

DIARIO TFG

30 de enero de 2023

Hoy he ojeado el curso de Moodle de ayuda para el TFG, sobre todo la parte que se centra en la búsqueda de información.

31 de enero de 2023

Hoy he comenzado con la búsqueda, en Google Scholar, de artículos que me puedan ser de utilidad. En concreto, he encontrado seis que me han parecido válidos.

1 de febrero de 2023

Primera reunión con Juan Pablo. De momento tengo las siguientes tareas:

- Completar la BBDD que realicé durante las navidades con la información disponible en los Challenge de Physionet
- Profundizar en el trabajo de Daniel Tosaus, así como en la parte de Matlab y Python que el realizó, probar a descargar algunas de las señales e intentar representarlas en Matlab. Mirar también lo del lector de header, etc.

Por otro lado, aún no sabemos que señales sería más conveniente emplear en el entrenamiento de la red neuronal, así como que formato deberán tener las mismas, duración, derivaciones, etc. Eso está pendiente de hablarlo con Antonio.

2 de febrero de 2023

Lo que he hecho esta tarde ha sido completar la lista de BBDD que nos puedan ser de utilidad para este estudio. He añadido 5 nuevas bases de datos a la tabla que previamente realicé.

3 de febrero de 2023

Lo primero que he hecho ha sido informar a los profesores sobre las actualizaciones realizadas en la Excel de las bases de datos que puedan ser de utilidad.

Para continuar, me he estado informando sobre el Waveform Database Software Package (WFDB) for MATLAB. Se trata de una colección de funciones de Matlab para leer, escribir, manipular, es decir, para procesar datos procedentes de Physionet. Creo que me podrá ser de utilidad ya que seguramente tenga que descargar muchas bases de datos y con estas funciones podré hacer esta tarea mucho más automática y rápida (o eso creo). De momento, en otro Word he ido apuntando las que he utilizado, aunque me faltaría volver a probarlas y guardar las gráficas obtenidas para enseñárselas a los profesores en la siguiente reunión.

Como aún me queda bastante que explorar sobre este toolkit, he pasado a ver que funciones había empleado Daniel en su TFG, que no hayan sido programadas por el claro. Menciona tres y cuando he comenzado a buscarlas he dado con otro kit de Physionet llamado ECG-Kit. Aún tengo que profundizar para ver que más hay en él. He visto que también hay vídeos de YouTube en los que se explica como instalar, usar las funciones de este kit, etc.

5 de febrero de 2023

Hoy he seguido cacharreando con el WFDB Toolkit. Finalmente, me lo he podido instalar bien, el problema estaba en que no había ejecutado Matlab como administrador. Dentro del mismo he estado trabajando con la función `physionetdb`, que, entre otras cosas, permite ver las bbdd disponibles, consultar los registros y descargarlas. Sin embargo, hay varios problemas. Solo hay disponibles 133 bbdd y creo que no está actualizado así que muchas de las que yo tengo apuntadas no aparecen. Por tanto, creo que lo mejor será descargar los .zip desde la URL correspondiente. He probado a descargar la bbdd más grande de todas y mi ordenador no ha sido capaz de descomprimir el archivo, probaré mañana.

6 de febrero de 2023

Esta tarde he comenzado con la descarga de algunas de las bbdd. De momento llevo descargadas 18 y he descargado incluso aquellas que tienen menos de 1000 pacientes por si en un futuro hiciesen falta. Esta tarea lleva bastante tiempo pues me descargo los .zip y después los tengo que descomprimir, y al ser de varios GB el proceso requiere de varios minutos. Aunque el otro día ya probé la función del WFDB Toolkit para representar funciones quiero probar a representar señales de todas las bbdd descargadas.

8 de febrero de 2023

Hoy he reorganizado la ubicación de las bbdd descargadas hasta el momento, ya que las tenía en una carpeta que está sincronizada con One Drive y se me subían todas a la nube ocupando una gran cantidad de espacio. He seguido con la descarga y representación, más en detalle, de señales de bbdd con más de 1000 pacientes. Por otro lado, también he probado a usar la función `readheader`. No estoy segura de si los datos como el sexo o la edad los lee, pero no sé si esto un problema puesto que no tengo claro si esto lo tendremos en cuenta para algo.

9 de febrero de 2023

Esta tarde he terminado de descargar las bbdd que quería (me he dejado muchas que son de menos de 1000 pacientes, de muy pocos y si me lo piden las descargaré) y he seguido representando señales. No estoy segura de que sea buena idea emplear la función `rdsamp` para representar las señales ya que va ligada a un link, que he intentado actualizar, pero no se sabido hacer, y no puede acceder a los challenge (solo a las bbdd que están en data). Es por esto que he ojeado el código de Daniel y he imitado su forma de representar los ECG. Mi idea era cargar los .mat o .dat de las señales y de ahí representarlos como solemos hacer en Matlab, pero tiene sentido, como hace el, emplear información proveniente de la cabecera, como la frecuencia de muestreo, el número de muestras, etc. Quiero preguntarles a los profesores como sería mejor, aunque en realidad yo creo que esto afecta más a la redacción del trabajo que al tratamiento de las señales, simplemente es la forma de visualizarlas y de comprobar los resultados obtenidos.

Mañana tenemos la primera reunión los cuatro juntos y les quiero comentar lo siguiente:

- Durante la lectura de las distintas bases de datos encontré el WFDB Toolkit, permite el procesamiento de señales de Matlab y por eso decidí instalarlo y utilizarlo. No sé si alguna de las funciones podría ser de utilidad. Con la función `rdsamp` he visualizado

diferentes señales de las distintas bbdd que he descargado. NO LO HAREMOS CON ESA, LO HAREMOS CON UNA QUE YA TIENEN ELLOS IMPLEMENTADA.

- Las bbdd las he descargado directamente de los link, el .zip, solamente una la he tenido que descargar desde el terminal de Windows con wget, creo que lo he hecho bien, pero no estoy segura porque la función readheader me da error. ¿Descargo todas o solo algunas? ME VAN A CREAR UNA CUENTA EN EL CLUSTER PARA DESCARGARME TODAS LAS BBDD AHÍ, LA PROGRAMACIÓN TAMBIÉN LA HARÉ AHÍ.

10 de febrero de 2023

Este viernes ha tenido lugar la primera reunión con los tres profesores. Primero les comenté lo que había hecho y algunas de las dudas que tenía (comentadas y resueltas arriba).

Aspectos relevantes:

- Me van a crear una cuenta en el clúster para poder descargarme todas las bbdd allí y no hacerlo en local. Para conectarme al mismo, de forma remota, en VSCode me he instalado ssh. Desde allí me conectaré.
- Resolver el problema de los .hea de la bbdd Ningbo. Una vez esté dentro y con todo descargado enviárselo a Antonio. Él lo hará y después yo con un for lo replico para todos los ficheros.

Cosas a hacer:

- Leer los documentos que envió Antonio sobre NN.
- Una vez tenga la cuenta del clúster descargarme allí todas las bbdd de más de 1000 sujetos y comprobar que se pueden abrir las señales.

12 de febrero de 2023

Hoy he leído los dos documentos que envió Antonio. El primero de ellos, me ha costado leerlo y no lo he llegado a comprender así que quiero repasarlo. En cambio, en el segundo se da una explicación sobre las Redes Neuronales Recurrentes (RNN), me ha parecido muy claro y conciso y he conseguido entenderlo.

14 de febrero de 2023

Ayer por la tarde Julia me notificó que ya estaba creada la cuenta del clúster para mi uso y también recibí un correo por parte del i3A con explicaciones. Lo que he hecho hoy ha sido intentar acceder al cluster, según lo que he visto en Internet, pero no he podido hacerlo. Le he escrito a Antonio para ver si mañana podría ayudarme. Por otro lado, ayer rescaté los apuntes del tema 4 de APDS, Redes Neuronales, para repasarlos hoy (no lo he terminado).

15 de febrero de 2023

Esta mañana he quedado con Antonio para que me ayudase a conectarme al cluster. Además, me ha explicado y ayudado a instalar cosas básicas que necesito para poder llevar a cabo el desarrollo de código dentro del cluster. Mientras le esperaba he repasado el paper sobre RNN.

16 de febrero de 2023

Hoy me he conectado al clúster y lo primero que he hecho ha sido crearme dos nuevas carpetas dentro del directorio Work, una llamada Data, donde almacenaré las bbdd que descargue y otra llamada Experiments, donde intentaré replicar la práctica 4 de APDS (lanzando scripts de python .py con `condor_command`), de momento. Después lo usaré para las NN, etc. El siguiente paso ha sido crear, dentro de Data, una carpeta para almacenar los datos que descargue de cada una de las bbdd, de momento solo he creado una carpeta para aquellas bbdd que tengan más de 1000 registros. He querido lanzar el comando `wget` para empezar a descargar las bbdd, pero estaban todos los nodos prácticamente ocupados.

17 de febrero de 2023

Hoy en el laboratorio Juan Pablo me ha ayudado y me ha dicho un comando que me permite solicitar un nodo, y una vez me lo asignan, allí ya puedo lanzar el comando `wget` para descargarme la bbdd seleccionada. He estado toda la mañana descargando bbdd. Al principio me sorprendió el tiempo de descargar, pero es que debo tener en cuenta la longitud de las señales, no será la misma en todos los casos. Además, me he creado una cuenta de Github y he estado buscando algunas cosas para poder cargar las señales y representarlas en Python. Por la tarde he seguido descargando. También he buscado el WFDB Toolbox que previamente instalé en Matlab, pero en Python. Lo he instalado, pero creo que no me funciona bien o no lo estoy lanzando correctamente. Dentro del mismo también hay una función para poder leer los .hea.

19 de febrero de 2023

Hoy he finalizado la descarga y organización de todas las bbdd de más de 1000 pacientes. También he comenzado con la visualización de alguna de las señales descargadas. Para ello, he instalado el equivalente en Python del WFDB Toolbox. Aún me faltan funciones por explorar para el procesamiento de los ECGs.

21 de febrero de 2023

Esta tarde he terminado de visualizar las señales de las bbdd descargadas, salvo de una de ellas, `icentia11k`, ya que sigue descargándose. Me he enterado mejor del manejo de las funciones del package instalado. Por ejemplo, como leer canales específicos de los registros, por si solo nos interesase una derivación concreta, o como leer de la muestra `x1` a la muestra `x2`. También he comprobado que los .hea se pueden leer.

22 de febrero de 2023

Hoy he estado repasando el paper sobre RNNs y CPC.

23 de febrero de 2023

He terminado de repasar el paper sobre CPC uniéndolo con la explicación del TFM que me proporcionó Antonio.

24 de febrero de 2023

Hoy he repasado el tema 4 de APDS sobre redes neuronales. También he terminado de leer bien la parte que me pareció más relevante del TFM que me pasó Antonio. Por otro lado, he probado a leer algunas anotaciones de la bbdd icentia11k, ya que es la única que tiene, con la función disponible para ello en el WFDB package de Python. He repasado parte del cuaderno 1 de la práctica 4 de APDS y he comenzado a mirar el remuestreo, pero no he avanzado mucho.

25 de febrero de 2023

Esta tarde he terminado de mirar el cuaderno 1 y he mirado el cuaderno 2.

26 de febrero de 2023

Hoy he repasado el cuaderno 3. También he estado remuestreando (downsampling y upsampling) pequeños ejemplos, primero en un notebook luego en un script de Python, y visualizando las gráficas. Me falta pasarles los rdos y hacer el for para toda la base de datos.

26 de febrero de 2023

Hoy he estado mirando el remuestreo, probado pequeños ejemplos para cada una de las bbdd.

27 de febrero de 2023

Hoy he seguido avanzando con el remuestreo y generando gráficas para comprobar que la cosa iba bien.

28 de febrero de 2023

Hoy les he enviado a los profesores lo que había avanzado y me han dado nuevas indicaciones:

- Leer más trozo de la señal para comprobar los efectos de borde
- Realizar una interpolación lineal en lugar de no lineal al remuestrear

2 de marzo de 2023

Esta tarde he estado con el remuestreo de las señales, planteando el bucle para la bdd challenge_2017. En principio ha funcionado bien.

3 de marzo de 2023

Hoy he estado mirando lo que se me indicó en el correo y sacando nuevas representaciones para mostrarlas en la reunión.

Reunión:

- **Remuestreo:** cambiamos de función, ya no uso `signal.resample` sino `signal.resample_poly`

https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.resample_poly.html

! Cuando pruebe a cambiar de función en los scripts con los ejemplitos sencillos, volver a generar las gráficas y enviarlas a los profesores para ver si sale bien la cosa.

- **Bucles:**
 - o **Autonomic:** Como he visto en los .dat hay tanto ECGs como una señal de presión arterial (NIBP), así que debo tener cuidado con lo que leo. No decido que canal leer como lo estaba haciendo hasta ahora, sino que paso a acceder al campo de la cabecera donde se indica el nombre de las señales contenidas en el .dat. Cuando salga la palabra ECG lo leo, sino no. Puedo buscar una cadena de texto dentro de ese campo que contiene el nombre de las señales, buscar como hacerlo en Internet. Remuestreo con la nueva función y ya lo guardo como me han indicado. En una nueva carpeta, aislada de las señales sin procesar, de la siguiente forma:
NOMBREBDD_NOMBREFICHERO_FS_SA_FECHA_IDENELHEA(MAYÚSCULAS).
Para guardarlo no empleo `wrsamp` sino que lo guardo directamente con la función que ha pasado hoy por correo Antonio.
Cuando hay 2 ECGs no sabemos si son el mismo o diferentes. Los represento conjuntamente y miro la correlación. Pasarlo también a los profesores.
 - o **Challenge_2017:** Aquí debo cambiar el remuestreo, es decir, la función que usamos, así como la forma de almacenar las señales remuestreadas. Las guardo directamente en otra carpeta y con el nombre que he indicado anteriormente.
 - o **Icentia11k:** Para hacer lo mismo que estoy haciendo en challenge 2017, necesito un fichero RECORDS adecuado, lo creo. De nuevo debo remuestrear, almacenar correctamente en la nueva ubicación y con el nombre adecuado.
- **Señales en nueva ubicación:**

Todas las señales de las bdd las paso a una nueva ubicación. Por ejemplo, si en mis .dat o .mat tengo 12 derivaciones, voy a guardar cada una de ellas, en crudo (emplear función de Antonio), con el siguiente nombre:
NOMBREBDD_NOMBREFICHERO_FS_SA_030323_DERIVACIÓN(MAYÚSCULAS).

Cada una de estas irá dentro de una carpeta llamada NOMBREBBDD, pero no mantengo la estructura de carpetas inicial. En DERIVACIÓN pondré el nombre de la misma o el nombre que esta reciba en el .hea.

11 de marzo de 2023

Esta semana he tenido un parcial, no he podido trabajar en el TFG. Hoy, después del examen, he estado modificando el remuestreo, en concreto la función que empleábamos para ello. He estado sacando diferentes gráficas y creo que el efecto de borde que aparecía anteriormente ya no sucede. Por otro lado, he comenzado a modificar el bucle de remuestreo para la bbdd autonomic aging cardiovascular aunque sin éxito.

12 de marzo de 2023

Hoy he estado haciendo los bucles para algunas de las bbdd. También he tenido que mover las señales de algunas de ellas a la carpeta adecuada, así como preparar los ficheros de RECORDS. Aún me queda bastante.

13 de marzo de 2023

AUTONOMIC AGING CARDIOVASCULAR

- Señales remuestreadas (downsampling) y escritas en Preprocesado

CHALLENGE 2017

- Señales remuestreadas (upsampling) y escritas en Preprocesado

CPSC 2018

- Señales escritas en Preprocesado

CPSC 2018 EXTRA

- Señales escritas en Preprocesado

PTB XL

- Señales escritas en Preprocesado

ICENTIA11K

- NO TERMINADO POR PROBLEMAS AL HABER TANTAS SEÑALES NO PUEDO ENTRAR A LA CARPETA

14 de marzo de 2023

GEORGIA

- Señales escritas en Preprocesado

NINGBO

- Señales escritas en Preprocesado

15 de marzo de 2023

ECG ARRHYTHMIA

- He olvidado la mala idea que tuve y sigo la misma implementación que para las bbdd que hice ayer

ICENTIA11K

- Me falta sacar todas las señales fuera y llamar a la escritura

AUTONOMIC AGING CARDIOVASCULAR

- He leído de nuevo la información de physionet, pero no se indica que es ECG1 y ECG2 en ningún momento
- He representado varias señales juntas para averiguarlo, las añado a la presentación

He comenzado a hacer el pwp para la reunión de mañana.

16 de marzo de 2023

Hoy he terminado el pwp para la reunión de esta tarde en la EINA.

Reunión: (también he anotado alguna cosa en el pwp)

- Enviar tabla Excel a los profes para pensar que pequeñas tareas haremos con las bbdd que no he descargado hasta el momento. Esta es la tarea sobre la que mediremos las prestaciones.
- **Autonomic aging cardiovascular:** escribir a los autores para saber qué son ECG1 y ECG2. A la hora de realizar el remuestreo y la escritura de las señales en la nueva ubicación no discriminar y leer también las señales de NIBP.
- **Icentia11k:** volver a meter las señales que saqué a sus respectivas subcarpetas. Hacerlo con los comandos que estuve viendo con Antonio al final de la reunión.

A hacer en todas las bbdd descargadas:

- **Filtros:**
Hacemos un filtro IIR paso alto a 0.3 eliminando la línea de base.
Comprobar los *extremos* también.

Estaría bien repasar la P1 de APDS, pues el filtro que empleo es uno que se empleó en dicha práctica.

- **Amplitudes:**

A la señal que leo le resto lo que esté entre paréntesis, baseline (si no hay nada será 0), y la divido por `adc_gain`, así las tendré en las unidades que se indique en el `.hea`.

Al final el procedimiento sería, remuestrear (si es necesario), amplitudes a mV o uV, filtrar para eliminar la línea de base, convertir a float32 y escribir en la nueva ubicación. Otra cosa que creo que es importante es no meter dentro de Preprocesado las carpetas de cada una de las bbdd, las saco fuera y hago una para cada uno de ellos.

También debería comprobar a leer los ficheros `.bin`, representarlos y ver que todo está bien.

19 de marzo de 2023

Hoy he rescatado la Excel con las bbdd que envié a los profesores en su día, ya que la que tengo y está con anotaciones. También he escrito un pequeño borrador preguntando a los autores de la bbdd `autonomic aging cardiovascular` qué son ECG1 y ECG2. Este también se lo enviaré mañana a los profesores.

20 de marzo de 2023

Hoy he enviado un correo a los profesores con la Excel que contiene las bbdd, el borrador del mail a enviar a los autores y le he pedido a Juan Pablo lo de los filtrados de las señales.

22 de marzo de 2023

Hoy he recibido respuesta de los profesores, lo que debo hacer es lo siguiente:

- Modificar el borrador del correo según me lo han indicado los profesores:
 - o Lo de que fueron grabados al mismo tiempo, creo que se puede saber simplemente viéndolos, ¿no? Si están sincronizados los latidos, es seguro que fueron grabados a la vez. Mira unos pocos al azar y con eso sales de dudas.

Pasarles el documento modificado previamente a enviarlo. Enviarlos poniendo a Juan Pablo y a Julia en copia.

- Responder a la pregunta de Julia. No he podido avanzar, debido al examen.
- Responder al otro correo de Juan Pablo en relación a los filtrados.

23 de marzo de 2023

Hoy he enviado el correo a los autores de la bbdd Autonomic Aging Cardiovascular. También he trabajado en la bbdd Autonomic Aging Cardiovascular. Ahora ya no discrimino entre ECGs y NIBPs, leo todas las señales que hay dentro de cada registro.

Por otro lado, el procesado que debo realizar es:

- Leer las señales
- Remuestrearlas
- Filtrarlas, aunque esta parte aun no la he mirado
- Pasar las amplitudes a mV o uV, según venga en el .hea, ¿normalizo?

He mirado también la bbdd del challenge 2017, pero tampoco la he terminado.

24 de marzo de 2023

Hoy he trabajado con las bbdd autonomic aging cardiovascular, challenge 2017, cpssc_218, cpssc_2018_extra, ptb-xl y ningbo. En la primera de ellas me ha surgido un problema con los filtrados. Primero, lo que hago es leer las señales que hay dentro de cada uno de los registros, después, si es necesario, remuestreo para que la frecuencia de dichas señales sea de 500Hz. El siguiente paso que realizo es modificar las amplitudes de las señales, ponerlas en mV o uV, restando baseline y dividiendo por adc_gain. Llegados a este punto, y antes de escribir las señales en los ficheros binarios lo que hago es filtrar, con un filtro IIR paso alto de Butterworth con una frecuencia de corte de 0.3Hz. Es aquí donde me aparece el error. Probando con diferentes señales he podido ver que después de filtrar me sale que la señal es NaN.

En el resto de bbdd cuando he ido representando pequeños ejemplos, no me han aparecido problemas. Además, en algunos de ellos se podía apreciar muy bien cómo se eliminaba la línea de base.

Debido a lo comentado, y a que les quiero preguntar a los tutores si debo normalizar las señales, que se quedó a medias en la última reunión, no voy a escribir las señales hasta que no esté segura sobre los filtrados.

25 de marzo de 2023

Hoy he mirado Georgia y Ecg arrhythmia.

28 de marzo de 2023

Hoy les he enviado el correo a los profesores con las siguientes dudas:

- Autonomic, filtro y me salen NaNs, es porque la señal que leo tiene valores NaNs. ¿Los elimino? ¿¿¿¿Borro NaNs de todas????
- ¿Normalizo? No, eso lo haremos después en la red.
- ¿Propuesta TFG? La estamos haciendo para entregarla mañana-

31 de marzo de 2023

Reunión:

- ✓ Autonomic Aging Cardiovascular:
 - Dentro del bucle debo añadir la creación de un fichero de texto donde se me guarde la siguiente información de cada uno de los registros de la bbdd: cuántas muestras NaNs hay y sus posiciones dentro de la señal. Si solo se encuentran al principio o al final, las podríamos borrar sin mayor problema. En cambio, si se encuentran en medio deberíamos poner en su lugar la media de las muestras vecinas. En este último caso no nos tomaríamos tantas molestias, no usaremos esta bbdd y ya está.
- ✓ En realidad, un fichero así me gustaría obtenerlo para todas las bbdd. Querré indicar cosas como la Fs, si hay muestras NaNs cuantas hay y cuáles son sus posiciones, que las unidades sean mV o uV, etc. Pensar más cosas que pueda poner aquí. Es como un .csv, por columnas y separando en comas.
- ✓ ¡¡Las amplitudes no las toco!! Con la función wfdb.rdsamp ya las tengo en las unidades adecuadas, es decir, mV o uV. Si estuviesen en uV las tendría que pasar a mV.
- ✓ A la hora de representar las señales debo poner el eje x en tiempo, segundos, no en muestras. Probar a obtener nuevas representaciones. También me gustaría mirar trozos por en medio de la señal, no en el inicio, ya que el primer latido es raro.
- ✓ Challenge 2017:
 - Les comenté que yo creo que el validation set es parte del training set, por cómo se describe esta bbdd en Physionet, y que por tanto esas señales estarían repetidas. Para poder comprobarlo, me propusieron ver los 10 primeros valores y la duración para ver si las señales son iguales. Además, creo que los identificadores coincidían.
- ✓ Donde me mando registrarme Julia, tengo que exportar la tabla y ver cuáles de las bbdd serían más asequibles de acceder.
- Tendrían que ir redactando algo de lo que he hecho.
- × Enviar recordatorio sobre correo ECG1 y ECG2 a los autores. Como no vamos a usar esa bbdd, no lo voy a enviar.

Después en la red, excluiré las primeras 500 muestras iniciales y finales.

7 de abril de 2023

Hoy he comenzado con las tareas pendientes de la reunión del día 31 de marzo. Para empezar, he estado estudiando la bbdd Autonomic, ya que era la que daba problemas por tener muestras NaNs. He generado un fichero .csv donde almaceno para cada señal de cada registro, su frecuencia de muestreo, unidades, duración en muestras, muestras que tomen valor NaNs y posición de las mismas. Muchas de las señales (las que me importan son los ECGs) tienen una gran cantidad de muestras NaNs, por medio, no en los extremos. No podemos usar esta bbdd.

Este mismo fichero me interesa generarlo, y comprobarlo, para el resto de bbdd. He empezado con la siguiente base de datos del Excel, cpssc_2018_extra. He generado el fichero .csv, parece que todas las señales están en mV y no tienen muestras NaN. Por otro lado, he obtenido nuevas representaciones que ya he añadido a la presentación de la siguiente reunión.

He seguido con el resto de bbdd, me faltan 3 que miraré el domingo:

- ✓ Challenge 2017: debo comprobar si las señales del validation set coinciden con algunas del training set, yo creo que sí. Para ello, voy a comparar los identificadores, la duración en muestras y las represento juntas para ver si son iguales. En caso de que lo sean, no uso el validation set, ya que tendríamos señales repetidas.
Me falta terminar el procesado normal y completar el pwp.
- ✓ Ecg-arrhythmia: Ya saqué las señales, no las voy a volver a meter. Me falta el procesado que he hecho para el resto de bbdd.
- lcentia11k: primero debo solucionar los problemas de congestión que tenía reintroduciendo todas las señales a sus respectivas carpetas. Después, hago el mismo procesado que con las demás.
- ✓ *Tengo que volver a enviar la Excel, que las notaciones en rojo de la derecha están mal colocadas.*

De momento no voy a escribir las señales procesadas en las carpetas, quiero esperar a que me den el visto bueno. después de escribir querré probar a leer las señales.

8 de abril de 2023

Hoy he terminado de obtener las representaciones para las señales de la bbdd ptbxi. También he avanzado un poco con el training set del Challenge 2017 y he hecho lo que faltaba de la bbdd ecg-arrhythmia.

9 de abril de 2023

Hoy solamente he arreglado las representaciones que obtuve ayer de la bbdd del Challenge 2017. Como hago un remuestreo, upsampling, la duración de las señales se modifica, aumenta. A la hora de representarlas juntas no queda bien. Las he representado por separado, cada una con su eje de tiempos.

10 de abril de 2023

Lo primero que he hecho ha sido comprobar para la bbdd Challenge 2017 que el validation test coincide con un conjunto de señales, 300, del training set.

He empezado también con las comprobaciones del número de muestras NaNs y las unidades de los ficheros .csv generados, pero no lo he terminado.

He comenzado a pasar a Excel la tabla que mencionó Julia de la página euCanSHare.

11 de abril de 2023

Hoy he terminado de exportar la tabla de bbdd de euCanSHare. He añadido también los procesos de acceso, pero debo mirarlos bien.

También he filtrado los ficheros .csv generados con la información de cada una de las señales de las bbdd para poder comprobar si hay NaNs o unidades distintas a mV.

14 de abril de 2023

Reunión:

- ✓ Mandar 5 primeras señales de las bases de datos ecg-arrhythmia y georgia a los profes para que las lean y vean si a ellos también les aparecen muestras nans. También de Mimic-iv
- ✓ Mirar JS43632 para ver si las representé al revés o está bien la imagen que presenté en la tutoría
- ✓ Mandar Excel con las bbdd de euCanSHare que les puede servir para su proyecto
- ✓ Añadir a los ficheros que contienen la información importante filtrada el **número de muestras totales** de cada uno de los registros
- **Escribir** las señales en los ficheros y en las carpetas correspondientes
- TAREA CONTROL:
 - ¿Qué derivación es de las 12? Bbdd 12 y 13 de la excel
 - ¿Detectar AF? Bbdd 15 de la Excel
 - Anotaciones: 2 o 3 capas, leer aleatoriamente con la función read32 que pasó Antonio. 1,5 segundos ECG/latido y pico. Reproducir resultados en el clúster
 - Leer TFG Daniel + mirar su código

Para pedir un nodo con gpu: `condor_ssh -c 1 -r 1024 -a GPU3090`

26 de abril de 2023

En la reunión pasada el tema principal fue determinar qué hacer con aquellas bbdd que contuviesen señales con muestras NaNs. Se quedó en que la bbdd autonomic aging cardiovascular no la íbamos a emplear ya que contiene 84 señales que contienen muestras NaN y además no sabemos que es ECG1 y que es ECG2, pues no hemos recibido respuesta al correo que enviamos a los autores de la bbdd en cuestión. Por tanto, esta queda descartada.

Por otro lado, había otras tres bbdd que también tenían algunas señales con muestras NaNs. En concreto, Georgia (solo 8 señales), Ecg-arrhythmia y Ningbo (104 señales ambas dos). Fue entonces cuando nos dimos cuenta de que estas dos últimas bbdd coincidían, los autores son los mismos, Ningbo esta sacada de ecg-arrhythmia, por ello, Ningbo tampoco la usaremos. Pasamos a tener **7 bbdd con las que trabajar** en lugar de 9.

BBDD A ESCRIBIR:

1. ~~Challenge 2017~~
2. ~~Cpsc 2018~~
3. ~~Cpsc 2018 extra~~
4. Ecg arrhythmia
5. Georgia
6. Icentia 11k

~~7. Ptb-xl~~

Hoy he comenzado con la escritura de las bbdd Challenge 2017, cpsc 2018, cpsc 2018 extra y ptb xl. Después he seguido con icentia11k, haciendo los ficheros.

Para las bbdd que señalo en amarillo primero modifco los ficheros .csv y voy a esperar la respuesta de los profes. Para ver si a ellos les aparecen los mismos problemas. Si es así, borraré las señales afectadas y escribiré el resto.

También he descargado las bbdd necesarias para la tarea de control (12, 13 y 15) y las he procesado como el resto, en una de ellas, mimic-iv, también me aparecen señales con muestras NaN. Me falta obtener las representaciones de algunas señales de cada bbdd.

28 de abril de 2023

Hoy he enviado el correo a los profesores para actualizarles sobre cómo van las cosas y he obtenido representaciones para las nuevas bbdd descargadas que emplearé en la tarea de control. También he mirado los ficheros de icentia11k, parece que también hay señales que tienen muestras NaNs.

29 de abril de 2023

BBDD A ESCRIBIR:

- ~~1. Challenge 2017~~
- ~~2. Cpse 2018~~
- ~~3. Cpse 2018 extra~~
4. Ecg arrhythmia
5. Georgia
6. Icentia 11k
- ~~7. Ptb-xl~~

Hoy he terminado con los ficheros de la bbdd icentia11k. También aparecen señales que tienen muestras NaN. Tendré que informar a los profesores.

He seguido con la representación de las señales de la bbdd afpdb, me he dado cuenta de que puede que en otras ocasiones no lo haya hecho bien, debo revisarlo para todas las bbdd que necesiten remuestrear.

30 de abril de 2023

Hoy voy a comenzar con la tarea de control. La dividimos en dos partes, vamos a preparar un ejemplo que consiste en determinar qué derivación de las 12 derivaciones típicas del ECG es, y otro ejemplo donde se debe detectar si la señal presenta AF (Atrial Fibrillation).

1. ¿Qué derivación de las 12 es?

2. ¿AF?

La base de datos que se va a emplear en esta tarea es *PAF Prediction Challenge Database* (afpdb), la número 15 de la Excel.

Descripción de los datos: esta base de datos se divide en un set de aprendizaje, formado por aquellos registros que comiencen por la letra n o por la letra p, y un set de testeo, formado por los registros que comienzan con la letra t.

Set de aprendizaje -> consiste de 50 record set, cada uno de los cuales contiene dos registros de 30 minutos llamados con número consecutivos y otros dos registros de 5 minutos cuyos nombres son iguales, pero terminando en c. Estos cuatro registros de cada record set son extractos de grabaciones de ECG de mayor duración de un solo sujeto. Estos 50 record set vienen de 48 sujetos.

Los registros que comiencen por la letra p vienen de sujetos que tienen PAF y el segundo registro (número par) de cada par de registros de duración de 30 minutos contienen el trozo de ECG inmediatamente anterior al episodio de PAF. Esto puede ser verificado examinando el registro de continuación (5 minutos) que se encuentra numerado de la misma forma, que contiene el episodio de PAF. En cambio, el primer registro de 30 minutos (el impar) contiene un segmento de grabación alejado de cualquier episodio de PAF (al menos 45 minutos), esto se comprueba observando el registro de continuación, que tampoco tiene PAF.

Por tanto, los impares no tienen PAF, el par de 30 minutos es justo antes y el par de continuación sí que tiene el episodio de PAF. Note: Please be aware that a few of the 30-minute records in this group may contain very short bursts of PAF that escaped notice while the learning set was being compiled.

Los registros que comienzan por la letra n vienen de sujetos que no tienen AF.

Set de testeo -> construido por 50 record set que provienen de 50 sujetos diferentes a los 48 anteriores, aquí no existen registros de continuación. Estos registros también van por pares. Aproximadamente la mitad de los record sets vienen de sujetos que tienen PAF.

PARTE DEL RETO ES IDENTIFICAR ESTOS RECORD SETS (LOS QUE TIENEN PAF) Y OTRA PARTE ES IDENTIFICAR QUE REGISTRO DE CADA PAR PRECEDE AL EPISODIO DE PAF.

<https://www.nature.com/articles/nrcardio.2016.118>

<https://physionet.org/files/afpdb/1.0.0/paf.shtml>

El objetivo general del Computers in Cardiology Challenge 2001 es caracterizar los cambios en el electrocardiograma de superficie inmediatamente antes del inicio de la fibrilación auricular paroxística (FAP), en un esfuerzo por desarrollar un algoritmo de predicción fiable para la arritmia.

1 de mayo de 2023

Condor_ssh ipython para cuando no esté segura de lo que estoy haciendo o con los cuadernos.

14 de mayo de 2023

Juan Pablo ya me ha aclarado lo que debo hacer con las bbdd en las que detectamos muestras NaNs. Voy a arreglar las bbdd en las que encontré muestras NaNs y a obtener representaciones de las mismas para comprobar que he hecho bien el proceso. Por otro lado, las procesaré y las escribiré, como ya hice para el resto de bbdd, una vez reciba el visto bueno de los profesores.

Las bbdd en las que me aparecían muestras NaNs son:

- Autonomic
- Georgia
- Ecg-arrhythmia
- Icentia 11k
- Mimic-iv

15 de mayo de 2023

Hoy he arreglado el problema de las muestras NaNs en las bbdd Ecg-arrhythmia, Georgia y Mimic-iv. He modificado los scripts de procesado de cada una de ellas para corregir las muestras NaNs una vez se detecten en las señales, y también he obtenido representaciones para algunas de las señales, donde se ve la señal antes de corregir las muestras NaNs (trozos en blanco) y tras cambiar estas por -32768 (en mV). Las he añadido al power point.

21 de mayo de 2023

Hoy he arreglado el problema de las muestras NaNs en las bbdd Icentia-11k y Autonomic. También he revisado las gráficas del pwp que puse en su día para dos de las bbdd que forman parte de la tarea de control, afpdb y ptbdb.

22 de mayo de 2023

He comenzado escribiendo el recordatorio a los autores de la bbdd autonomic, ya que no me han respondido todavía y ahora sí que podríamos usar esa bbdd.

Por otro lado, he terminado de arreglar las etiquetas de las gráficas del pwp, ayer por la noche me di cuenta de que algunas estaban mal puestas.

He seguido con la **tarea de control de AF** (en rojo se muestran mis dudas):

➤ **0_prepare_data:**

El fichero [run1_prepare_lists.py](#) no lo he empleado, ya que en esta base de datos ya vienen definidas que señales conforman el training set y cuales el test set (como explico en la página 19; Descripción de los datos: esta base de datos se divide en un set de

aprendizaje, formado por aquellos registros que comiencen por la letra n o por la letra p, y un set de testeo, formado por los registros que comienzan con la letra t). Así que he realizado la división “a mano”.

A la hora de generar los ficheros que contienen las etiquetas, también me he basado en las explicaciones de Physionet. Los ficheros que comiencen por la letra p vienen de sujetos que tienen PAF y los que comienzan por la letra n vienen de sujetos que no tienen PAF. Por otro lado, las etiquetas del test set no se proporcionan, pero indica que aproximadamente la mitad de ellos tienen PAF. ¿¿¿PUEDO NO TENER ESTAS ETIQUETAS??? ESTA BASE DE DATOS ES DEL CHALLENGE DE 2001, EN PHYSIONET ESTÁN LAS SOLUCIONES ASÍ QUE SACO LAS ETIQUETAS DE ALLÍ

➤ **1_1_conv1d_avgpool:**

He añadido explicaciones en [net.py](#).

También he analizado el código de [gen.py](#) y he añadido las funciones necesarias para la lectura de ficheros binarios de forma aleatoria y no aleatoria en [fe.py](#). No sé si debo añadir funciones para leer los ficheros .csv, de momento creo que no.

Por último, he estudiado el fichero principal [run1_train.py](#). En primer lugar, he modificado el fichero [config.json](#) poniendo mis ficheros de datos y etiquetas, después solo he añadido explicaciones. Dentro de la carpeta models, [model.json](#) también lo debo modificar.

23 de mayo de 2023

He seguido con la **tarea de control de AF** (en rojo se muestran mis dudas):

➤ **1_2_conv1d_avgpool:**

He comenzado modificando la carpeta de las listas, `lst`, y he añadido la mía, [ecglst](#), dejando solamente los ficheros buenos, que serían 7. Por otro lado, he analizado la red, [net.py](#). He visto que es igual que la anterior, con la única diferencia de que en este caso tenemos menos capas en el self.encoder. El resto de ficheros: [gen.py](#), [fe.py](#), [run1_train.py](#), [config.json](#) y [model.json](#), los he copiado de la carpeta anterior para mantener los comentarios, pues eran iguales.

➤ **2_1_conv1d_lstm_avgpool:**

He comenzado modificando la carpeta de las listas, `lst`, y he añadido la mía, [ecglst](#). Respecto a la red, [net.py](#), es diferente a las anteriores. Esta tiene menos capas convolucionales que las anteriores y, además, antes de la capa completamente conectada emplea una capa LSTM bidireccional. Leí sobre las mismas hace meses, en uno de los papers que me pasó Antonio.

Dentro de la misma el método forward también ha cambiado debido a la capa LSTM, hay que reordenar las dimensiones para la misma, pero el fundamento básico es el mismo.

El resto de ficheros: [gen.py](#), [fe.py](#), [run1_train.py](#) y [config.json](#), los he copiado de la carpeta anterior para mantener los comentarios, pues eran iguales.

➤ **3_1_conv1d_transformer_avgpool:**

He comenzado modificando la carpeta de las listas, lst, y he añadido la mía, [ecglst](#). La red, [net.py](#), es diferente a las anteriores porque incluye un bloque de codificador de modelo Transformer, [transformer.py](#), antes de la capa completamente conectada. Me ha llamado la atención la función de activación GELU, similar a la RELU.

El resto de ficheros: [gen.py](#), [fe.py](#), [run1_train.py](#) y [config.json](#), los he copiado de la carpeta anterior para mantener los comentarios, pues eran iguales.

➤ **3_2_conv1d_transformer_clstok:**

He comenzado modificando la carpeta de las listas, lst, y he añadido la mía, [ecglst](#). La red, [net.py](#), es similar a la anterior, pero ahora tengo un token, este se emplea en el módulo del codificador del transformer. Por otro lado, [transformer.py](#), es similar al anterior.

El resto de ficheros: [gen.py](#), [fe.py](#), [run1_train.py](#) y [config.json](#), los he copiado de la carpeta anterior para mantener los comentarios, pues eran iguales.

24 de mayo de 2023

He seguido con la **tarea de control de 12 LEADS:**

En esta tarea se emplean dos bbdd, mimic-iv y ptbdb. Lo que he hecho ha sido preparar los ficheros con las señales binarias y las etiquetas de estas dos bbdd tratándolas como una sola.

Primero, copio los ficheros binarios de cada bbdd a la carpeta ecgdata dentro de la tarea de control correspondiente, y después está la vuelvo a mover dentro de 0_prepare_data. Ahora, debo preparar los ficheros que contienen los nombres de las señales que se emplearan para train, test o dev. Esto lo hago ejecutando directamente el script [run1_prepare_list.py](#):

```
sartal@nv3090-17 0_prepare_data $ python run1_prepare_lists.py
total files: 16143
train files: 15143
dev files: 1000
test files: 1000
```

Una vez tengo las listas con los ficheros que forman cada uno de los conjuntos de datos que empleará la red neuronal debo preparar los respectivos ficheros con las etiquetas. DEBEN ESTAR ALINEADOS. Para las etiquetas emplearé la derivación del ECG de la que se trate, que recordemos aparece en el nombre de cada uno de los ficheros. Por tanto, debo leer cada una de las líneas, ver de qué derivación del ECG se trata y escribir esta en la misma posición, línea, en el fichero de etiquetas. Esto lo haré para cada conjunto de datos.

Después de preparar los ficheros de 0_prepare_data, es necesario que cambie algunas cosas con respecto al example_AF y es que ahora los ficheros de 0_prepare_data/ecglst se llaman diferente, por ello, cambio, [model.json](#), [config.json](#) y [gen.py](#).

Retomo la **tarea de control de AF** pues me quedaba un fichero de etiquetas por generar:

➤ **0_prepare_data:**

Me he dado cuenta de que esta bbdd era de un Challenge, el de 2001, y he buscado las soluciones del test set, así que ahora ya puedo tener las etiquetas para esa parte.

[Predicting Paroxysmal Atrial Fibrillation/Flutter: The PhysioNet/Computing in Cardiology Challenge 2001 v1.0.0](#)

Los sujetos del grupo A tienen PAF mientras que los sujetos del grupo N no tienen PAF. Hablan sobre 2 eventos:

○ **Evento 1 (filtración PAF):**

La idea es determinar si los sujetos del grupo A pueden ser distinguidos de aquellos del grupo N, es decir, si los individuos que tengan riesgo de sufrir PAF pueden ser identificados basándonos en su ECG.

La solución a este evento es: [event-1-answers - Predicting Paroxysmal Atrial Fibrillation/Flutter: The PhysioNet/Computing in Cardiology Challenge 2001](#)

t01 A	t51 A
t03 N	t53 A
t05 A	t55 N
t07 A	t57 A
t09 A	t59 N
t11 A	t61 A
t13 N	t63 N
t15 N	t65 A
t17 A	t67 N
t19 N	t69 N
t21 A	t71 A
t23 A	t73 N
t25 N	t75 A
t27 A	t77 A
t29 A	t79 A
t31 N	t81 A
t33 N	t83 N
t35 N	t85 A
t37 N	t87 A
t39 A	t89 A
t41 N	t91 N
t43 N	t93 N
t45 N	t95 N
t47 A	t97 A
t49 A	t99 A

○ **Evento 2 (predicción PAF):**

Este evento quiere determinar si los sujetos del grupo A tienen cambios distinguibles y detectables en sus ECGs inmediatamente antes de un episodio de PAF, es decir, es el comienzo de un episodio de PAF predecible en un individuo que sabemos que tiene riesgo de sufrirlo. Al final, se quiere

determinar que record del grupo A, recordemos que van por parejas como ya expliqué arriba, precede a ese episodio de PAF.

Por lo que he entendido con las etiquetas del evento 1 me sirve, porque lo que se hace en el mismo es identificar a los sujetos del grupo A, es decir, aquellos que tengan PAF. Como explico más arriba, van por pares, así que entiendo que si t01 pertenece al grupo A, es decir, es de un sujeto que tiene PAF, t02 también. Después lo que se hace en el evento 2 es identificar, para aquellos sujetos que tienen PAF, de sus dos grabaciones cual es la que precede al PAF y se identifica con la letra A (precede inmediatamente) y cual es la que contiene el PAF, que se identifica con la letra N (no precede inmediatamente).

Cuando he generado las etiquetas a partir del documento con las respuestas he asociado la letra A, que el sujeto tiene PAF, con el número 1 y para los sujetos que no tienen PAF, letra N, he asignado el número 0, como ya hice anteriormente.

TAREA DE CONTROL 12 LEADS:

Comienzo con el modelo más sencillo, el que tiene menos capas. Este era el segundo, pero he cambiado los nombres para que esté el primero.

1_1_conv1d_avgpool: Este primer modelo ha sido entrenado con los datos de las bdd mimiv y ptb.

➔ Como veo en el tfg de Daniel él fue aumentando progresivamente el volumen de datos que se empleaban en entrenar el modelo, yo eso no lo estoy haciendo (de momento).

Ejecuto el modelo y veo los resultados. ME DA ERROR, AUNQUE HE ARREGLADO VARIAS COSAS QUE QUIERO ARREGLAR EN EL RESTO DE MODELOS.

25 de mayo de 2023

TAREA DE CONTROL 12 LEADS:

1_1_conv1d_avgpool: Resulta que ayer me daba un error donde decía que los archivos que quería leer no existían, y es que no me había dado cuenta de que la carpeta de los ficheros binarios, ecgdata, también debe estar dentro de cada uno de los modelos implementados, al igual que la carpeta con los nombres de los ficheros y las etiquetas, ecglst, cosa que ya estaba haciendo.

Antes de ponerme a ejecutar los modelos y evaluar los resultados voy a dejarlos arreglados.

1. He tenido que arreglar las carpetas **ecgdata** y **ecglst** en cada uno de ellos. Ecgdata porque estaban también los ficheros con las derivaciones VX, VY y VZ, las cuales he querido quitar por no meter más complejidades al diseño. Por otro lado, ecglst porque he eliminado los ficheros mencionados antes y porque en los modelos seguían los ficheros para la otra tarea de control, ya que hice una copia pega de toda ella.

```
sartal@hermes2 0_prepare_data $ python run1_prepare_lists.py
total files: 14496
train files: 13496
dev files: 1000
test files: 1000
```

2. **gen.py** tenía una serie de fallos, es el mismo para todos los modelos, lo reutilizo.

Parece que ha funcionado, voy a tratar de interpretar los resultados obtenidos. En primer lugar, debo entender que para el entrenamiento hago 50 iteraciones (epochs) y en cada una de las mismas tengo 4000 ficheros, es decir, en cada epoch paso por completo ese conjunto de 4000 ficheros de entrenamiento, pero estos ejemplos se presentan a la red neuronal en lotes, en este caso, de tamaño 32 (batch), y se calcula la pérdida o error entre las salidas de la red y las etiquetas conocidas. Luego, se realiza la retropropagación del error para ajustar los pesos de la red y mejorar el rendimiento del modelo.

Voy a ir ejecutando los modelos implementados y después los comparo ejecutando **./results_best.sh**, donde me sale el mejor resultado de todos los epochs de cada modelo.

1_2_conv1d_avgpool: Aumento el número de capas.

1_3_conv1d_avgpool_softmax: Añado una capa final de tipo softmax.

2_1_conv1d_lstm_avgpool:

3_1_conv1d_transformer_avgpool:

3_2_conv1d_transformer_clstok:

He añadido en el pwp todas las explicaciones sobre el proceso que he seguido para esta tarea de control de las 12 derivaciones. Tengo bastantes dudas que espero poder aclarar mañana.

TAREA DE CONTROL AF:

He seguido añadiendo al pwp la explicación de generación de ficheros para esta tarea de control y ahora debo terminar de ejecutar los distintos modelos, y plantear uno nuevo con las explicaciones que di previamente para añadirlo a la presentación de mañana.

He estado ejecutando algunos de los modelos y mirando la función de pérdidas.

26 de mayo de 2023

Hoy he terminado de añadir algunas gráficas al pwp, pero es que creo que no me funciona bien porque no estoy normalizando los datos antes de dárselos a la red, me he dado cuenta justo ahora, antes de la reunión, así que no lo voy a poder probar. De todas formas, seguro que hay que hacerlo y lo preguntaré, entiendo que lo debo hacer en **gen.py** o en **run1_train.py**.

Reunión:

- ✓ Arreglo muestras NaNs. Está bien, sustituimos todos los NaNs por saturaciones negativas, -32768. A la hora de escribir las señales las guardaré filtradas y sin filtrar, esto debo hacerlo para las que ya tenía escritas también.

En cuanto a las tareas de control:

✓ **Tarea de control 1. Clasificación de las 12 derivaciones del ECG (clasificación multiclase excluyente).**

- ✓ Los ficheros de **test** y **dev** no deberían ser iguales, debo revisar el código para ver cómo se generan, [run1_prepare_lists.py](#).
- ✓ Como formato de las **etiquetas** empleamos los números de canal, no el nombre.

En cuanto a los **modelos**, puedo probar de nuevo a ir variando los parámetros con algo más de lógica, para lo cual debería informarme más sobre el optimizador, las capas convoluciones, etc, pero parece que los resultados que estaba obteniendo no son malos. Además, tampoco estamos buscando que sean mucho mejores, cosa que yo no sabía, pues estas tareas las aplicaremos tras el modelo de red no supervisado y es entonces cuando tendremos que ver una mejora significativa.

- ✓ Por otro lado, en todos ellos, dejo como dimensión de salida 12, es decir, el número de clases posible, que se corresponde con el número de derivaciones del ECG convencional. Y sobre las dimensiones de los kernel y stride no hace falta que toque nada porque no estamos buscando afinar mucho tampoco. Después, sí que deberemos adaptar las dimensiones para poder pasar por la red las salidas del modelo no supervisado.

Desde el fichero [config.json](#) es desde donde debo ir variando mis parámetros, no importan que estos tomen un valor distinto a los de gen.py, ya que los del fichero de configuración los machacaran. Los de gen.py están por si te olvidas de ponerlos en el fichero de configuración y que haya un valor con el que trabajar. Aquí hay dos parámetros que debo subir:

- ✓ nb_samples epoch=13496
- ✓ epochs=200, ¿las de test y dev las subo también? Las he subido

La normalización no hace falta que la hagamos en esta parte, eso entendí que será en la tarea no supervisada.

- ✓ Sobre el modelo en el que añadí al final una capa de tipo softmax, no es necesario, en el método loss la función de CrosEntropía ya lo está haciendo.

Para interpretar los resultados no solo debo emplear la función de pérdidas, esto lo explicaré después, pero es que además debo tener en cuenta que tengo 12 clases. Por lo que si mi tasa de error en test está en torno a 0.5, no está tan mal!! ¡¡Acierto la mitad!! *Si hacemos: $\ln(12)=2.48$, todo lo que esté en torno a esto es malo, pero lo que ya se encuentre por debajo está bien, no estamos tirando una moneda al aire.*

✓ **Tarea de control 2. Detección de AF (clasificación binaria excluyente).**

- ✓ La **división de los ficheros** no está bien, me olvido del conjunto de validación, dev, y la repito. Solamente hago un conjunto de train (300 señales) y otro de test (200 señales). Por otro lado, debo volver a leer la explicación de PhysioNet para poder entenderlo bien y ver si efectivamente debo quitar los ficheros de continuación (los que terminan en c).

Sobre los **modelos** lo de tener 1 neurona o 2 a la salida realmente no importa, no hay apenas diferencia entre los resultados. Ahora sí que nos interesa tener una capa de tipo softmax.

Para interpretar los resultados no solo debo emplear la función de pérdidas, esto lo explicaré después, pero es que además debo tener en cuenta que ahora tengo solo 2 clases y por eso ya de por sí la función de pérdida será más pequeña. Si hacemos: $\ln(2)=0.69$, todo lo que esté en torno a esto es malo, pero lo que ya se encuentre por debajo está bien, no estamos tirando una moneda al aire.

Por otro lado, esta bbdd es de un Challenge, el de 2001, y hay **papers** donde se muestran los resultados obtenidos, podría compararlos para ver si es aceptable lo que yo estoy obteniendo. Estos papers no emplean redes neuronales sino segmentación.

- ✓ **Evaluación de las prestaciones.** Como ya he mencionado, no solo debo centrarme en observar la función de pérdidas, tampoco es muy representativa de lo bien o mal que lo estoy haciendo. Me han propuesto varias representaciones, las cuales puedo encontrar en sklearn:
 - ✓ Matriz de confusión para la tarea 1. https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html
 - ✓ ROC + AUC o DET mejor para la tarea 2. Esto lo usa Daniel en su TFG. https://scikit-learn.org/stable/auto_examples/model_selection/plot_det.html
 - ✓ Especificidad y sensibilidad para la tarea 2. https://scikit-learn.org/stable/modules/model_evaluation.html#balanced-accuracy-score
 - × ¿Tasa de acierto top 5? De esto no me enteré bien, pero se nombró dos veces en la reunión.
 - ✓ Además, puedo observar los mejores **errores en test** de cada uno de los modelos para compararlos entre sí.
- × **Transformers.** Puedo mirar el paper que me pasó Antonio, pero debería buscar algo más didáctico para poder entenderlo mejor. Al final, si no me siento cómoda y no los empleo no pasa nada. **NO LOS USO.**
- ✓ **Contrastive Predictive Coding.** EN ESTE SÍ QUE DEBERÍA CENTRARME. Leerlo antes de la próxima reunión para la tarea no supervisada.

Nota: Como criterio de representación, y creo que también para pasarle las muestras a la red *no supervisada*, emplearemos la amplitud pico a pico del QRS. Lo que esté por encima de 3 mV o por debajo de 0.25 mV se desecha. La normalización también se hará en esta tarea no supervisada.

29 de mayo de 2023

Hoy he comenzado con el arreglo de las bbdd que contienen muestras NaNs, después de que los tutores me dieran el visto bueno. Por otro lado, me pidieron que guardase las señales sin filtrar también, así que eso lo debo hacer para todas las bbdd, incluso las que ya tenía escritas (con el filtro). He guardado las señales filtradas y no filtradas en carpetas con distinto nombre.

También he arreglado los ficheros de la tarea de control 1, tanto los de los nombres como los de las etiquetas.

30 de mayo de 2023

Sigo con el arreglo de las bbdd que tienen muestras NaN, solo me queda icentia11k, que como es tan grande está tardando mucho.

Paso a arreglar los ficheros que se van a emplear en la segunda tarea de control, la de detección de AF.

He seguido con la tarea de control 1, añadiendo la matriz de confusión y obteniendo representaciones para los distintos modelos propuestos. Todos los resultados están en el pwp.

Por otro lado, he comenzado con la tarea de control 2, he añadido la representación de la curva DET para poder interpretar los resultados de distintas maneras, pero creo que me sale muy mal. Me he quedado atascada. También debería mirar la especificidad y la sensibilidad.

6 de junio de 2023

Comienzo con la escritura de la bbdd icentia11k, pero es que se me queda atascado. Sigo con la segunda tarea de control, detección de AF.

He empezado de cero, modificando la función de pérdidas que debo emplear en mis redes, pues se trata de una tarea de clasificación **binaria**.

Por otro lado, he modificado el fichero run1_train.py, en él, calculo la curva DET, la exactitud, especificidad y sensibilidad. También he vuelto a modificar el conjunto de ficheros que voy a emplear, ya que había eliminado todos los de continuación, incluso aquellos que no tienen PAF, creo que en eso me equivoqué. Ahora tengo 300 señales de train y 200 de test.

LA CURVA DET ME SALE EN BLANCO, DEBO REVISARLO.

7 de junio de 2023

Comienzo arreglando el problema de la curva DET, me sale en blanco, cosa que no me pasaba con los modelos anteriores. No consigo solucionarlo, al final la represento "a mano". Los resultados que obtengo son regulares.

He añadido las explicaciones con el código implementado, así como las representaciones obtenidas al pwp.

Sigo con la escritura de icentia11k. Lo voy haciendo por partes, ya que se me queda colgado.

¡¡¡HE SOLUCIONADO EL PROBLEMA QUE TENÍA CON LA CURVA DET!!! Resulta que para calcularla estaba usando las etiquetas predichas en lugar de las probabilidades a posteriori.

8 de junio de 2023

Hoy les he enviado a los profesores los resultados de la segunda tarea de control.

9 de junio de 2023

Antes de la reunión de hoy a la 13 voy a leer de nuevo los papers de CPC y LSTM, ya que estos serán importantes para el mismo.

Reunión:

- ✓ **Tarea de control 1**, clasificación 12 derivaciones del ECG:

Arreglar ficheros de train, test y validación. Volver a sacar los resultados.

Primero divido el total de mis datos, x , en x_1 y test, 1000 señales para este último y el resto para x_1 . Después x_1 lo vuelvo a dividir en x_{train} y dev, 1000 señales para este último y el resto para x_{train} . Al final, me quedan 12496 señales para train, 1000 para test y 1000 para dev.

```
sartal@nv3090-17 0_prepare_data $ python run1_prepare_lists.py
total files: 14496
train files: 12496
dev files: 1000
test files: 1000
```

Tras la nueva división, que es la correcta, vuelvo a generar los ficheros con mis etiquetas.

Vuelvo a lanzar cada uno de los modelos y a ver los resultados. He tenido que cambiar también el número de señales por epoch, son 12496, no 13496.

Por lo que estoy viendo los ahora los resultados mejoran al ir subiendo la tasa de aprendizaje, cosa que antes no sucedía. En el caso de la red con una capa lstm es al contrario.

- ✓ **Tarea de control 2**, predicción AF:

- ✓ Subir número de muestras de la señal que ve la red, pasamos de 1500 muestras, es decir, 3 segundos, a 5000 muestras, que equivalen a 10 segundos. Julia me pasará un código para fijar los umbrales de sensibilidad, especificidad y exactitud (creo).
- ✓ Volver a sacar los resultados tras estos cambios.
- ✓ **MODELO NO SUPERVISADO**
<https://www.baeldung.com/cs/cnn-receptive-field-size> para saber que valores de kernel o stride emplear, y no ir probando a lo loco.
- ✓ Net.py: poner print para ir viendo los tamaños.
- ✓ Básicamente, ir mirando el código explicándolo y buscando información.

Nota: Como criterio de representación, y creo que también para pasarle las muestras a la red *no supervisada*, emplearemos la amplitud pico a pico del QRS. Lo que esté por encima de 3 mV o por debajo de 0.025 mV se desecha. La normalización también se hará en esta tarea no supervisada.

Por la tarde he seguido con la escritura de icentia11k.

10 de junio de 2023

Hoy he arreglado la tarea de control 1 y he obtenido los nuevos resultados, he hecho un nuevo pdf con ellos.

12 de junio de 2023

He comenzado mirando la tarea de control 2, cambio, en todos los ficheros gen.py el número de muestras de señal que va a leer la red, paso de 1500 muestras (3 segundos) a 5000 muestras (10 segundos). Me falta fijar los umbrales para especificidad, etc, a 0.75 como dijeron en la reunión, le he pedido el script a Julia.

Sigo mirando el modelo no supervisado. He ojeado los códigos y me he centrado en explicar el de la red, net_cpc.py, pues es lo único diferente. También he puesto prints para ir viendo los tamaños. Por otro lado, he estado leyendo sobre el campo receptivo y como calcularlo, pero he encontrado una nueva página así que seguiré mañana.

13 de junio de 2023

He empezado mirando los tamaños de las variables z, c y p, así como de la salida de la red poniendo prints y ejecutando el fichero net_cpc.py. He conseguido entender cómo funciona y he planteado las situaciones correspondientes a las dos tareas de control, aunque no sé si esto tiene algo de sentido. En uno de estos casos, al calcular, la función de pérdidas obtengo error. Creo que se debe a que le estoy pasando una longitud de secuencia de 1500 muestras y n es 8, por lo que no me sale un número entero de ventanas de 8 muestras.

He seguido con la segunda tarea de control, cambiando el umbral para generar las etiquetas predichas a 0.75 en lugar de a 0.5. Después, he obtenido nuevas representaciones, me falta terminarlas y comparar los modelos.

También he lanzado, con `condor_command` la escritura de las señales de icentia11k sin filtrar (765586).

14 de junio de 2023

Ayer lancé, con `condor_command`, la escritura de las señales filtradas de la bbdd icentia11k y ya ha terminado. Luego he lanzado la escritura de las señales sin filtrar que quedaban, ya ha terminado.

Después he terminado el pwp con los resultados que he obtenido para la segunda tarea de control, tras aumentar el número de muestras que le presento a la red a 5000 y fijar el umbral a la hora de generar las etiquetas predichas a 0.75, en lugar de a 0.5. Esto último que me pidieron no se si realmente se hace así, pero quería avanzar con ello para poder centrarme en el modelo no supervisado.

Paso a revisar lo que me falta del modelo no supervisado antes de plantearles un correo con dudas a los profesores. Primero he buscado más información sobre el campo receptivo y después lo he calculado, se me queda en 3262 muestras, no sé si sería necesario aumentarlo hasta 5000 muestras, es decir, 10 segundos de señal. Que una neurona de la capa final pueda ver la señal por completo.

Les he mandado un correo con las dudas que me han surgido.

Por la tarde me he centrado en entender mejor la función de pérdidas `ctc_loss` y también he mirado el código que me ha pasado Julia para los umbrales de la segunda tarea de control.

16 de junio de 2023

Hoy paso a mirar la tarea de control 2, solo falta lo del umbral según lo indicado en el correo de Antonio y el código de Julia.

He definido las funciones que me faltaban, TPR y TNR, a partir del siguiente ejemplo: <https://github.com/mokhanna/Diabetes-Prediction/blob/main/Diabetes%20Prediction.ipynb>.

Le he mandado un correo a Julia preguntándole. Al final, Julia me ha pasado el código de las funciones TPR y TNR. Ejecuto y me sale un nuevo error, me faltaba el método `predict_window`. Se lo pido a Julia, estas dos funciones las defino en mi red. He ejecutado los modelos y obtenido nuevos resultados. Aún no los he añadido a la presentación.

17 de junio de 2023

Paso a mirar el modelo no supervisado.

Preparación de los datos:

- ✓ Mover las señales a emplear a una carpeta dentro del modelo -> cuento con 1.602.185 señales, ~~me falta 11k~~.
- ✓ Generar las divisiones en los ficheros de train, test y dev -> para poder hacer las pruebas después con el loader realizo la siguiente división. Un 70% de las señales van para train, mientras que un 15% va para test y otro 15% para dev.
- ✓ No debo generar los ficheros con las etiquetas, ¿no? Correcto, en un modelo no supervisado no se utilizan etiquetas o categorías como información de entrenamiento. La red neuronal se entrena únicamente con los datos de entrada sin ningún tipo de supervisión externa.

Entrenamiento:

- ✓ Copiar las carpetas ecgdata y ecglst a los modelos planteados, 1_1_cpc_type 1 y 1_1_cpc_type2
- ✓ **gen_mod.py**: Lo he modificado para no tener en cuenta las etiquetas, aquí voy a ir probando el loader de las señales no supervisadas, que extraiga aleatoriamente un segmento, aunque sean largas, y representar para ver qué pinta tienen.
- ✓ **run1_train_cpc.py**: empleo config_cpc.json, net_cpc.py y gen_mod.py.

DUDAS

Preparación de los datos:

- ✓ **0_prepare_data**
¿Cómo realizo la división de las señales en los conjuntos de train, test y dev? ¿Qué % destino a test y dev? ¿Hay que tener en cuenta qué derivación del ECG es a la hora de repartirlas para tener una división de clases equilibrada en estos conjuntos?

Entrenamiento:

- **1_1_cpc_type1**
- ✓ **config_cpc.json**

```
{
  "seed": 42,
  "device": "cuda",
```

Elimino los ficheros con las etiquetas.

```
"file_wav_train": "ecglst/train.bin.list",
"file_wav_dev": "ecglst/dev.bin.list",
"file_wav_test": "ecglst/test.bin.list",
"model": "models/model_cpc",
"metrics": "results/metrics_cpc",
"nb_workers": 6,
```

¿Qué valores uso? ¿Cómo muestras todo el conjunto de train, cuantas epochs?

```
"nb_samples_epoch": 4000,
"nb_epochs": 50,
"nb_epochs_dev": 5,
```

```
"nb_epochs_test": 10,  
"batch_size": 64,  
"lr": 0.001,  
"lr_step": 20,  
"lr_gamma": 0.95,  
"clip_grad": 1.0,
```

¿No empleamos dimensión de salida? En realidad, si no le pongo valor a ese parámetro en config, se usará el que hay en net_cpc.py, es decir, 2. **¿Esto es lo que queremos?**

```
"input_dim": 1,  
"hidden_dim": 1000  
}
```

gen_mod.py

¿Normalizamos las señales antes de pasarlas a la red como entrada?

He probado a hacerlo dentro del script de gen_mod.py, en concreto, en el método getitem, para todos los conjuntos de datos.

¿Descartamos en función de la amplitud de las muestras?

Esto también lo podría hacer dentro del método anterior, getitem, para todos los conjuntos de datos.

REPRESENTACIONES SEÑALES PARA VER QUE EL LOADER FUNCIONA CORRECTAMENTE.

✓ run1_train_cpc.py

- **¿Añadimos otras métricas como la matriz de confusión, ROC, DET, ¿etc?**
Creo que no, que comprobaremos los cambios que hagamos (experimentos) con las tareas de control.

18 de junio de 2023

Como aún no han terminado de escribirse las señales de la bdd icentia11k, paso a la tarea de control2.

- ✓ Repaso + entender
- ✓ PWP
- ✓ **DUDAS:**
 - **¿Debo hacer lo del umbral en el bucle de test, no, o al final del todo?**
 - Además, he cambiado el código, no hago determinados bucles. Explicarlo en persona.
 - **¿Qué valor empleo para random_cut_len, he usado 1500, como Julia?**

Paso a mirar el modelo no supervisado. Se ha terminado de escribir la base de datos icentia11k, ya he movido todas. En total cuento con **1602185 señales**.

Tras haberlas movido todas a la carpeta ecgdata dentro de 0_prepare_data, realizo una división de prueba en los conjuntos de train, test y dev.

```
sartal@nv3090-16 0_prepare_data $ python run1_prepare_lists.py
total files: 1602185
train files: 1121531
dev files: 240327
test files: 240327
```

Al ser un modelo supervisado no debo generar los ficheros con las etiquetas.

19 de junio de 2023

He comenzado repasando la tarea de control 2, predicción de PAF. He apuntado bien mis dudas y he terminado de arreglar el pwp para la próxima reunión.

Por la tarde he estado haciendo el pwp del modelo no supervisado, añadiendo el código que he cambiado y comentando las dudas que tengo, me falta la representación de las señales en gen_mod.py para ver que el loader lo hace bien. Como aún no se han escrito las señales en la carpeta ecgdata dentro de 1_1_cpc_type1 no lo he podido probar.

20 de junio de 2023

De momento se han copiado 704420 señales a la carpeta de mi modelo y hay 1602185 señales en total. Paso a generar las líneas de código para representar las señales cargadas por el DataLoader.

He tenido un problema con el ordenador y no he podido avanzar más.

21 de junio de 2023

Ya han terminado de escribirse las señales a la carpeta ecgdata dentro de 1_1_cpc_type1 (ha costado dos días prácticamente). Hoy termino lo que me habían mandado del modelo no supervisado, probar el DataLoader. Repaso dudas y leo los correos intercambiados.

Redacción TFG:

He hecho la portada del TFG siguiendo el modelo que dan en la EINA, también he planteado una primera versión de la estructura (me he basado en el TFG de Daniel, ya que es más o menos parecido) y comenzado con la redacción de la introducción.

22 de junio de 2023

Hoy he estado trabajando en la introducción.

23 de junio de 2023

Por la mañana antes de la reunión he seguido con la introducción.

Reunión:

- **Tareas de control:**

- ✓ **Tarea de control 1 (clasificación 12 derivaciones del ECG)**

Dentro del código de la red, `net.py`, debo modificar el encoder, ya que los tamaños de kernel y stride están igual que para el problema de audio y esto no es adecuado para una señal de ECG.

Kernel = 3

Stride = 2

Creo que en la tabla donde se presentan los mejores resultados no debo añadir nada, ya que aquí solo calculaba la función de pérdidas y la matriz de confusión solamente.

Repito los resultados y se lo mando a los profesores.

- ✓ **Tarea de control 2 (predicción de AF)**

Debo hacer varios cambios:

`net.py`: debo modificar el encoder, ya que los tamaños de kernel y stride están igual que para el problema de audio y esto no es adecuado para una señal de ECG.

Kernel = 3

Stride = 2

Además, no empleo el método `predict_window` de Julia, sino `predict`, que es como lo había hecho yo hasta ahora. Paso por mi método `forward` y después por la sigmoide (tengo una neurona, no dos como ella).

`config.json`: elimino el parámetro `random_cut_len`.

`run1_train.py`: En la función `test` empleo el método `predict`, no `predict_window`. La matriz de confusión que obtengo es para el 50%, la quiero para el 75%. La debo hacer con el umbral que obtengo a partir del código de Julia, `Cutoff`.

En la tabla de `results_best.sh` debo añadir los valores de auc y de sp de comprobación (para una sensibilidad del 75%)

Repito los resultados y se lo mando a los profesores.

- **Modelo no supervisado:**

- ✓ **División de los ficheros.** Tengo 1.602.185 señales en total, me quedo con 1.600.000 señales para train y el resto, 2.185 señales, para test. No hago dev.

- **`net_cpc.py`.** Debo plantear distintos modelos, con diferente número de capas. A Julia le funcionaba de 4 a 6 capas. También podría añadir más capas en el encoder. En todos los modelos debo modificar los valores de kernel y stride en el encoder:

- Kernel = 3

- Stride = 2, stride total de 16 (4 capas), 32 (5 capas) o 64 (6 capas).

La “tarea” consiste en adivinar el segmento de continuación de la señal que se le presenta a la red, es decir, como continúa esa señal.

Los distractores ($n=8$), serán trozos anteriores de esa misma señal, de forma que esos trozos tengan la misma periodicidad. No me quedo muy claro si también habrá distractores que serán trozos de otras señales (estos no tendrán la misma periodicidad), creo que sí. Entre las señales que se le presenten como posible respuesta, se encontrará el fragmento de señal correcto.

* Para poder sacarle más partido, lo que se puede hacer es darle la vuelta a la señal y pedirle que diga la continuación, pero desde el principio de la señal (que ahora es el final).

- ✓ [config_cpc.py](#). Utilizo todo el train como número de muestras por epochs, 1.600.000, y subo el número de epochs (200 en train y 10 en test). Como dimensiones ocultas puedo tener 1000 o más. De momento dejo 1000. Añado aquí el parámetro para indicar el número de distractores que queremos.
- ✓ [gen_mod.py](#). No normalizamos ya que podríamos perder aspectos fisiológicos importantes de la señal. Lo que sí que haremos será descartar los segmentos de señal que se le presentan a la red en función de la amplitud pico a pico del QRS. Es decir, hacemos el valor absoluto de la señal, y todo lo que esté por encima de 3 mV, o por debajo de 0.25 mV, será descartado. Imprimimos un mensaje de error, descartamos ese segmento y nos vamos a coger otro.
- ✓ [run1_train_cpc.py](#). Tengo que quitar también las etiquetas. Como representaciones tenemos la función de coste (loss) y la tasa de acierto, entiendo que la de error. Algo razonable estaría entre el 60 y el 90%.

ESTOS PESOS ME LOS LLEVARÍA A LA TAREA DE CONTROL Y TENDRÍA QUE VER QUE MEJORAN LOS RESULTADOS. CREO QUE LA RED QUE HAY EN LAS TAREAS DE CONTROL HABRÍA QUE MODIFICARLA.

- **EXPERIMENTOS.** Primero vamos a ver que la red consigue hacer razonadamente bien y después haremos algunos experimentos. Podría pensar en comprobar que RF va mejor, uno pequeño, otro mediano y otro más grande.

24 de junio de 2023

Escritura introducción.

25 de junio de 2023

Sigo con la escritura de la introducción.

26 de junio de 2023

Hoy he medio arreglado el planteamiento de la estructura del TFG y he seguido con la introducción. También he ejecutado la tarea de control 1 con los últimos cambios de la reunión, me falta revisar bien lo que he puesto en el pwp.

27 de junio de 2023

Termino la tarea de control 1.

28 de junio de 2023

Introducción. He intercambiado algunos correos sobre la tarea de control 2.

29 de junio de 2023

Hago la tarea de control 2. Termino la introducción y la leo.

30 de junio de 2023

Les he enviado un correo a los tutores con los resultados de las dos tareas de control, la estructura planteada para el TFG y la introducción.

Empiezo hoy a mirar el modelo no supervisado (reunión 23/06/23). No he avanzado prácticamente nada.

1 de julio de 2023

Sigo con el modelo no supervisado. Comienzo realizando los cambios hablados en la pasada reunión, para cada uno de los modelos que he ido planteando.

1_0_cpc_type1:

- ecgata: Señales copiadas
- ecglst: División de los ficheros correcta
- net:
 - CPC, entender dimensiones
 - 4 capas convolucionales en el encoder (stride_total = 16)
 - Longitud final secuencia = 94 (inicial 1500)
 - RF = 79
- config_cpc: 32 trabajadores y dimensión de salida igual a n (8)
- gen_mod: límites de amplitud y loader probado
- run1_train_cpc.py ejecutado

1_1_cpc_type1:

- ecgdata: Señales copiadas
- ecglst: División de los ficheros correcta
- net:
CPC. método generar_distractores y método loss.
5 capas convolucionales en el encoder (stride_total = 32)
Longitud final secuencia = 47 (inicial 1500)
RF = 159
- config_cpc: 32 trabajadores
- gen_mod: límites de amplitud y loader probado
- run1_train_cpc.py ejecutado

1_2_cpc_type1:

- ecgdata: Señales copiadas
- ecglst: División de los ficheros correcta
- net:
CPC. método generar_distractores y método loss.
6 capas convolucionales en el encoder (stride_total = 64)
Longitud final secuencia = 24 (inicial 1500)
RF = 319
- config_cpc: 32 trabajadores
- gen_mod: límites de amplitud y loader probado
- run1_train_cpc.py ejecutado

Tras realizar los cambios que sí he sabido hacer, me he puesto a tratar de entender la arquitectura CPC. He leído el TFM que me pasó Antonio, no me queda muy claro como generar los distractores y la muestra positiva.

2 de julio de 2023

Leo entero el TFM e intento aprovechar la información de este link:
https://github.com/loeweX/Greedy_InfoMax/blob/master/GreedyInfoMax/audio/models/loss_InfoNCE.py

Por lo que he podido deducir, me faltaría hacer bien el método que me permite obtener los distractores o muestras negativas. Por otro lado, también debería realizar cambios en el script de entrenamiento.

DUDAS:

- **net_cpc.py**
 - encoder: termino con batchnorm y relu,
 - bloque AR: cuantas capas en la gru (solo tengo una, si pongo más me da error)
 - función para generar los distractores, no sé bien cómo hacerlo

- método loss y función ctc_loss, creo que debido a como genero los distractores luego no me cuadran las dimensiones
- **config_cpc.json**
 - número de trabajadores = 32 o mas
 - tamaño del lote = 256 o menos, como hay tantas señales....
 - dimensión de salida = 8
 - dimensiones ocultas, lo dejo en 1000 de momento
 - Añado el parámetro de número de distractores
- **gen_mod.py**: estoy discriminando bien en función de la amplitud pico pico del QRS¿¿¿¿
- **run1_train_cpc.py**: debo hacer cambios en relación con la función los de mi red, no¿¿¿

3 de julio de 2023

He respondido el correo de los profesores sobre lo último que les envié y también les he pedido tutoría porque tengo bastantes dudas en el modelo no supervisado.

Resumen del estado actual del TFG:

- ✓ **1. PROCESADO SEÑALES. HECHO.**
- **2. TAREAS DE CONTROL.** Para la primera tarea de control, añadido simplemente el cálculo de la exactitud (también en la tabla)
- **3. MODELO NO SUPERVISADO.** Avanzar gracias a la tutoría.
- **4. REDACCIÓN TFG.** Hay que corregir la estructura y la introducción (~~eliminar dos partes, reordenar para que queden bien las figuras~~ y corregir 1.4 y 1.5). Además, ya puedo redactar los capítulos 2 y 3.

Actualizo la estructura del TFG con los cambio de Julia, también he arreglado parcialmente la introducción.

Por la tarde sigo con la primera tarea de control y el modelo no supervisado.

4 de julio de 2023

Añado simplemente la exactitud a la primera tarea de control. NO LO HE PODIDO HACER PORQUE IBA SUPER LENTO AL HABER TANTA GENTE TRABAJANDO EN EL CLUSTER, LO DEJO PARA OTRO MOMENTO.

Reunión:

- ✓ **Tarea de control 1. Clasificación 12 derivaciones del ECG.**
Añadir el cálculo de la precisión en train, dev y test, así como a la tabla de comparación entre modelos. Repetir resultados y mandarlo a los profesores.
- **Modelo no supervisado.**
NO DEBÍA AÑADIR NADA MÁS.
 - ✓ Arreglar **net_cpc.py** y entender las **dimensiones**.
 - a. Codificador no lineal sin terminar en batchnorm y relu, última dimensión es la dimensión oculta, no la dimensión de salida. En consecuencia, hay que modificar el resto de dimensiones

- b. En la GRU de momento dejamos una capa
- c. Función generar distractores no hacía falta
- d. Dejo el método loss como estaba
- ✓ Arreglar fichero de configuración
 - e. nb_samples_epoch = 10000 (para tener feedback más rápido)
 - f. nb_workers = 12
 - g. batch_size = 256 bien (podría subir más pero no podemos saber a priori hasta cuanto)
 - h. output_dim = 1
 - i. hidden_dim = 1024
- 2. Arreglar generador, no estaba discriminando bien y añadir lo del **índice** que ha dicho Antonio. Sí que lo uso tanto en test como en train.
- 3. EJECUTAR PROBANDO DISTINTAS CAPAS EN EL ENCODER, DISTINTAS DIMENSIONES OCULTAS Y DISTINTO NÚMERO DE PREDICCIONES**

- ✓ **Corrección capítulo 1.**
- **Redacción capítulo 2 y 3.**

Después de la reunión he arreglado la tarea de control 1.

5 de julio de 2023

Les envío a los profesores la tarea de control 1 tras añadir el cálculo de la precisión para los distintos modelos planteados.

Paso a arreglar las partes de código del modelo no supervisado.

1_0_cpc_type1:

- ecgdata: Señales copiadas
- ecglst: División de los ficheros correcta
- net:
 - CPC, entender dimensiones
 - 4 capas convolucionales en el encoder (stride_total = 16)
 - Longitud final secuencia = 94 (inicial 1500)
 - RF = 79
- config_cpc: 12 trabajadores, 10000 muestras por epoch, lote de 256 señales, 1 dimensión de salida y 1024 dimensiones ocultas
- gen_cpc: límites de amplitud y loader probado (falta lo del índice para cubrirnos las espaldas)
- run1_train_cpc.py arreglado y **ejecutado** (se me quedaba en la cola porque el fichero de configuración no era el correcto).

1_1_cpc_type1:

- ecgdata: Señales copiadas
- ecglst: División de los ficheros correcta
- net:

CPC, entender dimensiones

5 capas convolucionales en el encoder (stride_total = 32)

Longitud final secuencia = 47 (inicial 1500)

RF = 159

- config_cpc: 12 trabajadores, 10000 muestras por epoch, lote de 256 señales, 1 dimensión de salida y 1024 dimensiones ocultas
- gen_cpc: límites de amplitud y loader probado (falta lo del índice para cubrirnos las espaldas)
- run1_train_cpc.py arreglado y **ejecutado** (se me quedaba en la cola porque el fichero de configuración no era el correcto).

1_2_cpc_type1:

- ecgdata: Señales copiadas
- ecglst: División de los ficheros correcta
- net:
CPC, entender dimensiones **ME DA ERROR**
6 capas convolucionales en el encoder (stride_total = 64)
Longitud final secuencia = 24 (inicial 1500)
RF = 319
- config_cpc: 12 trabajadores, 10000 muestras por epoch, lote de 256 señales, 1 dimensión de salida y 1024 dimensiones ocultas
- gen_cpc: límites de amplitud y loader probado (falta lo del índice para cubrirnos las espaldas)
- run1_train_cpc.py arreglado y **ejecutado** (se me quedaba en la cola porque el fichero de configuración no era el correcto).

Les he enviado un correo a los profesores con los resultados del primer modelo, creo que son bastante malos. Antonio ha dicho que antes de probar un entrenamiento largo, con 2000 o 3000 epochs, pruebo con 500-1000 epochs y más diezmado. El problema es que los modelos 1_1 y 1_2 no se me ejecutan directamente.

Paso a la parte de redacción, a corregir la introducción. No lo he terminado.

Resumen del estado actual del TFG:

- ✓ **1. PROCESADO SEÑALES.** HECHO.
- **2. TAREAS DE CONTROL.** HECHO.
- **3. MODELO NO SUPERVISADO.** En proceso.
- **4. REDACCIÓN TFG.** Hay que corregir la estructura y la introducción (eliminar dos partes, reordenar para que queden bien las figuras y corregir 1.4 y 1.5). Además, ya puedo redactar los capítulos 2 y 3.

6 de julio de 2023

Corrijo la introducción y se lo envío a los profesores junto con las dudas del modelo no supervisado. Hago los cambios indicados por los profesores y vuelvo a lanzar los tres modelos planteados.

Paso a redactar el capítulo 2 y 3. He redactado el capítulo 2, el 3 aún no lo he terminado.

7 de julio de 2023

Resumen del estado actual del TFG:

- ✓ **1. PROCESADO SEÑALES.** HECHO.
- **2. TAREAS DE CONTROL.** HECHO.
- **3. MODELO NO SUPERVISADO.** En proceso, nuevos parámetros.
- **4. REDACCIÓN TFG.** Hay que corregir la estructura y la introducción (eliminar dos partes, reordenar para que queden bien las figuras y corregir 1.4 y 1.5). Ya puedo redactar los capítulos 2 y 3.

Comienzo mirando los resultados de los 3 modelos (v1) lanzados ayer, parecen mejores, pero aun así la exactitud es muy baja, llega hasta 0.003 y pico en los 3 casos. Voy a lanzarlos con un n menor, 4 en lugar de 8. He lanzado 1_2_cpc_type1_v2, el resto de versiones 2 (con n=4) aun no las puedo lanzar porque no tengo todas las señales copiadas en las carpetas ecgdata correspondientes. Le envío los resultados que tengo de momento a Antonio.

Paso a redactar el capítulo 3. No he terminado, me falta bastante.

8 de julio de 2023

Ya se han copiado todas las señales a las carpetas ecgdata de los modelos 1_0_cpc_type1_v2 y 1_1_cpc_type1_v2. Los lanzo con n=4.

Paso a terminar la primera parte del capítulo 3, el pre-procesado de señales. Creo el índice de figuras y tablas del capítulo 4. Se lo envío todo por correo a Julia para que lo vaya mirando, la introducción corregida, el capítulo 2, la primera parte del capítulo 3 y el índice.

Antonio ha respondido, subo el número de muestras por epoch a 50000 (en las versiones 1, de momento) y lo vuelvo a lanzar.

Sigo escribiendo el capítulo 3.

9 de julio de 2023

Respondo al correo de Julia. Las tareas que lancé ayer aún no han terminado así que no les puedo mandar los resultados.

Lanzó los modelos, segundas versiones, pero con 50000 muestras por epoch.

Sigo con la escritura del capítulo 3.

10 de julio de 2023

Los modelos aún no han terminado, creo que les queda bastante. De momento algunos llevan más de dos días.

Sigo con la redacción del capítulo 3.

11 de julio de 2023

Los modelos aún no han terminado, sigo con la redacción del capítulo 3.

12 de julio de 2023

Los modelos aún no han terminado, sigo con la redacción del capítulo 3.

13 de julio de 2023

Los modelos aún no han terminado, sigo con la redacción del capítulo 3.

Resumen del estado actual del TFG:

- ✓ **1. PROCESADO SEÑALES. HECHO.**
- **2. TAREAS DE CONTROL.**
- **3. MODELO NO SUPERVISADO.** En proceso, nuevos parámetros.
- **4. REDACCIÓN TFG.** Hay que corregir la estructura y la introducción (eliminar dos partes, reordenar para que queden bien las figuras y corregir 1.4 y 1.5). Ya puedo redactar los capítulos 2 y 3 (falta explicar los modelos de la tarea no supervisada).

Como lo de la tarea no supervisada está yendo bastante lento y no queda mucho tiempo, he decidido que todavía no voy a añadir más explicaciones ni a hacer esquemas de los modelos planteados.

Reviso lo que he escrito estos días y les envío un correo a los profesores:

- **Redacción TFG.** Les explico lo que he hecho. Sobre la parte de resultados, pongo solo los modelos que han obtenido el mejor resultado y el resto los mando al anexo. QUIEREN QUE HAGA LAS GRÁFICAS DE PEDICCIÓN EN TEST REPITIÉNDO LA EJECUCIÓN DE LAS TAREAS NO SUPERVISADAS.
- **Modelo no supervisado.** Les comento como va y les traslado mi preocupación.
- **Les aviso de los viajes que tengo al final de julio y en agosto.**

14 de julio de 2023

Resumen del estado actual del TFG:

- **2. TAREAS DE CONTROL.** Las repito para conseguir las gráficas con la evolución de la precisión en test.
- **3. MODELO NO SUPERVISADO.** Nuevas indicaciones de Antonio para probar el proceso por completo.
- **4. REDACCIÓN TFG.**
 - Capítulo 3. Tarea no supervisada. Falta explicar las señales que se utilizan y cada uno de los modelos implementados, así como hacer con canva los esquemas de

los mimos. Si al final no pongo las gráficas con la precisión en test tengo que eliminarla de las explicaciones.

- Ya puedo empezar con la redacción del capítulo 4 (resultados).

Las ejecuciones de los modelos no supervisados todavía no han terminado, así que me pongo con otras cosas.

Empiezo repitiendo las tareas de control y redactando los resultados en el capítulo 4 y los anexos. Le he preguntado a Antonio porque las gráficas no me salían bien y no sé si merece la pena perder más tiempo en eso. NO SACAMOS ESAS GRÁFICAS.

Redacto pues el capítulo 4 de resultados con las representaciones que ya obtuve.

15 de julio de 2023

Sigo añadiendo cosas al capítulo 3. Comienzo con la redacción del capítulo 4.

16 de julio de 2023

Resumen del estado actual del TFG:

- **3. MODELO NO SUPERVISADO.** Nuevas indicaciones de Antonio para probar el proceso por completo.
- **4. REDACCIÓN TFG.**
 - **Capítulo 3. Tarea no supervisada.** ~~Falta explicar las señales que se utilizan~~ y cada uno de los modelos implementados, así como hacer con canva los esquemas de los mismos. Si al final no pongo las gráficas con la precisión en test tengo que eliminarla de las explicaciones.
 - **Capítulo 4. Resultados + Anexos.**

Termino de redactar los anexos del capítulo 4, hasta lo que tengo hecho (tareas de referencia).

17 de julio de 2023

Corrijo y termino la redacción del capítulo 4 hasta lo que tengo hecho. Lo leo todo y se lo envío a los profesores.

JULIA ME HA ENVÍADO SU CORRECCIÓN.

Resumen del estado actual del TFG:

- **3. MODELO NO SUPERVISADO.** Nuevas indicaciones de Antonio para probar el proceso por completo. ¿QUÉ COMBINACIONES HAGO? PREGUNTAR
- **4. REDACCIÓN TFG.**
 - **Capítulo 1. Introducción:** Correcciones Julia.
 - **Capítulo 2. Materiales:** Correcciones Julia.
 - **Capítulo 3. Métodos. Tarea no supervisada.** ~~Falta explicar las señales que se utilizan~~ y cada uno de los modelos implementados, así como hacer con canva

los esquemas de los mismos. Si al final no pongo las gráficas con la precisión en test tengo que eliminarla de las explicaciones (tengo que quitar esto sí).
Correcciones Julia.

- **Capítulo 4. Resultados + Anexos.** Faltan los resultados de los modelos de las tareas no supervisadas y los resultados conjuntos.

Paso a mirar las tareas no supervisadas, el proceso completo.

Me surgen dudas con lo de congelar la red no supervisada. He añadido el bucle que ha pasado Antonio en el bucle principal de train y el resto lo he dejado igual (fuera del bucle, es decir, que si que optimizo los gradientes del modelo de referencia). También tengo la duda de si debo poner el modelo no supervisado que cargo en evaluación, ya que no lo voy a entrenar más.

Me sale error al cargar el modelo, le he explicado por correo a Antonio todo lo que he hecho.

YA FUNCIONA

18 de julio de 2023

Sigo mirando la tarea semi-supervisada. Ya han terminado el resto de modelos. Paso a probar combinaciones.

Probaré el mejor modelo de cada tarea con el mejor modelo no supervisado (para k 8 y k 4) y pasándole a la red como entrada tanto z como c. pruebo además distinto número de capas convolucionales, hago un barrido.

Hoy dejo hecha la tarea 1.

19 de julio de 2023

Resumen del estado actual del TFG:

- ✓ **4. TAREA SEMI-SUPERVIDADA.** Terminar de obtener representaciones y enviarlas a los profesores
- **5. REDACCIÓN TFG.**
 - **Capítulo 1. Introducción:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras.
 - **Capítulo 2. Materiales:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las tablas.
 - **Capítulo 3. Métodos. Tarea no supervisada.** Falta explicar cada uno de los modelos implementados, así como hacer con canva los esquemas de los mismos.
Correcciones Julia (Si al final no pongo las gráficas con la precisión en test tengo que eliminarla de las explicaciones (tengo que quitar esto sí)). Arreglar formato de los títulos y poner títulos a las figuras y tablas.
 - **Capítulo 4. Resultados + Anexos.** Faltan los resultados de los modelos de las tareas no supervisadas y los resultados conjuntos. Arreglar formato de los títulos y poner títulos a las figuras y tablas.

Hoy termino de obtener todas las representaciones de la tarea semi-supervisada y se lo envío a los profesores.

Antonio me propone lanzar nuevos modelos no supervisados durante estos días, con más n , más capas y más canales. En los de tipo 1_3 pruebo con 7 capas y 1024 canales. Hago dos versiones una con n igual a 12 y otra con n igual a 16. En los de tipo 1_4 pruebo con 8 capas y 1024 canales, hago dos versiones una con n igual a 12 y otra con n igual a 16. **También me propone realizar un entrenamiento más específico, quedándome solo con la derivación I, que es la de la segunda tarea de control.**

20 de julio de 2023

Resumen del estado actual del TFG:

- **TAREA NO SUPERVISADA.** Hago pruebas para ir viendo si subiendo k , el número de capas o el número de canales mejora la cosa. Los lanzo con pocos epochs. El que merezca la pena lo lanzo durante 2000-3000 epochs (tardará una semana o más), así se ejecuta mientras estoy fuera.
- **TAREA SEMI-SUPERVISADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Capítulo 1. Introducción:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras.
 - **Capítulo 2. Materiales:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las tablas.
 - **Capítulo 3. Métodos. Tarea no supervisada.** Falta explicar cada uno de los modelos implementados, así como hacer con canva los esquemas de los mismos.
Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras y tablas.
 - **Capítulo 4. Resultados + Anexos.** Faltan los resultados de los modelos de las tareas no supervisadas y los resultados conjuntos.
Arreglar formato de los títulos y poner títulos a las figuras y tablas.

Parto de los modelos no supervisados que había elegido como mejores, estos tienen 6 capas, 512 canales, $k=8$ o 4.

1_2_cpc_type1_v3/v4 -> aumento k , pruebo con 12 y con 16.

1_3_cpc_type1_v1/v2 -> aumento el número de capas, pruebo con 7 capas convolucionales (mantengo $k=8/4$ y 512 canales). De momento parece que más capas sí que le va bien.

1_4_cpc_type1_v1/v2 -> aumento el número de canales, pruebo con 1024 (mantengo $k=8/4$ y 6 capas). Creo que más canales no le viene bien, avanza muy poco acc.

Ahora utilizo solamente la derivación I, filtro mis ficheros para quedarme solo con esa. Los he lanzado durante 300 epochs, mañana poder ver los resultados este fin de semana.

Mientras, me pongo a terminar de redactar lo que ya tengo, y si conseguimos mejoras las añadiré. Redacto el capítulo 3 y el capítulo 4.

24 de julio de 2023

Resumen del estado actual del TFG:

- **TAREA NO SUPERVISADA.** Hago pruebas para ir viendo si subiendo k , el número de capas o el número de canales mejora la cosa. Los lanzo con pocos epochs. El que merezca la pena lo lanzo durante 2000-3000 epochs (tardará una semana o más), así se ejecuta mientras estoy fuera.
- **TAREA SEMI-SUPERVISADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Capítulo 1. Introducción:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras.
 - **Capítulo 2. Materiales:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las tablas.
 - **Capítulo 3. Métodos. Tarea no supervisada.** Falta explicar cada uno de los modelos implementados, así como hacer con canva los esquemas de los mismos.
Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras y tablas.
 - **Capítulo 4. Resultados + Anexos.** Faltan los resultados de los modelos de las tareas no supervisadas y los resultados conjuntos.

Arreglar formato de los títulos y poner títulos a las figuras y tablas.

Reflexiono sobre los nuevos modelos no supervisados que lancé en su día.

La primera prueba que hice fue lanzar el mejor modelo de los primeros que probé, pero aumentando en número de predicciones, $k=12$ y $k=16$. Creo que esta estrategia no le viene bien, además, en el TFM que tengo de referencia sucedía esto mismo, aunque con señales de audio. Por tanto, yo descartaría subir k y lo dejaría en 4 predicciones futuras.

La siguiente prueba que hice fue subir el número de capas convolucionales que se emplean en el codificador no lineal. Había probado con 4, 5 y 6, esta vez probé con 7 (y $k=4$). Durante alguna de las iteraciones se llega a alcanzar una precisión en test de 0.0041, cosa que no había pasado con ninguno de los modelos anteriores.

La otra prueba que hice fue aumentar el número de canales, pasando de 512 a 1024. Los lancé con $k=8$ y 4. Los resultados obtenidos al realizar este cambio son los peores. Descartaría subir el número de canales, lo dejaría en 512.

- Les he enviado un correo a los profesores con los resultados y explicaciones.
- **TENDRIA QUE HABERLOS PROBADO CON LA SEGUNDA TAREA DE REFERENCIA Y A PARTIR DE ESO ELEGIR.** El que mejores resultados obtiene al probarlos con 5 capas es aquel que tiene 1024 canales y $k=8$.

Lanzó, durante 3000 iteraciones, un modelo que tiene 6 capas convolucionales, 1024 canales y hago varias versiones donde varío k. Pruebo con 8, 12 y 16. LOS ENTRENO SOLAMENTE CON LA DERIVACIÓN I.

Paso a mirar la redacción del capítulo 3 y del capítulo 4. Añado todo a la corrección de Julia, arreglo la bibliografía, y lo envío a los profesores.

31 de julio de 2023

Resumen del estado actual del TFG:

- **TAREA SEMI-SUPERVIDADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Capítulo 1. Introducción:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras.
 - **Capítulo 2. Materiales:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las tablas.
 - **Capítulo 3. Métodos. Tarea no supervisada.** Falta explicar cada uno de los modelos implementados, así como hacer con canva los esquemas de los mismos.
Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras y tablas.
 - **Capítulo 4. Resultados + Anexos.** Faltan los resultados de los modelos de las tareas no supervisadas y los resultados conjuntos.
Arreglar formato de los títulos y poner títulos a las figuras y tablas.

Comienzo con las correcciones de la redacción de TFG, me centro en la introducción.

1 de agosto de 2023

Compruebo como van los modelos no supervisados y sigo con la corrección de la introducción.

2 de agosto de 2023

No puedo acceder al cluster, debe ser un problema suyo. Termino la corrección de la introducción. Paso a la corrección del capítulo 2. Comienzo con la corrección del capítulo 3.

3 de agosto de 2023

Sigo sin poder entrar al cluster, ya le he preguntado a Antonio, me ha dicho que no le funciona a nadie, no sabe cuándo lo arreglarán.

Sigo corrigiendo.

Resumen del estado actual del TFG:

- **TAREA SEMI-SUPERVIDADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - ✓ **Capítulo 1. Introducción:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras.
 - ✓ **Capítulo 2. Materiales:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las tablas.
 - **Capítulo 3. Métodos. Tarea no supervisada.** Falta explicar cada uno de los modelos implementados, así como hacer con canva los esquemas de los mismos.
Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras y tablas.
 - **Capítulo 4. Resultados + Anexos.** Faltan los resultados de los modelos de las tareas no supervisadas y los resultados conjuntos.
Arreglar formato de los títulos y poner títulos a las figuras y tablas.

4 de agosto de 2023

Resumen del estado actual del TFG:

- **TAREA SEMI-SUPERVIDADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - ✓ **Capítulo 1. Introducción:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras.
 - ✓ **Capítulo 2. Materiales:** Correcciones Julia. Arreglar formato de los títulos y poner títulos a las tablas.
 - **Capítulo 3. Métodos. Tarea no supervisada.** Falta explicar cada uno de los modelos implementados, así como hacer con canva los esquemas de los mismos.
Correcciones Julia. Arreglar formato de los títulos y poner títulos a las figuras y tablas.
 - **Capítulo 4. Resultados + Anexos.** Faltan los resultados de los modelos de las tareas no supervisadas y los resultados conjuntos.
Arreglar formato de los títulos y poner títulos a las figuras y tablas.

7 de agosto de 2023

Termino de corregir el capítulo 4 y los anexos.

8 de agosto de 2023

Hago el resumen. Leo todo entero, reviso las correcciones y se lo envío a los profesores.

Resumen del estado actual del TFG:

- **TAREA SEMI-SUPERVIDADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Resumen**
 - ✓ **Capítulo 1. Introducción**
 - ✓ **Capítulo 2. Materiales**
 - **Capítulo 3. Métodos. Tarea no supervisada:** faltan los resultados de los modelos que todavía están ejecutándose (aunque ahora no porque no va el cluster) y explicar algo más.
 - **Capítulo 4. Resultados + Anexos:** faltaría poner la tabla con los tiempos promedio de ejecución que mencionó Antonio, así como hacer el cronograma y ver qué información subo a GitHub para poner el correspondiente link.
 - **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link), pero es que no tengo todos los resultados todavía.

14 de agosto de 2023

El cluster sigue sin funcionar. De momento solamente tengo las correcciones de Julia, las hago. Antonio no ha respondido.

Resumen del estado actual del TFG:

- **TAREA SEMI-SUPERVIDADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Resumen**
 - ✓ **Capítulo 1. Introducción**
 - ✓ **Capítulo 2. Materiales**
 - **Capítulo 3. Métodos. Tarea no supervisada:** faltan los resultados de los modelos que todavía están ejecutándose (aunque ahora no porque no va el cluster) y explicar algo más.
 - **Capítulo 4. Resultados + Anexos:** faltaría poner la tabla con los tiempos promedio de ejecución que mencionó Antonio, así como hacer el cronograma y ver qué información subo a GitHub para poner el correspondiente link.
 - **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link), pero es que no tengo todos los resultados todavía.
- **6. SUBIR CÓDIGO A GITHUB** (si es que se arregla el cluster)

15 de agosto de 2023

El cluster sigue sin funcionar. Paso a mirar las correcciones de Julia para el capítulo 3.

16 de agosto de 2023

El cluster ya funciona, pero no me salen los procesos que tenía ejecutándose. Le he preguntado a Antonio.

Sigo mirando las correcciones.

17 de agosto de 2023

Lanzo los 3 modelos no supervisados que tenía corriendo:

- 1_4_cpc_type1_v1 iba por la epoch 2029
- 1_4_cpc_type1_v2 iba por la epoch 2026
- 1_4_cpc_type1_v3 iba por la epoch 2029

Me han empezado todos en la epoch 2025.

18 de agosto de 2023

Reviso los modelos no supervisados que lancé ayer. Han ido avanzando bien, les quedan unos 3 días.

Paso a corregir el capítulo 3 y el capítulo 4. En los anexos he ido haciendo la tabla que me faltaba, aunque aún no la puedo completar del todo, y he hecho el cronograma.

Resumen del estado actual del TFG:

- **TAREA SEMI-SUPERVIDADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Resumen**
 - **Capítulo 1. Introducción**
 - **Capítulo 2. Materiales**
 - **Capítulo 3. Métodos. Tarea no supervisada:** faltan los resultados de los modelos que todavía están ejecutándose (aunque ahora no porque no va el cluster) y explicar algo más.
 - **Capítulo 4. Resultados + Anexos:** faltaría poner la tabla con los tiempos promedio de ejecución que mencionó Antonio, así como hacer el cronograma y ver qué información subo a GitHub para poner el correspondiente link.
 - **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link)
- **6. SUBIR CÓDIGO A GITHUB**

19 de agosto de 2023

He revisado los modelos no supervisados, siguen avanzando bien. Creo que terminarán mañana, no sé cuándo los podré probar. Paso a hacer el anexo III (aunque no lo puedo dejar terminado).

Resumen del estado actual del TFG:

- **TAREA SEMI-SUPERVIDADA.** Pruebo los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Resumen**
 - **Capítulo 1. Introducción**
 - **Capítulo 2. Materiales**
 - **Capítulo 3. Métodos. Tarea no supervisada:** faltan los resultados de los modelos que todavía están ejecutándose (aunque ahora no porque no va el cluster) y explicar algo más.
 - **Capítulo 4. Resultados + Anexos:** faltaría poner la tabla con los tiempos promedio de ejecución que mencionó Antonio, así como hacer el cronograma y ver qué información subo a GitHub para poner el correspondiente link.
 - **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link)
- **6. SUBIR CÓDIGO A GITHUB**

20 de agosto de 2023

Reviso los modelos no supervisados, van bien, mañana habrán terminado. Paso a corregir el resumen.

Resumen del estado actual del TFG:

- **4. TAREA SEMI-SUPERVIDADA.** Tengo que probar los nuevos modelos no supervisados en la tarea de control 2, para ver si ahora sí que conseguimos mejoras, ya que con los que teníamos hasta ahora no las habíamos conseguido.
- **5. REDACCIÓN TFG.**
 - **Resumen**
 - **Capítulo 4. Resultados + Anexos:** tengo que añadir los resultados que faltan de los modelos que todavía están ejecutándose y dar alguna explicación más. Una vez termine la tarea semi-supervisada de la segunda tarea de control también debo añadir esos resultados y las correspondientes explicaciones y gráficas. En los anexos faltaría poner la tabla con los tiempos promedio de ejecución que mencionó Antonio y ver (y preguntar) qué información y cómo subo a GitHub para poner el correspondiente link.
 - **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link)

22 de agosto de 2023

Sigo con la corrección del resumen.

23 de agosto de 2023

Los modelos no supervisados ya terminaron el lunes. Apunto los resultados en la memoria (capítulo 4) y añado alguna explicación.

Paso a probar estos nuevos modelos en la tarea semi-supervisada de diagnóstico de FAP. Añado los resultados y explicaciones correspondientes a la memoria.

Termino la tabla con los tiempos que mandó Antonio, no sé dónde incluirla dentro de los anexos.

Resumen del estado actual del TFG:

- **5. REDACCIÓN TFG.**

- **Resumen**

- **Anexos:**

- En los anexos faltaría poner la tabla con los tiempos promedio de ejecución que mencionó Antonio y ver (y preguntar) qué información y cómo subo a GitHub para poner el correspondiente link. También debo mover, aunque no sé muy bien cómo organizarlo, las cosas que sugiere Antonio.

- **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link)

24 de agosto de 2023

Miro el resumen y lo dejo terminado.

25 de agosto de 2023

Hago las correcciones de Antonio (lo que pueda). Me quedan algunas dudas sobre qué mover a los anexos, se las mando en un correo a los profesores

Resumen del estado actual del TFG:

- **5. REDACCIÓN TFG.**

- **Capítulo 3.** Correcciones Antonio

- **Capítulo 4.** Correcciones Antonio

- **Anexos:**

- En los anexos faltaría poner la tabla con los tiempos promedio de ejecución que mencionó Antonio y ver (y preguntar) qué información y cómo subo a GitHub para poner el correspondiente link. También debo mover, aunque no sé muy bien cómo organizarlo, las cosas que sugiere Antonio.

- **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link)

26 de agosto de 2023

Termino de corregir el capítulo 3 y el capítulo 4. También he puesto la tabla con los tiempos en los anexos, en los apartados de los entrenamientos.

Resumen del estado actual del TFG:

- **5. REDACCIÓN TFG.**

- **Anexos:**

En los anexos faltaría ver (y preguntar) qué información y cómo subo a GitHub para poner el correspondiente link. También debo mover, aunque no sé muy bien cómo organizarlo, las cosas que sugiere Antonio.

- **Capítulo 5. Conclusiones:** falta hacerlo (y poner el link de GitHub)

28 de agosto de 2023

Redacto el capítulo 5 con las conclusiones y las líneas futuras. Por otro lado, voy subiendo cosas a GitHub, como la Excel con la tabla de las bbdd, el diario, etc. Comienzo también con la descarga del código desde VSCode.