Proyecto de Sistemas de Gestión y Business Intelligence:

Reconstrucción de Audio a partir de MRI

Pablo González Santamarta

universidad de león

Escuela de Ingenierías
Industrial, Informática y Aeroespacial

universidad de león

Índice:

Introducción:

Con el auge de las inteligencias artificiales generativas se nos ha encargado en esta asignatura desarrollar el proyecto de una aplicación que utilice las ventajas de éstas para resolver un problema a nuestra elección. En la asignatura se utiliza OSF para documentar el desarrollo del proyecto y finalmente se sube el resultado a final del curso a github, desde donde se hace público.

El objetivo final de la asignatura es adquirir conocimientos de las vanguardias informáticas actuales e idealmente sembrar la semilla de un proyecto que en un futuro se coseche como un trabajo académico formal.

El desarrollo de éste trabajo tuvo múltiples etapas distintas en las que seguí varias direcciones avanzando poco a poco hasta encontrar el problema que quería resolver y el método que utilizar para ello. Cada etapa del proyecto me aportó un conocimiento esencial para evolucionar a la siguiente y la cronología de éstas etapas se explica en el próximo capítulo.

Problema a resolver:

Problemas planteados:

Enfoque de problemas buscados a resolver:

La decisión del problema a resolver para el proyecto fue difícil de escoger debido al amplio abanico al que se puede aplicar las inteligencias artificiales generativas. Supe desde el principio que quería hacer un proyecto ambicioso sobre un tema que fuese de especial interés para mí. Siempre me ha interesado la aplicación de la inteligencia artificial para revolucionar la sociedad inspirado en ficción como *Sueñan los Androides con Ovejas Eléctricas - Philip K. Dick (1968)*, *Ghost in the Shell - Masamune Shirow (1989-1991)* o *Yo, Robot - Isaac Asimov (1950)*.

Los sueños infantiles de androides, cyborgs y conceptos que difuminan la barrera que separa el hombre de su creación se concretaron en un interés por la aplicación de la IA a los prostéticos y la robótica, temas para los cuales mi conocimiento aún es inmaduro. Éste proyecto es para mí la oportunidad de crecer considerablemente en la dirección que me apasiona.

Ésta pasión me llevó a adentrarme en el mundo del Deep Learning y la Visión por Computador, de la cual adquirí conocimientos con clases en la Universidad de Granada y lecturas. En concreto: *Neural Networks from Scratch in Python - Harrison Kinsley, Daniel Kukieła (2020)*, *Deep Learning for Coders with fastai & PyTorch - Sylvain Gugger, Jeremy Howard (2020)*, *Computer Vision: Algorithms and Applications, 2nd Edition - Richard Szeliski (2022)* y *Multiple View Geometry in Computer Vision 2nd Edition - Richard Hartley, Andrew Zisserman (2004)*. Contando ya con éstos conocimientos mi mejor baza era centrarme en problemas que requiriesen visión artificial o pudiesen expresarse con esas técnicas.

Otro factor que me empujó a centrarme en la visión por computador fué el artículo [1] que el profesor Enrique López González me envió cuando le comuniqué mi interés por la visión. Aquí surgió mi primera idea de proyecto: el uso de inteligencias artificiales generativas multi-modales (MMICL) para el diagnóstico utilizando imágenes e informes médicos.

No fué hasta que, de nuevo gracias a Enrique López González, encontré el artículo [2] cuando me decidí a centrarme en imágenes neurales. La neurología es un campo de la medicina con mucho por descubrir y prometedor que me interesa y tiene una relación que se está estrechando cada vez más con la Inteligencia Artificial como podemos ver en casos como NeuraLink. Una vez me decidí por esto comenzó mi viaje por el campo de la Neurología.

Contando ya con un modelo y un tema sobre el que desarrollar necesitaba encontrar un dataset para escoger el problema que resolver, ya que esto era más sencillo que inventarme el problema y luego buscar los datos. Pronto encontré el repositorio de datos experimentales OpenNeuro.

Primer problema planteado: Herramienta de ayuda en la diagnosis de dislexia en base a escáneres de función cerebral fMRI durante lectura en voz alta

En OpenNeuro se guardan datos experimentales de neurología en una jerarquía estandarizada llamada BIDS y cómo leer las imágenes de los experimentos. Explorando el dataset que me llamó la atención fue MorphoSem [3], el cual guarda datos de un experimento en el que los sujetos deben leer en voz alta mientras se realiza el escáner en MRI, los sujetos son 40, siendo 20 de ellos disléxicos. Además de guardar los datos estándar para el formato BIDS también tiene guardadas grabaciones de las lecturas de los sujetos.

Con éste dataset la primera idea que tuve fue hacer una **herramienta de ayuda en la diagnosis de dislexia en base a escáneres de función cerebral durante la lectura en voz alta**, la cual daría una predicción de si el paciente es disléxico al introducir los escáneres realizados.

Inmediatamente me dí cuenta de que los datos experimentales contienen muy poca información en formato de texto, y de cada paciente no se dice nada más que si es hombre o mujer y su edad. Por tanto, para resolver éste problema una arquitectura como MMICL es innecesaria. Y mi siguiente objetivo fue construir mi propio modelo que codifique la información visual de los escáneres y envíe el vector resultante a un perceptrón que de la predicción final.

Segundo problema planteado: reconstruir audio de grabaciones de lectura a partir de escáneres de función cerebral fMRI del lector

Con el tiempo aprendí sobre las arquitecturas de transformadores [5], las cuales se originaron en el ámbito del procesamiento de lenguaje natural y tuvieron un gran éxito. Los transformadores tienen una propiedad muy interesante y es que su efectividad también es notoria codificando información visual [6].

Aprender sobre los transformadores de visión (ViT) abrió para mí una nueva madriguera de investigación en la que aprendí sobre múltiples arquitecturas basadas en los ViT. Entre ellas, por supuesto está MMICL [1]. Al final, acabé encontrando una aplicación de los transformadores aún más fascinante: La reconstrucción y síntesis de información.

Éste descubrimiento y la presencia de las grabaciones inspiró la idea de tratar de reconstruir las grabaciones de audio en lugar de diagnosticar dislexia, es decir, **reconstruir audio de grabaciones de lectura a partir de escáneres de función cerebral fMRI del lector**.

Problema final: reconstrucción de sonidos naturales a través de escáneres de actividad cerebral fMRI

A medida que comprendía el dataset MorphoSem y los fMRI en general el problema de reconstrucción de la lectura a partir de escáneres de función cerebral me parecía tener menos sentido. El audio reproducido al leer en voz alta es más dependiente de la anatomía del sujeto que de su proceso cerebral. Además los fMRI tienen una resolución temporal demasiado baja como para captar con precisión la actividad neuronal de una actividad tan compleja como el habla.

Volví a revisar OpenNeuro en busca de un dataset que tuviese mejores datos para resolver el problema y finalmente encontré algo que consideré mucho mejor. El dataset “High-res gradient echo EPI and 3D GRASE data of auditory cortex” [7].

Este dataset contiene datos experimentales de tres sujetos que se han sometido a múltiples escáneres fMRI durante los cuales se han reproducido audios de un segundo de un conjunto de 144 audios. De los escáneres realizados se proporcionan exclusivamente volúmenes en los que sólo es visible el córtex auditivo. Este dataset era ideal para resolver el problema de **reconstrucción de sonidos naturales a través de escáneres de actividad cerebral fMRI**.

Definición del problema:

El problema que definitivamente se trata de resolver a lo largo de éste proyecto es la reconstrucción de sonidos naturales a partir de escáneres fMRI. Esto quiere decir que se desarrollará una aplicación inteligente la cual generará un sonido a partir de una imágen fMRI de un cerebro que se corresponderá con el sonido que estaba escuchando el sujeto en el momento de la adquisición de la imágen. Éste es un problema del cuál no he encontrado papeles que hablen directamente de su solución.

Que el sonido se le llame natural quiere decir que los sonidos son del tipo que uno puede escuchar en la vida cotidiana, como sonidos de animales, obras, herramientas, voces…

Existen dos restricciones al alcance del problema:

1. Los sonidos reconstruidos durarán 1 segundo, al igual que en el dataset [7].
2. Los sonidos se reconstruirán a partir de una única imágen 2d perteneciente a un volumen obtenido por fMRI.

Para resolver éste problema hay múltiples pasos que se deben seguir:

1. Procesar y representar el audio de forma reproducible y única
2. Procesar y representar las imágenes fMRI de forma reproducible y única
3. Encontrar las imágenes fMRI que corresponden a cada audio
4. Reconstruir audio a partir de su representación
5. Encontrar las correspondencias entre las representaciones fMRI y las representaciones de audio
6. Reconstruir audio a partir de la representación de fMRI que le corresponde

Generación de Ideas

Para representar el audio de forma reproducible y única debo representar éste en forma matricial como puede ser un espectrograma o una onda y extraer de ésta representación un conjunto de características más simples que permitan identificarlo perdiendo la mínima información posible. Para ello debo escoger el espectrograma adecuado, procesarlo y extraer las características adecuadas. La extracción de las características se puede hacer utilizando técnicas de deep learning como el uso de redes neuronales.

Para preprocesar las imágenes fMRI y representarlas de forma reproducible y única debo aprender cómo se lee un archivo fMRI, qué datos contiene y qué significado tienen. Después puedo utilizar mis conocimientos de Visión por Computador para extraer sus características. Entre éstas principalmente se usarán técnicas de deep learning como CNN o al anteriormente mencionado ViT

Para encontrar las imágenes fMRI que corresponden a cada audio debo conocer qué es una imágen fMRI, cómo se obtiene y cuándo se reproducen los audios. Esto último viene incluído dentro del dataset.

Para la reconstrucción del audio debo encontrar un método inverso que me devuelva el espectrograma original a partir de las características extraídas del audio.

Para encontrar las correspondencias entre las representaciones debo investigar métodos que permitan hacer las dos representaciones de características de audios y fMRI correspondientes iguales.

Una vez consiga hacer las representaciones de audios y fMRI correspondientes iguales, al igual que se reconstruye el audio a partir de su representación, podría hacer reconstruir el mismo audio a partir de la representación de fMRI.

Identificación final de las Ideas Generadas

Para llevar a cabo la representación del audio es necesario que conozca procesamiento de señales, espectrogramas, transformaciones de fourier y otras técnicas necesarias. Además de eso debo conocer casos de deep learning en los que se trabaje con datos de audio exitosamente.

En el procesamiento de las imágenes fMRI, debo conocer métodos y arquitecturas de procesamiento de imágenes efectivas. También tengo que aprender cómo leer archivos de fMRI e interpretar y extraer los datos contenidos en ellos y procedimientos comunes de procesamiento de dichas imágenes para deep learning. Además debo conocer casos de deep learning y arquitecturas que han sido especialmente efectivas con imágenes de fMRI.

En la correspondencia de las imágenes y los audios debo saber desarrollar un algoritmo que permita corresponder las imágenes y los sonidos según el tiempo. La información necesaria para esto está guardada en los propios archivos del dataset y debo aprender a obtenerla e interpretarla correctamente para el desarrollo del algoritmo.

En la reconstrucción del audio debo de aprender métodos de deep learning que permitan reconstruir un audio a partir de una representación de sus características, como arquitecturas encodificador-decodficador, van, vqgan, redes adversariales, etc., y saber cómo escribir audio en un archivo para que pueda ser escuchado.

Para las correspondencias entre las representaciones de imágenes y audio debo de haber aprendido cómo representar ambos y técnicas de deep learning que ayuden a enseñar a una red neuronal a generar uno a partir del otro como el transfer learning.

Finalmente debo aprender a desarrollar y entrenar una arquitectura que permita generar audio a partir de input visual.

Formulación de los objetivos del aprendizaje

Debo de conocer y aprender a utilizar los siguientes conceptos y herramientas

1. Herramientas de Programación:
    a. Google Colab
    b. Librerías de python
        i. nibabel
        ii. py-bids
        iii. librosa
        iv. numpy

        v. matplotlib
        vi. pytorch
        vii. pandas
    c. Matlab
2. Procesamiento de señales de Audio
    a. Transformaciones de Fourier
    b. Espectrograma
    c. Espectrograma de MEL
    d. MFCC
3. Diseño y entrenamiento de arquitecturas generativas
    a. Transformadores de visión
        i. Capas de atención propia
    b. Codificadores
    c. Decodificadores
    d. Diccionario de códigos
    e. Transfer Learning
    f. Congelación de capas
    g. Redes Adversariales
    h. Arquitectura VAN
    i. Arquitectura VQGAN
4. Procesamiento de archivos fMRI
    a. Estructura BIDS
    b. Formato NIFTI
    c. Señal BOLD
    d. Interpretación de metadatos
    e. SPM
5. Comprensión de datasets de neurología:
    a. Formato
    b. Modalidades
    c. Tipos de estudios

Investigación

Partiendo de los conocimientos ya adquiridos mientras acababa de definir el problema que quería resolver, empezó mi búsqueda de qué son los MRI, cómo se obtienen, qué miden y cómo se interpretan.

Llegado este punto ya había aprendido que es BIDS a través de la página web (https://bids.neuroimaging.io). Y sabía dónde encontrar los archivos MRI y qué pretendían mostrar cada uno de ellos. Pero estos se encuentran en un formato que no se puede abrir con una aplicación cualquiera de visualización de imágenes, este es el formato NIfTI, del cual aprendí más en (https://nifti.nimh.nih.gov). Tras una rápida búsqueda en google descubrí que para visualizar tales archivos se debe utilizar librerías de programación o software especializado. En el caso de python existe la librería nibabel que permite abrir estos archivos y acceder a la imágen en forma de matriz y a sus metadatos como un diccionario. Para procesar y visualizar las imágenes neurales no descubrí hasta más tarde que es común utilizar el software SPM (https://www.fil.ion.ucl.ac.uk/spm/), demasiado tarde

como para poder utilizarlo, pero imprescindible, me habría ahorrado mucho trabajo y además le habría dado mucha mayor calidad.

Tras abrir los archivos de MRI pude visualizarlos utilizando matplotlib y ver qué características tenían en el diccionario de metadatos. Ahí descubrí características esenciales para la solución del problema como el SliceTiming, que indica cuánto tiempo pasa entre la obtención de cada imágen de cada volúmen y es un indicador de la resolución temporal del escáner y permite corresponder las imágenes con el momento en el que se reproducen los sonidos que queremos reconstruir.

Sin embargo, en la serie de videos de youtube de introducción a los MRI (https://www.youtube.com/watch?v=ZL-Tr1KSMKY&list=PLfXA4opIOVrGHncHRxI3Qa5GeCSudwmxM) de los cuales vi del 1 al 10b muestra qué mide el tipo de MRI con el que trabajo en éste proyecto y cómo eso refleja la actividad cerebral. A grandes rasgos éstas dos cosas son la concentración de oxígeno y la señal BOLD. Cuando una parte del cerebro se activa se desplaza una mayor cantidad de sangre rica en oxígeno hacia ella, lo cual se refleja en el MRI como un ligero aumento de brillo en la zona activada. Éste aumento de brillo no alcanza su máximo hasta pasados 5 segundos de media. Con esto y lo anterior por fin tenemos una guía de qué imágen coger para cada sonido.

Otra parte que necesitaba aprender es la lectura del sonido y su representación en transformada de fourier, espectrogramas y mfccs. La serie de videos (https://www.youtube.com/watch?v=iCwMQJnKk2c&list=PL-wATfeyAMNqIee7cH3q1bh4QJFAaeNv0) fue especialmente útil para entender de forma práctica cómo utilizar éstas representaciones. Además gracias a esa lista descubrí la librería librosa para python, que implementa los algoritmos para leer y transformar el audio de forma que se dispone de ellos utilizando una sola línea de código.

Lo que queda por investigar tras aprender esto es qué arquitectura es mejor utilizar y cómo se implementa y entrena. Como los problemas de audio suelen tener soluciones similares a los problemas de visión en deep learning [8], busqué artículos sobre un problema muy similar, la reconstrucción de imágenes naturales a partir de escáneres MRI, del cual oí hablar por las noticias el verano de 2023. Encontré el artículo [9] que plantea múltiples métodos que se han utilizado para resolver el problema utilizando deep learning. Gracias a éste artículo aprendí sobre las utilidades de las arquitecturas codificador-decodificador y las redes adversariales para la resolución de problemas de síntesis de imágenes. La que mayor interés me produjo fue la que en el artículo es nombrada como RenD-VAE/GAN [10], una especialización de la arquitectura VAE-GAN (Varational AutoEncoder Generative Adversarial Networks) [11].

Buscando ejemplos de implementaciones de esta arquitectura encontré una variación de esta llamado VQ(VAE)-GAN (Vector Quantized Varational AutoEncoder Generative Adversarial Networks) [12]. Arquitectura que se presenta como solución a problemas que tenía VAE-GAN.

Modifiqué la implementación (https://www.youtube.com/watch?v=_Br5WRwUz_U) en base a trabajo que encontré utilizando VQGAN para síntesis de audio a partir de imágenes [13] y

[10]. En ambos artículos se mencionan mejoras a la arquitectura que implementaré en el futuro.

Solución al Problema

La solución al problema acaba siendo una arquitectura VQGAN en la que se usando dos codificadores en el entrenamiento para explotar la técnica de transfer-learning. Ésta solución está implementada en un notebook de google colab en python sin finalizar.

El dataset de entrenamiento se procesa antes del entrenamiento si así se indica en los parámetros del programa.

En el procesamiento se leen todos los archivos de audio del experimento y se crea su espectrograma de MEL. Después, se añaden las columnas de acolchamiento necesarias para que el tamaño del audio sea divisible entre 16 (2 elevado al número de capas de downsample en el codificador). Todos ellos se guardan en el sistema de archivos para ser leídos en el entrenamiento y ahorrar memoria de la GPU.

El procesamiento de los MRI consiste en encontrar el slice correcto para cada sonido descrito en la tabla de eventos de cada ejecución en cada sesión de cada sujeto. Los slices son recortados al tamaño seleccionado (208) y se guardan en el sistema de archivos para ser leídos en el entrenamiento y ahorrar memoria de la GPU.

Durante este proceso se crea una lista con los nombres de los archivos de los MRI procesados, cuando se requiere obtener un dato para una iteración se lee la imágen MRI y el audio que le corresponde que está indicado en el propio nombre del archivo MRI.

En base a una serie de parámetros seleccionados por el usuario, se crea un modelo y ,si el usuario lo desea y existen, carga los parámetros de un modelo igual entrenado. El modelo entrena en tres fases 5 épocas recorriendo todo el dataset (debido a la escasez de datos y la informalidad y muy probable fracaso del proyecto no preparé el dataset para evaluar el modelo). Cada iteración del entrenamiento se pasa el espectrograma de MEL generado y el original a un discriminador cuyo objetivo es discernir cuál fue generado. Esto junto con una medida de la diferencia respecto la imágen original evaluarán la efectividad del modelo en esa iteración y éste se actualizará según éstas dos métricas.

En la primera fase el modelo aprende a reconstruir los espectrogramas de MEL de cada audio. En la segunda se sustituye el codificador de audio por el codificador de MRI y se congela el diccionario y el decodificador para entrenar el codificador de MRI. En la última fase se descongela el diccionario y el decodificador y se congela el codificador para finalizar entrenar el generador.

Exposición de los resultados

El modelo resultado final tiene errores relacionados con la generación de sonido a partir de la fase 2, ya que genera matrices del tamaño de las imágenes MRI en lugar de imágenes del tamaño deseado del sonido. Ésto es debido a que para un input de tamaño (CxWxH) se crea un vector de (KxW/2^4xH/2^4), donde K es la dimensión del diccionario y W y H son divididos en dos 4 veces a causa de las cuatro capas dowsample del encodificador. Una solución a ésto es introducir el input como un vector de tamaño fijo de secciones de tamaño fijo de la imágen original, dando una reconstrucción de tamaño fijo. Ésta solución también permite eliminar preocupaciones como hacer que los espectrogramas de MEL sean de dimensiones divisibles entre 4, lo cual requiere crear información falsa (acolchamiento de 0)

Además de los problemas que nos impiden siquiera probar la reconstrucción de audio a partir de MRI, el propio reconstructor de Audio es poco efectivo. Esto se debe a la severa escasez de datos, a una posible mala configuración de los parámetros y a que el discriminador utilizado, VGG, está entrenado con imágenes naturales y no con espectrogramas. En el artículo [13] se presenta una alternativa mejor.

Los futuros problemas que se espera encontrar en el desarrollo son una mala reconstrucción del audio a partir de los MRI debido a que no están procesados adecuadamente y la falta de datos.

Bibliografía

1. Zhao, H., Cai, Z., Si, S., Ma, X., An, K., Chen, L., ... & Chang, B. (2023). Mmicl: Empowering vision-language model with multi-modal in-context learning. *arXiv preprint arXiv:2309.07915*.
2. Jin K. Kim, Michael Chua, Mandy Rickard, Armando Lorenzo,
3. ChatGPT and large language model (LLM) chatbots: The current state of acceptability and a proposal for guidelines on utilization in academic medicine, Journal of Pediatric Urology, Volume 19, Issue 5, 2023, Pages 598-604, ISSN 1477-5131, https://doi.org/10.1016/j.jpurol.2023.05.018. (https://www.sciencedirect.com/science/article/pii/S1477513123002243)
4. Eddy Cavalli and Valérie Chanoine and Johannes C. Ziegler (2023). MorphoSem. OpenNeuro. [Dataset] doi: doi:10.18112/openneuro.ds004786.v1.0.1
5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.
6. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z. H., ... & Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 558-567).
7. Michelle Moerel and Essa Yacoub (2023). High-res gradient echo EPI and 3D GRASE data of auditory cortex. OpenNeuro. [Dataset] doi: doi:10.18112/openneuro.ds004814.v1.0.0
8. Gugger, S., Howard, J. (2020). Deep Learning for Coders with fastai and Pytorch. O'Reilly Media.
9. Front. Neurosci., 20 December 2021 Sec. Brain Imaging Methods Volume 15 - 2021 | https://doi.org/10.3389/fnins.2021.795488
10. Ren, Z., Li, J., Xue, X., Li, X., Yang, F., Jiao, Z., & Gao, X. (2021). Reconstructing seen image from brain activity by visually-guided cognitive representation and adversarial learning. NeuroImage, 228, 117602.
11. Razghandi, M., Zhou, H., Erol-Kantarci, M., & Turgut, D. (2022, May). Variational autoencoder generative adversarial network for Synthetic Data Generation in smart home. In *ICC 2022-IEEE International Conference on Communications* (pp. 4781-4786). IEEE.
12. Oord, A. V. D., Vinyals, O., & Kavukcuoglu, K. (2017). Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*.
13. Iashin, V., & Rahtu, E. (2021). Taming visually guided sound generation. *arXiv preprint arXiv:2110.08791*.