

Trabajo Fin de Grado

Implementación y evaluación de un sistema de diagnóstico de patologías cardíacas basado en redes neuronales a partir de señales electrocardiográficas

Implementation and evaluation of a neural network based system to diagnose cardiac pathologies from ECG signals

Autor

Daniel Tosaus Lanza

Directores

Juan Pablo Martínez Cortés
Antonio Miguel Artiaga

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2021

Índice

Resumen	1
Índice de figuras	2
Índice de tablas	3
CAPÍTULO 1. INTRODUCCIÓN	4
1.1 Motivación.....	4
1.2 Fundamentos teóricos del corazón. Anatomía Básica y Sistema de Conducción. .	4
1.3 La señal ECG.....	6
1.3.1 Formación y partes del ECG	6
1.3.2 Registro del ECG. Tipos de derivaciones.....	7
1.3.3 El ECG como herramienta de diagnóstico	9
1.3.4 Degradaciones comunes en la señal ECG	10
1.4 Introducción al uso de redes neuronales.....	11
1.4.1 Estado del arte	11
1.4.2 Fundamentos de redes neuronales	11
1.5 Objetivos del trabajo.....	13
1.6 Organización de la memoria.....	13
CAPÍTULO 2. MATERIALES Y MÉTODOS	14
2.1 Datos de estudio. PhysioNet / Computing in Cardiology Challenge	14
2.2 Tratamiento del ECG para extracción de características.....	15
2.2.1 Lectura, remuestreo y filtrado de valores no fisiológicos del ECG.....	15
2.2.2 Primera delineación y eliminación de línea de base.....	16
2.2.3 Cálculo de índices globales de QRS.....	17
2.2.4 Segunda delineación y formación de señales sintéticas	19
2.2.5 Generación de imágenes	20
2.3 Análisis de las etiquetas.....	21
CAPÍTULO 3. DISEÑO DE LA RED NEURONAL Y ENTORNO DE ENTRENAMIENTO	23
3.1 Construcción de un DataLoader	23
3.2 Diseño de un modelo sencillo de red.....	25
3.3 Entorno de Entrenamiento.....	28
3.4 Primer entrenamiento del Modelo 0	29
3.5 Segundo entrenamiento del Modelo 0	29
3.6 Tercer entrenamiento del Modelo 0.....	29
3.7 Variaciones en el modelo	30
3.7.1 Modelo 1.....	31
3.7.2 Modelo 2.....	32
3.7.3 Modelo 3.....	33
3.7.4 Modelo 4.....	34
CAPÍTULO 4. RESULTADOS	36
4.1 Consideraciones iniciales	36
4.2 Entrenamiento del Modelo 0 con el Subset 1	36
4.3 Entrenamiento del Modelo 0 con el Subset 2	37
4.4 Entrenamiento del Modelo 0 con el Subset 3	38
4.5 Entrenamiento de los Modelos 1, 2, 3 y 4 con el Subset 3	39
CAPÍTULO 5. CONCLUSIONES Y LÍNEAS FUTURAS.....	40
5.1 Conclusiones.....	40
5.2 Líneas futuras	41
Bibliografía.....	42

Resumen

El Electrocardiograma es una de las herramientas más sencillas y utilizadas en el diagnóstico de patologías cardíacas. Esta es una prueba no invasiva que registra las señales eléctricas del miocardio mediante electrodos colocados en la superficie del torso y las extremidades, y permite observar algunos aspectos del funcionamiento del corazón.

Este trabajo de fin de grado aborda el diagnóstico automático de patologías cardíacas mediante el ECG y el uso de redes neuronales. Para ello se ha implementado y evaluado un sistema de procesado del ECG y un modelo de red neuronal de clasificación.

En primer lugar, nos hemos centrado en cómo extraer información del ECG y cómo representarla de forma apropiada para una red neuronal. En este trabajo se ha optado por una representación gráfica en forma de imagen. De este modo podremos usar estas imágenes para entrenar una Red Neuronal que sea capaz de dar un diagnóstico cuando se le presente un ECG procesado de esta forma. En segundo lugar, se han diseñado y utilizado redes neuronales convolucionales para la clasificación de los ECGs. Como datos, se han tomado señales electrocardiográficas de diferentes bases de datos internacionales etiquetadas por expertos según distintos tipos de patologías.

Y finalmente, se ha evaluado la efectividad del procesado en cuanto a la extracción de características del ECG en cada patología, así como la precisión del diagnóstico automático.

Como conclusión del trabajo, podemos decir que el procesado de los ECG ha conseguido extraer con éxito las características de las patologías escogidas, gracias a las cuales se ha podido llevar a cabo una clasificación satisfactoria de las patologías con una red de tipo convolucional.

Índice de figuras

<i>Figura 1.2.1: Estructura de conducción del corazón.....</i>	5
<i>Figura 1.3.1: Formación del ECG a partir de la propagación del potencial de acción.</i>	6
<i>Figura 1.3.2: Conjunto de ondas que conforman el ECG.....</i>	7
<i>Figura 1.3.2.1: Posición de los electrodos para captar las derivaciones frontales.</i>	8
<i>Figura 1.3.2.2: Posición de los electrodos para captar las derivaciones precordiales.</i>	8
<i>Figura 1.3.2.3: Proyecciones de la actividad cardíaca en las distintas derivaciones....</i>	9
<i>Figura 2.2.1.1: Derivaciones I, AVF y V5 originales.</i>	15
<i>Figura 2.2.2.1: Derivaciones I, AVF y V5 eliminada su línea de base</i>	17
<i>Figura 2.2.3.1: Ventanas de Hanning asociadas al primer QRS en las 12 derivaciones.</i>	18
<i>Figura 2.2.3.2: Coincidencias en los índices de QRS y umbral de decisión de índices globales.....</i>	18
<i>Figura 2.2.4.1: Derivación II de ECG real con las principales ondas delineadas automáticamente.....</i>	19
<i>Figura 2.2.4.2: Señales sintéticas de QRS, onda P y onda T solapadas.....</i>	20
<i>Figura 2.2.5.1: Imágenes resultantes del procesado de las 12 derivaciones y las 3 señales sintéticas.</i>	21
<i>Figura 2.3.1: Apariciones conjuntas de las distintas patologías a lo largo de todos los datasets.....</i>	22
<i>Figura 3.1.1: Diagrama de flujo del cargador de datos.....</i>	24
<i>Figura 3.2.1: Proceso de convolución con filtro al atravesar capa convolucional.....</i>	25
<i>Figura 3.2.2: Diagrama de flujo y capas del modelo base de la red.....</i>	26
<i>Figura 3.2.3: Diagrama de flujo y capas del Modelo 0 de la red.....</i>	28
<i>Figura 3.7.1.1: Diagrama de flujo y capas del Modelo 1 tanto sin stride (izquierda), como con stride (derecha).</i>	31
<i>Figura 3.7.2.1: Diagrama de flujo y capas del Modelo 2.....</i>	32
<i>Figura 3.7.3.1: Diagrama de flujo y capas del Modelo 3.....</i>	33
<i>Figura 3.7.4.1: Diagrama de flujo y capas del Modelo 4_16_1.5 (izquierda) y Modelo 4_16_2.5 (derecha).....</i>	34
<i>Figura 3.7.4.2: Diagrama de flujo y capas del Modelo 4_32_1.5 (izquierda) y Modelo 4_32_2.5 (derecha).....</i>	35
<i>Figura 4.2.1: Evolución del coste y la precisión en test del Modelo 0 durante el entrenamiento con el Subset 1</i>	36
<i>Figura 4.3.1: Evolución del coste y la precisión en test durante el entrenamiento del Modelo 0 con el Subset 2.....</i>	37
<i>Figura 4.4.1: Curvas ROC y AUC de las 4 patologías a clasificar por el Modelo 0</i>	38
<i>Figura 4.5.1: AUC obtenida frente a número de parámetros para cada versión del modelo</i>	39

Índice de tablas

<i>Tabla 2.1.1: Grabaciones de los distintos datasets utilizados en el trabajo.....</i>	<i>14</i>
<i>Tabla 3.5.1: Ficheros de entrenamiento para cada patología en el Subset 2.....</i>	<i>29</i>
<i>Tabla 3.6.1: Distribución de los ficheros de entrenamiento que conforman el Subset 3 por patología y dataset.....</i>	<i>29</i>

CAPÍTULO 1. INTRODUCCIÓN

1.1 Motivación

Las patologías cardíacas representan una de las principales causas de mortalidad en la población mundial, siendo en 2020 la primera causa de muerte en el mundo la cardiopatía isquémica [1]. Adicionalmente, estas afecciones habitualmente pueden causar otras condiciones mortales, haciendo que su impacto se refleje en millones de personas en todo el mundo. Por este motivo, un gran número de profesionales se encuentran dedicados a la prevención y tratamiento de estas patologías cardíacas; siendo un aspecto esencial para ello un diagnóstico temprano y preciso de la condición de cada paciente.

Este trabajo surge del interés personal del autor de relacionar e integrar conocimientos de procesado de señal con la resolución de problemas del ámbito médico y de las señales biomédicas. Dada la naturaleza del mismo, este agrupa aspectos de varias disciplinas como la Fisiología, las Señales Biomédicas y su procesado, y el diseño y manejo de Redes Neuronales. Este carácter interdisciplinar ha supuesto uno de los retos más importantes en la realización de este trabajo.

1.2 Fundamentos teóricos del corazón. Anatomía Básica y Sistema de Conducción.

El corazón es un órgano que actúa como una bomba, recibiendo sangre oxigenada y enviándola a las células del resto del cuerpo. Tiene dos tipos de cavidades, llamadas aurículas y ventrículos. Las aurículas, situadas en la parte superior del corazón se encargan de dejar entrar la sangre en el corazón y de hacer pasar esta sangre a los ventrículos. Posteriormente, los ventrículos la bombean al resto del cuerpo.

Todo este movimiento de la sangre entre aurículas, ventrículos, y después el resto del cuerpo, es posible gracias a la contracción de las fibras musculares que conforman el corazón, llamadas miocitos. En situación de reposo los miocitos presentan una diferencia de potencial negativa entre el interior y el exterior de la célula. Cuando estas reciben una estimulación eléctrica, entran iones positivos en la célula gracias a la apertura de un conjunto de canales iónicos. Este fenómeno se denomina despolarización y provoca que las células del miocardio se contraigan. Dicha despolarización se propaga como una onda entre células vecinas, cambiando su potencial de acción y provocando la contracción sincronizada del miocardio [2].

El conjunto de procesos ordenados que tienen lugar en el transcurso de un latido es el siguiente:

Primero se genera un impulso eléctrico de activación en la aurícula derecha en una región denominada nodo sino-auricular (SA), cuya frecuencia de activación está modulada por el sistema nervioso autónomo. Este impulso eléctrico se va propagando entre células musculares vecinas estimulando ambas aurículas y provocando su contracción. Esta contracción hace que la sangre pase de las aurículas a los ventrículos a través de las válvulas auriculoventriculares. Tras la contracción, las células de las aurículas se polarizan de nuevo volviendo a su estado de reposo y cerrándose las válvulas auriculoventriculares.

Sin embargo, la señal eléctrica no se propaga directamente a los ventrículos, debido a que entre aurículas y ventrículos existe un tejido aislante que hace que la propagación solo sea posible en una zona concreta. Esta zona es el nodo auriculo-ventricular (AV) y en él el impulso eléctrico se propaga lentamente, sufriendo un retardo de unos 0.1 segundos antes de pasar a los ventrículos. A continuación, el estímulo eléctrico se propaga por unas células especializadas para la conducción y que conforman un camino llamado Haz de His. El Haz de His se divide en dos ramas, izquierda y derecha, que terminan en las llamadas células de Purkinje, las cuales activan las células del miocardio y provocan la contracción simultánea de los ventrículos, bombeando la sangre hacia el exterior del corazón [2].

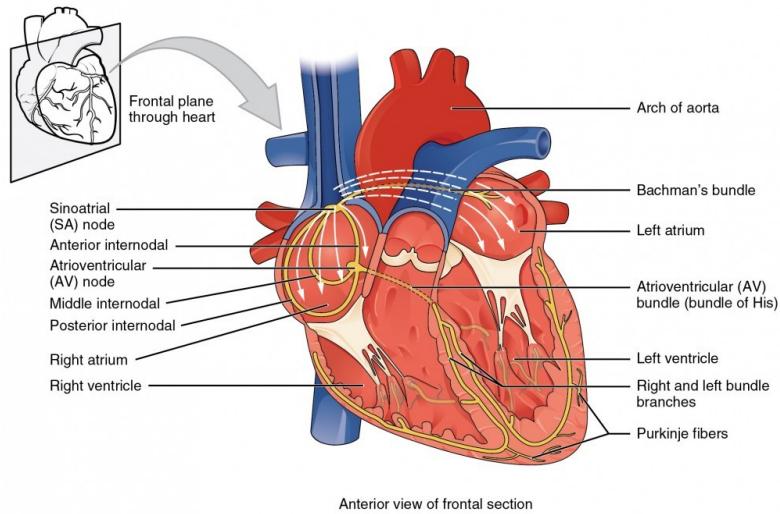


Figura 1.2.1: Estructura de conducción del corazón.

Imagen obtenida de: <https://courses.lumenlearning.com/suny-ap2/chapter/cardiac-muscle-and-electrical-activity/>

Posteriormente la relajación de aurículas y ventrículos también se debe a una corriente de repolarización, que se produce cuando el potencial de acción (la diferencia de potencial entre el interior y el exterior de la célula) vuelve a los valores de reposo (negativos). Tras un periodo refractario, las células son capaces de despolarizarse nuevamente, en un nuevo ciclo cardíaco.

Dado que el funcionamiento del corazón está guiado por la propagación de impulsos eléctricos que generan corrientes, la medición de esta actividad eléctrica nos permite comprender el funcionamiento del corazón, tanto en condiciones fisiológicas como en condiciones anómalas, es decir, en presencia de patologías cardíacas.

Para poder medir esta actividad de manera no invasiva, se colocan electrodos sobre el torso o las extremidades del paciente, de manera que cuando una onda de despolarización (contracción, el potencial de acción de células pasa a ser positivo) se acerca a un electrodo positivo se registra una deflexión positiva en el ECG. Con las señales de repolarización tendríamos el caso análogo, observando una deflexión negativa en el ECG [2].

1.3 La señal ECG

1.3.1 Formación y partes del ECG

En circunstancias no patológicas, la señal ECG durante un ciclo cardíaco completo (Figura 1.3.2) está compuesta por varias ondas. La llamada onda P registra el impulso eléctrico que provoca la contracción auricular y, por tanto, el paso de la sangre de las aurículas a los ventrículos. A la onda P le sigue un conjunto de ondas que, usualmente, se estudian conjuntamente. Este es el complejo QRS, compuesto por las ondas Q, R y S, que es reflejo de la propagación del impulso eléctrico que provoca la contracción de los ventrículos. Este complejo es la componente del ECG que presenta mayores frecuencias y energía, ya que la contracción ventricular debe ser rápida y fuerte, lo que facilita su identificación automática en la señal ECG.

Tras la contracción ventricular existe una pausa que queda representada en el ECG como una parte plana de la línea basal llamada segmento ST. Siguiendo a esta pausa aparece la onda T, que es la onda producida por la repolarización de las células de los ventrículos.

Cabe destacar aquí que las aurículas también se repolarizan, volviendo al potencial de reposo, y por tanto generan una onda eléctrica asociada a la repolarización de las células del miocardio. Sin embargo, esta onda de repolarización auricular suele coincidir en el tiempo con la contracción ventricular, por lo que en el ECG queda enmascarada por el complejo QRS.

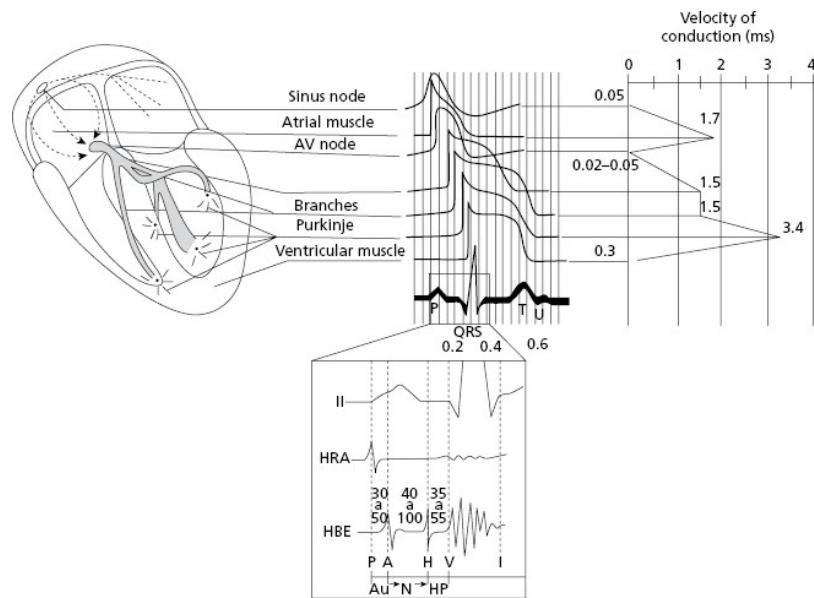


Figura 1.3.1: Formación del ECG a partir de la propagación del potencial de acción.

Imagen obtenida de: <https://thoracickey.com/2-the-ecg-curve-what-is-it-and-how-does-it-originate/>

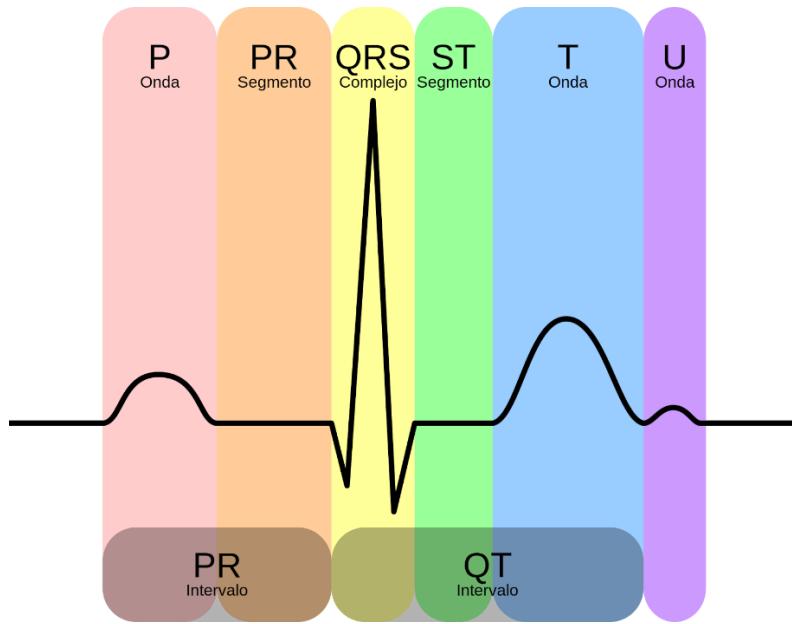


Figura 1.3.2: Conjunto de ondas que conforman el ECG.

Imagen obtenida de: <https://medicinageneraluniversal.blogspot.com/2017/01/que-es-un-electrocardiograma.html>

1.3.2 Registro del ECG. Tipos de derivaciones

Para captar la señal electrocardiográfica anteriormente descrita, se colocan electrodos sobre la piel del sujeto. La colocación de estos electrodos define distintas proyecciones de la misma actividad cardíaca que se denominan derivaciones.

El sistema de derivaciones más utilizado es el de “12 derivaciones estándar”, que es el que utilizan las señales usadas en ese trabajo. Como su nombre indica, se cuenta con 12 señales diferentes, obtenidas a partir de 12 combinaciones diferentes de 10 electrodos [2]. Estas derivaciones se pueden dividir en derivaciones frontales y derivaciones precordiales. En las derivaciones frontales se colocan electrodos en los brazos izquierdo y derecho y en la pierna izquierda, formando un triángulo (denominado triángulo de Einthoven) (Figura 1.3.2.1). Cada lado de este triángulo conforma una derivación (I, II, III), obteniéndose el ECG como la diferencia de potencial entre cada par de electrodos.

Otras derivaciones frontales son las AVR, AVL y AVF. En estas derivaciones se colocan electrodos en brazos y piernas y se calcula la diferencia de potencial entre uno de ellos (considerado de polaridad positiva) y un punto común calculado como el promedio de los tres electrodos. En la AVR es el brazo derecho, en la AVL el izquierdo, y en el AVF el pie izquierdo. La orientación de las derivaciones AVR, AVL y AVF corresponde a la dirección desde el centro del triángulo hasta el electrodo considerado positivo en cada caso. Al tener una orientación espacial diferente a las I, II y III, permiten observar la actividad cardíaca desde otras tres direcciones distintas.

Dicho de otro modo, cada derivación frontal toma un registro desde distinto ángulo, por lo que cada derivación puede verse como una proyección diferente de la misma actividad cardíaca.

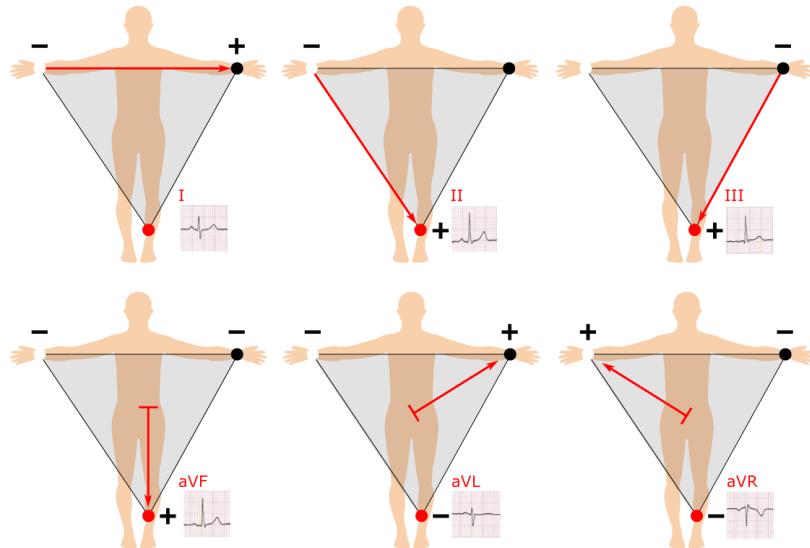


Figura 1.3.2.1: Posición de los electrodos para captar las derivaciones frontales.

Imagen obtenida de: <https://www.cablesandsensors.com/pages/12-lead-ecg-placement-guide-with-illustrations>

Por otro lado, para obtener las derivaciones precordiales se colocan electrodos positivos en 6 puntos distintos del tórax. Recordemos que el hecho de que estos electrodos sean positivos hará que cualquier onda de despolarización que avance hacia ellos provoque una deflexión positiva en el ECG. Se puede pensar en las derivaciones precordiales como proyecciones horizontales que se cruzan en el nódulo AV, considerando que la polaridad negativa de las mismas está en la parte posterior del sujeto. Las derivaciones precordiales son V1, V2, V3, V4, V5 y V6.

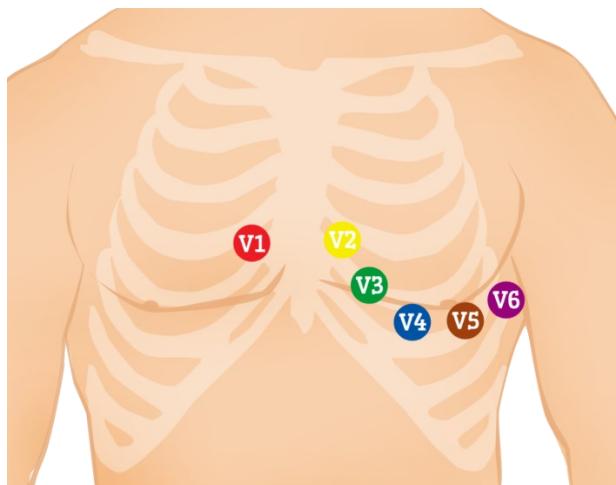


Figura 1.3.2.2: Posición de los electrodos para captar las derivaciones precordiales.

Imagen obtenida de: <https://www.cablesandsensors.com/pages/12-lead-ecg-placement-guide-with-illustrations>

Estas 12 derivaciones conforman 12 proyecciones simultáneas diferentes del mismo fenómeno cardíaco, por lo que entre distintas derivaciones tenemos tanto información complementaria como redundante.

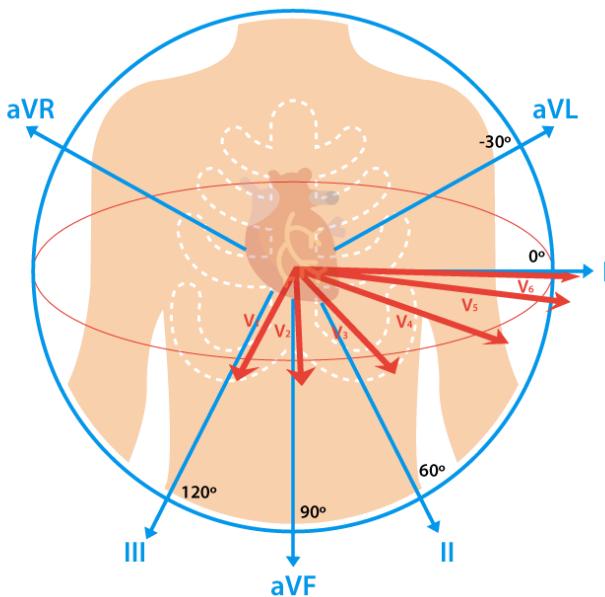


Figura 1.3.2.3: Proyecciones de la actividad cardíaca en las distintas derivaciones.

Imagen obtenida de: <https://www.cablesandsensors.com/pages/12-lead-ecg-placement-guide-with-illustrations>

1.3.3 El ECG como herramienta de diagnóstico

Debido a que el ECG recoge toda la actividad eléctrica que modela el funcionamiento del corazón, es posible detectar patologías en un paciente observando ciertas características o patrones en su ECG. Algunos aspectos que pueden indicar la presencia de patologías cardíacas son la amplitud, duración y forma de las ondas que conforman el ECG en las diferentes derivaciones, o el ritmo de aparición y la separación entre las mismas.

Cuando la forma de las ondas es normal pero su ritmo de aparición no lo es, pueden existir patologías relacionadas con el ritmo cardíaco como la taquicardia (ritmo superior al normal) o la bradicardia (ritmo inferior al normal)¹. Entre las patologías que tienen que ver con alteraciones del ritmo cardíaco también podemos encontrar la Arritmia sinusal, el marcapaso migratorio² o la fibrilación auricular.

Estas variaciones en el ritmo cardíaco pueden ser consecuencia de problemas de conducción en el mecanismo de contracción del corazón. Entre los problemas de conducción que puede sufrir un corazón podemos destacar desde alteraciones de la activación del nodo SA, hasta focos ectópicos de activación³, o bloqueos de las ramas de conducción de impulsos eléctricos. Estos problemas de conducción suelen manifestarse en el ECG alterando la forma de las ondas que lo componen. Algunas de las patologías más comunes relacionadas con problemas de conducción son los bloqueos, ya sean sinusales, auriculoventriculares, de rama, etc.

¹ Un ritmo cardíaco fisiológicamente normal está, en reposo, entre 60 y 100 latidos por minuto [2].

² El marcapaso migratorio es un ritmo variable debido al cambio del punto de activación de las células que inician el latido. En estos casos el proceso de contracción auricular es irregular, traduciéndose en cambios de forma de la onda P.

³ Un foco ectópico es una célula o conjunto de células que inician un impulso eléctrico independiente del nodo SA. Eso puede provocar la aparición de ondas anormales antes de lo esperado en el ECG.

1.3.4 Degradaciones comunes en la señal ECG

Debido al modo de captación de las señales eléctricas, el electrocardiograma suele estar afectado por varias fuentes de ruido e interferencias. Debido al movimiento ligero de los electrodos, por movimiento del paciente o simple respiración, se producen variaciones de la impedancia en la unión de los electrodos con la piel, provocando la aparición de una línea de base sobre la que está la señal del ECG. En condiciones normales y de reposo esta variación suele tener frecuencias menores a 1 Hz, aunque puede ser mayor en mediciones específicas como las pruebas de esfuerzo, o los registros Holter.

Por otro lado, puede existir ruido debido a la actividad de otros músculos cercanos, es decir, ruido electromiográfico (EMG), y su presencia en el ECG depende del movimiento y nivel de esfuerzo del sujeto (nivel de contracción muscular) y del equipo médico utilizado para su registro, que también puede provocar la aparición de ruido electrónico.

Otra interferencia común es la interferencia por acoplamiento eléctrico del equipo médico (PLI, Power-Line Interference noise). En este último caso, los cables que llevan las señales del ECG desde los electrodos hasta el equipo médico de grabación pueden sufrir interferencias electromagnéticas del entorno. Esto puede ser especialmente problemático debido a que las frecuencias de red (50 Hz - 60 Hz) están dentro de la banda de la señal ECG útil [3].

Por todo ello, uno de los primeros pasos del procesado del ECG que deberemos llevar a cabo es realizar filtrados con el fin de limpiar la señal al máximo de estos ruidos interferentes, y eliminar en lo posible la línea de base que presenten las señales del ECG [4].

1.4 Introducción al uso de redes neuronales

1.4.1 Estado del arte

La historia de las redes neuronales comienza en 1958 con la creación por parte de F. Rosenblatt del Perceptron [5]. Estas estructuras tienen entradas y salidas binarias, y por sí mismas, solo sirven para modelar decisiones binarias sencillas, o funciones lógicas con AND o OR.

Posteriormente, la composición de redes multicapa a partir de unidades básicas, (MLP, Multilayer Perceptron) mediante el algoritmo FeedForward (conectar salida de unas capas con entrada de otras) o el algoritmo de back-propagation [6] para entrenar de forma eficiente, permitieron desarrollar la rama del aprendizaje automático basada en redes neuronales, desarrollando la base teórica de las redes neuronales profundas actuales.

Unos años después Yann LeCun crearía la primera red neuronal convolucional (CNN, Convolutional Neural Network) [7], usándola para el reconocimiento de caracteres manuscritos, y sentando las bases de una de las familias de modelos con más éxito en la última década. Este tipo de redes son una herramienta muy potente para el tratamiento de datos en forma de imágenes y por ello ha sido la elección en cuanto a arquitectura para este trabajo.

En la actualidad estas técnicas han evolucionado, dando lugar a redes neuronales de gran complejidad y número de parámetros denominadas redes profundas (Deep Learning [8]); e incluso más recientemente a arquitecturas novedosas más allá de las redes convolucionales [9], algunas de ellas capaces de generar imágenes muy realistas casi indistinguibles de imágenes reales [10].

1.4.2 Fundamentos de redes neuronales

Una red neuronal pretende resolver un problema tomando valores como entrada y generando una salida a partir de estos. Los problemas más habituales suelen ser de regresión o de clasificación. En cualquier caso, la red debe ser capaz de procesar los datos de entrada y, mediante un diseño adecuado de sus dimensiones, generar una salida satisfactoria.

Para hacerlo, estas redes están formadas por diferentes capas, las cuales procesan los datos de salida de la anterior capa y sirven de entrada para la siguiente. Después de cada capa se aplica una función no lineal llamada función de activación. Esto permite que la red pueda modelar fenómenos más complejos y se puede conseguir usando una gran variedad de funciones de activación.

Las capas pueden procesar los datos de diversas formas: convolucional, lineal...; y cada una tiene unos parámetros internos cuyos valores se ajustarán durante el entrenamiento.

El entrenamiento se lleva a cabo introduciendo datos de ejemplo en la red y comparando su salida con la salida deseada mediante una función de coste que mide el grado de ajuste entre estos dos valores. Posteriormente, se aplica el algoritmo de back-propagation, que calcula de forma eficiente el gradiente de la función de coste, y nos indica como deberíamos modificar cada parámetro de la red para que el coste en la salida sea menor. Normalmente esto se hace de forma iterativa, mediante una actualización de los valores sucesiva, es decir, se realiza una optimización de tipo gradiente descendiente [6].

En las técnicas llamas SGD (Stochastic Gradient Descent) cada vez que realizamos esta operación de evaluación del coste, cálculo del gradiente y actualización de los parámetros, utilizamos un subconjunto de datos diferente. De esta manera, y entrenando con muchos ejemplos distintos, idealmente podemos hacer que los parámetros de las capas que componen la red converjan a los valores óptimos que hacen que la red en su conjunto modele el comportamiento deseado, es decir, que aprenda a generalizar incluso en datos no vistos en el proceso de ajuste o entrenamiento.

Teniendo en cuenta la gran cantidad de ejemplos que puede necesitar una red para entrenarse, una práctica habitual es utilizar para las actualizaciones de gradiente un conjunto pequeño (varias decenas) de ejemplos de manera simultánea. Este bloque de ejemplos se denomina batch (o lote) y es muy empleado para optimizar el entrenamiento y mejorar su estabilidad y velocidad de convergencia.

Las operaciones a realizar para calcular el gradiente y actualizar los pesos pueden verse como operaciones matriciales por lo que, computacionalmente, son tareas que se pueden parallelizar y realizar simultáneamente. Por este motivo se suelen usar GPUs durante el entrenamiento de redes neuronales, puesto que su arquitectura (enfocada al procesado de imágenes) está optimizada para el cálculo matricial en paralelo, permitiendo liberar ciclos de CPU para otras tareas.

Otros parámetros que se pueden elegir en el entrenamiento son el tipo de función de coste, que define como se calcula el parecido entre la salida real y la salida deseada; el optimizador, que gestiona como se actualizan los pesos y cuanto se avanza cada vez en la dirección del gradiente (learning rate, lr); y los ciclos de entrenamiento o epochs, que definen cuantas veces servimos a la red el conjunto completo de datos de entrenamiento.

1.5 Objetivos del trabajo

Con la realización de este trabajo se pretende estudiar la señal electrocardiográfica y su relación con distintas patologías del corazón, para después extraer en forma de imágenes las características propias de cada patología, de forma que una red neuronal sea capaz de aprender estas características y ofrecer una clasificación de patologías satisfactoria.

Para conseguir esto se ha perseguido completar las siguientes metas parciales:

- Diseñar un sistema de procesado de señales que genere imágenes que representen las características propias de un ECG concreto.
- Diseñar una red neuronal convolucional que tome como entrada estas imágenes y pueda clasificar los ECG según las patologías que presentan.
- Entrenar y validar esta red para distintas variaciones sobre un diseño básico.
- Estudiar posibles mejoras para el sistema y futuras aplicaciones.

1.6 Organización de la memoria

El planteamiento de la estructura de la memoria coincide con el flujo de los datos a lo largo del sistema que se ha diseñado. Por esto, tras el resumen y la introducción previa, se van a ir tratando los siguientes puntos.

En primer lugar, los aspectos relativos al procesado de las señales ECG se expondrán en el Capítulo 2. En él se explicarán los pasos seguidos para el remuestreo, filtrado y demás procesado requerido para extraer la información del ECG y plasmarla en imágenes.

Después, el Capítulo 3 describirá el diseño de la red neuronal y su entrenamiento. Aquí se explicará la metodología y las decisiones de diseño tomadas en este ámbito.

Seguidamente, el Capítulo 4 abordará la discusión los resultados obtenidos. Se tratarán las condiciones de las pruebas realizadas, así como su efecto en los resultados finales.

Y, por último, el Capítulo 5 cerrará la memoria con las conclusiones a extraer, así como posibles ampliaciones o líneas futuras de trabajo.

CAPÍTULO 2. MATERIALES Y MÉTODOS

2.1 Datos de estudio. PhysioNet / Computing in Cardiology Challenge

Las señales ECG sobre las que trabajaremos proceden de una serie de datasets internacionales, recopilados por la entidad PhysioNet para organizar el Computing in Cardiology Challenge 2020. En este concurso se abordaba un problema de clasificación de ECG por patologías, teniendo datos de ECG muy diferentes para el entrenamiento en cuanto a frecuencia de muestreo, duración, localización geográfica, etc. [11].

Una tabla con las características de los distintos datasets puede verse a continuación (Tabla 2.1.1).

Dataset	Nº Grabaciones	Duración Media (seg)	Edad media (años)	Sexo (masculino/femenino)	Frecuencia de Muestreo (Hz)
CPSC	6877	15.9	60.2	54% / 46%	500
CPSC-2	3453	15.9	63.7	53% / 47%	500
INCART	72	1800.0	56.0	54% / 46%	257
PTB	516	110.8	56.3	73% / 27%	1000
PTB-XL	21837	10.0	59.8	52% / 48%	500
G12EC	10344	10.0	60.5	54% / 46%	500

Tabla 2.1.1: Grabaciones de los distintos datasets utilizados en el trabajo.

En nuestro caso, se hizo uso de los datasets CPS, CPS-2, INCART, PTB, PTB-XL y G12EC. La premisa original del concurso era lograr clasificar con el mayor índice de acierto un grupo de 27 patologías concretas. Sin embargo, en los datasets hay datos de hasta 111 patologías diferentes.

Todos los datos están en formato WFDB y cada grabación tiene un fichero binario Matlab v4 para los datos de la señal, y un fichero de texto con formato WFDB de cabecera, que describe las características de la grabación y los datos del paciente, incluyendo el diagnóstico.

Estos ECGs son los datos que se procesarán para extraer sus características y plasmarlas en forma de imágenes, que posteriormente puedan servir de entrada a una red neuronal.

2.2 Tratamiento del ECG para extracción de características

Los ficheros binarios con la señal de cada ECG estaban almacenados en forma de matriz, donde cada fila contenía la señal grabada en cada derivación.

Estas grabaciones presentan (o pueden presentar) alguna o varias de las degradaciones que hemos explicado al hablar del ECG. Por ello antes de hacer ninguna extracción de características es preciso limpiar la señal.

A modo de ejemplo iremos siguiendo el procesado de las señales fijándonos en las derivaciones I, AVF y V5. Se han escogido estas tres derivaciones para ilustrar el procesado de los ECG debido a que cada una pertenece a un plano de proyección distinto de la actividad cardíaca. I proyecta de brazo derecho a izquierdo, AVF proyecta hacia los pies y V5 proyecta en horizontal desde la espalda hacia el frente.

2.2.1 Lectura, remuestreo y filtrado de valores no fisiológicos del ECG

En primer lugar, cargaremos el fichero .mat que contiene esta matriz, y leeremos su fichero de encabezado .hea. Después convertiremos su amplitud a uV y comprobaremos que su frecuencia de muestreo es de 500 Hz.

Por simplicidad se ha decidido trabajar con las señales muestreadas a 500Hz, ya que es una frecuencia suficiente para capturar la información relevante, y es conveniente que la red tenga a la entrada señales con una misma frecuencia de muestreo. Esto supone que se deberán remuestrear todas aquellas señales que estén a 275 Hz o a 1000 Hz.

Partimos de las señales ECG originales (Figura 2.2.1).

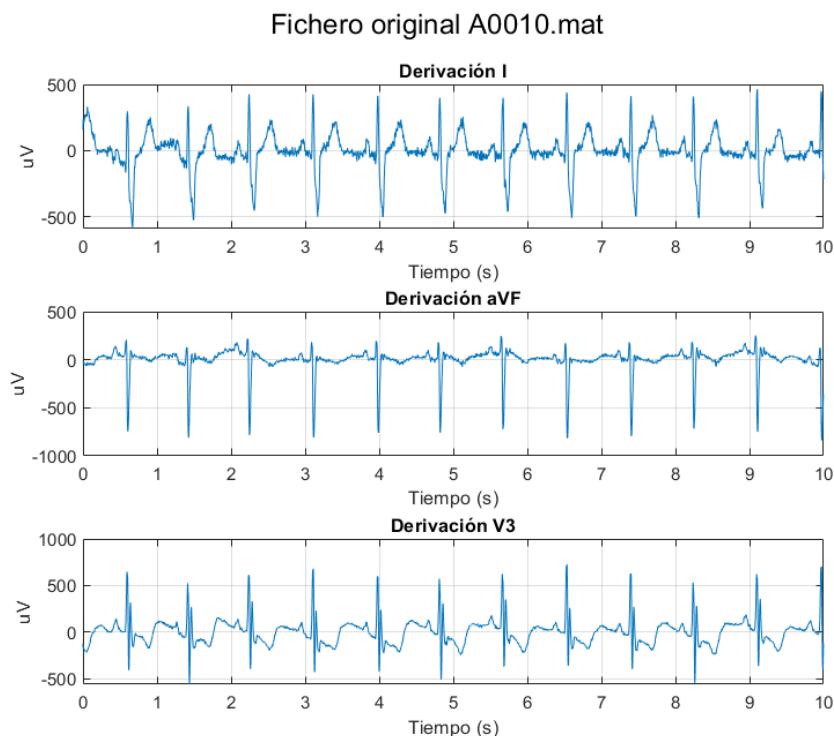


Figura 2.2.1.1: Derivaciones I, AVF y V5 originales.

En el ejemplo, las señales originales estaban muestreadas a 500 Hz, pero entre los ficheros utilizados podemos encontrar señales muestreadas también a 275 o 1000 Hz, realizándose en estos casos un remuestreo.

Después de esto, analizaremos si la señal presenta valores de amplitud no fisiológicos. Estos artefactos pueden ser causados por un error de codificación de la señal o del equipo de medida de la misma y en ocasiones pueden ser eliminados al diezmar la señal en el remuestreo (cuando pasamos de 1000 Hz a 500 Hz). Sin embargo, estos errores pueden persistir en las señales tras el remuestreo, generándose picos de amplitud de unas pocas muestras. Para detectar estos casos comprobaremos en cada derivación si la diferencia de amplitud máxima entre muestras es superior a 5000 uV. Este es un límite generoso, ya que la máxima variación de amplitud fisiológica en un ECG está entre los 2,5 mV y los 3 mV [12].

En el caso de detectar variaciones que superen este rango dinámico tan amplio en una derivación, se aplica un filtro de mediana de 5 muestras a toda la señal. El filtro de mediana ha sido empleado en casos similares a este en los que se desea eliminar artefactos poco frecuentes de la señal. Esto genera una versión suavizada de la señal y se ha demostrado su superioridad frente a un filtro lineal a la hora de preprocesar otras señales biológicas como los potenciales evocados del cerebro (ERPs) [13].

2.2.2 Primera delineación y eliminación de línea de base

Una vez eliminadas la mayoría de las componentes no fisiológicas, es necesario localizar los latidos y las ondas correspondientes a cada fase de los mismos. Para ello, se realiza una primera delineación de cada una de las derivaciones.

La delineación es el proceso por el cual, mediante algoritmos y procesado de señal, se obtienen las localizaciones de cada tipo de evento en el ECG. Es decir, es el proceso que generan unas anotaciones indicando dónde está cada onda del ECG.

En nuestro caso haremos uso de un delineador desarrollado en el grupo de investigación y descrito en [14], llamado Wavedet. Wavedet es un software basado en la transformada wavelet que permite analizar la señal en distintas escalas para detectar en ellas las diferentes componentes de ECG. Pudiendo trabajar con señales a 250 Hz, 500 Hz y 1000Hz. Implementa la detección de QRS como primer paso antes de detectar el resto de ondas, sus inicios y finales, pero también permite el uso de anotación externa de QRS (en este modo, le indicamos donde están los QRS, y el delineador intenta encontrar el resto de ondas de cada latido). Delinearemos cada una de las 12 derivaciones obteniendo la posición de los eventos más importantes de cada señal ECG.

Con la posición de los QRS podemos hacer uso de una función para limpiar la variación de la línea de base de las derivaciones (Figura 2.2.2.1). Era preciso delinejar el ECG previamente debido a que esta función toma como parámetro la posición temporal de los intervalos PR, es decir, el tramo entre el final de la onda P y el inicio del complejo QRS. En este punto puede ocurrir que el delineador no sea capaz de delinejar correctamente alguna de las derivaciones. En estos casos se sigue adelante, obviando la limpieza de las variaciones de la línea de base, ya que posteriormente será posible volver a delinejar estas señales y obtener unas anotaciones válidas.

Fichero A0010.mat sin línea de base

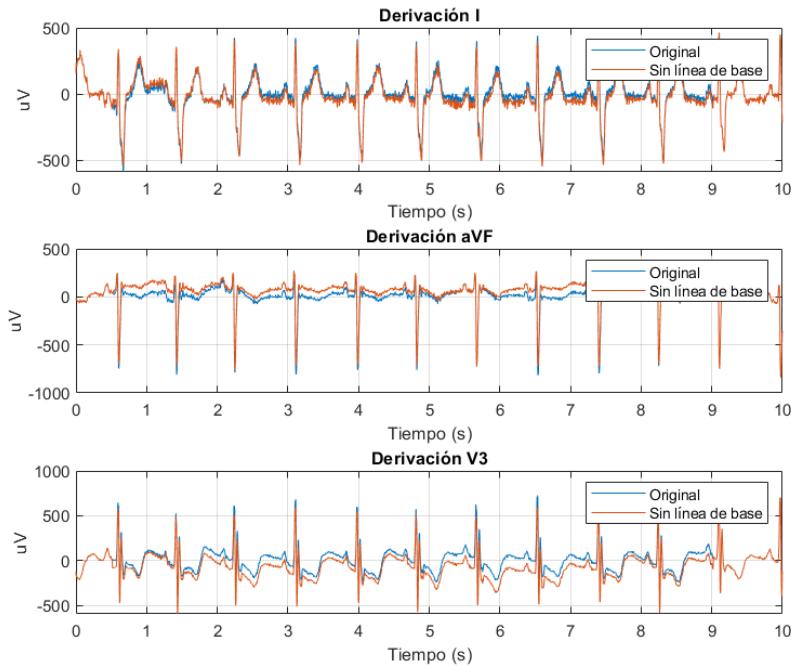


Figura 2.2.2.1: Derivaciones I, AVF y V5 eliminada su variación de línea de base.

2.2.3 Cálculo de índices globales de QRS

Debido a que tenemos 12 derivaciones, el delineador nos dará los índices de ocurrencia de cada evento del ECG en cada una de ellas. Es de esperar que sean diferentes, pues las proyecciones son diferentes, y además puede haber errores del delineador en unas derivaciones y no en otras. Como todas las derivaciones son descripciones simultáneas de la misma actividad cardíaca podemos usar esa redundancia para calcular unos índices que llamaremos globales, resultado de la información presente en todas las derivaciones. Este cálculo lo realizaremos inicialmente para la detección de QRS.

Los índices globales de QRS son las anotaciones que nos dirán en qué muestras están los QRS haciendo uso de la información de todas las derivaciones. Para calcularlos tomaremos para cada derivación sus índices de QRS y generaremos un tren de deltas, poniendo una delta en cada detección de QRS. Posteriormente estos trenes de deltas se convolucionarán con una ventana de Hanning de duración 100 ms⁴.

En este punto tendremos 12 trenes de ventanas de Hanning, que sumaremos para obtener una señal en cuyos máximos tendremos algo similar al centro de masas del índice de detección de cada latido⁵. Localizado el primer QRS en cada derivación, el índice global de este QRS será posición del máximo al sumar todas las ventanas de Hanning solapadas. (Figura 2.2.3.1).

⁴ 100 ms es la duración aproximada normal de un QRS [15].

⁵ Este método puede asemejarse a un método de estimación con ventanas de kernel, abordado en otras publicaciones [16].

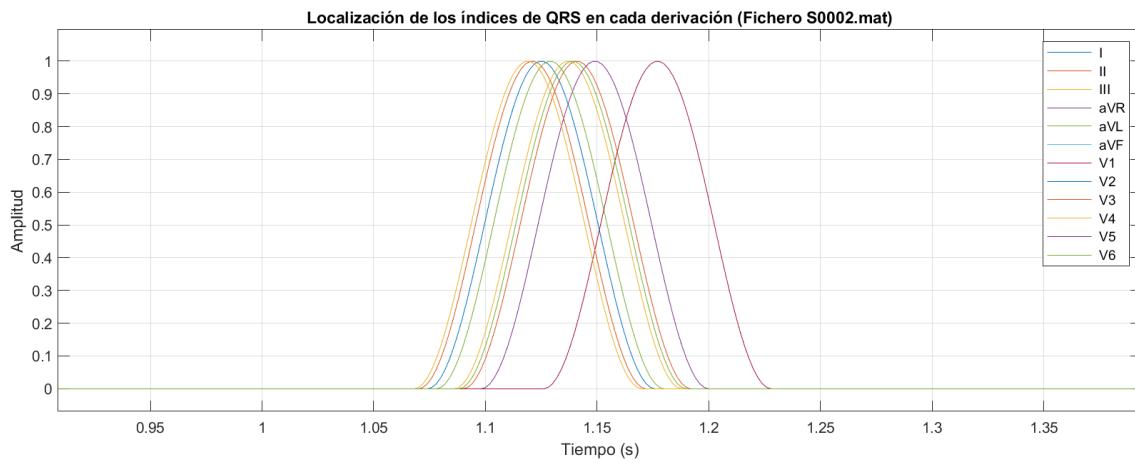


Figura 2.2.3.1: Ventanas de Hanning asociadas al primer QRS en las 12 derivaciones.

Tras realizar esta suma, calculamos el valor de amplitud umbral, a partir del cual se consideró que un máximo correspondía con un índice de QRS global. El umbral es necesario para filtrar aquellos casos en los que se detecta un QRS erróneamente, ya que, si no, tomariamos como índice global de QRS también los outliers⁶ que pudiera haber.

Inicialmente consideramos un umbral fijo de entorno a un valor 8 de amplitud, ya que se pensó que podíamos exigir que al menos 8 derivaciones coincidieran aproximadamente en el instante del latido. Sin embargo, existían muchos casos en los que, por variaciones en los instantes de detección del QRS entre derivaciones no bastaba con ese umbral. En vista de esto se optó por un umbral dinámico que se adaptase a cada caso. Después de experimentar con varios umbrales, fijos y variables, se acabó calculando el umbral como el 60% de la amplitud del pico mayor.

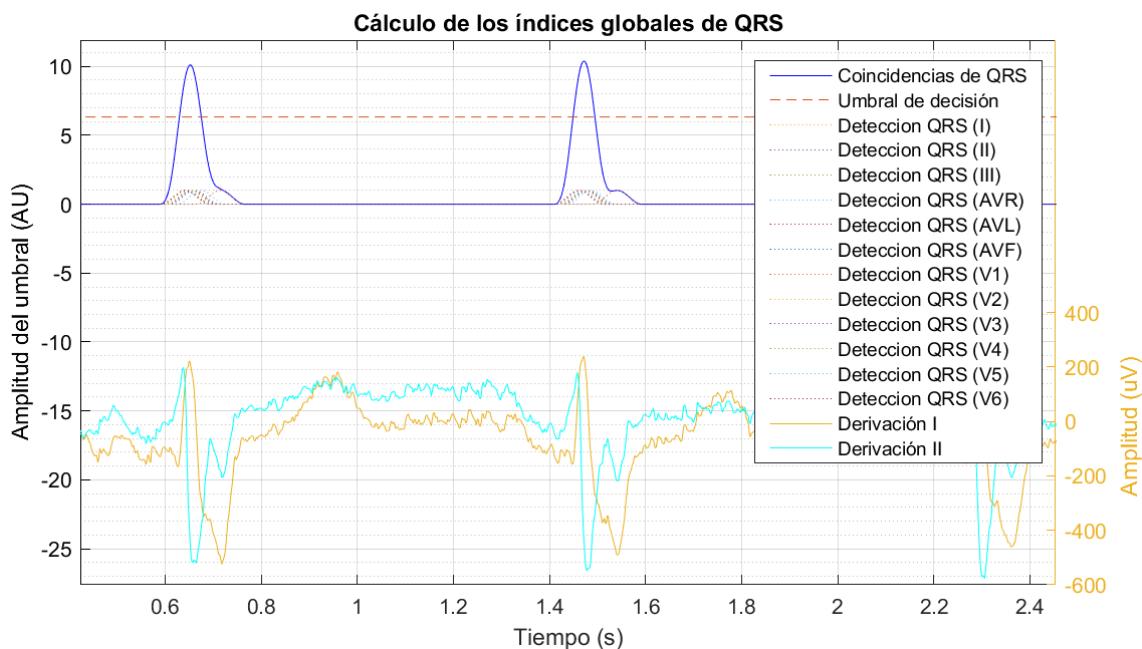


Figura 2.2.3.2: Coincidencias en los índices de QRS y umbral de decisión de índices globales.

⁶ Llamamos outliers a aquellas detecciones de QRS que están muy separadas del evento real y del resto de anotaciones que lo identifican. Esto puede ocurrir si el delineador comete algún error o si la señal presenta alguna distorsión.

Las posiciones de los máximos de los picos que superan el umbral serán también la posición de los QRS, la cual guardaremos en un fichero. (Figura 2.2.3.2).

2.2.4 Segunda delineación y formación de señales sintéticas

A continuación, podemos volver a delinear todas las señales para encontrar los inicios y finales de cada onda, pero en este caso le daremos al delineador las posiciones de los QRS como entrada externa, indicándole la ruta al fichero de anotaciones globales que acabamos de guardar.

Esto hace que el delineador se vea forzado a generar anotaciones para el mismo número de latidos en todas las delineaciones, y permite delinear señales problemáticas para las cuales el delineador habría fallado. Esto es así ya que en la mayoría de los casos observados en los que Wavedet fallaba en la delineación esto se debía a un error en la detección de los QRS. El cálculo de los índices globales de QRS y la posterior delineación con anotaciones externas permite proporcionar al delineador la posición de los QRS, evitando así la mayoría de los errores.

Con estas anotaciones se procedió a generar tres señales sintéticas binarias que señalasen las posiciones de los complejos QRS, las ondas P y las ondas T. Las tres tendrían valor nulo, salvo entre el inicio y el fin de una fase concreta del ECG, donde valdrían 1. La primera tendría pulsos que señalan el intervalo entre el inicio y fin de los complejos QRS, la segunda entre inicio y fin de las ondas P, y la tercera entre inicio y fin de ondas T (Figura 2.2.4.2).

Estas señales sintéticas contienen características que hemos extraído de la señal original de QRS y que le pueden servir a la red neuronal para relacionar las patologías con ciertos patrones presentes en dichas señales. Para formarlas hicimos uso de las anotaciones que nos indican el inicio y fin de los eventos QRS, onda P y onda T; para cada derivación.

Ejemplo de índices de inicio y final de QRS, onda T y P (Figura 2.2.4.1).

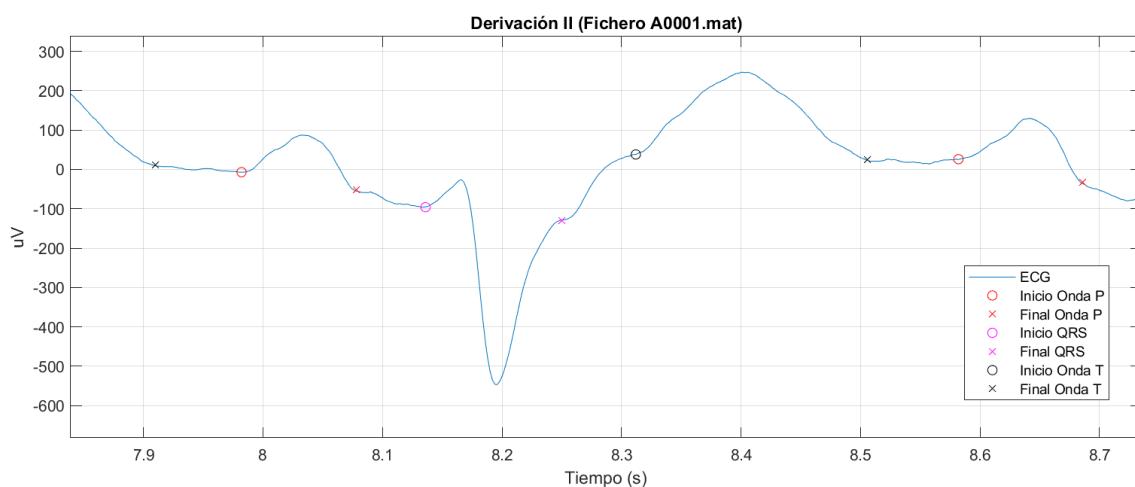


Figura 2.2.4.1: Derivación II de ECG real con las principales ondas delineadas automáticamente.

Pongamos de ejemplo la generación de la señal sintética que modela los inicios y finales de los QRS. Como inicio de cada QRS se toma la anotación de inicio de QRS menor de entre todas las derivaciones (detección más temprana), y como final del QRS la anotación

de fin de QRS mayor (última detección). Para evitar outliers se fuerza a cada inicio escogido a estar menos de 12 ms antes de al menos otros 3 inicios, y al final escogido a estar menos de 12 ms después de al menos otros 3 finales.

Estas reglas de protección garantizan que no tomamos como inicio o final un valor que se da solo en una derivación y está alejado del evento real (que se verá casi simultáneamente en la mayoría de las derivaciones).

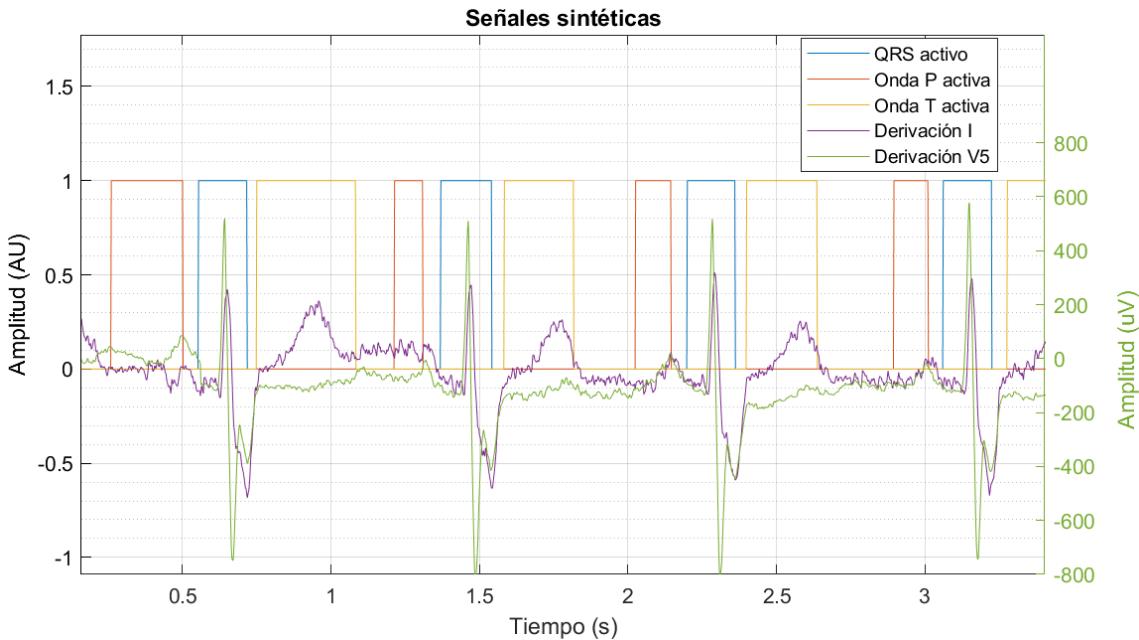


Figura 2.2.4.2: Señales sintéticas de QRS, onda P y onda T solapadas.

Generadas estas tres señales binarias, las vamos a considerar como señales adicionales que nos “aumentan” la información presente en el propio ECG; y ya podemos formar las 15 imágenes que servirán de entrada para nuestra red neuronal (12 para las derivaciones y 3 para las señales binarias).

2.2.5 Generación de imágenes

El concepto clave en la generación de imágenes es que en cada fila de cada imagen veamos una ventana temporal general concreta tomando como referencia un latido diferente (para ello, usaremos los índices de QRS globales como puntos de referencia de cada latido). De esta forma, en cada imagen veremos, de cada derivación o señal binaria, los valores que hay en una ventana temporal común a todas las imágenes. Cada fila de la imagen corresponde con una ventana temporal que comienza medio segundo (250 muestras) antes de la detección del QRS, y termina un segundo después de la misma detección (500 muestras). De esta manera, cada fila de las imágenes es una ventana temporal de 750 muestras o 1.5 s, lo cual permite observar la onda P, el complejo QRS, la onda T, y en general, también la onda P y complejo QRS del siguiente latido. Los diferentes latidos quedan alineados en la imagen con respecto al índice del QRS global. El número de filas de las imágenes vendrá dado por el número de latidos detectados en el ECG, es decir el número de índices globales de QRS detectados.

Cabe destacar en este punto que, para garantizar que todas las ventanas fueran de 750 muestras, se ha tenido que excluir algunos latidos al inicio y final de algunas grabaciones. También se valoró hacer relleno con ceros en la señal para poder tomar de manera genérica la ventana de 750 muestras a partir de cada latido detectado. Esta última implementación se descartó, puesto que se consideró que el relleno podía ser un agente de confusión, al añadir muestras irreales que no aportan información clínica real sobre la señal.

Las imágenes formadas siguiendo el procedimiento descrito sirven a la red para extraer las características de los ECG y asociarlas con las patologías. A simple vista es posible ver algunas de estas características en las imágenes. Debido a que las ventanas temporales cubren, en general, un latido y parte del siguiente, se pueden advertir en las imágenes características del ECG como la variabilidad entre las señales latido a latido, o la propia variación del ritmo cardíaco (en especial en la zona en la que se muestran los segundos latidos de cada ventana).

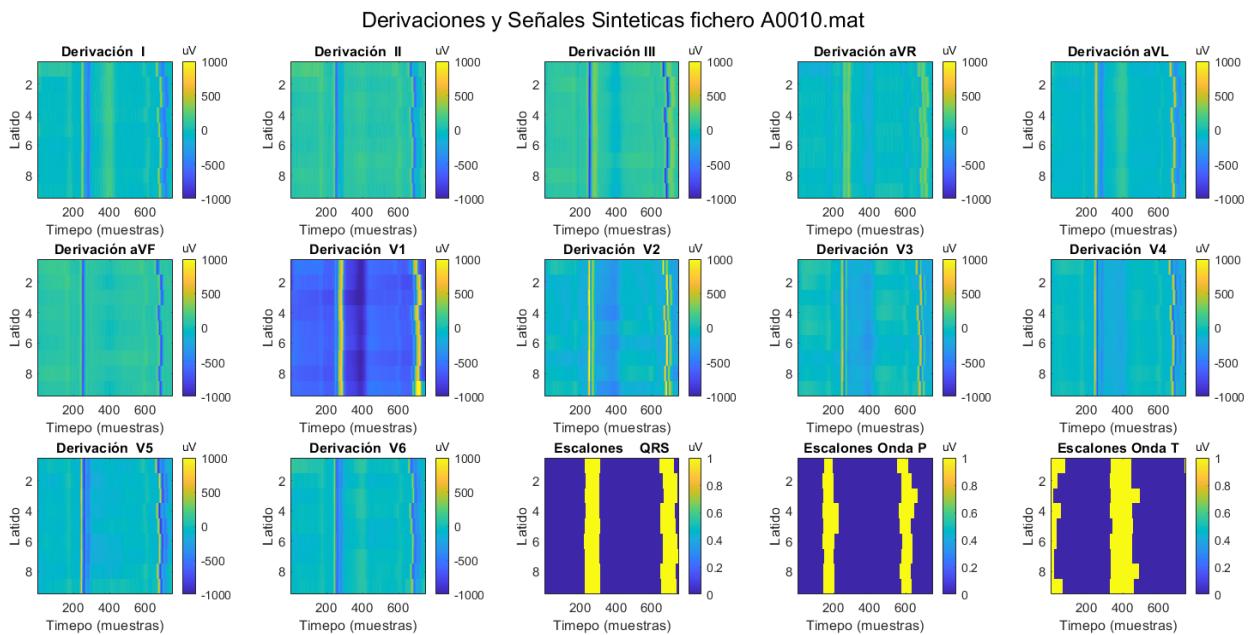


Figura 2.2.5.1: Imágenes resultantes del procesado de las 12 derivaciones y las 3 señales sintéticas.

2.3 Análisis de las etiquetas

Posteriormente, se ha llevado a cabo un análisis de las etiquetas, observando las coincidencias entre patologías, para tratar de encontrar patrones que simplificasen la clasificación o etiquetas que no pudieran darse a la vez que otras. El ánimo de esto era conformar un subconjunto del total de ficheros disponibles, de manera que la tarea de clasificación fuera muy sencilla inicialmente, para después aumentar la complejidad.

Para ello hicimos una comparación cruzada en los diferentes datasets, observando cuántas veces coincidía cada par de patologías a lo largo de todas las grabaciones. El número de coincidencias de cada par de patologías en función del dataset puede consultarse en el [Anexo II](#). Las coincidencias entre patologías a lo largo de la suma de datasets puede verse a continuación (Figura 2.3.1).

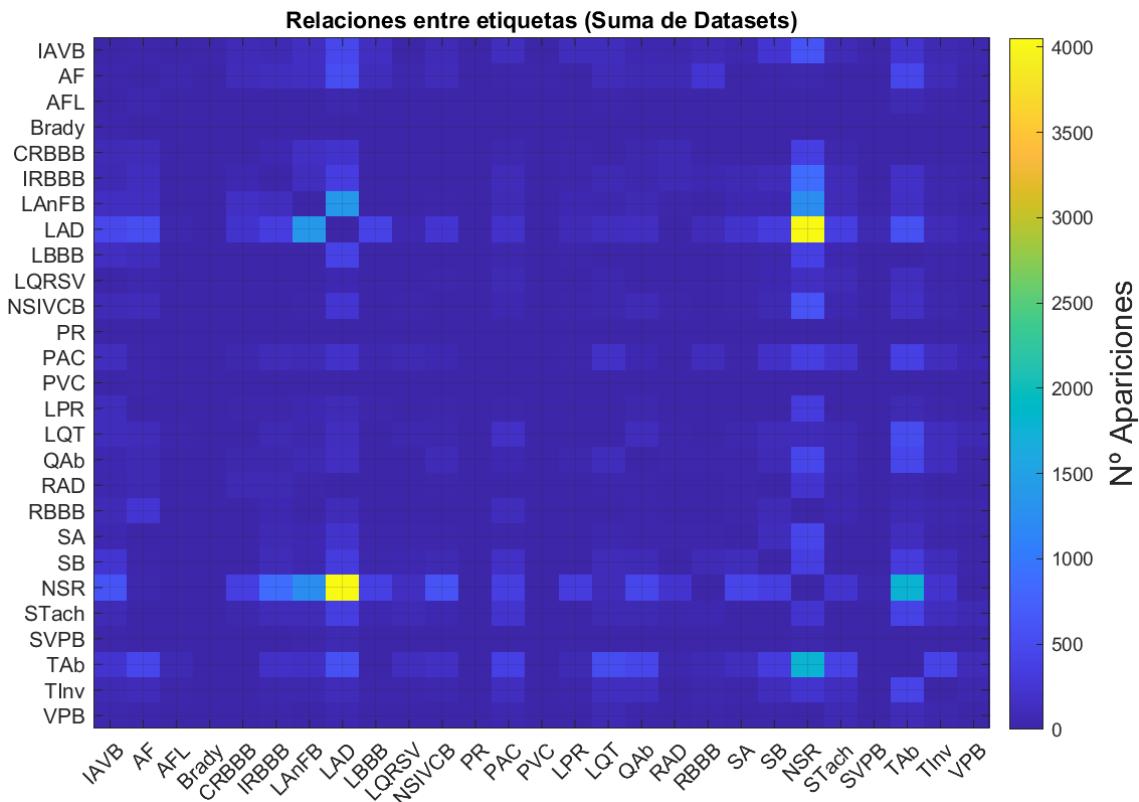


Figura 2.3.1: Apariciones conjuntas de las distintas patologías a lo largo de todos los datasets.

Tras este primer análisis no se detectó un patrón suficientemente generalizado de dependencia entre patologías, pero sí se advirtió un gran número de ejemplos de ECG en los que la única patología presente era SNR (ritmo sinusal) o AF (fibrilación auricular).

Por este motivo, como primer paso, se conformó un subconjunto simple que contuviera ficheros cuya única patología presente fuera SNR o AF. El SNR o Ritmo Sinusal indica que como el marcapasos natural del corazón está actuando el Nodo Sinoauricular. Esto corresponde al comportamiento normal del corazón por lo que esta etiqueta identificaría a un paciente preliminarmente sano. Por otro lado, AF significa Fibrilación Auricular, que es una de las arritmias tratadas más comunes entre las poblaciones europea y norteamericana [17] [18] [19] (en especial en las de mayor edad).

Por estas razones se eligieron estas patologías para conformar un primer clasificador binario que fuera un detector de AF. Posteriormente se introdujeron más patologías, haciendo un clasificador multiclase con clases excluyentes. Para esto se recogieron los ficheros que tenían como única patología alguna de las siguientes: SNR (ritmo sinusal), AF (fibrilación auricular), IAVB (bloqueo auriculo-ventricular de primer orden) o RBBB (bloqueo de rama derecha). Estas patologías fueron escogidas por ser las que más ejemplos presentaban en ECGs con una única patología o etiqueta.

Finalmente, también se formó un subset con todos los ficheros de ECG que entre sus etiquetas contuvieran alguna de las clases anteriores. Esto se hizo con el fin de entrenar un modelo de red neuronal que reconociera la presencia de estas 4 patologías, de forma no excluyente, pudiéndose dar una, varias de ellas o ninguna en la misma señal.

CAPÍTULO 3. DISEÑO DE LA RED NEURONAL Y ENTORNO DE ENTRENAMIENTO

3.1 Construcción de un DataLoader

Como primera tarea antes de empezar con la red neuronal, se planteó el diseño de un cargador de datos que fuera capaz de leer las imágenes y etiquetas generados en Matlab y servírselos a la red en su entrenamiento con gran rapidez.

La idea principal del dataloader es que proporcione a la red las distintas imágenes descritas en el capítulo anterior de tamaño 10x750, es decir, 10 ventanas temporales de cada fragmento de señal. Debido a que hay imágenes con más de 10 latidos (ventanas temporales) y con menos de 10, se trata cada caso recortando 10 ventanas contiguas partiendo de una fila aleatoria de la imagen, en el caso de ser suficientes; o replicando las ventanas existentes hasta que podamos tomar 10 de ellas en caso contrario. En el capítulo anterior se describen 15 imágenes según la derivación y el procesado realizado para obtenerlas. La forma más adecuada de representarlas para el uso que les vamos a dar con redes neuronales es apilarlas en un tensor que tendrá dimensión 15x10x750.

Hubo que tener en consideración que debido al gran tamaño del dataset, no era posible cargar todos los datos en memoria, por lo que la carga se va realizando conforme la red necesita más datos. En un inicio se planteó realizar variaciones del número de ventanas temporales a escoger, para comprobar si había cambios en la precisión de diagnóstico de ciertas patologías, pero, finalmente y por cuestión de tiempo, se decidió realizar los entrenamientos con imágenes de 10x750, siendo esto un estudio preliminar y pudiéndose explorar más casos en futuros trabajos.

Para la programación del cargador de datos, se creó una clase que heredaba propiedades de la clase Dataset del paquete utils.data de la librería Pytorch. Posteriormente también se hizo uso de la clase DataLoader, de la misma librería, para gestionar la carga de datos. Su utilización tiene la ventaja de que permite el uso de varios procesadores para cargar de forma simultánea los datos y que esto no suponga un cuello de botella para el proceso de entrenamiento, ya que si la carga no es paralela es bastante probable que la GPU tenga muchos tiempos de espera sin actividad esperando a que se lean los datos.

El proceso de carga de un tensor, ilustrado en la Figura 3.1.1, es el siguiente: En su creación esta clase toma un parámetro que es una lista de los nombres de los ficheros que conforman el dataset y con ello actualiza sus variables internas. Cuando se solicita un tensor de datos se ejecuta el método `__getitem__()` el cual toma como argumento el índice del fichero a cargar dentro de la lista de nombres.

Tras una comprobación de la integridad de los datos y asegurarnos de que el tensor tiene al menos 10 ventanas temporales, el programa devuelve un tensor de tamaño 15x10x750 con las 15 imágenes apiladas que la red ya puede utilizar como entrada.

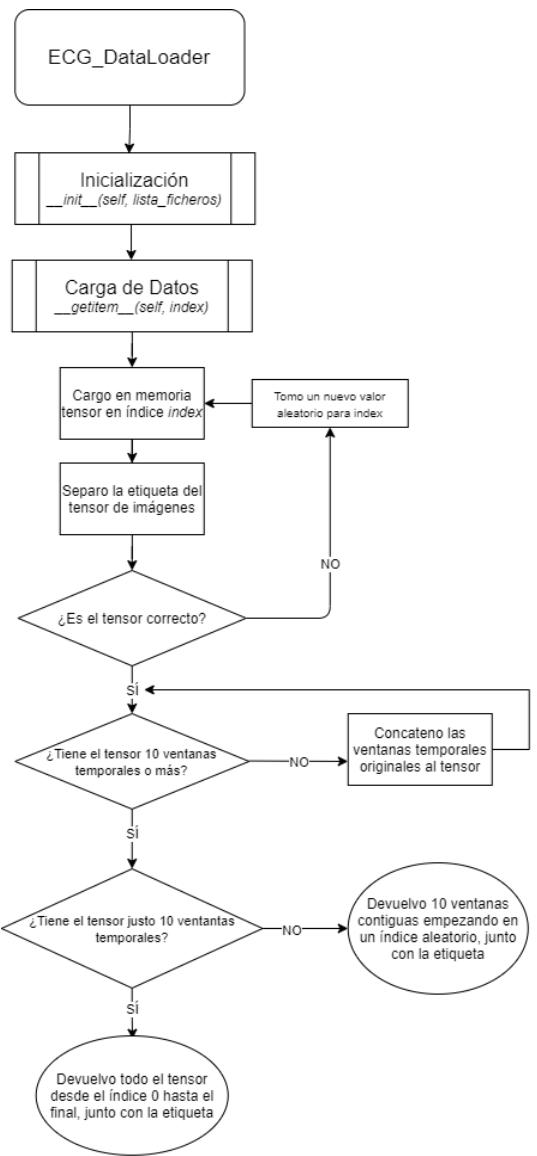


Figura 3.1.1: Diagrama de flujo del cargador de datos.

3.2 Diseño de un modelo sencillo de red

Debido a que nuestros datos de entrada a la red son imágenes se decidió diseñar una red de arquitectura convolucional. Tradicionalmente, esta ha sido la estructura más usada para trabajar con imágenes, siendo en un inicio muy común separar las imágenes en tres canales de color: RGB; que después se apilaban para servir de entrada a la red.

En nuestro caso son 15 los canales que contienen la información del ECG, como hemos visto en el capítulo anterior. Por lo tanto, presentaremos como entrada a la red las 15 imágenes apiladas. Como las imágenes obtenidas del ECG mediante el proceso de enventanado son de dimensión 10x750, el dato de entrada a la red tendrá como dimensiones (15x10x750).

En una red convolucional las imágenes de entrada irán atravesando varias capas en las cuales se realizarán convoluciones 2D. Estas convoluciones se realizarán entre las imágenes de entrada y un filtro 2D propio por cada canal de entrada y salida. Como ejemplo observamos en la Figura 3.2.1 la señal a la entrada de la primera capa convolucional, que tiene 32 canales de salida, esto implica que la capa convolucional tiene como parámetros 32×15 filtros distintos. La señal a la salida de esta primera capa tendrá como dimensión (32x10x750), es decir 32 imágenes apiladas del tamaño inicial. Para favorecer la generalización y reducir el número de operaciones es habitual tras la convolución realizar diezmados de la imagen, que permiten reducir sus dimensiones, después de los cuales se aplica la función de activación antes de la siguiente capa.

Los parámetros de estas capas convolucionales, tales como los valores del filtro, sesgos, etc.; se irán optimizando durante el entrenamiento de manera que estas capas se especialicen en extraer las características que identifican a las imágenes.

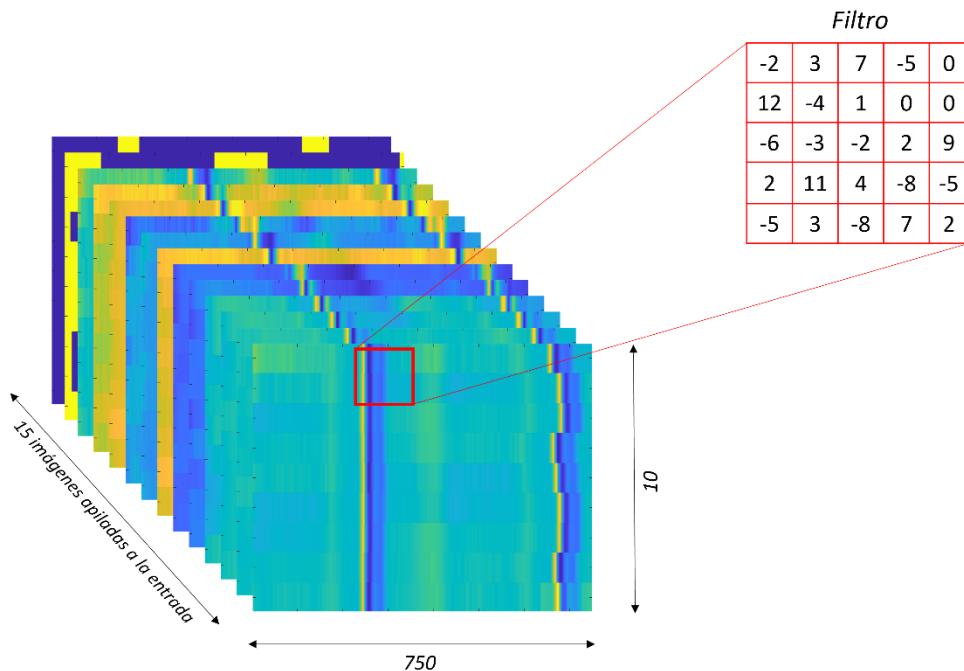


Figura 3.2.1: Proceso de convolución con filtro al atravesar capa convolucional.

Posteriormente estas características extraídas de las imágenes de entrada suelen pasar por otros tipos de capas que realizan la tarea de clasificación. En nuestro caso este papel lo harán capas lineales, traduciendo esas características de las imágenes en un vector de probabilidades a posteriori que nos indican lo “segura” que está la red de la presencia de las posibles patologías. Podemos ajustar diversos parámetros de todas estas operaciones

como las dimensiones del filtro 2D, los tamaños de kernel de la convolución, o el nivel de diezmado de la imagen que queremos. Todo esto da lugar a multitud de redes y configuraciones distintas. Estudiaremos algunas tendencias en los resultados como consecuencia de estas decisiones de diseño en la parte experimental.

En la Figura 3.2.2 podemos ver estos pasos en un ejemplo de diseño concreto. El paso de las capas convolucionales a las lineales requiere una reducción de las dimensiones espaciales de las imágenes para favorecer que el sistema realice una clasificación global independientemente de dónde se den lugar los eventos dentro de la señal. Este paso puede realizarse de distintas formas. En este trabajo se ha optado por una de las más extendidas que consiste en aplicar una reducción de tipo promedio en la dimensión que nos interesa reducir, es decir la dimensión temporal (Average Pooling), de forma que la salida de la última capa convolucional que consiste en 128 imágenes de 5x12 (128x5x12) reduce su dimensión temporal mediante un promedio, 5, siendo ahora de dimensión (128x12).

Aunque pueda parecer que se han perdido detalles la tarea de la red al optimizar sus parámetros será codificar en las dimensiones restantes la información relevante para clasificar el ejemplo independientemente de la posición temporal de los detalles que diferencian a la imagen de otras. Posteriormente, se obtienen los datos en forma unidimensional reagrupando la información de esta “imagen” en un vector, mediante una operación matriz a vector (en el módulo de programación se conoce la función como reshape). Es decir, pasamos de dimensión (128x12) a un vector de dimensión (1x1536), al que posteriormente se aplican un par de capas de red convencional (como un MLP, Multilayer Perceptron) con una no linealidad entre ambas.

Si la tarea se puede definir como clases excluyentes el objetivo es que la red acabe dando como resultado una estimación de la probabilidad a posteriori de cada clase dada la entrada. Esto se hace gracias a una capa final de tipo softmax que obliga a que las salidas sumen 1 y sean positivas. En el caso del ejemplo de la figura este número de clases es 4. Si las clases no son excluyentes entre sí, sino que son varias decisiones binarias independientes la salida no se hace pasar por una softmax sino que se suele aplicar una función logística o sigmoide para obtener cada una de las decisiones de forma independiente.

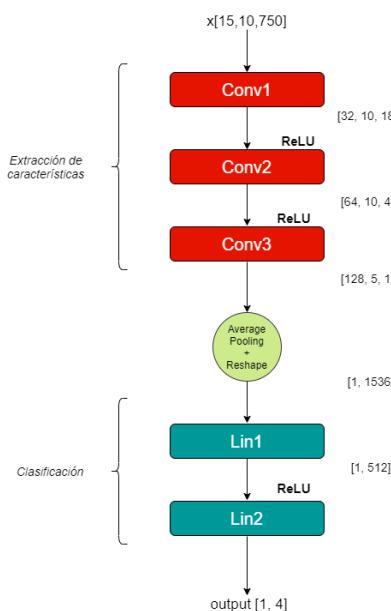


Figura 3.2.2: Diagrama de flujo y capas del modelo base de la red.

La función de activación escogida para toda la red ha sido de tipo ReLU (Recified Linear Unit) que actúa sobre los valores negativos de la señal, poniéndolos a 0, mientras que los valores positivos conservan su magnitud.

Por otro lado, como criterio de optimización se utilizaron dos tipos de funciones de coste en función del tipo de tarea. Si la tarea es una clasificación con clases excluyentes y el objetivo es obtener una probabilidad a posteriori que sume 1, usando una operación de tipo softmax, el criterio de entrenamiento usado suele ser la función de CrosEntropía.

La CrosEntropía es una medida, surgida del campo de la Teoría de la Información, que cuantifica la diferencia entre distribuciones de probabilidad dado un conjunto de eventos o variables aleatorias. La distribución objetivo es normalmente una probabilidad 1 en la clase etiquetada y cero en resto, por lo tanto, para disminuir la distancia entre distribuciones la salida de la red debe maximizar la probabilidad de la clase objetivo y disminuir las del resto. La Ecuación 3.2.1 corresponde con la expresión de la CrosEntropía, siendo $p(x)$ la distribución objetivo y $q(x)$ la salida de la red.

$$H(p, q) = - \sum_{x \in X} p(x) \log q(x)$$

Ecuación 3.2.1: Expresión de la crosentropía entre la distribución objetivo $p(x)$ y la salida de la red $q(x)$.

En el caso de que las salidas sean varios detectores binarios independientes, entonces se usa la función de CrosEntropía binaria entre las etiquetas y las salidas de la red, las cuales tienen que ser valores entre 0 y 1, para lo cual se ha aplicado una función logística o sigmoide como se ha comentado anteriormente. Esta función se aplica a cada detector binario independiente y se optimiza la suma de estos costes, lo cual se puede entender como un entrenamiento multiobjetivo.

Posteriormente introdujimos capas de normalización de batch⁷ entre todas las capas ya existentes. También introdujimos una capa de dropout⁸ con probabilidad 0.5 que usábamos tras las capas convolucionales.

Sobre el diseño básico existe la posibilidad de aplicar muchas mejoras. En concreto aplicamos algunas que no implican grandes incrementos en los costes computacionales. Por ejemplo, utilizar capas de normalización de batch nos permite mejorar y estabilizar la convergencia de los pesos de cada capa durante el entrenamiento. Esto sumado a la capa de dropout, que desecha un porcentaje de los datos de entrenamiento, nos permite evitar casos de overfitting y tener un modelo más robusto que converja en menos tiempo [20].

Aplicando estas mejoras sobre el modelo básico de la Figura 3.2.2 anterior definimos el llamado en adelante Modelo 0 (Figura 3.2.3) que sirvió como base para las demás variaciones realizadas.

⁷ Las capas de normalización de batch son capas intermedias que adaptan la entrada de la siguiente capa, normalizándola y reescalándola, con el fin de optimizar la velocidad y la estabilidad de la red.

⁸ Las capas de dropout funcionan activando y desactivando (con la probabilidad indicada, en nuestro caso de 0.5) neuronas aleatorias de la red durante el entrenamiento. Es una de las técnicas que se utilizan para hacer la red más robusta.

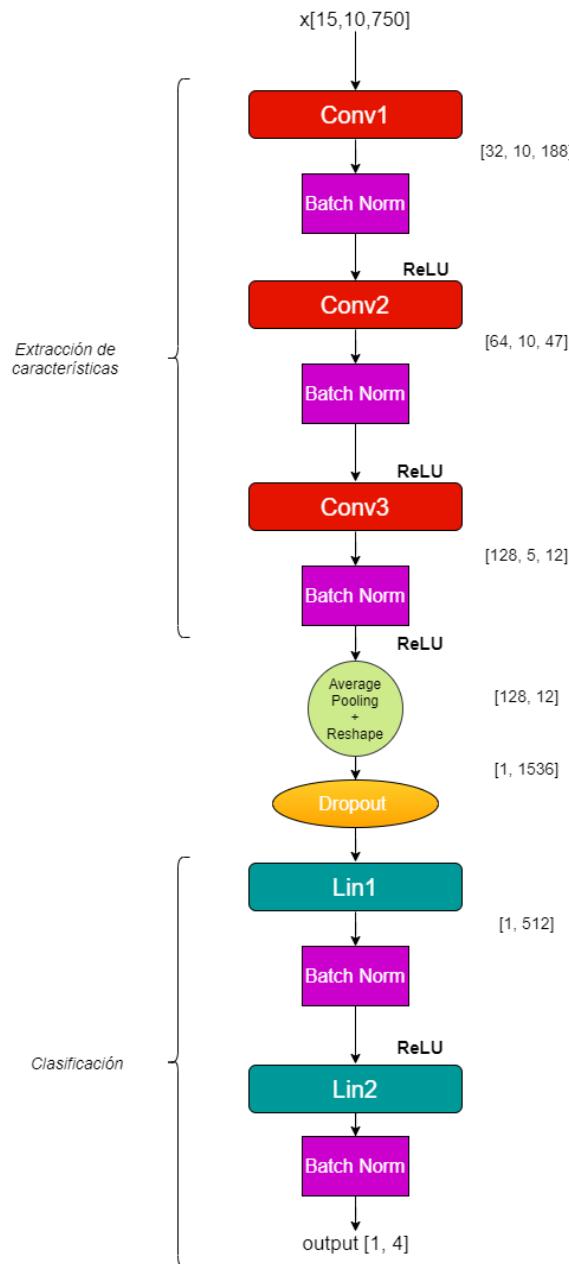


Figura 3.2.3: Diagrama de flujo y capas del Modelo 0 de la red.

3.3 Entorno de Entrenamiento

Para el entrenamiento y la evaluación de los diferentes modelos de red que se realizaron se contó con acceso al Cluster del grupo de investigación Vivolab⁹ y compuesto por 13 nodos de cálculo y 4 de almacenamiento.

Toda la programación se realizó en Python y se ejecutó y evaluó dentro de dicho cluster.

⁹ <http://vivolab.unizar.es/>

3.4 Primer entrenamiento del Modelo 0

Inicialmente el modelo con el que trabajamos fue entrenado con un subset de datos muy sencillo. Los ficheros correspondían con pacientes que podían tener una de dos patologías: SNR o AF. Estos ficheros fueron extraídos del dataset original CPS y contábamos con 918 ejemplos de SNR y 976 ejemplos de AF. Desde este punto nos referiremos a este subset concreto de ficheros como Subset 1.

3.5 Segundo entrenamiento del Modelo 0

En el segundo modelo quisimos aumentar la complejidad e introducir 2 patologías más, a fin de formar un clasificador de 4 clases excluyentes¹⁰. Los ficheros para el entrenamiento tendrían una única patología de las 4 posibles. Las 2 patologías que añadimos a SNR y AF son: el Bloqueo Auriculoventricular(IAVB) y el Bloqueo de Rama Derecha (RBBB).

Inicialmente se entrenó un modelo con ficheros solo pertenecientes al dataset de origen CPS, por lo que teníamos el siguiente número de ejemplos para cada clase: 918 de SNR, 976 de AF, 686 de IAVB y 1533 de RBBB; los cuales conformaban el que llamaremos Subset 2.

	SNR	AF	IAVB	RBBB	Total
Número de ficheros	918	976	686	1.533	4.113

Tabla 3.5.1: Ficheros de entrenamiento para cada patología en el Subset 2.

3.6 Tercer entrenamiento del Modelo 0

Lo siguiente que hicimos fue realizar un nuevo entrenamiento con muchos más datos, ya que los que habíamos usado eran tan solo los pertenecientes al dataset de origen CPS.

Por ello realizamos una recopilación de ficheros con estas patologías a lo largo de todos los datasets de Computing In Cardiology Challenge 2020. En esta ocasión reunimos todos los ECG que entre sus patologías contuviesen alguna de las cardiopatías a clasificar. Este último subconjunto de ficheros será referido en el resto de la memoria como Subset 3. En la Tabla 3.6.1 se puede ver el detalle de su composición.

	SNR	AF	IAVB	RBBB
CPS	918	1.221	722	1.675
CPS-2	4	152	106	1
G12EC	1.752	487	693	542
PTB	80	15	0	0
PTB-XL	17.441	1.514	788	0
INCART	0	2	0	2
Total	20.195	3.391	2.309	3.219

Tabla 3.6.1: Distribución de los ficheros de entrenamiento que conforman el Subset 3 por patología y dataset.

¹⁰ Cabe destacar que las patologías escogidas no tienen por qué ser fisiológicamente excluyentes, pero el objetivo del entrenamiento es validar la extracción de características realizada en la primera parte del trabajo.

De esta manera, con un pequeño cambio en la evaluación se adaptó la red para que pudiera seleccionar múltiples clases si estas eran detectadas de forma independiente. La idea era trabajar con el valor de salida de la red para cada clase. Debido a que usamos la función de coste basada en CrosEntropía binaria, y salidas del modelo con la función sigmoide o logística estos valores de salida pueden interpretarse como la probabilidad a posteriori de la clase en cuestión para cada detector binario. Partiendo de esta base fue necesario realizar un análisis de estas probabilidades con el fin de encontrar, si es posible, un umbral que maximizase las detecciones y evitase los falsos positivos y negativos.

Para realizar este análisis recurrimos a la curva ROC, y utilizamos como parámetro de precisión el área encerrada bajo la misma (AUC, Area Under the Curve). La curva ROC (Receiver Operating Characteristic) es una gráfica muy utilizada en el ámbito de la biomedicina para calcular el umbral más apropiado en un test. Esta curva es una medida de la sensibilidad y especificidad clínicas del sistema para cada posible umbral. Por ello, el área bajo la curva ROC nos da una idea de la utilidad de nuestro test (cuanto mayor es, más cerca estará el mejor umbral posible de la clasificación perfecta), y nos permite comparar clasificadores entre sí [21]. Por este motivo se utilizó la AUC para comparar las prestaciones de diferentes variaciones de nuestro modelo.

3.7 Variaciones en el modelo

A continuación, se planteó realizar modificaciones en el modelo en cuanto a número de capas, número de canales y factor de crecimiento de canales. Tomando para cada versión del modelo los mismos datos de entrenamiento y de test se trató de alcanzar un compromiso entre precisión en las predicciones y complejidad del modelo.

Para realizar estas pruebas se cambió el modelo de diferentes formas y se evaluaron sus prestaciones para poder representar la AUC media obtenida frente al número de parámetros que conforman cada modelo.

Como dataset de entrenamiento para todos ellos se utilizó el Subset 3, dado que presenta el mayor número de ejemplos y variedad de patologías.

3.7.1 Modelo 1

En primer lugar, se probó a añadir una capa convolucional más, llegando a una profundidad de 256 canales, como se puede ver en la Figura 3.7.1.1. Esto pretendía aumentar la complejidad de la fase de extracción de características de la red, para comprobar si esto se traducía en una clasificación más precisa.

Se realizaron dos entrenamientos, tanto con stride (factor de diezmado) en esta capa añadida, como sin él.

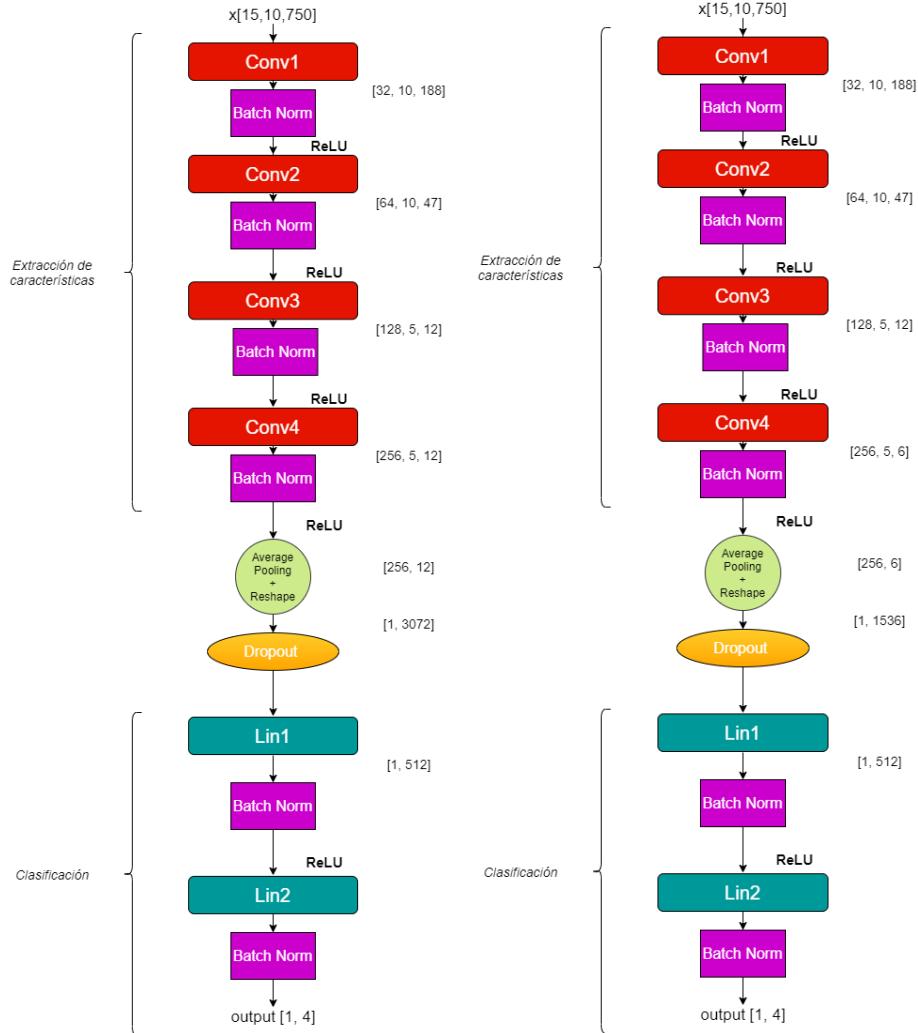


Figura 3.7.1.1: Diagrama de flujo y capas del Modelo 1 tanto sin stride (izquierda), como con stride (derecha).

3.7.2 Modelo 2

Seguidamente, se volvió al modelo inicial y se añadió una capa lineal más al final del modelo. El objetivo con esto era reducir el número de canales de 512 hasta 4 de manera más progresiva, aportando la nueva capa un paso intermedio de 512 a 128 canales, representado en la Figura 3.7.2.1.

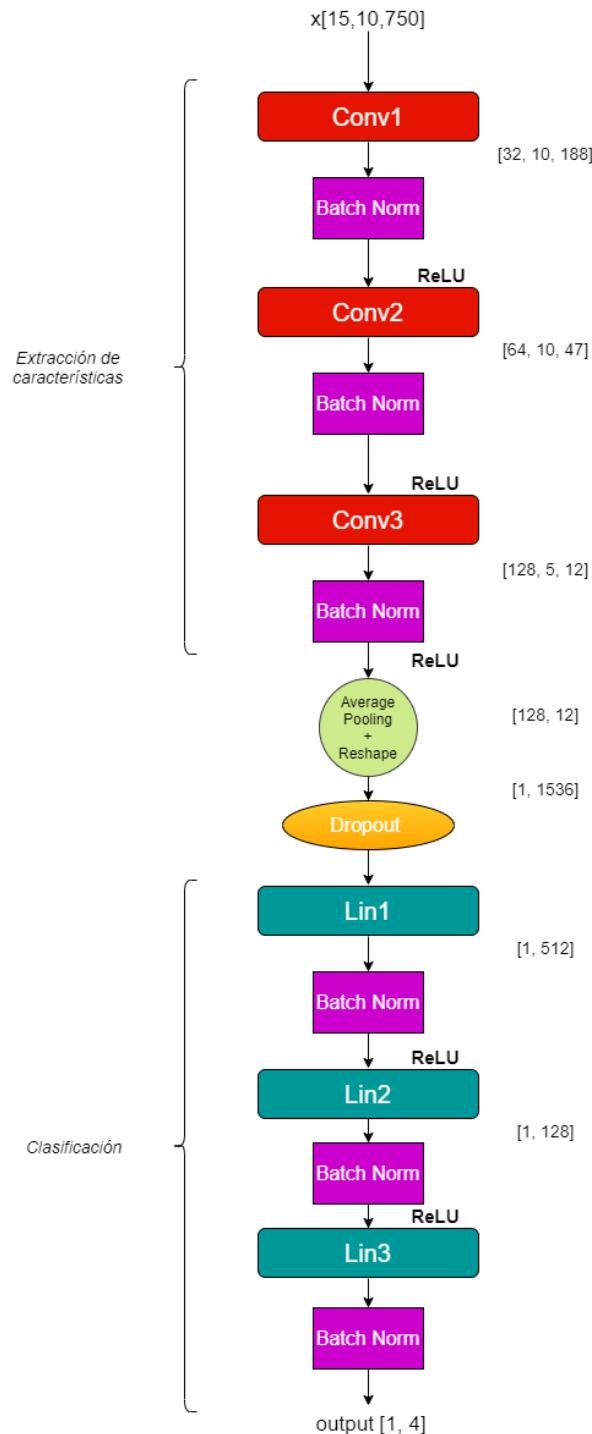


Figura 3.7.2.1: Diagrama de flujo y capas del Modelo 2.

3.7.3 Modelo 3

Después de esto se combinaron las dos versiones anteriores (Modelos 1 y 2), y se añadió además una capa lineal más tras las capas convolucionales, Figura 3.7.3.1. Por lo que teníamos una capa convolucional en cuya salida había 256 canales, una capa lineal a continuación, que permitía ayudar a reducir los canales progresivamente; y la capa lineal al final comentada en el Modelo 2.

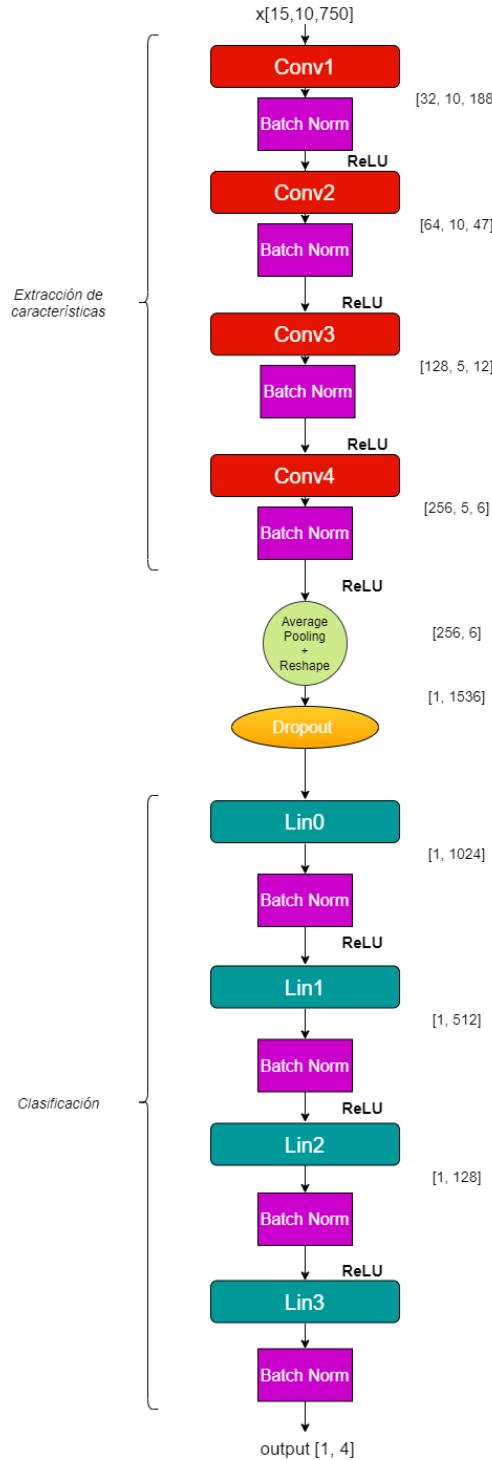


Figura 3.7.3.1: Diagrama de flujo y capas del Modelo 3.

3.7.4 Modelo 4

A continuación, se realizaron varias modificaciones en la dimensionalidad de las capas del modelo inicial. Para ello se programó el modelo de manera que partiera de un número de canales indicado en la primera capa y fuera aumentando y reduciendo el número de canales en función de un factor de crecimiento también aportado como parámetro. Este tipo de estrategias permiten una exploración más eficiente que variar libremente todos los números de canales [22].

Las variaciones de dimensionalidad realizadas fueron partir de 32 canales para factores de crecimiento 1.5 y 2.5; y posteriormente, usar estos mismos factores de crecimiento partiendo de 16 canales, como se observa en las Figura 3.7.4.1 y 3.7.4.2.

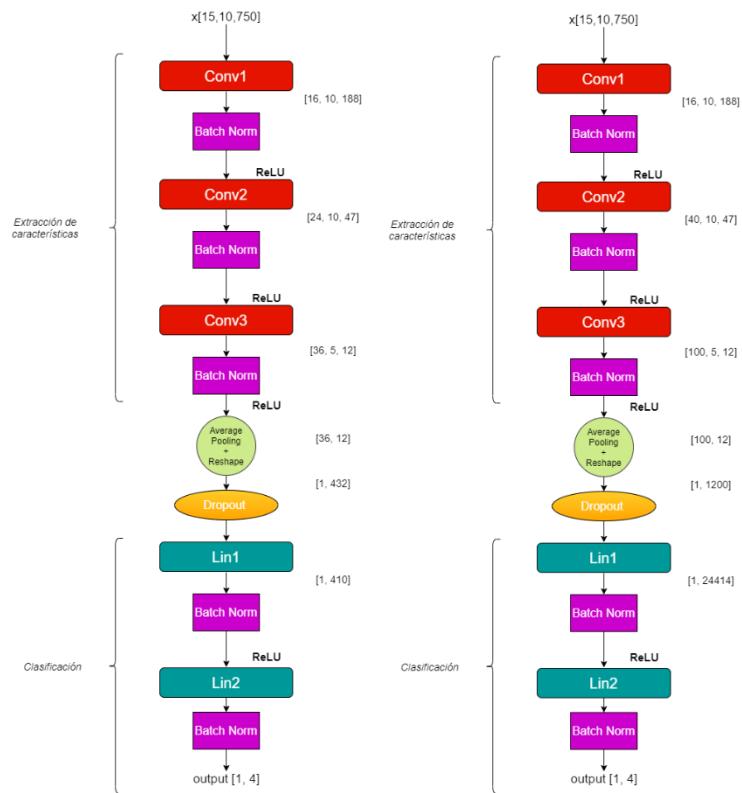


Figura 3.7.4.1: Diagrama de flujo y capas del Modelo 4_16_1.5 (izquierda) y Modelo 4_16_2.5 (derecha).

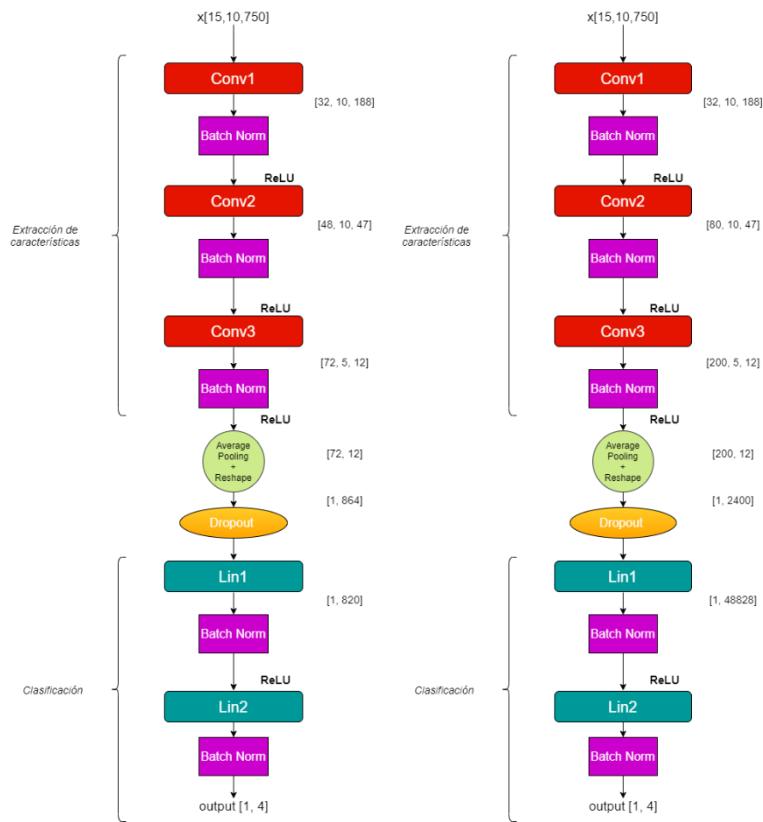


Figura 3.7.4.2: Diagrama de flujo y capas del Modelo 4_32_1.5 (izquierda) y Modelo 4_32_2.5 (derecha).

CAPÍTULO 4. RESULTADOS

4.1 Consideraciones iniciales

Para la realización de todos los entrenamientos se utilizó un optimizador basado en el algoritmo Adam con un learning rate de 0.0001 [23]. En todos los entrenamientos se ha reservado un porcentaje de los datos como datos de test, con los cuales estimar la precisión del modelo. En todos los entrenamientos realizados se ha utilizado un procesado de los datos en lotes de 32 (tamaño de batch 32). En el caso de los entrenamientos realizados con los Subsets 1 y 2, el porcentaje de datos reservados para test ha sido del 30%; mientras que en los entrenamientos con el Subset 3 (mucho mayor en número de ficheros) se reservó el 20% de datos para test.

En el caso de los entrenamientos que consideraban las clases excluyentes (entrenamientos con los Subsets 1 y 2), para la evaluación tras cada epoch se tomaban los datos de test y se usaba el modelo entrenado para realizar una predicción de la patología. La precisión resultante corresponde con el número de predicciones correctas respecto del total.

En el caso de los entrenamientos con el Subset 3 (en los que permitimos clases concurrentes), utilizamos las curvas ROC y su AUC como parámetro de precisión del modelo.

Todos los resultados y figuras obtenidos en las distintas pruebas realizadas pueden consultarse en su totalidad en el [Anexo I](#).

4.2 Entrenamiento del Modelo 0 con el Subset 1

Entrenamos durante 50 epochs con los datos del Subset 2 y obtuvimos una precisión de más del 96% pasados unos 10 epochs de entrenamiento, y de más del 98% hacia el final de este. También pudimos apreciar como la tendencia del coste se mantenía descendente entre epochs. Como el propósito de este primer modelo era actuar como verificación del proceso de extracción de los datos a partir de los ECGs y depurar errores una vez se consiguió una precisión razonable se pasó a trabajar con el resto de conjuntos de datos con mayor complejidad.

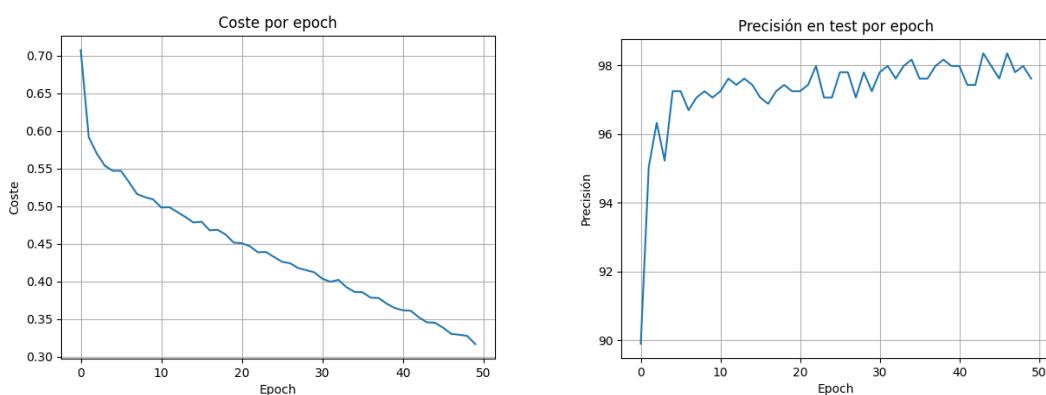


Figura 4.2.1: Evolución del coste y la precisión en test del Modelo 0 durante el entrenamiento con el Subset 1.

4.3 Entrenamiento del Modelo 0 con el Subset 2

Entrenamos el modelo durante 50 epochs con los datos del Subset 2, pudiendo observar que teníamos una precisión de más del 90% pasados unos 10 epochs de entrenamiento.

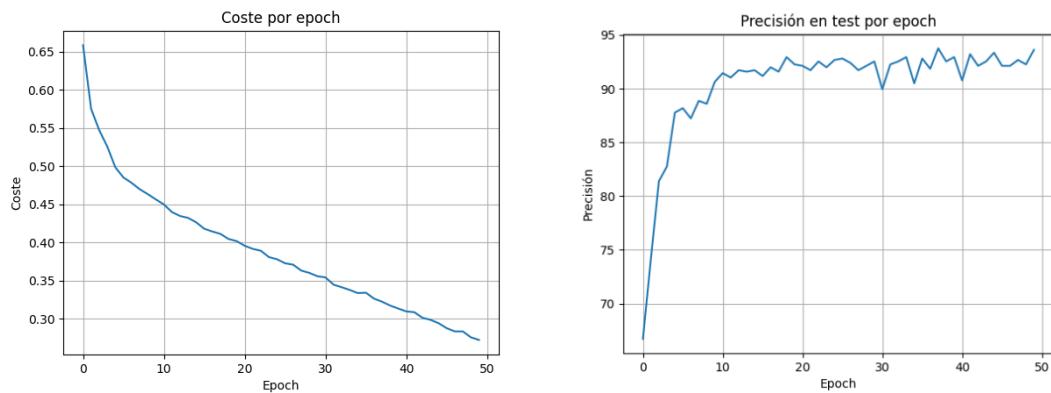


Figura 4.3.1: Evolución del coste y la precisión en test durante el entrenamiento del Modelo 0 con el Subset 2.

4.4 Entrenamiento del Modelo 0 con el Subset 3

En este caso los datos de entrenamiento eran los del Subset 3 y no se consideraban las clases excluyentes. Tras entrenar el modelo durante 20 epochs, se calcularon las AUCs de cada clase y la AUC media. Los resultados obtenidos se muestran en la Figura 4.4.1, donde se puede ver como las cuatro clases se detectan con una alta precisión. La AUC media del modelo fue de 0.9733, teniendo el modelo entrenado un total de 927.628 parámetros.

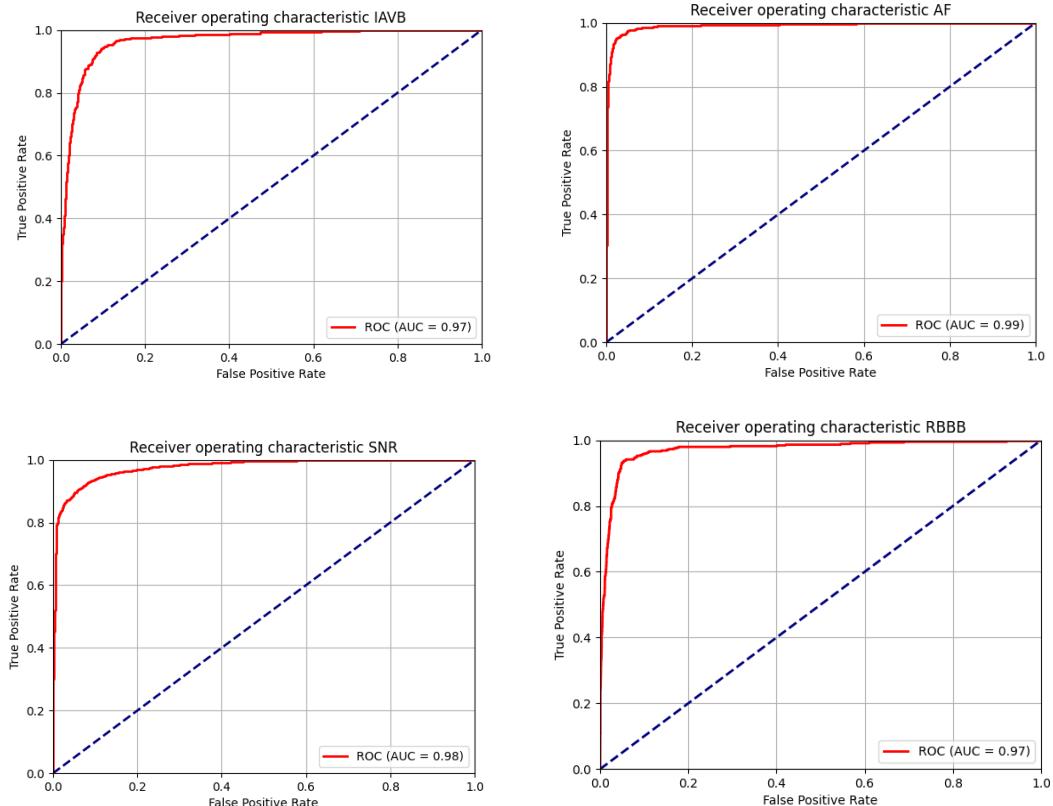


Figura 4.4.1: Curvas ROC y AUC de las 4 patologías a clasificar por el Modelo 0.

4.5 Entrenamiento de los Modelos 1, 2, 3 y 4 con el Subset 3

Todas las variaciones del modelo base se entrenaron durante 20 epochs con los mismos datos del Subset 3, para después ser evaluados con unos datos de test comunes a todos los modelos.

Los resultados de AUC media obtenida respecto al número de parámetros empleado por cada modelo puede verse en la Figura 4.5.1. Se puede observar que en la mayoría de los casos aumentar la complejidad del modelo no reporta mejoras en la AUC media, como puede verse al pasar del Modelo 2 al 3.

Otro caso en el que el aumento de complejidad no reporta mejoras de precisión es en los Modelos 4_32_2.5 y 4_16_2.5. En este caso, sin embargo, esto puede ser debido en parte a una reducción brusca de dimensiones que sufren los datos en la última capa lineal. Tampoco reducir mucho el número de capas parece mejorar las prestaciones, ya que podemos ver como en el Modelo 4_16_1.5 la AUC es más baja que la del Modelo 0, e incluso menor que la del Modelo 4_16_2.5.

De las variaciones realizadas las únicas dos que mejoran el modelo de partida son: aumentar muy ligeramente el número de canales (partiendo de 32 y con un factor de crecimiento 1.5), Modelo 4_32_1.5; y añadir una capa lineal final que permita una reducción más progresiva de los canales, Modelo 2.

En vista de que el número de parámetros es muy parecido en las dos variaciones, nos quedaríamos con la que tiene mayor AUC media en las pruebas, es decir con el Modelo 2.

Las tablas con los valores exactos de AUC, parámetros y curvas ROC puede consultarse en el [Anexo I](#).

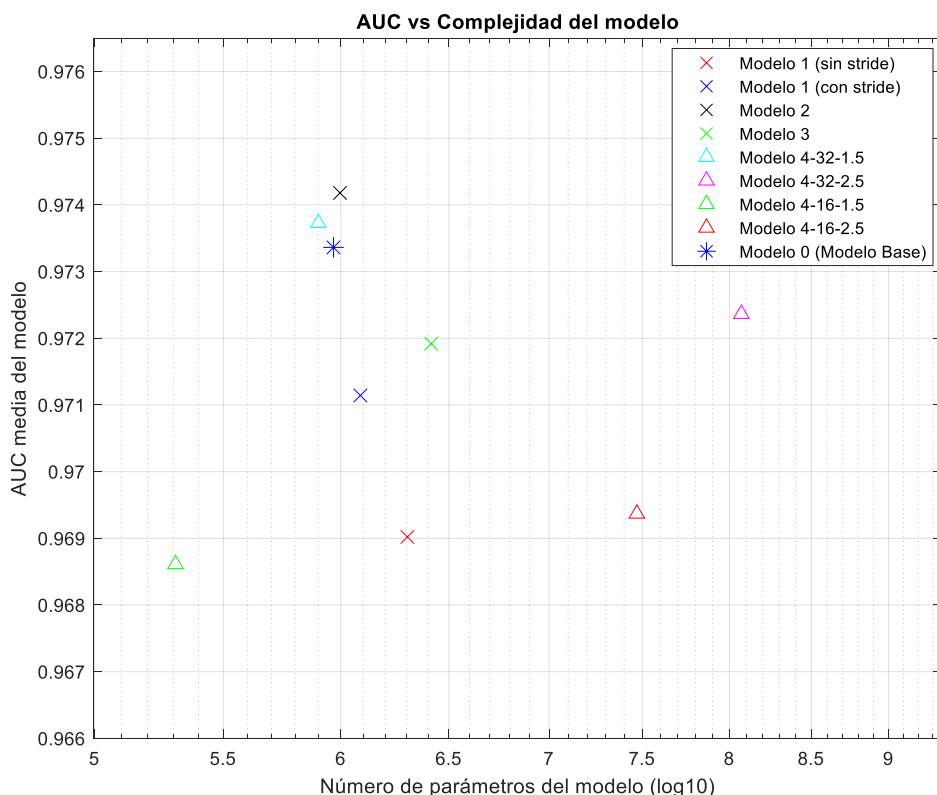


Figura 4.5.1: AUC obtenida frente a número de parámetros para cada versión del modelo.

CAPÍTULO 5. CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Este TFG ha abarcado distintas disciplinas relacionadas con el diagnóstico automático de patologías cardíacas. En primer lugar, se ha incidido en la fisiología del corazón, en las características de la señal ECG, y en cómo esta puede servir para identificar ciertas condiciones médicas.

A continuación, se procedió a procesar estas señales ECG con el fin de extraer las características que permitían distinguir señales procedentes de un corazón sin patologías de uno con alguna de las patologías con las que se trabajó. Este ámbito del procesado de señales abarcaba todo el tratamiento de los ECG desde los ficheros originales hasta las imágenes finales resultantes. Estas imágenes han mostrado ser una buena representación de las características, ya que, como se indica en el Capítulo 4, las tasas de acierto (en el caso de clases excluyentes) y la AUC (en el caso de no excluyentes) han resultado ser muy altas. Esto implica que el método de procesado empleado con el ECG organiza y dispone la información de una manera que después permite a la red neuronal extraer las características que identifican las diferentes patologías.

Finalmente, este trabajo ha realizado una aproximación al campo de las redes neuronales, en concreto de las redes convolucionales, para poder realizar el sistema de clasificación automática de patologías. Este último aspecto ha resultado ser uno de los más costosos, dada la falta de experiencia y la complejidad de este ámbito. A pesar de esto, se ha conseguido diseñar un modelo sencillo que cumple la función de clasificación con buenos parámetros de acierto (AUC), e incluso realizar diferentes variaciones para observar los cambios que genera la estructura de una red en su complejidad interna y su desempeño.

Como resultados concretos podemos destacar que se han obtenido precisiones en la clasificación superiores al 90% entrenando el Modelo 0 con los Subsets 1 y 2 al considerar las clases excluyentes; y valores de AUC media de 0.97 al entrenar dicho modelo con el Subset 3 y clases no excluyentes.

También se han variado tanto las dimensiones de las capas del modelo inicial como el número de capas, dando lugar a los Modelos 1, 2, 3 y 4. Al entrenar estos modelos con el Subset 3 hemos observado que, en la mayoría de los casos, los modelos con mayor número de capas y/o canales resultan tener menores valores de AUC media, y que los únicos modelos que mejoraban las prestaciones del Modelo 0 eran el Modelo 2 y el Modelo 4_32_1.5.

Todo el proceso de realización de este TFG en forma de diario, así como el código empleado, puede encontrarse recopilado en el siguiente repositorio (Anexo III):
https://github.com/755224unizar/TFG_Notes

5.2 Líneas futuras

Durante la realización de este trabajo se han ido tomando decisiones que limitaban ligeramente la complejidad del mismo para ajustarse al alcance de un TFG. Algunas de estas decisiones implicaron ajustes en aspectos que pueden mejorarse en un futuro.

Algunos de estos aspectos a mejorar podrían ser los siguientes:

- Utilizar datos de un mayor número de patologías. Esto nos permitiría aumentar la capacidad de diagnóstico de la red y comparar sus prestaciones con otros participantes del Computing in Cardiology 2020, los cuales diseñaron sistemas de diagnóstico para 27 patologías distintas.
- Trabajar con otras arquitecturas de red neuronal que presenten ventajas de algún tipo frente a las redes convolucionales en esta tarea de clasificación (mayor robustez, eficiencia, etc.) como los recientes modelos de tipo Transformer
- Además de clasificar los ECG en uno de los posibles diagnósticos, tratar de identificar las características de la señal cardíaca que han permitido realizar esos pronósticos y generar una explicación del proceso. Esto podría permitir descubrir relaciones ocultas en la señal ECG, mejorando el conocimiento de las patologías estudiadas.
- Variación de algunos de los parámetros del tratamiento de las imágenes, como el tamaño de las ventanas temporales con las que se generan las imágenes o el número de ventanas temporales que procesa la red neuronal (en nuestro caso solo se probó con 10 ventanas). De esta manera podríamos estudiar los efectos de estos cambios en el sistema de diagnóstico.

Bibliografía

- [1] OMS, Organización Mundial de la Salud. www.who.int. 9 de 12 de 2020. <https://www.who.int/es/news-room/fact-sheets/detail/the-top-10-causes-of-death>.
- [2] Dr. Dale Dubin. «Electrocardiografía Práctica. Lesión, Trazado e Interpretación.» En *Electrocardiografía Práctica. Lesión, Trazado e Interpretación*, 1-150. EMALSA. INTERAMERICANA. División de McGRAW-HILL, 1972.
- [3] Mr. Hrishikesh Limaye, Mrs. V.V. Deshmukh. «ECG Noise Sources and Various Noise Removal Techniques.» *International Journal of Application or Innovation in Engineering & Management (IJAIE)*, February 2016: 87-89.
- [4] Leif Sörnmo, Pablo Laguna. «Bioelectrical Signal Processing in Cardiac and Neurological Applications.» *Academic Press*, 2005.
- [5] Frank Rosenblatt. «The perceptron: A probabilistic model for information storage and organization in the brain.» *Psychological Review*, 1958: 386–408.
- [6] David E. Rumelhart, Geoffrey E. Hinton & Ronald Williams. «Learning Representations by back-propagating errors.» *Nature*, 1986: 533-536.
- [7] Yann LeCun, Léon Bottou, Yoshua Bengio & Patrick Haffner. «Gradient-Based Learning Applied to Document Recognition.» *IEEE*, 1998: 1-46.
- [8] Yann LeCun, Y. Bengio, & G. Hinton. «Deep learning» *Nature*, 521, 2015: 436-444.
- [9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez & I. Polosukhin. «Attention is all you need» *Advances in neural information processing systems*, 2017: 5998-6008.
- [10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair & Y. Bengio. «Generative adversarial nets». *Advances in neural information processing systems* 27, 2014.
- [11] Erick A. Perez Alday, Annie Gu, Amit Shah, Chad Robichaux, An-Kwok Ian Wong, Chengyu Liu, Feifei Liu, Ali Bahrami Rad, Andoni Elola, Salman Seyed, Qiao Li, Ashish Sharma, Gari D. Clifford, Matthew A. Reyna. «Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020.» 2020.
- [12] Elentra, Faculty of Health Sciences, Queen's University. «elentra.healthsci.queensu.ca.». https://elentra.healthsci.queensu.ca/assets/modules/ECG/normal_ecg.html (último acceso: 19 de septiembre de 2021).
- [13] Hirooki Yabe. «Median method for eliminating infrequent artifacts and identifying the signals blurred by latency jitter and uncertain occurrence.» *Behavior Research Methods, Instruments, & Computers*, 1998: 68-77.
- [14] J.P. Martinez, R. Almeida, S. Olmos, A.P. Rocha, P. Laguna. «A wavelet-based ECG delineator: evaluation on standard databases.» *IEEE Transactions on Biomedical Engineering*, April 2004: 570 - 581.
- [15] Peter R. Rijnbeek, PhD, MD, PhD Gerard van Herpen, MD, PhD Michiel L. Bots, y et al. «Normal values of the electrocardiogram for ages 16–90 years.» *Journal of Electrocardiology*, 2014: 914-921.
- [16] M. Rosenblatt. «Remarks on Some Nonparametric Estimates of a Density Function.» *The Annals of Mathematical Statistics*, 27 (3), 1956: 832–837.
- [17] Yoko Miyasaka, Marion E. Barnes, Bernard J. Gersh, Stephen S. Cha, Kent R. Bailey, Walter P. Abhayaratna, James B. Seward, and Teresa S.M. Tsang. «Secular Trends in Incidence of Atrial Fibrillation in Olmsted County, Minnesota, 1980 to 2000, and Implications on the Projections for Future Prevalence.» *Circulation*, 2006: 119–125.

- [18] Juan José Gómez-Doblas, Javier Muñiz, Joaquín J. Alonso Martín, Gustavo Rodríguez-Roca, José María Lobos, Paula Awamleh, Gaietà Permanyer-Miralda, Francisco Javier Chorro, Manuel Anguita, Eulalia Roig. «Prevalencia de fibrilación auricular en España. Resultados del estudio OFRECEPrevalence of Atrial Fibrillation in Spain. OFRECE Study Results.» *Revista Española de Cardiología*, 2014: 259-269.
- [19] E. Benjamin, P. Muntner, A. Alonso, M. Bittencourt, C. Callaway, A. Carson, A. Chamberlain, A. Chang, S. Cheng, S. Das, et al., «Heart disease and stroke statistics 2019 update: A report from the American Heart Association. », *Circulation*, vol. 139, no. 10, e56, 2019.
- [20] A. K. Tushar, A. Ashiquzzaman & M. R. Islam, «Faster convergence and reduction of overfitting in numerical hand sign recognition using DCNN», *IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, 638-641.
- [21] Suzanne Ekelund. «acute care testing.org.» Enero de 2011. <https://acute care testing.org/en/articles/roc-curves-what-are-they-and-how-are-they-used> (último acceso: 7 de septiembre de 2021).
- [22] M. Tan, & Q. Le . «Efficientnet: Rethinking model scaling for convolutional neural networks», *International Conference on Machine Learning*, May 2019: 6105-6114, PMLR.
- [23] Diederik P. Kingma, Jimmy Lei Ba. «ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.» *International Conference for Learning Representations*. San Diego, CA., 2015.