

Modelo de predicción de la Progresión de la Fibrosis Pulmonar



Enric Rovira Meléndez

Máster Inteligencia Artificial -

Kschool

31-8-2020

1. Introducción:

- Motivación y Objetivos del proyecto:

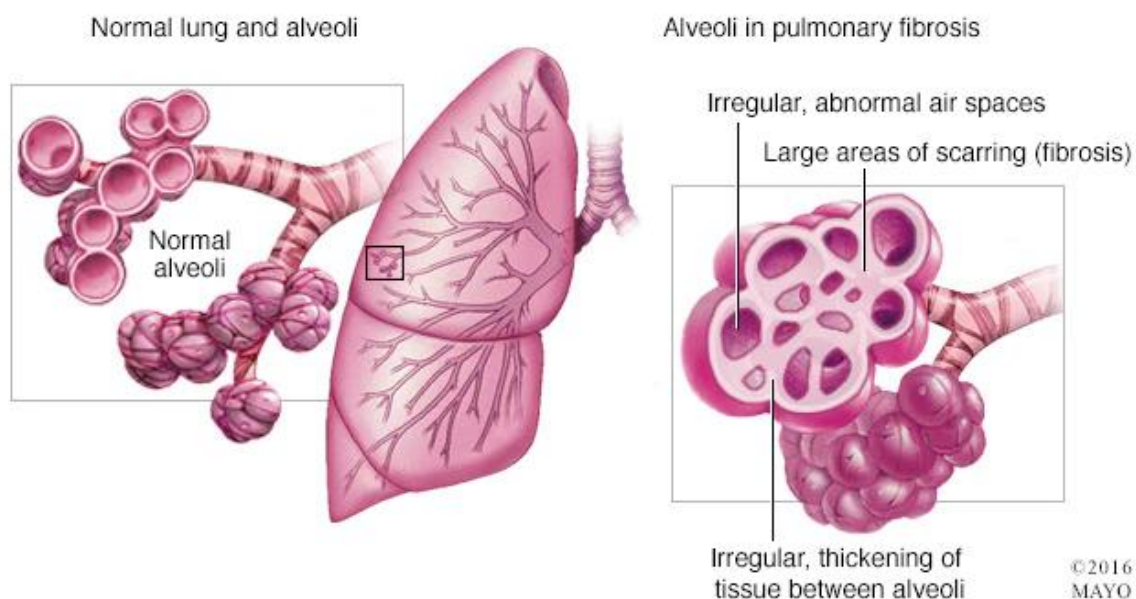
El Objetivo del proyecto es demostrar la viabilidad de un modelo de Inteligencia Artificial aplicado al ámbito médico. En concreto nos encontramos ante el objetivo de predecir el desarrollo y evolución de la fibrosis pulmonar en pacientes (en caso de haberla), en concreto predeciremos la capacidad pulmonar y como esta evoluciona a lo largo del tiempo.

La razón por la que he elegido este tema es porque la temática me llama mucho la atención y creo que puede aportar valor a la sociedad. Además, la manera de abordar el problema combina múltiples técnicas abordadas en el máster en todos los bloques.

- Contexto:

La Fibrosis Pulmonar es una enfermedad que actualmente es incurable, se da sobre todo a partir de los 50 años y el pronóstico es bastante desesperanzador, en un rango de 2-3 años se produce un empeoramiento en la capacidad pulmonar. Este empeoramiento sigue diversas tendencias y la capacidad de pronosticar dicha evolución puede ser clave en la detección precoz de y en las simulaciones de medicamentos experimentales.

Esta enfermedad se produce cuando el tejido pulmonar se daña, cicatrizando el tejido pulmonar, engrosando el tejido e impidiendo que los pulmones funcionen correctamente. A medida que la fibrosis pulmonar empeora, la capacidad pulmonar disminuye y los pacientes tienen más dificultades para respirar.



2. Descripción de los datos

Los datos están divididos en dos bloques, por cada paciente tenemos un escáner de sus pulmones e información básica del paciente (Sexo, Edad, Fumador/Exfumador/Nunca fumó) y la evolución de su capacidad pulmonar representada por la variable FVC (Full Vital Capacity) medida con un espirómetro y representada en ml. Tenemos 176 pacientes, los dividiremos en 171 pacientes de entrenamiento y 5 de validación.

Metadatos del Paciente:

La descripción de los metadatos del paciente se adjunta en la siguiente tabla, así como información relevante sobre los mismos.

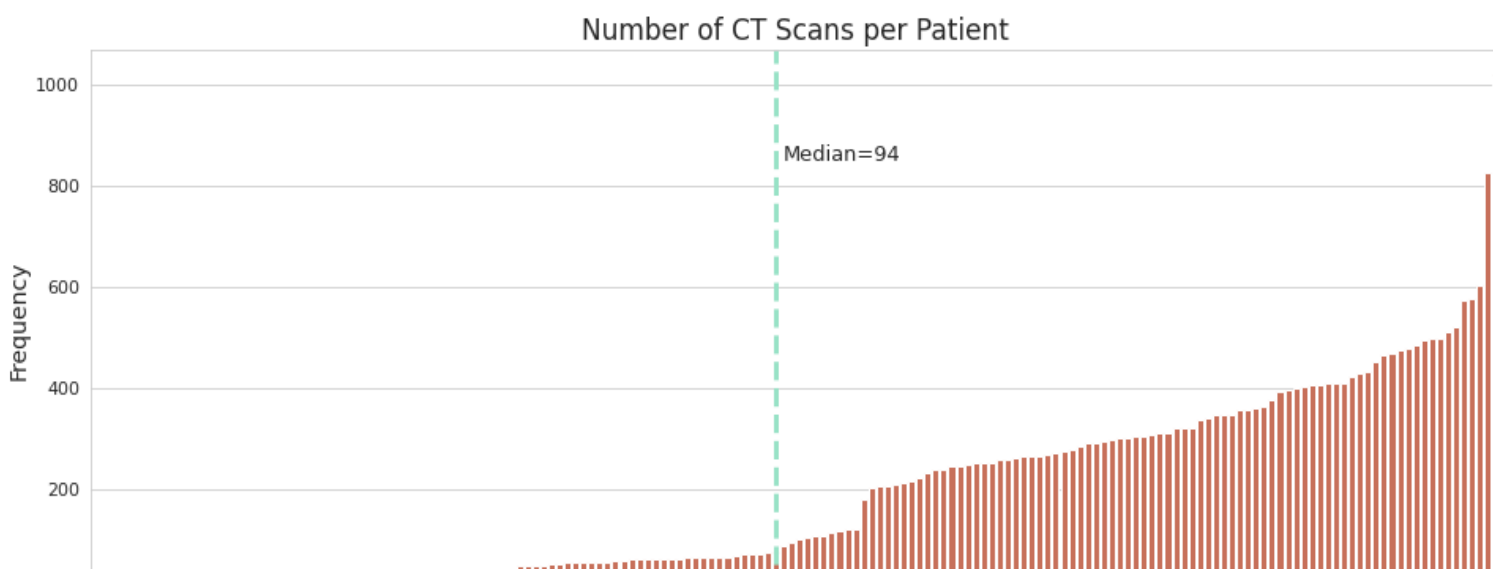
Variable	Descripción
Sex	Male / Female
Age	50 - 80
Smoke Status	ExSmoker - Smoker - Never Smoked
FVC	Capacidad pulmonar del paciente medida en ml. (800-6100)
Weeks	Semanas transcurridas desde la última visita. (0- 120)
Percent	FVC del paciente como porcentaje del FVC típico de un paciente de características similares

Para manejar los datos categóricos se han creado dos diccionarios (Sexo y Smoke Status) para codificar los niveles de dichas variables como enteros para posteriormente utilizar una capa de *Embedding*.

Por último, el valor de la Edad y *Percent* se han estandarizado restando la media y dividiendo por la desviación típica.

Escáner del Paciente:

Escáner 3D del paciente (CT-Scan) que engloba toda la zona de los pulmones, cada escáner está compuesto por un número indeterminado de rodajas como se muestra en la imagen inferior:



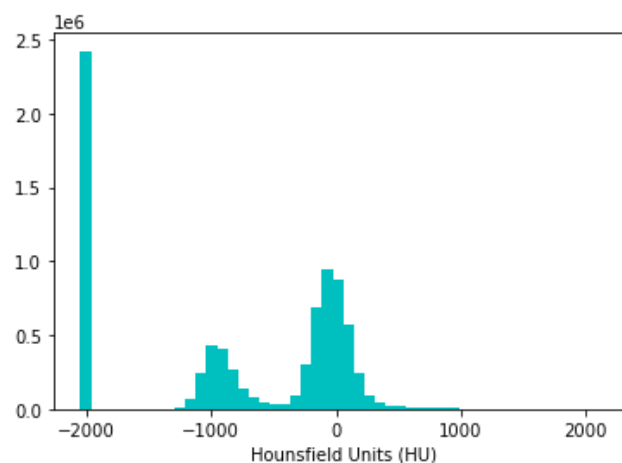
Los archivos CT-Scans se encuentran en un formato llamado **dicom** por tanto, para leerlos usaremos la librería *pydicom*, una vez leídos tenemos un objeto *Dataset* que contiene múltiples atributos del CT-Scan como se muestra en la imagen inferior:

```
Dataset.file_meta -----
(0002, 0000) File Meta Information Group Length  UL: 198
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID        UI: CT Image Storage
(0002, 0003) Media Storage SOP Instance UID      UI: 2.25.6932543653355114724157884519895943966
(0002, 0010) Transfer Syntax UID                UI: Explicit VR Little Endian
(0002, 0012) Implementation Class UID           UI: 1.2.276.0.7230010.3.0.3.6.1
(0002, 0013) Implementation Version Name        SH: 'OSIRIX_361'
(0002, 0016) Source Application Entity Title     AE: 'ANONYMOUS'
-----
(0008, 0005) Specific Character Set              CS: 'ISO_IR 100'
(0008, 0008) Image Type                         CS: ['ORIGINAL', 'PRIMARY', 'AXIAL']
(0008, 0018) SOP Instance UID                   UI: 2.25.6932543653355114724157884519895943966
(0008, 0060) Modality                           CS: 'CT'
(0008, 0070) Manufacturer                       LO: 'TOSHIBA'
(0008, 1090) Manufacturer's Model Name          LO: 'Activion16'
(0010, 0010) Patient's Name                     PN: 'ID00172637202238316925179'
(0010, 0020) Patient ID                         LO: 'ID00172637202238316925179'
(0010, 0040) Patient's Sex                      CS: ''
(0012, 0063) De-identification Method           LO: 'Table;'
(0018, 0015) Body Part Examined                 CS: 'Chest'
(0018, 0050) Slice Thickness                    DS: "10.0"
(0018, 0060) KVP                                DS: "120.0"
(0018, 1120) Gantry/Detector Tilt               DS: "0.0"
(0018, 1130) Table Height                       DS: "99.0"
(0018, 1140) Rotation Direction                 CS: 'CW'
(0018, 1151) X-Ray Tube Current                 IS: "205"
(0018, 1170) Generator Power                    IS: "24"
(0018, 1190) Focal Spot(s)                     DS: [1.6, 1.4]
(0018, 1210) Convolution Kernel                 SH: 'FC86'
(0018, 5100) Patient Position                   CS: 'FFS'
(0020, 000d) Study Instance UID                  UI: 2.25.55101252536187289024783023836238462290
(0020, 000e) Series Instance UID                UI: 2.25.134397266906956726621146943761679706211
(0020, 0010) Study ID                           SH: ''
(0020, 0013) Instance Number                    IS: "1"
(0020, 0020) Patient Orientation                 CS: ['L', 'P']
(0020, 0032) Image Position (Patient)           DS: [-214.8046, -220.8984, 1872.000]
(0020, 0037) Image Orientation (Patient)        DS: [1.00000, 0.00000, 0.00000, 0.00000, 1.00000, 0.00000]
(0020, 0052) Frame of Reference UID             UI: 2.25.30969998138159261957937341719543875636
(0020, 1040) Position Reference Indicator        LO: ''
(0020, 1041) Slice Location                     DS: "0.0"
(0028, 0002) Samples per Pixel                  US: 1
(0028, 0004) Photometric Interpretation          CS: 'MONOCHROME2'
(0028, 0010) Rows                               US: 512
(0028, 0011) Columns                            US: 512
(0028, 0030) Pixel Spacing                      DS: [0.761, 0.761]
(0028, 0100) Bits Allocated                     US: 16
(0028, 0101) Bits Stored                       US: 16
(0028, 0102) High Bit                           US: 15
(0028, 0103) Pixel Representation               US: 1
(0028, 1050) Window Center                      DS: "-500.0"
(0028, 1051) Window Width                      DS: "-1500.0"
(0028, 1052) Rescale Intercept                  DS: "0.0"
(0028, 1053) Rescale Slope                      DS: "1.0"
(7fe0, 0010) Pixel Data                         OW: Array of 524288 elements
```

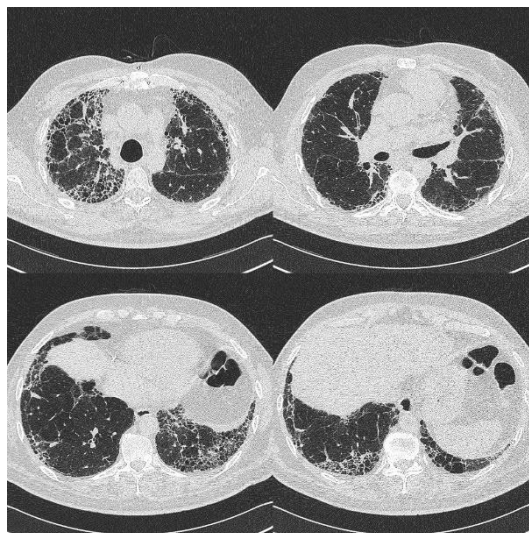
Con el atributo pixel data obtenemos el array de píxeles y con los atributos *RescaleIntercept* y *RescaleSlope* convertimos el array de píxeles a unidades de [Hounsfield](#). Estas unidades de Hounsfield representan el valor de la radiación del ecáner reflejada en los diversos tejidos, tal y como se muestra en la imagen inferior:

Substance	HU
Air	-1000
Lung	-500
Fat	-100 to -50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (cortical bone)

Esto nos será muy útil para diversas tareas. En la imagen siguiente podemos observar la distribución de píxeles del escáner de un Paciente, lo que nos puede dar una ligera idea de que y cuanto estamos viendo:

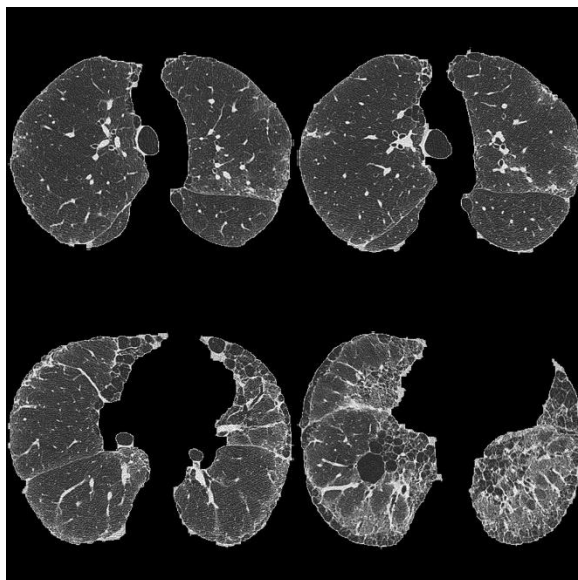


La imagen inferior representa 4 rodajas del escáner de un paciente, con los píxeles convertidos a unidades de [Hounsfield](#).



Cada CT-Scan tiene un origen completamente distinto al resto (hay ciertas similitudes) por tanto, tenemos formatos diversos, mientras unos CT-Scans tienen un grosor de 4mm otros tienen un grosor de 3, 2 o incluso 0.5mm, por tanto para no introducir ruido en el posterior modelo, debemos unificarlos, para ello nos aprovecharemos de los atributos *PixelSpacing* y *SliceThickness* del dataset dicom del paciente y unificamos todos los escáneres a un espaciado de [1, 1, 1] de tal manera que todos los escáneres representen 1mm del pulmón de un paciente. Es posible que para la fibrosis sea suficiente con 2mm, pero todas las pruebas se han llevado a cabo con 1mm.

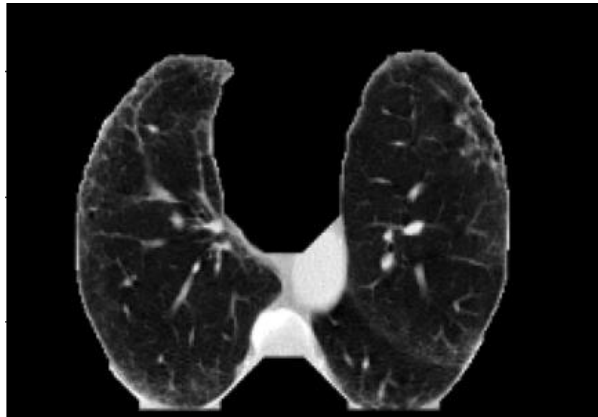
Dado que la mayoría de los elementos de dichas imágenes no son relevantes y para la mayor convergencia la solución propuesta incluye la segmentación del pulmón utilizando el algoritmo [Watershed](#). La razón por la cual se ha elegido este algoritmo en vez de utilizar un algoritmo de IA es el hecho de que no requiere dato etiquetado para entrenarse ya que se aprovecha de las unidades de Hounsfield para segmentar la parte deseada (El pulmón en este caso). Una vez realizada la segmentación obtenemos la imagen inferior:



NOTA: Este pulmón no es el mismo que la imagen anterior, pero se ha seleccionado por ser el de mayor nitidez.

Una vez tenemos todos los pulmones de los pacientes segmentados, aplicamos la última técnica. **Normalización por ventana.** Lo que haremos será tratar todas las partes del pulmón que tienen gradientes de grises pero que realmente para nosotros no es interesante de forma idéntica y potenciar aquellas que queremos dar más importancia, de nuevo para ellos utilizaremos los valores de la escala de *Hounsfield*. Por ejemplo: Para nosotros no es relevante si el aire se refleja como -900 o -950, para nosotros solo es relevante que en esa zona del pulmón hay aire.

Después de aplicar la normalización por ventana el pulmón segmentado quedaría como se representa en la imagen inferior.



Preprocesamiento de Secuencias:

Las Secuencias se han preprocesado aplicando las siguientes técnicas:

- Normalización de las medidas FVC restando la media y dividiendo por la desviación típica.
- Creación de 3 diccionarios clave:
 - **dict_ini_features**: Este diccionario contiene para cada paciente los metadatos y el FVC inicial del paciente. El fvc medido en la 1ª visita.
 - **dict_sequence_fvc**: Este diccionario contiene la secuencia de mediciones de FVC por cada paciente.
 - **dict_weeks_elapsed**: Este diccionario contiene por cada paciente las semanas transcurridas entre la última visita y la siguiente.

3. Análisis Técnico

En este apartado se explicará cada uno de los modelos propuestos, así como las diversas decisiones tomadas, en el siguiente apartado abordaremos los resultados y métricas, de esta manera podemos centrarnos en la explicación detallada de los modelos abordados.

3.1 Generador

El Generador de datos ha sido construido utilizando la API Sequence de Tensorflow ya que nos permite dinamizar la forma en la que devolvemos los datos y tiene soporte multiproceso (no utilizado).

El generador toma como entrada los diccionarios comentados en el apartado anterior, y devuelve por cada batch una tupla con las imágenes preprocesadas del Paciente y el diccionario **dict_ini_features** de dicho paciente. (en el caso del Autoencoder la salida será la misma que la entrada, las imágenes del paciente.)

Dado que tenemos muy pocos datos, cobra mucha importancia la aumentación de datos que podemos darle a las imágenes, en este caso al ser imágenes 3D podemos jugar de diversa manera. Finalmente se ha optado por aumentaciones aleatorias de flip, updown, cropping, invertir eje z (profundidad) para que la 1ª imagen sea la parte superior del pulmón o la parte inferior de forma aleatoria de tal manera que el modelo aprenderá a no depender de en que formato vienen las imágenes.

Hay que destacar que cada paciente tiene un N° indeterminado de rodajas en su escáner, pero para introducir esto en un modelo de Deep Learning necesitamos que el numero de rodajas sea constante. De esta manera he optado por crear ventanas de tamaño **num_frames_batch** (parámetro). Estas ventanas toman un índice aleatorio y devuelven las rodajas num_frames_batch siguientes. Como estas ventanas son aleatorias este también es un método de aumentar los datos, cuando un paciente tiene menos rodajas que este parámetro se crea un padding de ceros como relleno.

Dado que no todas las rodajas tienen información relevante, las primeras y las últimas prácticamente siempre contienen rodajas completamente negras (todos los píxeles a 0) o pequeñas cavidades de aire en el intestino o incluso la tráquea. De esta manera con un parámetro controlamos la cantidad de rodajas del paciente que queremos coger en nuestra ventana. Por ejemplo, si este parámetro llamado **Alpha** vale 0.9 esto equivaldría a mantener el 90% de las rodajas, eliminándose el 10% de ellas, concretamente un 5% superior y un 5% inferior.

El tamaño en el que se carga la imagen y el tamaño de cropping también son recibidos por parámetro y tenemos la posibilidad de realizar un center_crop antes del random_crop así nos aseguramos de que el random_crop no cree imágenes completamente negras.

Por último, hay es necesario resaltar que todas las aumentaciones realizadas mantienen la **coherencia del eje z** (profundidad) para que no superponga el pulmón derecho con el izquierdo, aunque nos ayudara con la aumentación de datos el modelo perdería de vista la cantidad de fibrosis que hay en los escáneres.

De esta manera el tamaño de salida del generador es:

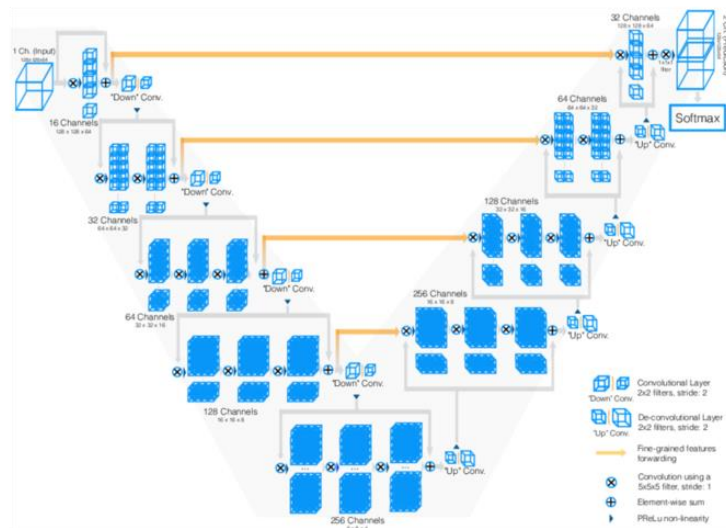
[batch_size, num_frames_batch, img_crop, img_crop, 1]

Se ha optado por un valor de num_frames_batch=32 lo que representa ventanas de 3,2cm.

3.2 Autoencoder

Para conseguir aumentar la velocidad del modelo de Secuencias vamos a construir un Modelo Autoencoder que nos permitirá obtener features enriquecidas con las que nutrir nuestro modelo y ahorrarnos tener que entrenar la gran cantidad de parámetros que tiene un modelo de imágenes 3d.

La arquitectura viene representada en la imagen inferior:



Sigue una arquitectura en forma de V en el que las salidas de las convoluciones de las capas superiores se concatenan en la fase de upsampling facilitando la reconstrucción de las imágenes.

En este apartado destacar 2 puntos:

1. Mediante un estudio de los escáneres observé que al ser de 1mm la diferencia entre una rodaja y otra era muy sutil, por tanto se me ocurrió, utilizar una capa AveragePooling como primera capa del modelo, pero solo realizamos el pooling sobre el eje z, es decir $\text{pool_size}=(2, 1, 1)$ de esta manera conseguimos conservar la integridad del escáner a la vez que reducimos enormemente la cantidad de parámetros y el tiempo de entrenamiento. Esto me permitió obtener grandes resultados.
2. Al tener muy pocos datos, es muy sencillo que el modelo produzca overfitting, por tanto, se ha recurrido a una regularización (en todos los modelos) bastante estricta, utilizando weigh_decay l2 y kernel_constraint MaxNorm, así como diversas capas de dropout y un learning rate bajo, para ayudar con la convergencia, así como evitar el overfitting.

Los parámetros de estos regularizadores son bastante restrictivos $1e-4$ para l2 y 0.1 para MaxNorm.

Este modelo entrenado se guarda para su posterior utilización en el modelo final.

3.3 Predictor FVC Inicial

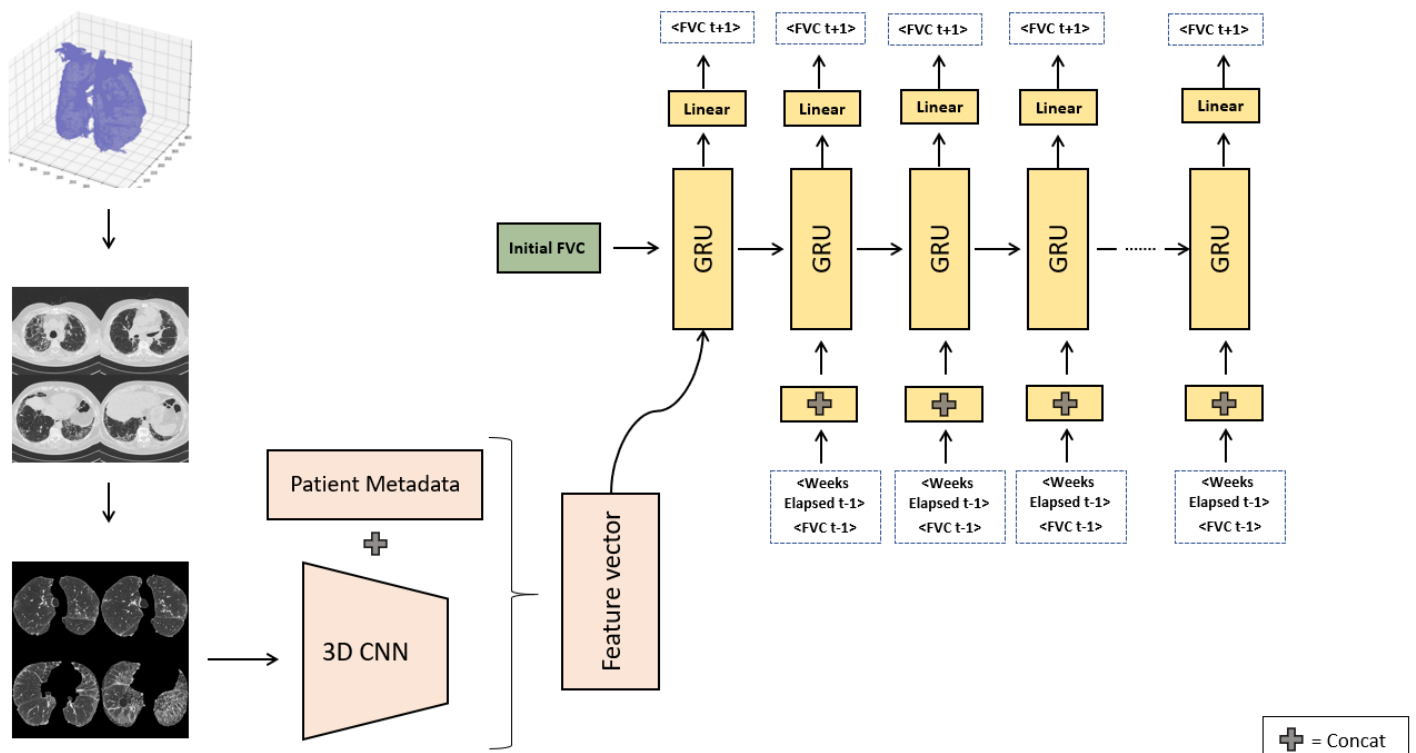
Este modelo no forma parte de la entrega final ya que se ha descartado. Intenté con los escáneres predecir el valor inicial del FVC para así poder obtener features mas enriquecidas que con el Autoencoder. Sin embargo, la poca cantidad de datos y la cantidad de ruido que hay entre un escáner y el valor medido por un espirómetro es muy alto. De esta manera aun utilizando mucha regularización para evitar el overfitting y mucha aumentación de datos el modelo aprendía poco más que a predecir la media global de entrenamiento con una desviación mínima.

Para este modelo utilicé una arquitectura resnet3D que si se adjunta en el repositorio por si en el futuro se dispone de mayor cantidad de datos y este modelo pueda funcionar.

3.4 Sequence Encoder-Decoder

Este modelo es el encargado de recibir como entrada todos los datos comentados en el apartado de Datos, así como el Autoencoder preentrenado y predecir la secuencia de mediciones de la capacidad pulmonar.

El diagrama de la arquitectura se muestra en la imagen inferior.



Este modelo ha sido desarrollado en tensorflow utilizando *subclassing* y entrenamientos personalizados, de esta forma conseguimos un gran dinamismo sobre el modelo.

A continuación, se explican los diferentes submodelos que engloba este modelo.

- **BackBoneTabularModel:**

Este modelo toma como entrada los metadatos del paciente y crea dos capas de embedding para el Sexo y el Smoke-Status, también recibe como entrada el valor de Percent y la Edad, concatena la salida de los embeddings y estos dos valores en un vector denso y aplica sobre él una capa dropout y una capa densa.

- **BackBoneImgModel:**

Carga el Autoencoder y a partir de la rodaja de imágenes de entrada produce el vector de características enriquecidas.

- **Encoder:**

Toma como entrada las salidas de los dos modelos anteriores. Convierte el batch de características enriquecidas en un vector de 2 dimensiones aplanando la profundidad al ancho produciendo un vector de la forma:

[batch_size, Depth*height*width, filters_output_autoencoder]

Por último, concatena a este vector la salida del modelo Tabular y aplica una capa de dropout y una capa densa.

- **Decoder:**

Este modelo es el encargado de recibir y construir la secuencia de predicciones a partir del vector recibido por parte del encoder, sobre el que aplica una capa de Atención, en concreto he utilizado la capa de tensorflow Bahdanu.

La salida de la capa de atención se concatena con los inputs del decoder que corresponden al fvc en t-1 y weeks_elapsed desde t-1, es decir, desde el instante de tiempo anterior al actual.

Este vector entra a una capa recurrente GRU (mismo rendimiento que LSTM pero menos parámetros) y que nos devuelve la secuencia completa más el estado oculto, por último conectamos la secuencia completa a una capa densa con una capa dropout y esta a una capa de salida, esta capa de salida consiste en una capa de tipo Distribution, en concreto he utilizado la distribución de Laplace ya que es la que nos indican los proveedores de los Datos y con la que se calcula la métrica, la salida de esta capa es la salida del Decoder junto con el estado oculto de la capa gru y los pesos de la capa de atención. El estado oculto lo utilizaremos en el entrenamiento y los pesos de la capa de atención los usaremos para la interpretabilidad del modelo.

- PulmonarFibrosis Sequence:

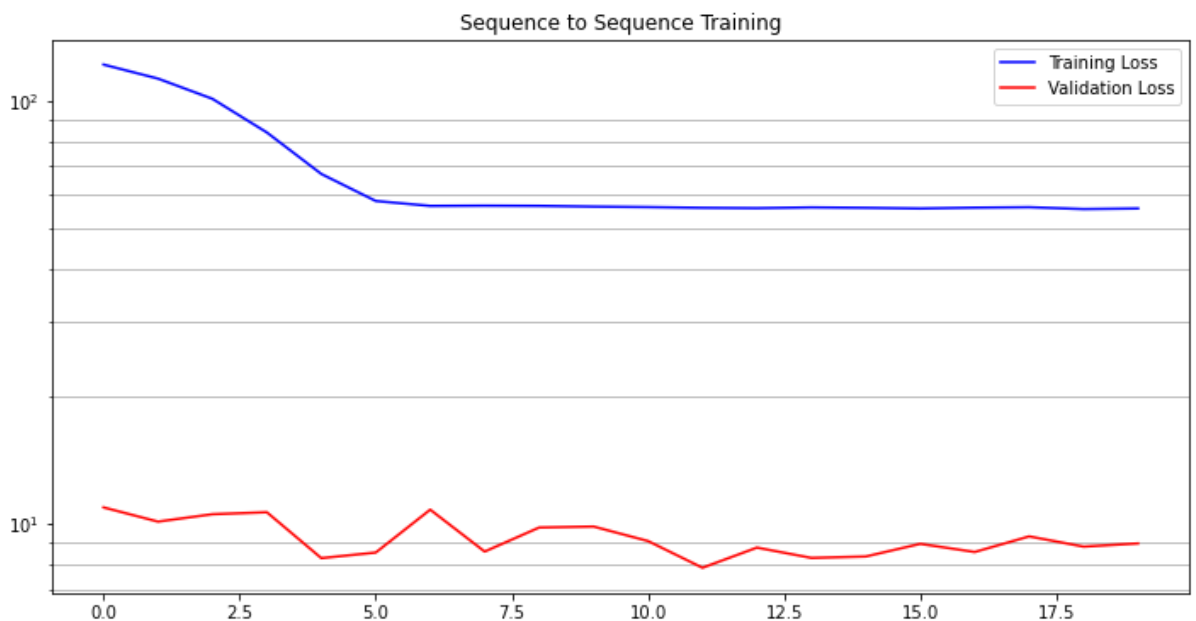
Este modelo toma como entrada los modelos anteriores y el encargado de realizar el entrenamiento. Destacar la utilización de teacher forcing para ayudar con la convergencia del modelo sobre todo en los primeros batches, aunque también he experimentado con un $\epsilon + \epsilon_{decay}$ teacher forcing (tomado como inspiración de DRL) pero con resultados similares.

Por cada paciente construye la secuencia de fvc para lo cual llama al decoder para cada instante de tiempo. En cada uno de estos instantes vamos introduciendo como input al decoder el valor del anterior FVC medido y las semanas transcurridas desde la última medición.

Con este modelo parametrizaremos diversos hiperparámetros tanto de los modelos mencionados anteriormente como los propios del aprendizaje. Asimismo, también guardaremos los pesos del modelo en cada epoch.

La función de pérdida mide la bondad de las predicciones y la confianza obtenidas con la capa de Distribución de Laplace. Explicado en el siguiente apartado.

Para el entrenamiento se ha utilizado una fuerte regularización (comentada en el apartado Autoencoder) y la convergencia es bastante rápida como se muestra en la imagen inferior.



4. Resultados & métricas

4.1 Resultados

Se han adjuntado en el repositorio, pero son las obtenidas en la imagen inferior:

Metrics

MSE	RMSE	MAPE	CustomMetric
13878	9.47	3.51	6.25

Baselines

- Baseline 1- Mean FVC from training to all predictions

MSE	RMSE	MAPE	CustomMetric
151977	14.26	12.11	9.82

- Baseline 2 - Initial FVC per Patient from each patient we take initial fvc and predict it over all timesteps.

MSE	RMSE	MAPE	CustomMetric
100013	12.98	8.22	8.92

- Baseline 3 - Last Timestep(Although this is a good baseline for general Sequence-to-Sequence purposes it is not for this case because we want to focus on the prognosis availability of the model).

MSE	RMSE	MAPE	CustomMetric
23886	10.13	4.25	6.83

La fórmula de “CustomMetric” que a su vez es la misma utilizada como función de pérdida en el proceso de entrenamiento es la siguiente:

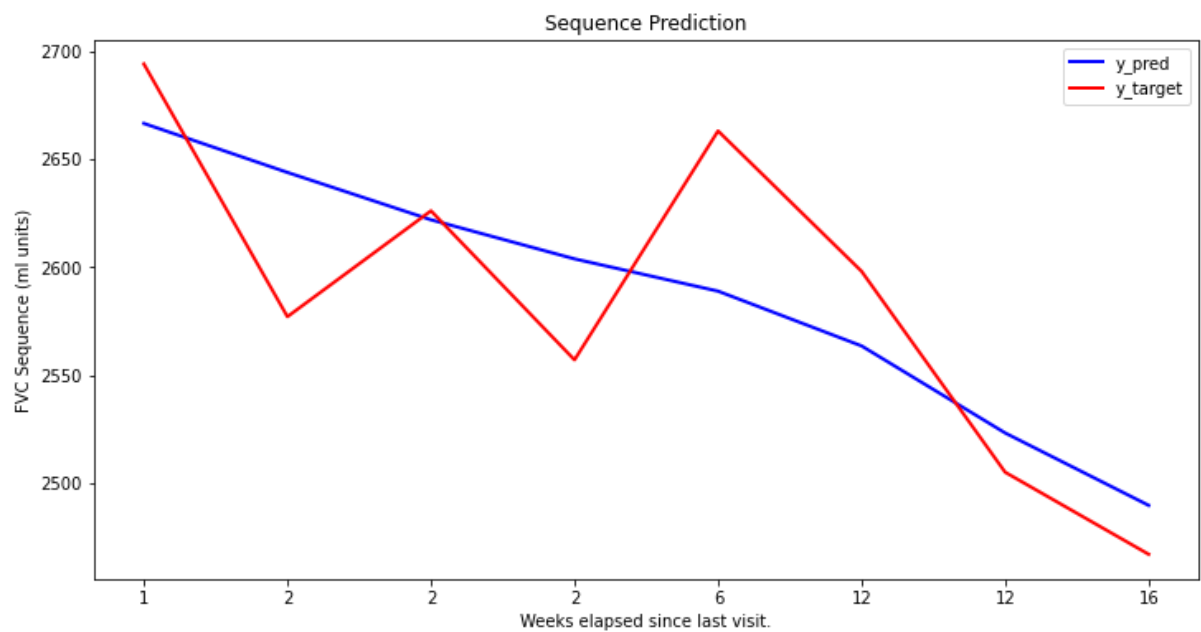
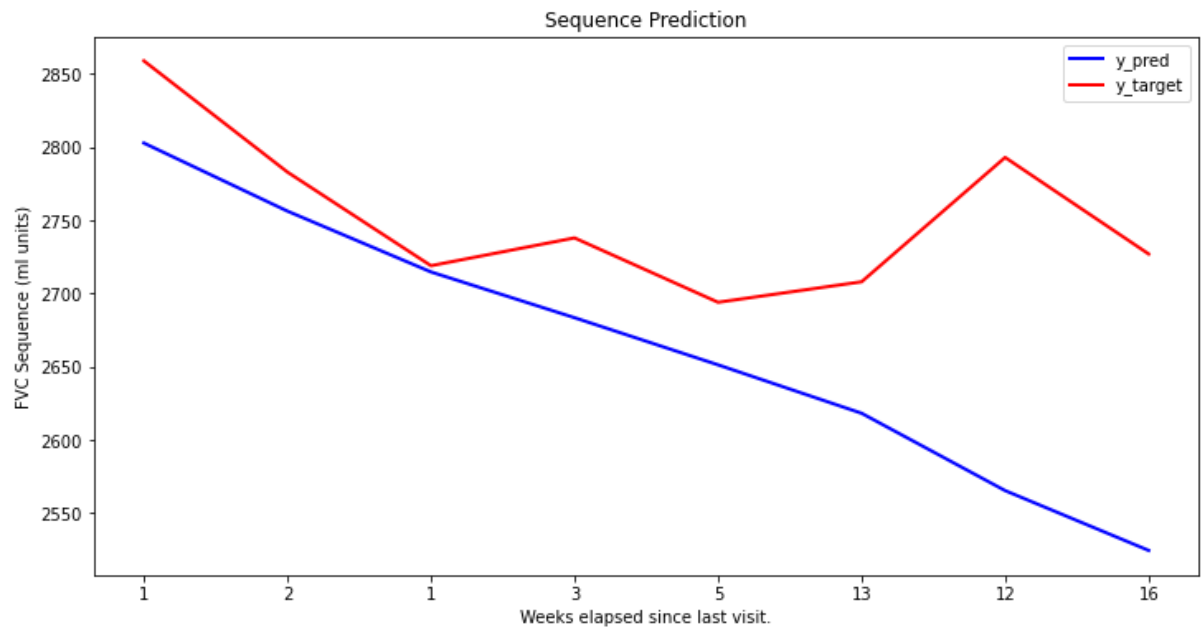
$$\sigma_{clipped} = \max(\sigma, 70),$$

$$\Delta = \min(|FVC_{true} - FVC_{predicted}|, 1000),$$

$$metric = -\frac{\sqrt{2}\Delta}{\sigma_{clipped}} - \ln(\sqrt{2}\sigma_{clipped}).$$

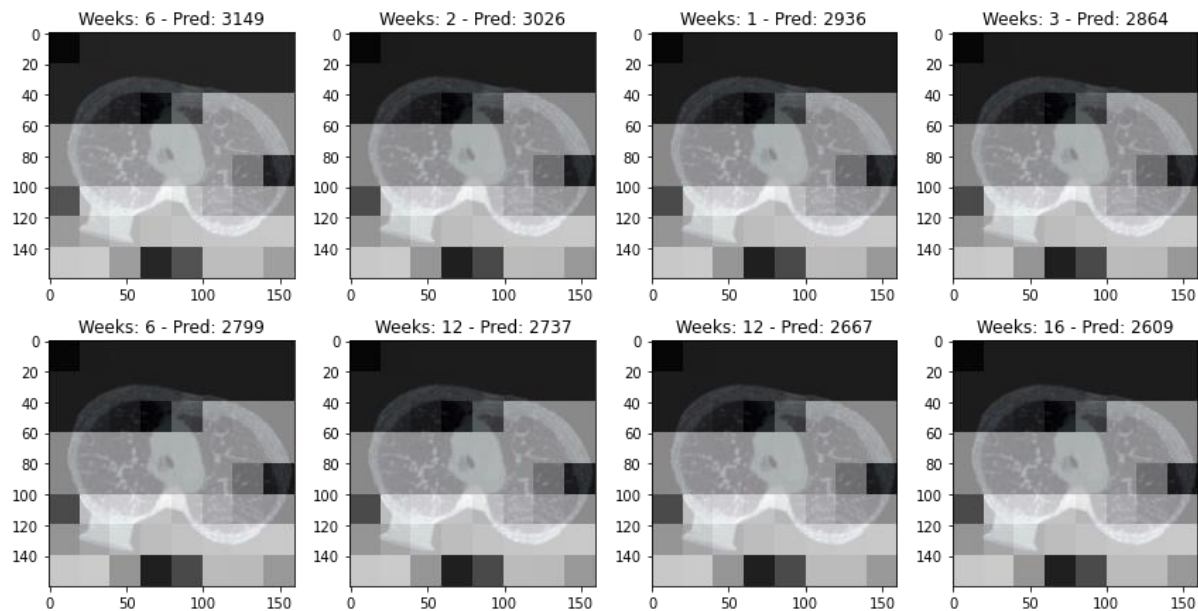
Siendo sigma la confianza de las predicciones.

Se adjuntan dos secuencias de predicción del modelo:



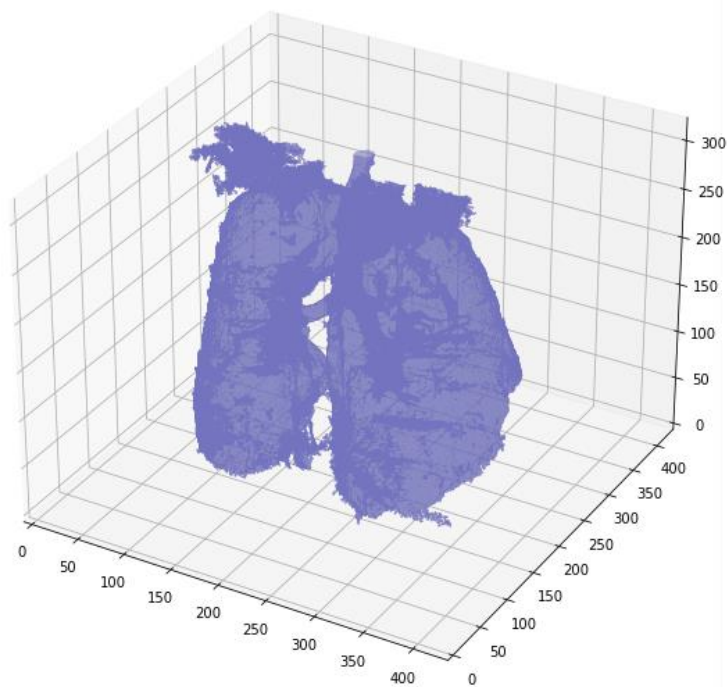
4.2 Interpretabilidad

Gracias a la capa de Atención y a los pesos obtenidos en el proceso de entrenamiento somos capaces de saber en que parte de los escáneres se está fijando más el modelo para realizar sus predicciones:



Como el escáner es en 3dimensiones lo que dificultaría la visualización de los datos. Se ha convertido a una imagen fija en dos dimensiones obteniendo el máximo en el eje z (profundidad) ya que se trata de los pixeles mas representativos. Los cuadros grises/blancos representa la cantidad de atención que le está dando la capa de atención a dicha parte de la imagen.

Por último, quiero comentar que se ha incluido un Notebook de Exploración de los Escáneres en 3D (Ejecución pesada). Con el que obtendremos visualizaciones como la siguiente:



5. Conclusiones

Se ha demostrado que la Arquitectura Encoder-Decoder es una buena opción a la hora de resolver estas tareas y gracias a la capa de atención podemos obtener interpretabilidad que relaciona las predicciones con los escáneres y así ayudar a la detección y prevención de la fibrosis pulmonar.

El modelo funciona correctamente para largas secuencias y largos valores de Weeks Elapsed que es lo que los médicos e investigadores requieren para prevenir y mitigar esta enfermedad, no les interesa la capacidad pulmonar dentro de unas pocas semanas sino su evolución con un largo horizonte.

El modelo es muy mejorable, desde hiperparametrización hasta las arquitecturas, pero con más datos conseguiríamos resultados mejores.