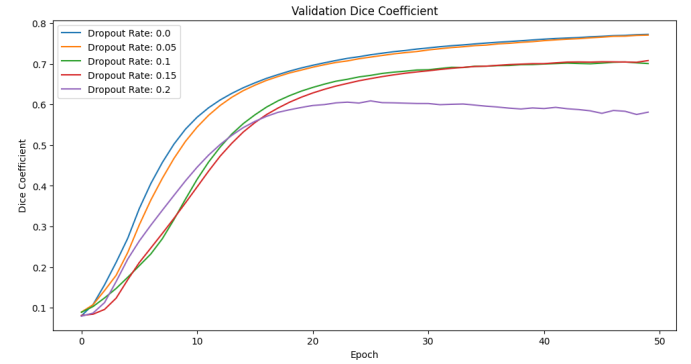


Fig. 2. Resultados del entrenamiento de CNN1

Además, se puede observar también en esta figura cómo al aumentar el valor de *dropout* en el entrenamiento se consigue un rendimiento menor. Esto podría deberse a un uso excesivo de las capas de *dropout* y que, por lo tanto, no sea necesario añadir tantas capas al modelo.

En cuanto al modelo CNN2, se puede comprobar en la figura 3 como todos los entrenamientos con diferentes configuraciones de dropout obtienen resultados similares, superiores a 0.9 en el coeficiente *dice*. También se puede observar que el utilizar un valor superior en la primera mitad de las etapas de entrenamiento ha supuesto que el este no sea progresivo, y por lo tanto le cueste estabilizarse.

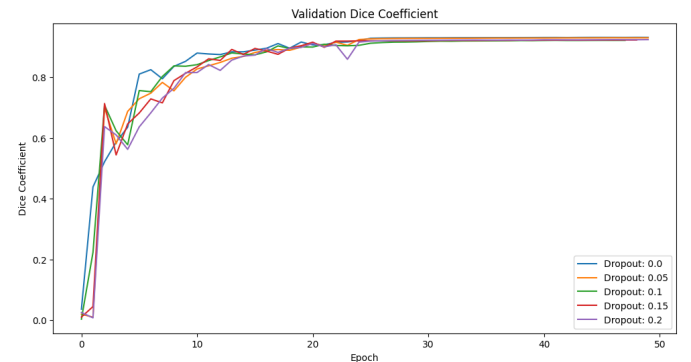


Fig. 3. Resultados del entrenamiento de CNN2

Para comprobar que esto no se debiera a un sobreentrenamiento del modelo, se ha realizado un entrenamiento con un menor número de etapas. En la figura 4 se puede observar una comparativa del rendimiento del modelo entrenado en 50 y 25 etapas, y se puede comprobar como con 25 etapas el modelo sigue sin mejorar progresivamente durante las primeras 12 etapas con mayor *learning rate* y, además, le cuesta mejorar su rendimiento tras estas.

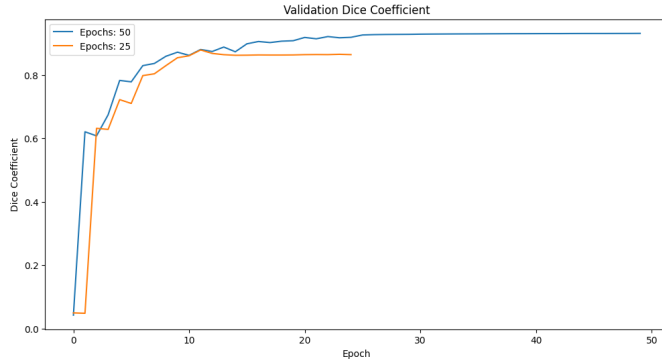


Fig. 4. Comparativa de CNN2 con diferentes etapas de entrenamiento

Aunque el modelo CNN2 con menor entrenamiento tenga un peor rendimiento que el modelo CNN2 entrenado con el doble de etapas, sigue demostrando ser un modelo prometedor, ya que con pocas etapas consigue superar el rendimiento del modelo CNN1.

En cuanto a las segmentaciones predichas, en la figura 5 se pueden comparar las predicciones de las mejores configuraciones de cada uno de los modelos diseñados.

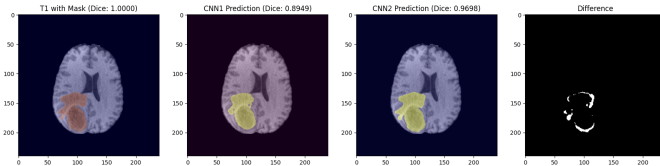


Fig. 5. Comparativa de las predicciones realizadas por ambos modelos.

En la predicción del primer modelo se puede comprobar como se ha perdido un poco de información con respecto a la máscara, ya que, como se ha comentado anteriormente, este modelo no es capaz de definir los bordes de las regiones tumorales de las MRI. Por otra parte, la predicción del segundo modelo muestra cómo es capaz de identificar y perfeccionar correctamente los contornos de los tumores con respecto a la máscara original. Estas diferencias pueden observarse más claramente en la imagen de la derecha de la figura.

#### IV. CONCLUSIONES

Los resultados obtenidos muestran que ambos modelos propuestos son capaces de superar el *baseline* establecido, aunque con diferencias notables en su rendimiento. El modelo CNN1, pese a su simplicidad, ha logrado un desempeño

aceptable, superando consistentemente el *baseline* salvo en una configuración con elevado *dropout*.

Por otro lado, el modelo CNN2 ha demostrado ser considerablemente más robusto y preciso, alcanzando valores del coeficiente dice superiores a 0.9 en todas las configuraciones probadas durante 50 etapas. Sin embargo, se ha observado que la estrategia de entrenamiento influye significativamente en la estabilidad del aprendizaje, especialmente en las primeras etapas con mayor *learning rate*.

No obstante, el *dropout* que se ha añadido a los modelos no ha demostrado ser especialmente efectivo, ya que la mejor configuración de ambos modelos ha sido con un *dropout* de 0.0.

A nivel de segmentación, CNN2 ha mostrado una mayor capacidad para identificar y delinear correctamente los contornos tumorales con comparación con CNN1, lo que demuestra que la CNN con arquitectura profunda ofrece una solución más precisa y fiable para este tipo de tareas.

Durante el desarrollo y evaluación de los modelos, se han identificado diversas dificultades que han influido en el proceso. Algunas de estas han estado relacionadas con la lectura de las imágenes y la creación de los distintos *datasets*, especialmente por el tratamiento de imágenes dentro del *notebook*. Por otra parte, el uso de Google Colab en su versión gratuita ha supuesto un reto, puesto que no se disponía del tiempo de computo suficiente como para ejecutar de seguido todo el *notebook* y se ha tenido que posponer las ejecuciones por los límites establecidos.

Finalmente, se proponen varias líneas de mejora y ampliación del trabajo. Entre ellas, se considera interesante explorar una adecuada configuración para aplicar *learning rates* variables en los entrenamientos, para poder entrenar rápidamente los modelos en las primeras etapas y posteriormente perfeccionar los resultados que se obtienen en etapas posteriores, ya que el valor de  $1 \cdot 10^{-4}$  se considera demasiado elevado. Por otra parte, también sería interesante analizar el efecto del *dropout* utilizando menos capas de este tipo, ya que puede que se haya llevado cabo un uso excesivo de esta técnica en los modelos y por ese motivo no se haya encontrado especialmente útil su uso.

En resumen, en este trabajo se ha conseguido diseñar dos redes neuronales convolucionales para la segmentación de imágenes médicas, ambas con un buen rendimiento en comparación al *baseline*. A pesar de las limitaciones y problemas encontrados, se ha logrado superar el rendimiento de referencia y establecer una base sólida para futuras mejoras.

#### V. ENLACES

Para este trabajo se ha utilizado el entorno de Google Colab, donde el código desarrollado puede encontrarse en este enlace.