Universidad de Buenos Aires

Análisis y procesamiento para la segmentación de fonocardiogramas

Facultad de Ingeniería

Tesista: Álvaro Joaquín Gaona
Director: Dr. Pedro Arini
Co-directora: Dra. Paula Bonomini

Una tesis presentada a la Universidad de Buenos Aires de acuerdo a los requisitos de la carrera de grado de Ingeniería Electrónica en la Facultad de Ingeniería.

Enero 2019

Palabras: 15420

RESUMEN

Esta tesis presenta el analisis de un segmentador de señales de fonocardiograma basado en el conjunto de técnicas conocidas como *Deep Learning*. Se analiza la segmentación de estas señales que pueden ser de naturaleza patológica y no patológicas, mediante una Recurrent Neural Network (RNN) (*Recurrent Neural Network*). Particularmente esta red neuronal se denomina *Long Short Term Memory* (Long Short-Term Memory (LSTM)). Esta aplicación basada en una red LSTM con métricas de performance cercanas a la del estado del arte pueden ser llevadas a cabo para la implementación de un segmentador en tiempo real. Para entender el funcionamiento del segmentador, se desarrollarán conceptos básicos acerca del modelado de la unidad básica de la red LSTM y las distintas capas aplicadas para la conformación del segmentador, seguido por una breve introducción del pre-procesamiento, extracción de features y de una potencial instancia de post-procesamiento. El concepto de redes neuronales recurrentes (*recurrent neural network*) es el núcleo de esta tesis y se explica cómo a partir de esta red neuronal es posible segmentar dichas señales que son relativamente amorfas a otras señales conocidas. Entre estas se encuentra, el electrocariograma (Electrocadiograma (ECG)), donde sus ondas están bien definidas. El algoritmo necesita información previa de un set de datos equilibrado entre señales patológicas y no patológicas para el entrenamiento. Complementariamente, es vital la delineación de los ECG extrayendo de esto la onda R y el fin de la onda T. La implementación se encuentra basada en MATLAB™ y se calcula el desempeño (*performance*) del sistema, basado en métricas estándar en la literatura de la clasificación y/o segmentación.

Índice general

Capítulo 1

Presentación

1.1. Motivación

El presente trabajo se hace en el marco de una tesis de grado de la *Facultad de Ingeniería de la Universidad de Buenos Aires* (FIUBA) en conjunto con los investigadores Dr. Pedro David Arini y la Dra. María Paula Bonomini.

Pedro y Paula se encuentran actualmente en el *Instituto Argentino de Matemática* (Instituto Argentino de Matemática (IAM)) del *Consejo Nacional de Investigaciones Científicas y Técnicas* (Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)) y en el *Instituto de Investigaciones Biomédicas "Alberto Sols"* (Instituto de Investigaciones Biomédicas (IIBM)). Pedro se especializa en el campo de *"Procesamiento digital de señales aplicado al cálculo y modelado estadístico de la actividad eléctrica cardíaca"* y Paula en el campo de *"Modelos matemáticos y tratamiento digital de la señal electrocardiográfica para predicción de riesgo cardíaco"*.

Por otro lado, el análisis de fonocardiogramas no es común en ambientes clínicos como hospitales. Dicho esto conseguir datos asociados de pacientes no es tarea sencilla. Por ende, el IAM se encuentra trabajando en el diseño e implementación de un equipo capaz de realizar la adquisición de estas señales con bajo error para realizar una base de datos propia dedicada a la investigación.

Más aún, las técnicas de análisis automático de fonocardiogramas no han sido desarrolladas como el análisis de electrocardiogramas. Así, el IAM tiene pendiente como tema de investigación la segmentación de fonocardiogramas para luego que sea ésta una base en la detección automática de enfermedades cardiovasculares.

Finalmente, este tema va a seguir llevado a cabo en áreas de estudio superiores con el objetivo de mejorar y aportar a la comunidad científica una propuesta nueva e innovadora.

1.2. Objetivos

La propuesta de investigación tiene como enfoque proponer una alternativa de segmentación diferente al estado del arte. Bajo esta disposición, se definen objetivos generales al tema de investigación y objetivos específicos del trabajo presentado en este documento.

El desempeño de la segmentación es fundamental en todo trabajo relacionado a la detección o estimación. De esta manera este trabajo ha sido llevado a cabo con ese objetivo. Por otro lado, se ha visto en la implementación de David Springer *et al.* [10] que el algoritmo de etiquetamiento comete errores, lo cual requiere un ajuste ad hoc de los parámetros. Ésto dispara la necesidad de un nuevo algoritmo y/o la extracción de marcas de otras señales complementarias.

De la mano del desempeño y del error de segmentación se encuentra la necesidad de ampliar la base de datos de entrenamiento.

Generales

- Mejorar el desempeño de la segmentación respecto al estado del arte.
- Proponer un algoritmo de etiquetamiento automático a partir de marcas del ECG.
- Proponer nuevas marcas asociadas a otras señales complementarias al Phonocardiogram (PCG) (*phonocardiogram*).
- Ampliar la base de datos utilizada para la segmentación.
- Realizar una implementación en tiempo real del segmentador propuesto.
- Proponer un clasificador para la detección de señales patológicas.
- Plantear nuevas lineas de trabajo que permitan continuar con el trabajo de investigación.

Específicos

- Extraer marcas de ECG asociados a un PCG.
- Acondicionar las señales de PCG.
- Implementar algoritmo de extracción de cuadros (*frames*) de las señales de PCG.
- Implementar red neuronal LSTM como clasificador.
- Implementar un entrenamiento del modelo de segmentación bajo el concepto de *Cross-validation*.
- Comparar métricas de desempeño entre los distintos métodos del estado del arte.
- Proponer mejoras a futuro.

1.3. Lineamientos generales de la tesis

- **Capítulo 1**: Se hace una presentación del trabajo. Explica el contexto, motivación y objetivos.
- **Capítulo 2**: Se da un marco teórico de la naturaleza de la señal y de la fisiología asociada al tema, dando un contexto de la biología humana involucrada.
- **Capítulo 3**: Se hace mención a las distintas bases de datos con fonocardiogramas, la comparativa entre ella y el por qué de la elección de una de ellas.
- **Capítulo 4**: Se explica la etapa de pre-procesamiento necesaria para el acondicionamiento de las señales (filtros, etiquetas, ente otras cuestiones).
- **Capítulo 5**: Se explica la etapa de procesamiento del trabajo, abordando principalmente la teoría de la extracción de atributos de los fonocardiogramas.
- **Capítulo 6**: Se aborda el tema de *deep learning* definiendo el marco teórico de los modelos y técnicas aplicados en este trabajo.
- **Capítulo 7**: Se muestra la implementación de la solución al problema en cuestión, mostrando algoritmos implementados, arquitectura del modelo.
- **Capítulo 8**: Se intenta ilustrar los resultados del trabajo y proponiendo algunos temas de discusión a partir de estos resultados obtenidos.
- **Capítulo 9**: Se toman conclusiones del trabajo, proponiendo mejoras a fin de igualar o superar el rendimiento de otros algoritmos del actual estado del arte.

Capítulo 2

Introducción

2.1. Actividad eléctrica del corazón

El corazón esta formado, su mayor parte, por tejido muscular que consta de células altamente diferenciadas para la función contráctil que forman las paredes de las cámaras auriculares y ventriculares. El resto está organizado en estructuras específicas implicadas en la generación y propagación de impulsos eléctricos. La actividad eléctrica comienza a nivel de células que generan de manera espontánea potenciales de acción, llamadas *células marcapaso*. Esta propiedad se encuentra más o menos desarrollada en diversas zonas del tejido especializado. El grupo celular que posee la frecuencia intrínseca más elevada es el *nódulo sinoauricular* (Keith, Flack. 1907 [1]), que constituye el marcapaso primario del corazón. Su frecuencia de descarga *in situ* es de alrededor de 1.25 Hz (75 impulsos por minuto). Situado a nivel de la unión de la vena cava superior y la aurícula derecha, está formado por un grupo de miocitos ramificados que se conectan con las células del miocardio auricular. De importancia fisiológica es el hecho de que las células del nódulo sinusal están sometidas a una acción de estiramiento que puede modular su frecuencia de descarga. Este estiramiento sería inducido por la presencia de la arteria del nódulo sinusal y por la distensión de las paredes de la aurícula durante el arribo de sangre proveniente de las venas cavas.

El *nódulo auriculoventricular*, localizado en el piso de la aurícula derecha, entre el seno coronario y la válvula tricúspide, contiene células similares a las del nódulo SA, pero su frecuencia intrínseca de descarga es menor. La parte superior contiene células de transición. Éstas generan un potencial lento, de poca amplitud, que se conduce a muy baja velocidad. La lenta conducción del impulso a través del nódulo AV engendra un retardo entre la activación auricular y la ventricular, lo que permite una contracción secuencial de estas cámaras. Las fibras inferiores del nódulo AV convergen y forman el *haz de His*. Las ramas del haz de His terminan en las *fibras de Purkinje*, cuyas propiedades eléctricas les confieren la mayor velocidad de conducción. Estas células forman una red subendocárdica, a partir de la cual transmiten el influjo eléctrico al músculo ventricular. El sistema de Prukinje asegura la casi sincrónica activación de toda la masa ventricular desde el endocardio al epicardio y desde el ápex hacia la base.

2.1.1. El potencial de acción cardíaco

El potencial cardíaco se define como la polarización y despolarización de la membranas de las células miocárdicas. Esto produce una diferencia de potencial trasmembranal que sirve para transmitir información entre células. Es producido por la llegada de un impulso eléctrico a la célula de modo que produce una redistribución de carga que genera estas diferencias de potenciales. Estos cambios derivan de la apertura y cierre subsecuente de los canales iónicos según una cinética particular para cada uno. El fenómeno de excitación dura alrededor de 0.5 ms en la fibra nerviosa, 5-10 ms en la fibra muscular esquelética y 250 ms en la fibra cardíaca ventricular. La bomba de Na^+-K^+ no desempeña ningún papel directo en la generación del potencial de acción, ya que su función se limita al mantenimiento de los gradientes de Na^+ y K^+, mientras que el gradiente de Ca^{2+} es mantenido por las bombas de Ca^{2+} situadas en la membrana celular y en la del retículo sarcoplasmático.

Este fenómeno se encuentra compuesta de 5 fases distinguibles que ayudan a mantener el potencial de acción. Cabe aclarar que este potencial de acción son las células del miocardio ya que las células marcapasos poseen un funcionamiento claramente diferente. En la Figura 2.1 se ilustra estas etapas.

- **Fase 0**: **corriente de sodio (I_{Na})**. La fase 0 constituye la despolarización rápida del potencial de acción. En las células ventriculares, así como en las células auriculares y de la red de His-Purkinje, esta rápida despolarización depende de la apertura de los *canales rápidos de* Na^+ similares a los presentes en el músculo esquelético y el nervio. Esta corriente de Na^+ (I_{Na}) es designada como rápida porque exhibe cinética de activación e inactivación muy rápida en comparación con otras corrientes. Debido a que los canales rápidos de Na^+ son operados por voltaje, luego de la despolarización la compuerta de *activación* se abre y I_{Na} llega a su pico en aproximadamente 1 ms. Luego de la activación pico de I_{Na} la corriente comienza a disminuir debido al cierre de los canales mediante el proceso de *inactivación*, que en este caso se debe a la interacción del extremo aminoterminal de la proteína que forma el canal con el poro de éste (inactivación de tipo N). Por lo tanto, I_{Na} es prácticamente cero en pocos milisegundos. Una vez que el canal se encuentra en estado inactivo no puede volver a abrirse a menos que se *reactive*. La reactivación o remoción de la inactivación sólo es posible a potenciales negativos y luego de un lapso determinado. El proceso depende del voltaje y del tiempo, tanto para la activación como la inactivación. Luego de que la mayoría de los canales de Na^+ se han inactivado durante la fase 0, el potencial de membrana debe repolarizarse para que los canales se reactiven y se encuentren disponibles para volver a abrirse y disparar un nuevo potencial de acción.
  Para que ocurra la rápida despolarización de la fase 0, la célula debe ser despolarizada hasta el valor de voltaje necesario para abrir los canales rápidos de Na^+ (aproximadamente -70 mV). Cuando los canales de Na^+ comienzan a abrirse, el Na^+ empieza a fluir a favor de su gradiente electroquímico hacia el interior de la célula. Esto hace que la célula se despolarice más, lo que abre canales de Na^+ adicionales. Una vez que se abren suficientes canales de Na^+ el proceso se hace regenerativo origina la rápida despolarización de la fase 0 y el potencial de membrana se acerca velozmente al potencial de

equilibrio para el Na^+ ($E_{Na} = +60$ mV). El voltaje al cual se abre un número de canales de Na^+ suficiente para iniciar el potencial de acción se representa el *umbral* necesario para disparar cualquier potencial de acción. A partir de este umbral, la corriente I_{Na} comienza a aumentar para luego declinar, una vez que va alcanzando el potencial de equilibrio para el Na^+ (E_{Na}). El potencial de membrana no llega nunca a E_{Na} por diversas razones: 1) a medida que el potencial de membrana se acerca a E_{Na}, la fuerza impulsora que promueve el ingreso de Na^+ disminuye; 2) los canales de Na^+ se inactivan inmediatamente luego de su apertura; 3) las corrientes repolarizadoras se empiezan a activar durante la etapa final de la fase 0. Por lo tanto, durante la fase 0, el potencial de membrana más positivo que se alcanza es de aproximadamente +35 mV, lo que determina un cambio de potencial de 110 a 120 mV, en solamente 2 ms.

- **Fase 1**: **corriente transitoria hacia afuera (I_{to})**. La fase 1 del potencial de acción cardíaco consiste en una rápida repolarización transitoria del potencial de membrana que sigue inmediatamente a la fase 0. La fase 1 varía de una especie a otra y además en diferentes regiones del corazón de una misma especie. También se ha descrito un ingreso de Cl^- durante la fase 1. La corriente responsable de la fase 1 se denomina *transitoria hacia afuera* (*transient outward*) y se la expresa comúnmente como I_{to}. Esta corriente se produce por la apertura de canales dependientes de voltaje y del tiempo. Además, estos canales presentan inactivación luego de pasar por el estado abierto. Estas características hacen que I_{to} sea transitoria. El umbral para su activación es de aproximadamente -30 mV. La rápida activación de I_{to} que genera la fase 1 es seguida por la inactivación que corresponde a la meseta (fase 2).
  La corriente transitoria hacia afuera esta compuesta por dos corrientes, I_{to1} e I_{to2}. La primera es una corriente de K^+ que es independiente de la concentración interna de Ca^{2+} para su activación y es sensible al bloqueador 4-aminopiridina. El segundo componente, depende del Ca^{2+} intracelular para su activación y el ion permanente es Cl^- es más importante durante la primera porción de la meseta del potencial de acción.

- **Fase 2**: **corriente de calcio (I_{Ca})**. La fase 2 corresponde a la meseta del potencial de acción cardíaco. Sigue a la repolarización temprana de la fase 1 y es un período durante el cual el valor del potencial de membrana se mantiene relativamente constante durante varios milisegundos. La presencia de esta prominente meseta es responsable de la larga duración del potencial de acción en las células cardíacas, el cual es la mayor diferencia entre los potenciales de acción de las células del cardíacas y las esqueléticas o nerviosas. La meseta es causada por un balance entre corrientes catiónicas hacia el interior de la célula (despolarizantes) y corrientes catiónicas hacia el exterior de ésta (repolarizantes). La principal corriente despolarizante de la fase 2 es una *corriente de Ca^{2+} (I_{Ca})*. La corriente de Ca^{2+} de la fase 2 se produce a través de canales operados por voltaje y que presentan inactivación. La cinética de inactivación de estos canales es más lenta que la de los canales rápidos de Na^+ y, por lo tanto, el tiempo durante el cual permanecen abiertos es más prolongado. Debido a esta característica se los denomina canales de Ca^{2+} de *tipo L* (por *long lasting*) o de larga duración (corriente I_{Ca-L}). Existen

también en las células cardíacas canales de Ca^{2+} con cinéticas de activación e inactivación más rápidas que los canales L. Son los llamados canales de Ca^{2+} de tipo T (por *transient*) o de apertura transitoria (corriente I_{Ca-T}).

- **Fase 3**: **corriente tardía de potasio (I_K)**. La fase 3 corresponde a la repolarización final del potencial de acción y se debe a la corriente generada por la lenta apertura de los canales de K^+ tardíos (I_K). La fase 3 es causada primariamente por el desbalance de las corrientes que estaban relativamente equilibradas durante la fase 2. Debido a la inactivación de los canales de Ca^{2+}, I_{Ca-L} disminuye mientras que I_K aumenta. Esta corriente es generada por los canales de K^+ operados por voltaje, pero que carecen de inactivación. Son canales de activación muy lenta que se activan gradualmente durante la despolarización sostenida de la fase 2 y que se encuentran plenamente abiertos durante la fase 3. A medida que el potasio sale de la célula a través de I_K, el potencial de membrana se repolariza y alcanza valores en los cuales se suprime la inhibición de I_{K1}. Por lo tanto, durante la fase final de la fase 3, I_{K1} contribuye a repolarizar el potencial de membrana y a regresar al potencial de reposo.

- **Fase 4**: **corriente de potasio con rectificación hacia adentro (I_{K1})**. El potencial de reposo de las células cardíacas es determinado fundamentalmente por la distribución asimétrica del ion K^+ y por la permeabilidad al potasio. El potencial de reposo de las células ventriculares y auriculares es determinado casi exclusivamente por la corriente de K^+ a través de los canales conocidos como *rectificadores hacia adentro* (I_{K1}), también denominados rectificadores anómalos. La característica más notable de I_{K1} es que exhibe rectificación hacia adentro. Es decir, deja pasar más corriente hacia adentro de la célula que hacia afuera. Sin embargo, la corriente de K^+ hacia afuera es la que atraviesa este canal en forma fisiológica, ya que el potencial de membrana raramente se hiperpolariza por debajo del potencial de equilibrio para el K^+ (E_K). A medida que la célula se despolariza desde cerca de su potencial de reposo hacia potenciales progresivamente más despolarizados, la corriente hacia afuera primero se incrementa para luego disminuir con la sucesiva despolarización. Por lo tanto, la contribución de I_{K1} a la repolarización tiende a ser menor a potenciales cercanos a la meseta (fase 2) y mayor a medida que el potencial de membrana se acerca al potencial de reposo.
  El potencial de resposo en el miocardio varía según el tipo celular: es de -60 mV en las células nodales, de -80 mV en las auriculares y ventriculares y de -90 mV en los otros tejidos de conducción (His y Purkinje). En las células nodales la densidad de canales de K^+ generadores de I_{K1} (número de canales por μm de membrana) es menor que en el miocardio ordinario. En consecuencia, la conductancia al K^+ es más baja y la permeabilidad a los otros iones (Na^+ y Cl^+) pesa relativamente más en la determinación del potencial de reposo que se aleja de E_K.

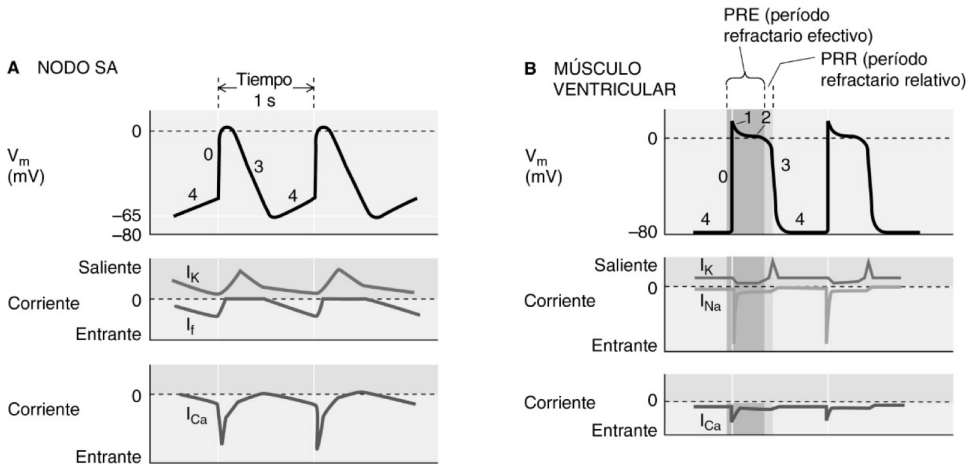


Figura 2.1: Fases de los potenciales de acción cardíacos. Los registros de esta figura son ideales. I_K, I_{Na}, I_{Ca} e I_f son corrientes a través de canales de K^+, Na^+, Ca^{2+} y catiónicos no selectivos, respectivamente. Figura extraída de [4]

2.1.2. Propagación del impulso

Existen variaciones regionales en la velocidad de conducción del impulso cardíaco.

El retardo de conducción no es proporcional a la distancia recorrida, lo cual indica que la velocidad de propagación no es uniforme. En contraste con el miocardio ordinario y la vía de conducción rápida, los potenciales de acción de las células nodales presentan menor amplitud y una fase 0 más lenta porque la corriente de Ca^{2+} constituye el elemento predominante de la excitación. El mayor retardo de conducción se observa entre el nódulo AV y el haz de His, a pesar de que sólo unos pocos milímetros separan a estas dos estructuras. Esto se debe al hecho de que un potencial lento genera corrientes locales débiles y, por lo tanto, es menos apto para la conducción. Además, las uniones intercelulares comunicantes o en hendidura son menos numerosas en el nódulo AV. Aunque cada miocardiocito está rodeada de una membrana aislante, el miocardio se comporta desde el punto de vista eléctrico como una célula única gigante (sincicio funcional) debido a la presencia de los discos intercalares. Estos discos son estructuras que separan las células vecinas e incluyen regiones especializadas en las que las membranas adyacentes se adosan y que contienen çanales"de gran diámetro. Estos permiten el paso de iones y moléculas de bajo peso molecular y constituyen vías de baja resistencia eléctrica. Estos canales (no selectivos) conducen la corriente del circuito local, de manera que el flujo eléctrico que parte del nódulo sinusal se propaga a toda la masa del tejido.

2.2. Actividad mecánica

El corazón posee una función de bomba, la cual se cumple mediante la contracción y relajación rítmica del músculo cardíaco que lo forma. Esta formado por dos corazones, el corazón izquierdo y el derecho. El corazón derecho impulsa sangre venosa que llega luego de haber atravesado los diferentes órganos. Esta desemboca en la aurícula derecha por dos grandes venas: cava superior y cava inferior. Pasa a

través de la válvula tricúspide al ventrículo derecho y éste la eyecta por la arteria pulmonar al circuito menor; allí se produce la hematosis. La sangre ya oxigenada y con una presión parcial de dióxido de carbono (P_{CO_2}) normal regresa por las venas pulmonares a la aurícula izquierda. Desde ella pasa a través de la válvula mitral al ventrículo izquierdo, el cual la expulsará por la aorta a toda la economía. Las válvulas auriculoventriculares, mitras y tricúspide, aseguran la dirección del flujo sanguíneo al evitar que durante la sístole la sangre que contienen los ventrículos refluya a las aurículas. Las válvulas sigmoideas, aórtica y pulmonar, también aseguran la dirección del flujo al impedir que la sangre refluya desde estos vasos a los ventrículos durante el intervalo diastólico.

A pesar que los dos ventrículos poseen un volumen interno similar y expulsan igual cantidad de sangre, el derecho debe generar unos $15 - 20$ mmHg de presión par abrir la válvula sigmoidea pulmonar, mientars que el izquierdo debe generar aproximadamente 80 mmHg para abrir la sigmoidea aórtica. La fuerza que cada uno de estas cavidades deben hacer explica los grosores de sus paredes. Cada fibra miocárdica realiza la misma fuerza pero el ventrículo izquierdo posee una mayor cantidad de estas fibras.

Los ventrículos de un adulto promedio expulsan aproximadamente 50 ml por latido. Sin embargo, dejan un volumen residual sin esxpulsar, que es igual o ligeramente inferior al expulsado. Por lo tanto, el volumen ventricular antes de la eyección es de unos 70 a 80 ml. Este volumen se denomina *volumen diastólico final* (VDL) y el porcentaje que se expulsa de este volumen se conoce como *fracción expulsada* (FE). Éste es de aproximadamente $60 - 70\,\%$.

Ciclo cardíaco

El ciclo cardíaco un proceso continuo y periódico, de frecuencia variable de acuerdo a las necesidades del cuerpo humano. Este ciclo variará su volumen de eyección, su frecuencia cardíaca dependiendo de eventos externos al sistema cardiovascular.

Fase isovolumétrica sistólica. Llamada anteriormente fase isométrica sistólica, pasó a denominarse isovolumétrica al apreciarse que durante ella se producían cambios en la forma de cavidades, aunque no en su volumen. El aumento de la presión intrventricular que se origina cuando comienzan a contraerse las fibras ventriculares origina el cierre de las válvulas auriculoventriculares. Como las válvulas se encuentran cerradas, el volumen ventricular no se modifica y por este período se denomina isovolumétrico. La presión intraventricular izquierda asciende desde aproximadamente 100 mmHg a 80 mmHg en menos de una décima de segundo. Esto proporciona una velocidad media de desarrollo de la presión intraventricular izquierda de alrededor de 700 $\frac{mmHg}{s}$. Cuando las presiones intraventriculares alcanzan los niveles de las presiones en la aorta o en la arteria pulmonar las válvulas sigmoideas se abren y esto marca el fin del periódo isovolumétrico.

Fase de eyección. Esta fase comienza al abrirse la válvula sigmoidea correspondiente y finaliza al cerrarse ésta.

Durante este período cada uno de los ventrículos eyecta aproximadamente $50-60$ ml. La eyección de la sangre esta dada por la contracción de las fibras del miocardio. Las presiones medias con las que debe realizar fuerza los ventrículos son de 100 mmHg y 16 mmHg para el el izquierdo y derecho respectivamente.

Fase isovolumétrica diastólica. Una vez que la expulsión finaliza, y que la presión en la aorta o la pulmonar iguala o supera a la existente en el ventrículo izquierdo o el derecho, las válvulas sigmoideas se cierran. Como las válvulas auriculoventriculares aún permanecen cerradas, la presión intraventricular disminuirá sin cambios de volumen. La presión descenderá de forma isovolumétrica hasta que la presión en las aurículas supere a la ventricular y se abran las válvulas auriculoventriculares.

Fase de llenado. La apertura de las válvulas auriculoventriculares señala el inicio de la fase de llenado, la cual se complementará con la contracción auricular, la cual dará por finalizado a esta fase.

En la Figura 2.2 se ilustra el ciclo cardíaco en un gráfico de presión-volumen donde queda definido claramente los diferentes hitos del ciclo.

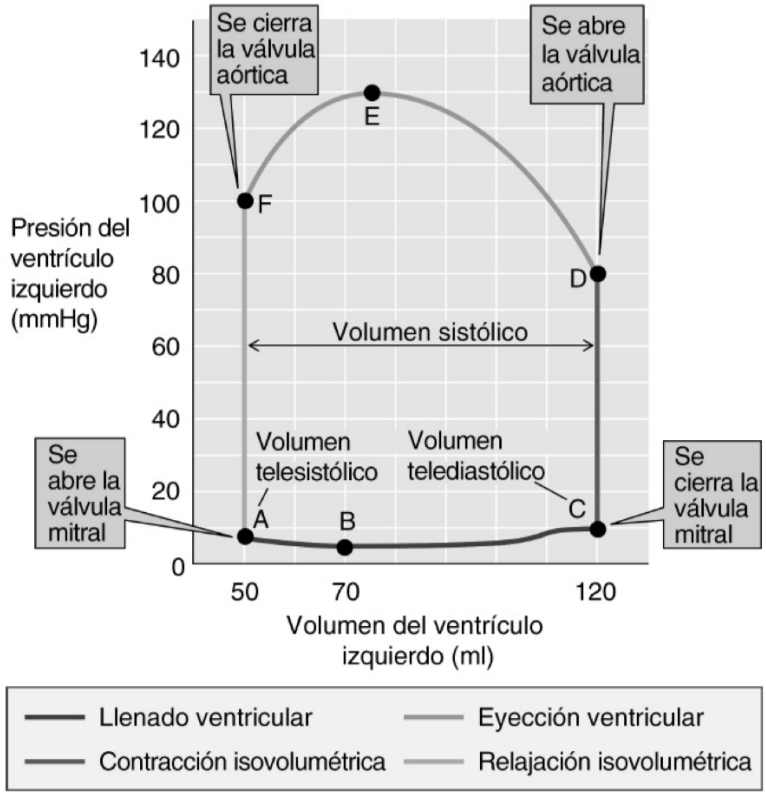


Figura 2.2: Curva de presión-volumen del ventrículo izquierdo. Figura extraída de [4]

2.3. El Fonocardiograma

La actividad mecánica del corazón genera movimientos vibratorios en las diferentes estructuras cardíacas que se propagan hacia la superficie torácica. Algunas de esas vibraciones son identificadas por el oído humano y constituyen los ruidos cardíacos. Los movimientos vibratorios de menor frecuencia, detectados por la visualización o por la palpación de expansiones o retracciones en determinadas regiones de la superficie, configuran los diferentes pulsos.

2.3.1. Ruidos cardíacos

En toda la superficie torácica próxima al corazón se auscultan dos ruidos cardíacos, definidos como el primero (1^o) y el segundo (2^o) ruido. Éstos coinciden con el comienzo y el final de la sístole ventricular, respectivamente, y brindan una idea secuencial de los fenómenos que ocurren durante el ciclo cardíaco. Con menor frecuencia se pueden auscultar otros dos ruidos que se reconocen como tercero (3^o) y cuarto (4^o).

Primer ruido. El primer ruido cardíaco se produce por movimientos vibratorios generados en las válvulas auriculoventriculares (mitral y tricúspide) y en las estructuras próximas a ellas. Las vibraciones tienen lugar en el momento del cierre valvular y coinciden con el comienzo de la contracción ventricular o sístole ventricular.

En este ruido predominan vibracioens de baja frecuencia que le confieren una característica auditiva grave.

La zona donde se ausculta meojr y se obtiene el mejor registro gráfico del primer ruido es la región próxima a la punta del corazón y el área correspondiente a la base del esternón.

El registro gráfico de ese ruido (intensidad de la frecuencia vibratoria en función del tiempo), conocido como fonocardiograma, muestra un grupo central de vibraciones con frecuencia vibratoria de 120 a 142 Hz, precedido y seguido por movimientos vibratorios de frecuencias menores, entre 30 y 100 Hz. El primer ruido tiene una duración promedio de $0.1 - 0.12$.

Segundo ruido. Al cerrarse las válvulas aórtica y pulmonar se generan movimientos vibratorios que determinan la generación del 2^o ruido. La conformación anatómica de esas estructuras le da al segundo ruido características auditivas diferentes: su duración es menor, entre $0.07 - 0.10$ s y frecuencias entre $150 - 170$ Hz.

Por razones de proximidad topográfica, y si bien se puede identificar en toda la región anterior del tórax, el sitio donde se ausculta y registra mejor es a nivel del segundo espacio intercostal, tanto a la izquierda del esternón como a la derecha.

Tercer ruido. El tercer ruido cardíaco es auscultado en casi todos los niños y adolescentes sin enfermedad cardíaca; su prevalencia auscultatoria es menor a medida que aumenta la edad en individuos sanos (23 % en una población de ambos sexos entre 36 y 37 años de edad).

La presencia de este rudio está directamente vinculada a las vibraciones de la pared ventricular durante el llenado ventricular rápido.

Cuarto ruido. El cuarto ruido, también de auscultación poco frecuente en adultos sanos, coincide con la última parte del llenado ventricular, secundaria a la contracción auricular. Es precedido por la onda P del registro del electrocardiograma y sigue inmediatamente la contracción auricular. Esto se puede observar en la Figura 2.3.

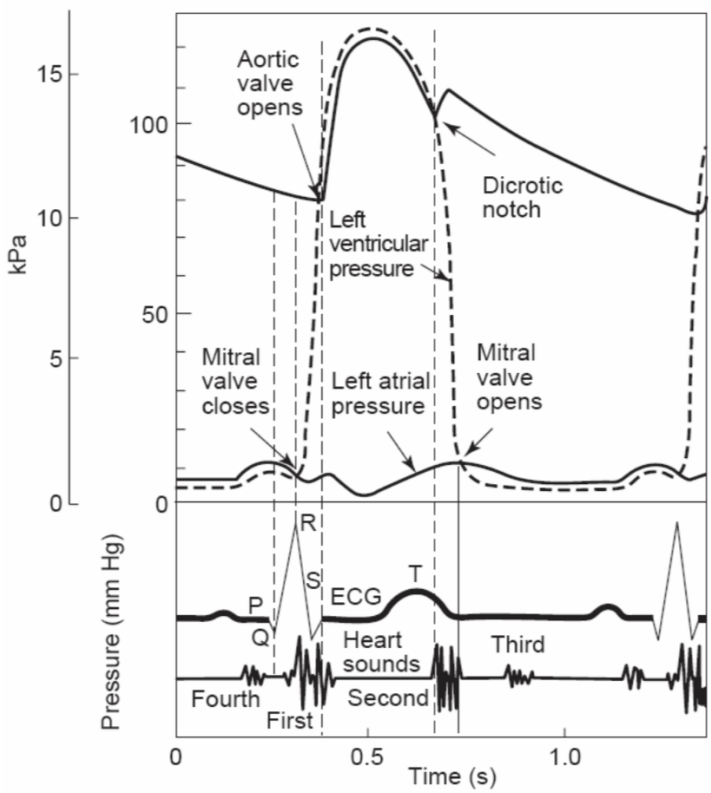


Figura 2.3: Correlación de los cuatro sonidos cardíacos con los eventos eléctricos y mecánicos del cíclo cardíaco en fase. Figura extraída de [9].

El Cuadro 2.1 resume los tipos y las características de los ruidos cardíacos. La identificación de los ruidos cardíacos normales que determinan la sístole y la diástole es la base de la auscultación. Dicho esto, resulta de suma importancia tener perfectamente caracterizada a la señal que surge de la auscultación o fonocardiografía.

RUIDO	LOCALIZACIÓN EN EL CICLO CARDÍACO	MECANISMO	IMPLICACIÓN CLÍNICA
4^{to} ruido	Presistólico, antes del 1^{er} ruido	Contracción enérgica de aurícula	Patológico, HTA, alteración de la distensibilidad del VI o del VD
1^{er} ruido	Inicio de la sístole ventricular	Cierre de válvulas AV	Fisiológico si es normal
Clics mesotelesistólicos	Mesotelesistólicos	Desplazamiento posterior hacia aurícula de valvas mitrales (posterior)	Patológicos. Aislados o antes de soplos sistólicos por insuficincia mitral
2^{do} ruido	Fin de la sístole	Cierre de válvulas sigmoideas, aórtica y pulmonar	Fisológico si es normal
Chasquidos de apertura	Inicio de la diástole, después del 2^{do} ruido. Inicio de la sístole, después del 1^{er} ruido	Apertura de la válvula AV enferma (reumática), aperturas sigmoideas enfermas	Patológicos. Estenosis mitral y tricuspídea. Estenosis aórtica y pulmonar
3^{er} ruido	Protodiastólico	Distensión ventricular súbita en la fase de llenado rápido	Fisiológico en niños y adultos jóvenes. Patológico en el resto.

VI: Ventrículo Izquierdo; VD: Ventrículo derecho; HTA: Hipertensión arterial; AV: auriculoventricular.

Cuadro 2.1: Tipos y características de los sonidos cardíacos.

2.3.2. Estado del arte

A partir de esto surgieron propuestas de procesar, clasificar y segmentar las señales derivadas del fonocardiograma. El PCG es una onda mecánica, particularmente sonido, que refleja el proceso del corazón como bomba cardíaca y en ella queda asentado el correcto funcionamiento del corazón, esto ya sea las válvulas cerrándose correctamente, el ritmo cardíaco, rigidez de las paredes cardiovasculares, entre otras características.

Aquí surgen varios desafíos. La clasificación de patologías cardíacas asociadas y segmentación de la señal en sus sonidos fundamentales. Todo esto en la práctica, al realizar la adquisición de estas señales se encuentra contaminado de distintos factores, externos al fonocardiograma. Estas fuentes de ruido pueden ser de naturaleza eléctrica, como la tensión de línea de $50 - 60$ Hz (dependiendo de la zona geográfica) o mecánica como el habla. Otros ejemplos contaminación es la respiración (de baja frecuencia), produce que la línea isoeléctrica se perciba como oscilante. Por otro lado, puede haber presencia de ruido muscular.

Hasta la actualidad varias personas han trabajado con estos problemas, ya sea tanto clasificación como segmentación. Liang et al. en [2], han trabajado con algoritmos determinísticos basados en la detección de picos con la energía y envolvente de Shannon. Esto ha mostrado una performance interesante pero ante señales muy corruptas puede haber picos que interfieran con la detección y reduzcan

la eficiencia de detección de los sonidos fundamentales. Luego, un segundo trabajo en [3] se utilizó la descomposición de wavelet y la energía de Shannon para detectar los picos en cada bloque y realizar la segmentación, así obteniendo una vez más performance altas aunque similares al trabajo anterior.

En 2010, Schmidt *et al.* [7] han propuesto un sistema de segmentación donde proponen el algoritmo Duration-dependent Hidden Markov Model (DHMM) (*Duration-dependant Hidden Markov Model*) logrando un alto desempeño a la hora de la clasificación de los sonidos fundamentales tanto para pacientes sanos como enfermos. Luego, en 2014, Abbas *et al.* [9] han propuesto una técnica de clasificación basada en *K-means clustering* mostrando una alta estabilidad en la detección y clasificación, donde juega un rol vital en la cuantificación e identificación de diferentes casos hemodinámicos. Hasta 2015, los distintos métodos de clasifiación debían estar acompañadas por señales adicionales como el ECG. En 2015, David Springer *et al.* [10] han publicado un algoritmo de segmentación de los sonidos S1 y S2 con un sólo canal de PCG sin referencia externa usando una modificación del algoritmo propuesto por Schmidt, así mejorando la perfomance del estado del arte.

Recientemente, Franceso Renna y Miguel Coimbra de la Universidad de Porto, Portugal, propusieron el uso de Convolutional Neural Network (CNN) (*Convolutional Neural Network*). Este enfoque fue utilizado en otros trabajos para la segmentación de imágenes. Asimismo, aplicaron a la salida de diferentes modelos temporales que generan una salida consistente y que responde a la naturaleza de transición de los estados del fonocardiograma, mejorando las métricas del trabajo de Springer.

Capítulo 3

Estudio de bases de datos

La selección de datos para la implementación del segmentador se extrajo de PhysioNet, organización apoyada por *National Institute of General Medical Sciences* (National Institute of General Medical Sciences (NIGMS)) y *National Institute of Biomedical Imaging and Bioengineering* (National Institute of Biomedical Imaging and Bioengineering (NIBIB)). Esta organización se dedica a ofrecer, de manera gratuita, acceso a un gran cantidad de señales fisiológicas y herramientas para el procesamiento de las mismas.

Particularmente, se extrajeron dos base de datos. Una asociada a a un desafío público en el año 2016 que se denominaba "*Classification of Normal/Abnormal Heart Sound Recordings: the PhysioNet/Computing in Cardiology Challenge 2016*", el cual consistía en diseñar e implementar un clasificador de fonocardiogramas normales y anormales bajo ciertas condiciones. En el Cuadro 3.1 se mencionan las características generales de la base de datos.

Set	Cantidad de archivos	Normal	Anormal	Duración promedio [s]	Máx. dureación [s]	Mín. duración [s]
training-a	409	117	292	32.56	36.5	9.27
training-b	490	386	104	7.98	8	5.31
training-c	31	7	24	49.44	122	9.65
training-d	55	27	28	15.15	48.54	6.61
training-e	2141	1958	183	23.07	101.67	8.06
training-f	114	34	114	33.12	59.62	29.38

Cuadro 3.1: Base de datos: Challenge 2016. Tipos y características de los sonidos cardíacos.

Señales recolectadas en ambientes clínicos y no clínicos, y de pacientes sanos y patológicos. El total es de 3126 fonocardiogramas para el set de entrenamiento (sólo el set A viene acompañado de señales del ECG y 300 fonocardiogramas para el set de validación.

Estos fonocardiogramas fueron extraídos del área aórtica, área pulmonar, área tricúspide y mitral. En cada subset de datos, se clasificaron en fonocardiogramas normales y anormales. Los fonocardiogramas normales fueron extraídos de pacientes sanos y los anormales de pacientes con problemas cardíacos. Los pacientes sufren

de distintas enfermedades cardíacas que no se encuentran detalladas en cada uno de los pacientes. Sin embargo, las enfermedades cardíacas típicas son deficiencias valvulares y enfermedades asociadas a las coronarias. Las deficiencias valvulares incluyen prolapso de la válvula mitral, regurgitación mitral, estenosis aórtica y cirugías valvulares. Todos los pacientes sanos y patológicos incluyen adultos y niños, y cada uno de éstos pueden aportar de uno a seis grabaciones . Adicionalmente, cada una de las grabaciones fue muestreada a 2000 Hz y una única derivación del fonocardiograma.

Es importante tener en cuenta, dado a que las grabaciones fueron hechas en ambientes no controlados, muchas de éstas se encuentan contaminadas por diferentes fuentes de ruido, como el habla, movimiento del estetoscopio, respiración y ruidos intestinales.

La segunda base de datos ha sido publicada por David Springer a partir del trabajo en [10] . Si bien no es exactamente la base de datos utilizada en su trabajo, fue aportada por él junto a una implementación hecha en MATLAB™.

A diferencia de la base del *Challenge 2016*, posee 792 grabaciones de fonocardiogramas muestreadas a una frecuencia de 1000 Hz. Éstas fueron extraídas de 135 pacientes, y más de una grabación por paciente. También, se han dividido grabaciones en varias secciones por inconsistencias de anotaciones.

Las anotaciones están basadas en electrocardiogramas (no presentes en el set de datos), las cuales involucran la onda R y el fin de la onda T muestreadas a 50 Hz. Estas anotaciones fueron extraídas de manera automática mediante un método de concordancia, el cual se detallará en la Sección 4.2.

Por otro lado, se utilizó una clasificación binaria para definir cada grabación como patológica o no patológica.

3.1. Selección y normalización de datos

Ya vimos que estas base de datos contienen datos que difieren entre si. Desde la cantidad de señales hasta los tipos de datos (ECG, anotaciones del ECG, frecuencia de muestreo).

Dicho esto es necesario decidir por una base de datos en una primera instancia. Para ponerlo en términos cuantitativos, por qué la elección de una base respecto de la otra, se menciona lo siguiente. La frecuencia de muestreo no es un problema, por lo tanto tanto 1 KHz como 2 KHz es más que suficiente para tener una buena resolución en tiempo. Ambas bases clasifican las señales respecto a si son patológicas o no patológicas, es decir, que es posible realizar algún tipo de entrenamiento para la clasificación de éstas, aunque no es el alcance de este trabajo y no afecta a la selección de los datos. Sin embargo, esto sirve para identificar que la base no se encuentre desbalanceada respecto a la proporción de sanos como enfermos. El punto crítico de la selección de la base se debe a las anotaciones del electocardiograma. La base de datos de Springer [12] ya contiene marcas extraídas automáticamente y filtradas por un algoritmo de concordancia. De lo contrario, en la base de datos del desafío se proveen los ECG, a los cuales es necesario extraerles las marcas correspondientes . Debido a que el alcance del trabajo es implementar un segmentador de PCG y no un extractor de anotaciones, se optó por utilizar la base de David Springer dado a que permite apoyarse sobre trabajos previos que ya han utilizado dichos datos, confiando en la naturaleza de estos datos y así realizar una mejor comparación en

el Capítulo 7. Asimismo, se referirá a la base seleccionada como "base de datos"para evitar confusión.

Los datos de la base de datos se encuentran en formato `.mat` (formato propietario de MATLAB™) y las anotaciones de las ondas R y fin de la T, se encuentran muestreadas a 50 Hz. Esto requiere de estandarizar el formato de los datos y acomodar la frecuencia de las anotaciones dado a que las señales se encuentran a una frecuencia de muestreo mayor. Para ello, se tomó como estándar el formato Waveform Database (WFDB) (*Waveform Database*), el cual contiene dos categorías estándar, *MIT Format* y *EDF Format*.

MIT Format

- Los archivos *MIT Signal* (`.dat`) son archivos binarios que contienen muestras de señales digitalizadas. Éstas almacenan las señales, pero no pueden ser interpretados sin los *header* files. Los archivos son de la forma: `{RECORD_NAME}.dat`.
- Los archivos *MIT Header* (`.hea`) son archivos de texto cortos que describen el contenido del archivo de la señal asociado.
- Los archivos *MIT Annotation* son archivos binarios que contienen anotaciones (etiquetas que generalmente refieren a muestras específicas de una señal asociada). Éstos deben ser leídos junto a los archivos *header* asociado. En caso de que en un directorio se vean archivos `{RECORD_NAME}.dat`, y/o `{RECORD_NAME}.hea`, cualquier otro archivo con otra extensión es un archivo de anotaciones. Por ejemplo,
- `{RECORD_NAME}.atr` es un archivo con anotaciones para esa señal.

EDF Format

- Los archivos EDF contienen señales digitalizadas
- almacenadads en formato internacional. Estos archivos guardan la información de encabezado al comienzo del
- archivo, a diferencia del formato MIT. A su vez pueden estar acompañados de archivos con anotaciones.
- Por ejemplo, si existe un archivo `*.edf`, un archivo con anotaciones asociado podría ser
- `*.edf.qrs`, donde `.qrs` es la extensión de las anotaciones en este caso (podría ser otro).
- Los archivos EDF+ son archivos EDF que contienen anotaciones
- codificadas como señales.

La categoría utilizada en este trabajo es *MIT Format*. Este formato debe ser leído con una librería especializada. *WFDB Software Package* es un conjunto de funciones para escribir y leer en formatos específicos de *PhysioBank databases* (entre otros). La

librería *WFDB* es *LGPLed*, y puede ser usada para programas escritos en *ANSI/ISO C*, *K&R C*, *C++*, o *Fortran*, corriendo bajo sistemas operativos para los cuales el compilador *ANSI/ISO* or *K&R C* se encuentre disponible, incluyendo todas las versiones Unix, MS-DOS, MS-Windows, the Macintosh OS, y VMS. Opcionalmente, la librería de WFDB puede ser compilada con HTTP o FTP como input sin tener que bajar localmente toda la base de datos para un procesamiento.

En este trabajo se utilizó *WFDB Toolbox for MATLAB™*, el cual provee acceso desde MATLAB™ a las aplicaciones de `WFDB`. Esta toolbox provee MATLAB™ y Java *wrappers* para las distintas aplicaciones y un instalador que instala la toolbox en sí misma y el *WFDB Software Package* precompilado. Esta toolbox corre en 64-bit MATLAB™ R2010b o mayor en *GNU/Linux*, *Mac OS X*, y *MS-Windows*.

El set de datos de Springer, fue convertido de *MAT Format* a *MIT Format*, donde el directorio tiene la estructura de la Figura 3.1. Los archivos `.dat` contienen la señal del PCG. La información ejemplo de los archivos `.hea` se observan en la Figura 3.2 y los archivos `.ann` contienen las anotaciones de las ondas R y el fin de la T, de acuerdo al estándar provisto por PhysioNet.

```
springer-db/
├── RECORDS
├── RECORDS-abnormal
├── RECORDS-normal
├── record1.dat
├── record1.hea
├── record1.ann
├── record2.dat
├── record2.hea
├── record2.ann
.
.
.
├── recordN.dat
├── recordN.hea
└── recordN.ann
```

Figura 3.1: Estructura del directorio de la base de datos. El archivo `RECORDS` contiene en un archivo de texto los nombres de la señales del set de datos, lo mismo para `RECORDS-abnormal` y `RECORDS-normal`, asociados a señales patológicas y no patológicas respectivamente.

.

```
recordi 1 1000 25000
recordi.dat 16 12.1052(3460)/mV 0 0 7531 -17169 0
# Comments
```

Figura 3.2: Información ejemplo del archivo header de una señal. En la primer línea se encuentra, de forma ordenada, el nombre de la señal, la cantidad de señales en el archivo `.dat`, la frecuencia de muestreo en Hz y la cantidad de muestras. La segunda línea contiene el nombre del archivo y a continuacíon información de adquisción.

En cuanto a la corrección de las anotaciones provistas en la base de datos que se encuentran submuestreadas a 50 Hz. Para ello, se recupera la frecuencia original a partir de la ecuación 3.1.

$$\boldsymbol{a}_{f_2} = \frac{f_2}{f_1} \cdot \boldsymbol{a}_{f_1} \tag{3.1}$$

Donde $f_1 = 50$ Hz y $f_2 = 1000$ Hz.

Capítulo 4

Preprocesamiento

4.1. Acondicionamiento de la señal

Las señales biológicas, como tantas otras, presentan contaminación en el momento de la adquisición. Éstas fuentes de ruido, ya las mencionamos antes, provienen de ruidos respiratorios, intestinales, habla, entre otras. Por varios motivos es recomendable realizar un pre-procesamiento de ellas para que los algoritmos que buscan características específicas de la señal en cuestión no "se confunda". Por ejemplo, los algoritmos de delineado de ECG, donde ubican las ondas fundamentales (P, Q, R, S, T_{on}, T, T_{off}) mejor se desempeñan cuanto más limpia o clara es la señal.

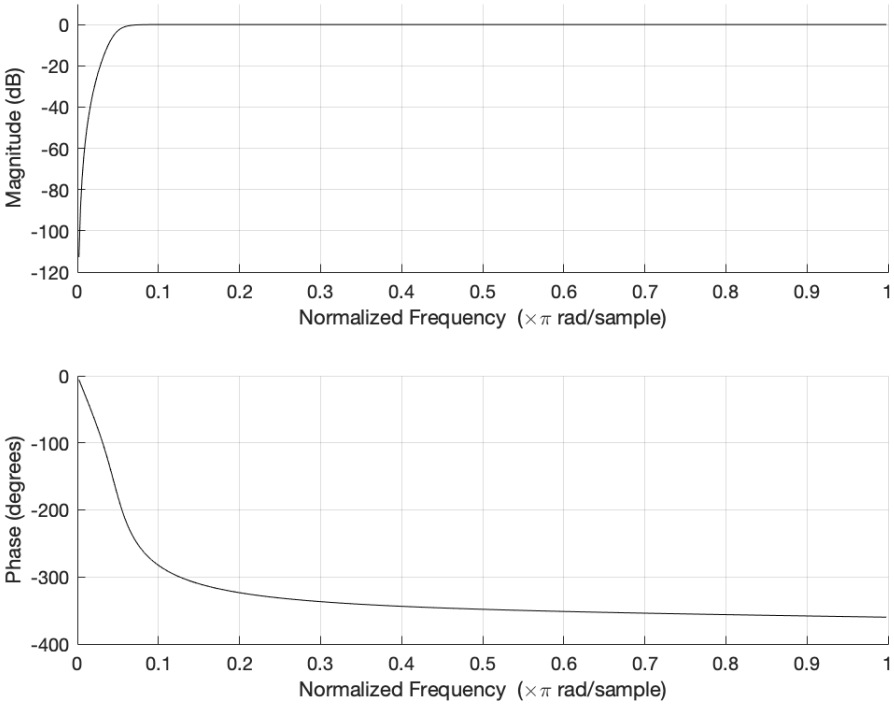
Un fonocardiograma presenta frecuencias entre $25 - 250$ Hz. Esto se debe a los sonidos fundamentales de la señal, S_1 y S_2. El contenido espectral de S_1 se encuentra entre $10 - 180$ Hz, mientras que S_2 entre $50 - 250$ Hz. En el primer caso, la mayor energía espectral se encuentra en componentes por debajo de los 130 Hz. Por supuesto, que la presencia de otras frecuencias que no son intrínsecas al fonocardiograma pueden estar relacionado a enfermedades de insuficiencia cardíaca como contaminación del entorno.

Para realizar el filtrado se utilizaron dos filtros lineales e invariantes en el tiempo de respuesta infinita al impulso. Un Butterworth pasa-altos y pasa-bajos. Las características de estos filtros se detallan más adelante. Esta etapa es importante dado a que elimina ruido de contenido alto en frecuencias, que se observa un "pasto"sobre la señal, y también remueve contenido de baja frecuencia que genera lo que se conoce como línea de base móvil (*baseline wandering*), causado por contenido de baja frecuencia como la respiración. Lo que ocurre es que la línea isoeléctrica (*baseline*) se encuentra desnivelada. Por otro lado, se remueve el valor medio (*offset*). A pesar del filtrado, algunas señales contienen ruido de baja frecuencia que no es posible eliminar con filtros lineales, de esta forma, se aplica el algoritmo de media móvil (*moving vverage*) con una ventana de largo determinado heurísticamente para suavizar los picos. También se aplica un algoritmo de eliminación de picos de los fonocardiogramas. Finalmente, se termina normalizando las señales para que contengan energía unitaria.

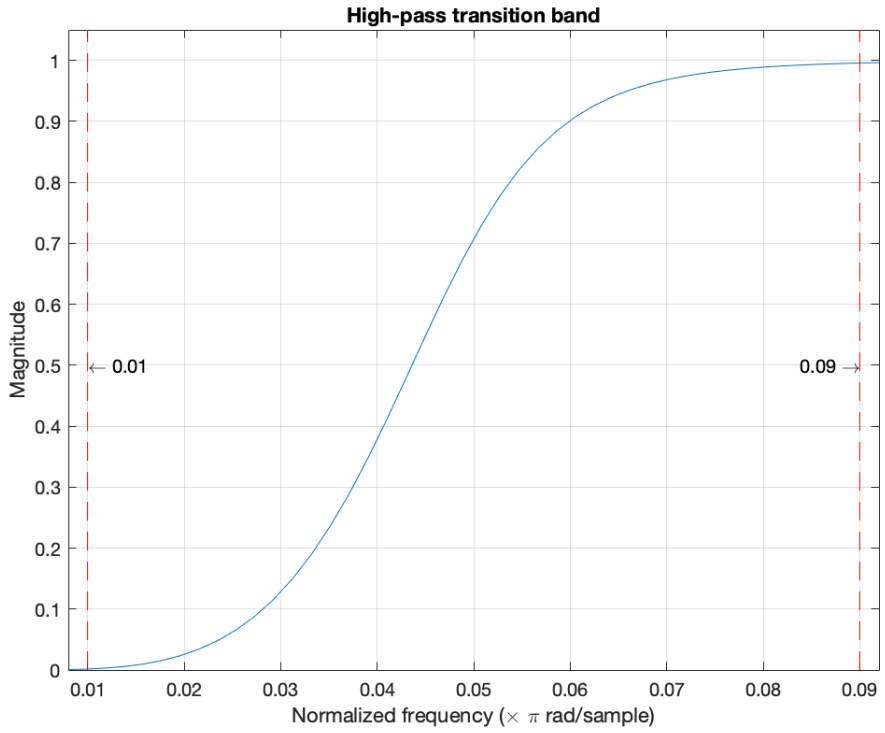
A continuación se observan las especificaciones de los filtros. En la Figura 4.1a y 4.2a se ilustran las magnitudes de los filtros y las fases, y en la Figura 4.1b y 4.2b las bandas de transición.

Filtro	**Orden**	$\mathbf{A}_+$	$\mathbf{A}_-$	ω_c	ω_+	ω_-
Pasa-altos	4	0.01	0.01	25 Hz	41 Hz	1.5 Hz
Pasa-bajos	4	0.01	0.01	400 Hz	467 Hz	345 Hz

Cuadro 4.1: Tabla de especificaciones de los filtros. A_+ es la atenuación de la banda de paso, $\mathbf{A}_-$ la atenuación de la banda de rechazo, ω_c la frecuencia de corte, ω_+ la frecuencia de paso y ω_- la frecuencia de rechazo.

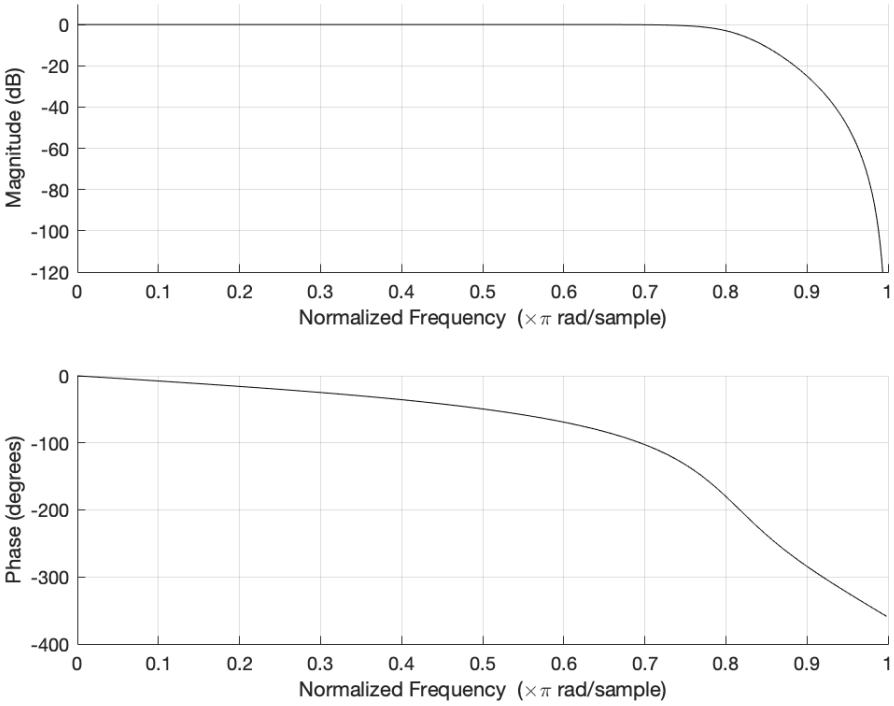


(a) Magintud y fase del filtro pasa-altos.

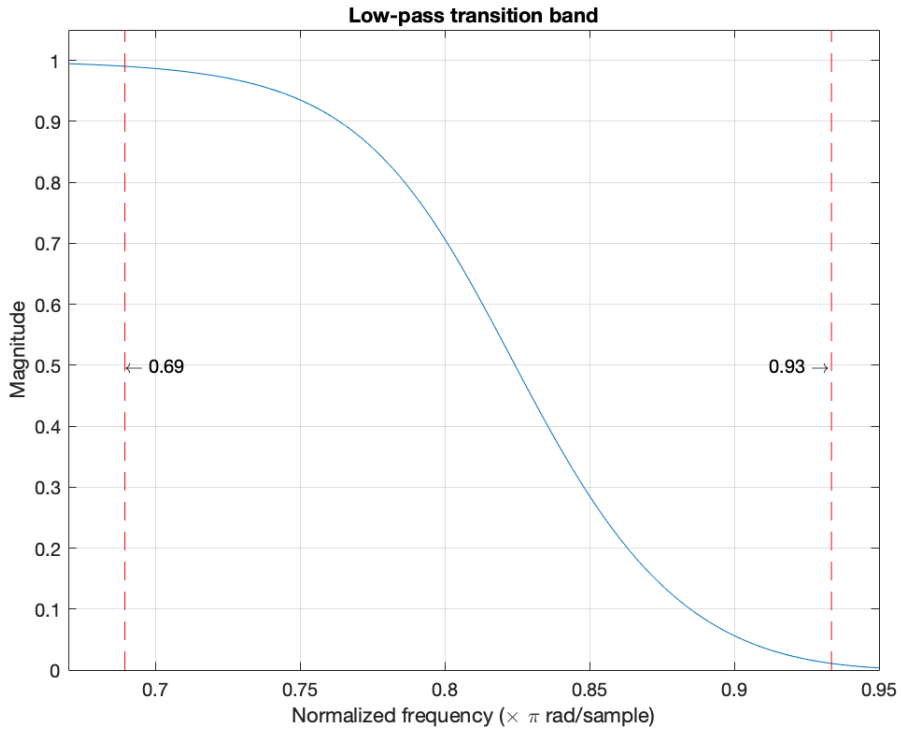


(b) Banda de transición del filtro pasa-altos.

Figura 4.1: Filtro pasa-altos.



(a) Magintud y fase del filtro pasa-bajos.



(b) Banda de transición del filtro pasa-bajos.

Figura 4.2: Filtro pasa-bajos.

Eliminación de picos de fricción

Los picos de fricción deben ser eliminados dado que estos picos generalmente pueden tener mayor amplitud que los sonidos fundamentales del fonocardiograma. A continuación se presenta el algoritmo para realizar dicho procesamiento.

Dada una señal de fonocardiograma $\boldsymbol{x}_i \in \mathbb{R}^N$ muestreada a una frecuencia de muestreo F_s, se elige una ventana de largo arbitrario, i.e. $L = \left[\frac{F_s}{2}\right]$.

La idea fundamental es dividir el PCG en ventanas de longitud L. Debido a esto si el módulo entre la longitud de la señal y el largo de la ventana definido no es cero, algunas muestras quedarán fuera de alguna de las ventanas. Estas muestras se denominan muestras residuales que no participan en el algoritmo y deben ser agregadas al final para recuperar la señal. El número de muestras residuales se definen en la siguiente ecuación.

$$r_s = \text{mod}(N, L) \tag{4.1}$$

Definimos la ventana $\boldsymbol{w}_i \in \mathbb{R}^L$, con $i = 0, 1, \ldots, \left\lfloor \frac{N}{L} \right\rfloor$. Luego, definimos al vector de los máximos, $\boldsymbol{m} \in \mathbb{R}^{\left\lfloor \frac{N}{L} \right\rfloor}$. Este vector contiene los valores máximos de las ventanas, denominados *Maximum Absolute Amplitude* (Maximum Absolute Amplitude (MAA)).

$$m_i = \max \boldsymbol{w}_i \tag{4.2}$$

Las componentes m_i dinámicamente irán cambiando de valor hasta que ninguna de ellas sea superior a 3 veces la mediana de $\boldsymbol{m}$. Esta es la condición donde el algoritmo reconstruye la señal filtrada.

Hasta que no se cumpla la condición antes mencionada, se busca entre las ventanas la que contenga el mayor pico.

$$i_{\boldsymbol{w}} = \arg\max_i(\boldsymbol{m}) \tag{4.3}$$

Una vez hecho esto es necesario encontrar la posición del pico en la ventana asociada.

$$i_{spike} = \arg\max_i(\boldsymbol{w}_{i_{\boldsymbol{w}}}) \tag{4.4}$$

El índice i_{spike} hace referencia al valor máximo del pico de fricción. Sin embargo, es necesario computar el comienzo y el fin del pico para eliminarlo. Para ello se necesitan computar los cruces por cero.

$$s_i = sgn(w_i) \tag{4.5}$$

Donde s_i son las componentes del vector que $\mathbf{s}_i \in \mathbb{R}^L$. Luego, se computan los cruces por cero, $\mathbf{z} \in \mathbb{R}^{L-1}$ en la ecuación 4.6

$$z_i = s_i - s_{i-1}, \quad 0 \leq i \leq L-2 \tag{4.6}$$

A partir de este momento sólo queda definir el inicio y el fin del pico. El inicio del pico se encuentra a partir del último cruce por cero hasta la posición i_{spike}. El final del pico, se encuentra con el primero cruce por cero despuées de i_{spike}. Las ecuaciones 4.7 y 4.8 reflejan el cálculo.

$$p = \max\Big\{n \in \mathbb{N}_0 \mid n \leq i_{spike} - 1\Big\} \tag{4.7}$$

$$q = \min\Big\{n \in \mathbb{N}_0 \mid i_{spike} \leq n \leq L-2\Big\} \tag{4.8}$$

Es posible que el vector de cruces por cero sea nulo, con lo cual los valores de p,q deberían coincidir con el primer índice de la ventana y el último respectivamente. Luego, simplemente queda por eliminar este pico.

$$w_{i,j} = \epsilon, \quad p \leq j \leq q \tag{4.9}$$

Siendo ϵ un número tan chico como sea defina. Por último, queda por recalcular el vector $\mathbf{m}$ con $\mathbf{w}_i$ actualizado.

Cuando se cumpla la condición de 3 veces la mediana de $\mathbf{m}$, se concatenan todas las ventanas y se agregan al final las muestras residuales.

Suavizador - Media móvil

En la Figura 4.3 se ilustra un fonocardiograma de un paciente sano en donde se ve que presenta ciertas oscilaciones, luego de ser filtrado por los filtros lineales anteriores.

Mediante un algoritmo suavizador, se aplica un ventaneo de media móvil, donde se desliza la ventana a lo largo de toda la señal calculando el promedio de cada una de las muestras. Además, se encuentra aplicado el algoritmo de remoción de picos de los PCG.

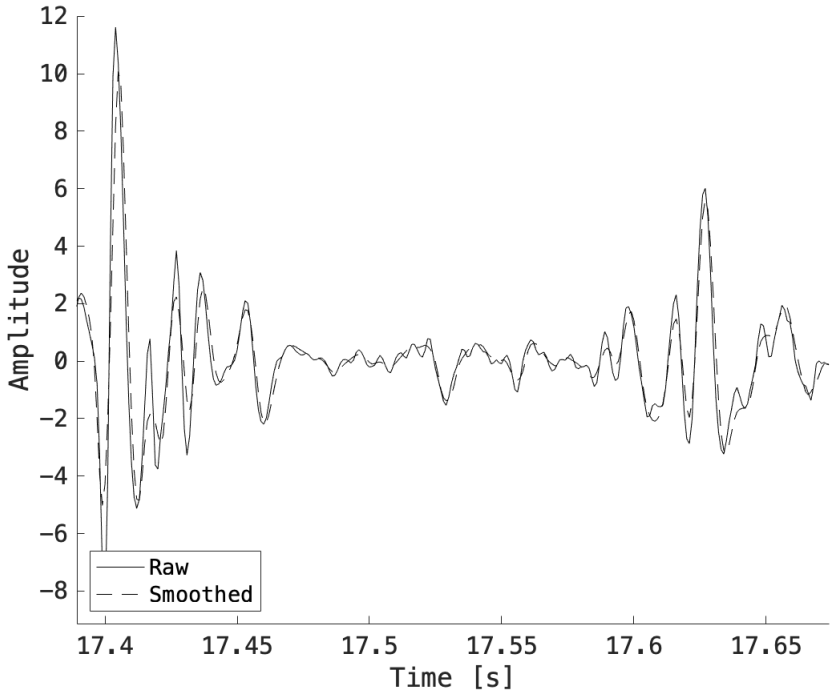


Figura 4.3: Segmento del fonocardiograma de un paciente sano. En línea sólida se muestra la entrada sin ningún tipo de acondicionamiento y en línea punteada la señal suavizada.

4.2. Extracción de marcas

El etiquetamiento de los fonocardiogramas de entrenamiento necesitan de las anotaciones de la onda R y del fin de la onda T. Esto fue propuesto por Schmidt *et al.* en [7], donde propone que los sonidos fundamentales del fonocardiograma, S_1 y S_2 tienen una media y un desvío asociado. Simplemente midió la duración de S_1 y S_2 para luego calcular la media y el desvío muestral, concluyendo que la duración se mantiene relativamente constante siendo de (122 ± 32) ms y (92 ± 28) ms con un 95 % de intervalo de confianza.

Para la extracción de marcas del ECG, éstas fueron hechas automáticamente por algoritmos de delineación. Para ello se compararon 4 detectores para la detección de la onda R y el fin de la onda T. El hecho de que haya ruido y artefactos en el ECG provocará que las anotaciones de los detectores no coincidan. Para ello, para asegurar que las anotaciones sean de calidad, se derivó un *Signal Quality Index* (SQI) mediante la concordancia entre los detectores.

Los detectores utilizados para la detección de la onda R fueron gQRS, jQRS (anteriormente utilizados en [14] y [15]), el algoritmo basado en lo que se conoce como *Parabolic fitting* utilizado en [13] y el delineador de onditas. Ahora para la detección del fin de la onda T se utilizó el algoritmo conocido como *ecgpuwave* (basado en la detección previa del complejo QRS), un método basado en maximizar el área de una ventana móvil entre sucesivas ondas R [16], el delineador de onditas y un algoritmo hecho por Vazquez-Seisdedos et al. [17]. Para decir que los detectores son consistentes entre si las anotaciones no deben estar más alejadas que 100 ms.

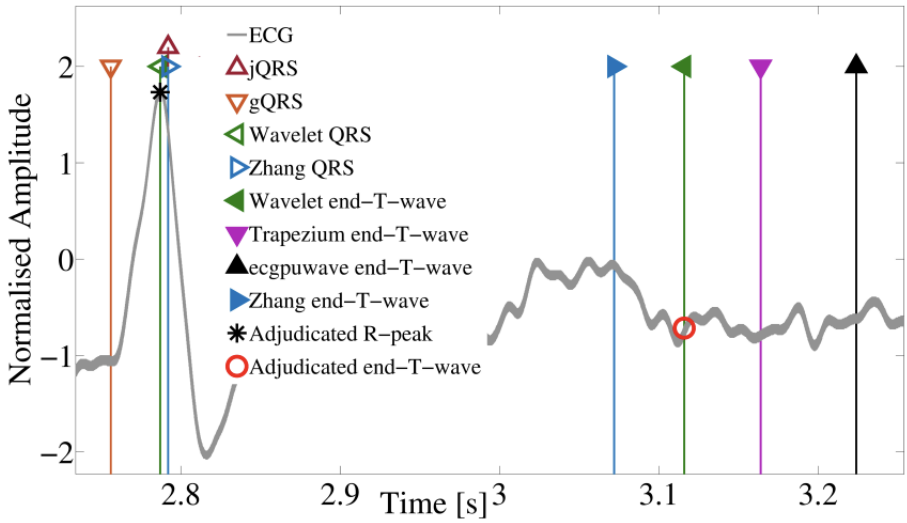


Figura 4.4: Ejemplo de las anotaciones de una señal de ECG con ruido . La posición de las anotaciones para cada uno de los detectores se muestran. Imagen extraída de [10].

4.3. Etiquetamiento

El etiquetamiento (*labeling*), corresponde a obtener etiquetas realizadas de forma manual o automática generalmente utilizadas en el entrenamiento del modelo propuesto. Este método es lo que se conoce como *aprendizaje supervisado*, donde se necesita tener catalogada la entrada para estimar los parámetros del modelo.

El etiquetamiento, como se ha explicado en el Sección 4.2, consiste en estimar los tiempos de duración de los sonidos. Sin embargo, para el conjunto de datos de este trabajo, las estimaciones de Schmidt *et. al* no dieron buenos resultados, dado a que en el momento del etiquetamiento los estados no quedaban bien definidos. Para ello, se modificaron las medias a (122 ± 22) ms y (152 ± 22) ms. Es necesario mencionar que el algoritmo propuesto por Springer *et. al* sea, tal vez, el responsable de la necesidad de adaptar los parámetros de manera ad hoc.

En la Figura 4.5 se observa una porción de una señal de PCG donde, en base a las marcas proveídas por la base de datos, se realiza el etiquetado de los estados de la señal [sístole (S_1), sístole isovolumético, diástole (S_2), diástole isovolumétrico].

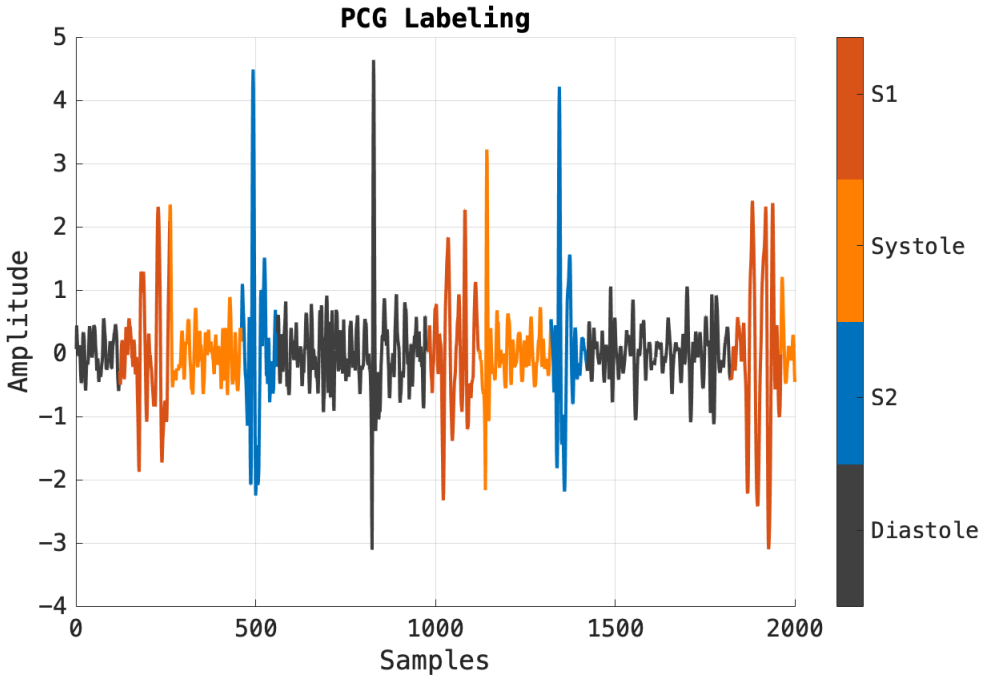


Figura 4.5: Segmento de una señal de fonocardiograma etiquetada. En color rojo se marcan los sonidos S_1, en naranja el sístole isovolumétrico, en azul los sonidos S_2 y en negro el diástole isovolumétrico.

Algoritmo etiquetador

Sea una señal envolvente del fonocardiograma $\boldsymbol{x}_i \in \mathbb{R}^{N_i}$ y las posiciones de las marcas del electrocardiograma asociadas $\boldsymbol{r}_i \in \mathbb{R}^{M_i}$, $\mathbf{t}_i \in \mathbb{R}^{M_i}$. Se definen las medias y desvíos de los sonidos, $(\mu_{S_1}, \sigma_{S_1})$ y $(\mu_{S_2}, \sigma_{S_2})$.

Se define $\mathbf{s}_i \in \mathbb{R}^{N_i}$ al vector de estados de la señal. Recordar que este vector sólo puede tomar 4 valores, $\mathbf{s}_i \in [1, 4]$.

A partir de aquí se mencionara el algoritmo para etiquetar cada uno de los cuatro estados.

Primer sonido (Estado #1)

Para el marcado del primer sonido o estado, se define un umbral superior en base a la posición de la onda R.

$$U_{b,i} = [\min(N_i, \mathbf{r}_i + \mu_{S_1})] \tag{4.10}$$

Donde $[x]$ es el entero más cercano a x.

$$\boldsymbol{s}_j = \mathbb{1}\left\{\max(1, \mathbf{r}_i) < j < \min(U_{b,i}, N_i)\right\} \tag{4.11}$$

Segundo sonido (Estado #3)

Para el etiquetado del segundo sonido o tercer estado, se requiere definir una ventana de búsqueda como así también dos umbrales, uno inferior y otro superior.

Los umbrales se definen de la siguiente manera y dependen de cada una de los finales de las ondas T.

$$U_{b,i} = \min(N_i, \lceil t_i + \lfloor \mu_{S_2} + \sigma_{S_2} \rfloor \rceil) \tag{4.12}$$

$$L_{b,i} = \max(t_i - \lfloor \mu_{S_2} + \sigma_{S_2} \rfloor, 1) \tag{4.13}$$

Se define los enteros superiores e inferiores según la siguiente notación.

$$\lceil x \rceil := \sup\Big\{ n \mid n \in \mathbb{Z},\ \ x \leq n \Big\}$$
$$\lfloor x \rfloor := \inf\Big\{ n \mid n \in \mathbb{Z},\ \ x \geq n \Big\}$$

Luego, se define la ventana de búsqueda en la ecuación 4.14. Para cada posición del final de la onda T se define una ventana. En ella se busca el índice del máximo valor.

$$\tilde{\boldsymbol{s}}_{w,i} = \prod_{k=L_{b,i}}^{U_{b,i}} \boldsymbol{x}_k \cdot \mathbb{1}\Big\{ \mathbf{s}_k \neq 1 \Big\} \tag{4.14}$$

$$l_i = \arg \max_{j \in [L_{b,i}, U_{b,i}]} \Big(\tilde{\boldsymbol{s}}_{w,i} \Big) \tag{4.15}$$

Una vez extraído la posición del máximo, es necesario calcular el centro de la ventana.

$$c_{w,i} = \min\Big(N_i, l_i + L_{b,i} \Big) \tag{4.16}$$

Diástole isovolumétrico (Estado #4)

De esta manera es posible calcular las muestras que corresponden al tercer estado. Antes es necesario redefinir el umbral superior.

$$U_{b,i} = \min\left(N_i, \left\lceil c_{w,i} + \frac{1}{2}\mu_{S_2} \right\rceil \right) \tag{4.17}$$

$$\mathbf{s}_j = 3 \cdot \mathbb{1}\left\{ \max\left(\lceil c_{w,i} - \frac{1}{2}\mu_{S_2} \rceil, 1 \right) < j < U_{b,i} \right\} \tag{4.18}$$

Con $j = 0, 1, 2, \ldots, M-1$.

Una vez definidos los estados 1 y 3, correspondientes a S_1 y S_2 respectivamente, es posible definir el cuarto estado.

Es necesario obtener la diferencia entre todos las posiciones de la onda R y la

onda T. Para ello se define el vector $\mathbf{d}_i \in \mathbb{R}^M$, donde sus componentes son $d_{j,i}$, con $i = 0, 1, 2, \ldots, M-1$ y $j = 0, 1, 2, \ldots, M-1$.

$$d_{j,i} = \begin{cases} \infty, & r_j - t_i < 0 \\ r_j - t_i, & \text{otro caso} \end{cases} \tag{4.19}$$

$$p_i = \begin{cases} N, & \sum_{j=1}^{M_i} \mathbb{1}\Big\{d_{j,i} < \infty\Big\} = 0 \\ t_{\arg\min\limits_j \mathbf{d}_i} - 1, & \text{otro caso} \end{cases} \tag{4.20}$$

$$\mathbf{s}_j = 4 \cdot \mathbb{1}\left\{\left\lceil c_{w,i} + \frac{1}{2}(\mu_{S_2} + \sigma_{S_2})\right\rceil < j < p_i\right\} \tag{4.21}$$

Inicio y fin de la señal

Dado a que todos los estados derivan de la posición de la onda R y del fin de la onda T, el primer estado en la señal siempre será 1 o 3. Por lo tanto, hasta ese estado, debe ser 2 o 4 respectivamente.

Tanto para el inicio como el fin de la señal es necesario obtener desde izquierda a derecha la posición del primer estado y el último estado en $\mathbf{s}$. Las ecuaciones 4.22, 4.23 reflejan como calcular dichos índices.

$$f_n = \min\Big\{n \in \mathbb{N}_0, \mid s_n \neq 0\Big\} \tag{4.22}$$

$$l_n = \max\Big\{n \in \mathbb{N}_0, \mid s_n \neq 0\Big\} \tag{4.23}$$

Luego, se definen los estados del inicio y del final de la señal.

$$s_j = \begin{cases} 2, & s_{f_n} = 3 \\ 4, & s_{f_n} = 1 \end{cases}, \quad \text{con } j = 0, 1, \ldots, f_n - 1 \tag{4.24}$$

$$s_j = \begin{cases} 2, & s_{l_n} = 1 \\ 4, & s_{l_n} = 3 \end{cases}, \quad \text{con } j = l_n + 1, l_n + 2, \ldots, M-1 \tag{4.25}$$

Sístole isvolumétrico (Estado #2)

Todos los estados las muestras que no han sido definidos son las correspondientes al estado 2.

$$s_j = \begin{cases} 2, & s_j = 0 \\ s_j, & s_j \neq 0 \end{cases}, \quad \text{con } j = 0, 1, \ldots, M-1 \tag{4.26}$$

Capítulo 5

Procesamiento

5.1. Identificación y construcción de características

Generalmente para la detección o clasificación se utiliza lo que se conoce como atributos (*features*). Estos atributos son necesarios para realizar estimaciones de todo tipo, desde casos sencillos para la estimación de la altura y peso promedio en una población hasta detección de obstáculos para los algoritmos de evasión. En este caso particular, para estimar dónde comienzan y finalizan los estados de un PCG, es necesario también contar con atributos que ayuden a los algoritmos de segmentación a definir éstos.

En el caso de señales en el tiempo o *time-series* existen distintos tipos de atributos que se pueden extraer. Ejemplos de estos son, amplitud, energía, contenido espectral, entre otros. Por supuesto, que no siempre todos los atributos de la señal son aptos para el tipo de estimación que se desea efectuar.

Los enfoques de clasificación de sequencias que se abordan aquí son: 1) sequencia a sequencia (*sequence-to-sequence*), 2) secuencia a etiqueta (*sequence-to-label*). La primera implica que la longitud de la entrada sea conocida, fija e igual que la salida del bloque de clasificación o segmentación y la segunda implica sólo tener una etiqueta de lo que la entrada representa. Los casos que se manejan en este trabajo son las técnicas de clasificación de señales sanas y patológicas. El caso más sencillo las etiquetas son dos: 1) sano, 2) patológico y las técnicas de segmentación en las cuales se basa el el presente trabajo, donde se quiere etiquetar o segmentar la secuencia de entrada (esto significa ponerle un valor a cada muestra de la secuencia en el tiempo).

Características base

Una importante etapa en el proceso comienza con la extracción de atributos propuestos por David Springer *et al.* [10] A continuación se listan algunos atributos:

1) Envelograma homomórfico (*homomorphic envelogram*): Este procedimiento es muy similar a la modulación AM. Ésta técnica ha sido usada por numerosos investigadores para la extracción de envolventes del PCG [23], [27], incluido el algoritmo de segmentación [7]. El envelograma homomórfico es derivado de aplicarle el logaritmo natural a la señal en cuestión $x(n)$. Esta señal se la modela como la multiplicación de una señal de baja frecuencia envolvente y una oscilación de más

alta frecuencia.

$$x(n) = a(n) \cdot o(n) \tag{5.1}$$

El logaritmo natural aplicado a esto implica que se pueda separar en una suma ambas componentes.

$$\ln(x(n)) = \ln(a(n)) + \ln(o(n)) \tag{5.2}$$

De esta manera con un filtro pasa-bajos, con una frecuencia de corte adecuada, se logra filtrar las osilaciones $o(n)$ y se obtiene $a(n)$ con la ecuación 5.3.

$$a(n) = \exp(\mathcal{H}(\ln(x(n))) \tag{5.3}$$

Donde el operador $\mathcal{H}$ es el filtro pasa-bajos.

$$\begin{aligned} a(n) &= \exp(\mathcal{H}(\ln(a(n)) + \ln(o(n))) \\ a(n) &= \exp(\ln(a(n))) \end{aligned}$$

2) Envolvente de Hilbert (*Hilbert envelope*): dada una función analítica $g(t)$.

$$g(t) = \sin(\omega t)\sin(\Omega t) \tag{5.4}$$

donde $\omega > \Omega$, la envolvente es posible construirla a partir del valor absoluto de la función analítica $\mathscr{A}(g(t))$. $\mathscr{A}$ se compone de la señal original $g(t)$ y su transformada de Hilbert $\tilde{g}(t)$ de la siguiente manera:

$$\mathcal{T}(g(t)) = g(t) + i\tilde{g}(t) \tag{5.5}$$

1. Lo primero es construir la transformada de Hilbert.

$$\tilde{g}(t) = -\int_{-\infty}^{\infty} [a(f)\sin(ft) - b(f)\cos(ft)]df \tag{5.6}$$

$$a(f) = \frac{1}{\pi}\int_{-\infty}^{\infty} g(t)\cos(ft)]dt \tag{5.7}$$

$$b(f) = \frac{1}{\pi}\int_{-\infty}^{\infty} g(t)\sin(ft)]dt \tag{5.8}$$

2. Empezar por construir $a(f)$.

$$\begin{aligned} a(f) &= \int_{-\infty}^{\infty} \sin(\omega t)\sin(\Omega t)\cos(ft)]dt \\ &= \frac{1}{2}\int_{-\infty}^{\infty} [\cos(\omega - \Omega)t - \cos(\omega + \Omega)t]\cos(ft)]dt \\ &= \frac{1}{2}[\delta(f - \omega + \Omega) - \delta(\delta(f - \omega - \Omega)] \end{aligned}$$

3. Luego por $b(f)$.

$$b(f) = \int_{-\infty}^{\infty} \sin(\omega t)\sin(\Omega t)\sin(ft)]dt$$

Usando las identidades trigonométricas

$$\sin(\alpha)\sin(\beta) = \frac{1}{2}[\cos(\alpha-\beta) - \cos(\alpha+\beta)]$$
$$\cos(\alpha)\sin(\beta) = \frac{1}{2}[\sin(\alpha-\beta) + \sin(\alpha+\beta)]$$

Entonces

$$b(f) = 0$$

4. Así substituyendo $a(f)$ y $b(f)$ en 5.6.

$$\tilde{g}(t) = -\sin(\Omega t)\cos(\omega t) \tag{5.9}$$

5. Obtener el valor absoluto de $\mathscr{A}(g(t))$.

$$\mathcal{T}(g(t)) = \sin(\Omega t)\sin(\omega t) - i\sin(\Omega t)\cos(\omega t) \tag{5.10}$$

$$|\mathcal{T}| = \sqrt{\mathcal{T}\mathcal{T}*} = |\sin(\Omega t)|$$

Messer et al. [18] y Kumar et al. [19] calcularon la envolvente del PCG usando la transformada de Hilbert. La transformada de Hilbert extrae la función analítica que excluye las frecuencias negativas de la señal original y su envolvente se obtiene calculando el módulo.

2) Envolvente de onditas (*wavelet envelope*): el análisis de onditas ha sido ampliamente explorado. Sin embargo, hay discusiones sobre cuál familia es la óptima para el filtrado, clasificación y segmentación del PCG. Algunos investigadores determinan que la familia Morlet es la que mejor concuerda en el análisis de fonocardiogramas [20], [21]. Otros destacan la familia de Daubechies [22], [23]. En este caso se ha utilizado la Discrete Wavelet Transform (DWT) con distintas familias y niveles (la familia de Morlet no ha sido utilizada dado a que DWT no es compatible con esta familia).

3) Envolvente de densidad espectral de potencia (*power spectral density envelope*): la mayoría del contenido espectral de S_1 y S_2 se encuentra por debajo de 150 Hz con un pico en 50 Hz. Basado en estas frecuencias, se extrae este atributo a partir entre 40 Hz y 60 Hz, en ventanas con 50 % de solapamiento y ancho de 0 .05 s. La PSD fue calculada utilizando ventaneo de Hamming y transformada de Fourier.

Cada uno de los 4 atributos extraídos se los normaliza en varianza y media. Luego, se realiza un submuestreo a 50 Hz por motivos computacionales.

Transformada sincronizada de Fourier

El anásis frecuencio-temporal y de escala de tiempo son herramientas estándar para el estudio de señales no estacionarias o determinísticas con variación frecuencial. En particular, señales multicomponentes, por ejemplo superposición de amplitud y ondas moduladas en frecuencia (Amplitude Modulation (AM)-Frequency Modulation (FM)), son bien analizadas con la Transformada de tiempo corto de Fourier (Fast Synchrosqueezed Transform (FSST)) [25] y la Transformada Continua de Onditas (Continous Wavelet Transform (CWT)) [26]. Es bien conocido que ambas transformadas para estas señales dibujan una especie de líneas en los planos de Time-frequency (TF) (tiempo-frecuencia) o Time-scale (TS) (escala de tiempo), alrededor de crestas que corresponden a la frecuencia instantánea de los modos que hacen a la señal.

La Synchrosqueezed Transform (SST) (Tranformada sincronizada), introducida en [24], es una suerte de reasignación al método que trata de ajustar la representación TS manteniendo la invertibilidad. Esta técnica se desarrollo en el contexto de la CWT pero sin ningún tipo de avances en el campo de la Short-time Fourier Transform (STFT).

Transformada de tiempo corto de Fourier y señales multicomponentes

Denotamos la transformada de Fourier con la siguiente notación $\hat{f}(\eta)$ para la función $f(t)$.

$$\hat{f}(\eta) = \int_{-\infty}^{\infty} f(t) e^{-2i\pi\eta t} dt \tag{5.11}$$

La STFT es una versión local de la transformada de Fourier obtenida por medio de una ventana deslizante g:

$$V_f(\eta, t) = \int_{-\infty}^{\infty} f(\tau) g(t-\tau) e^{-2i\pi\eta(t-\tau)} d\tau \tag{5.12}$$

La representación de $|V_f(\eta, t)|^2$ en el plano TF es lo que se conoce como *espectrograma* de la señal f.

De esta manera, se analiza señales multicomponentes AM-FM con la STFT.

$$f(t) = \sum_{k=1}^{K} f_k(t) = \sum_{k=1}^{K} A_k(t) e^{2i\pi\phi_k(t)} \tag{5.13}$$

Si se asume que las variaciones en amplitud y frecuencia son lentas, se puede escribir la siguiente aproximación alrededor de un tiempo fijo t_0, lo cual equivale a aproximar f por funciones puras.

$$f(t) \approx \sum_{k=1}^{K} A_k(t_0) e^{2i\phi[\phi_k(t_0) + \phi_k^{'}(t_0)(t-t_0)]} \tag{5.14}$$

La correspondiente STFT aproximada se escribe (cambiando t_0 por un t genérico).

$$V_f(\eta, t) \approx \sum_{k=1}^{K} f_k(t) \tilde{g}(\eta - \phi_k^{'}(t)) \tag{5.15}$$

Esto muestra que la representación de una señal multicomponente en el plano TF se encuentra concentrada alrededor de crestas definidas por $\eta = \phi_k^{'}(t)$. Si las frecuencias $\phi_k^{'}$ se encuentran lo suficientemente separadas cuando k varía, cada modo ocupa un dominio distinto del plano TF, permitiendo la detección, separación y reconstrucción.

Sincronización basada en Fourier

El objetivo de la SST tiene dos aspectos: proveer una representación concentrada de señales multicomponentes en el plano TF, y una descomposición que permite separar y demodular los diferentes modos.

Motivación-definición

A partir de la STFT V_f, la FSST mueve los coeficientes $V_f(\eta, t)$ según a la transformación $(\eta, t) \to (\hat{\omega}_f(\eta, t), t)$ donde $\hat{\omega}_f$ es la *frecuencia instantánea* definida por:

$$\hat{\omega}_f(\eta, t) = \frac{1}{2\pi}\partial_t \arg V_f(\eta, t) = \mathcal{R}e\left(\frac{1}{2\pi i}\frac{\partial_t V_t(\eta, t)}{V_f(\eta, t)}\right) \tag{5.16}$$

Este operador es simplemente la frecuencia instantánea de la señal a un dado tiempo t, filtrada en la frecuencia η. Esto es una buena aproximación local de la frecuencia instantánea $\phi_k^{'}(t)$. El segundo punto importante de la SST es la reconstrucción "vertical", que se encuentra en $L^2(\mathbb{R})$ siendo la ventana g continua y distinta de cero en $t = 0$.

$$f(t) = \frac{1}{g(0)}\int_{-\infty}^{\infty} V_f(\eta, t)d\eta \tag{5.17}$$

Ésto permite definir la FSST, lo cual consiste en restringir el dominio de integración de [5.17] al intervalo donde $\hat{\omega}_f(\eta, t) = \omega$, escribiendo:

$$T_f(\omega, t) = \frac{1}{g(0)}\int_{-\infty}^{\infty} V_f(\eta, t)\delta(\omega - \hat{\omega}_f(\eta, t))d\eta \tag{5.18}$$

A continuación se muestra la comparación entre una STFT y una FSST. El ejemplo consiste en una señal con dos componentes frecuenciales, dado $\Omega_0 = \frac{2}{5}\pi$. La señal $x(n)$ se describe en la ecuación 5.19

$$x(n) = \sin(\Omega_0 n) + 3\sin(2\Omega_0 n),\ n \in [0, 1023] \tag{5.19}$$

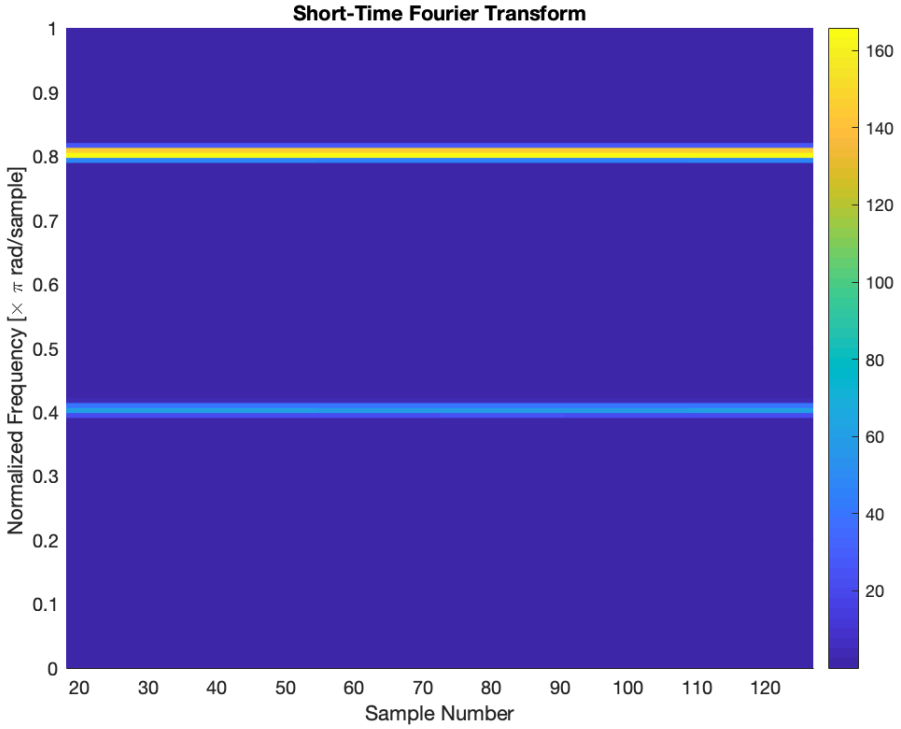


Figura 5.1: Espectrograma de una señal sinusoidal de dos tonos. Generación de una señal suma de dos sinusoidales de 1024 muestras con ruido blanco Gaussiano. La frecuencia más alta presenta una amplitud mayor que la de baja frecuencia.

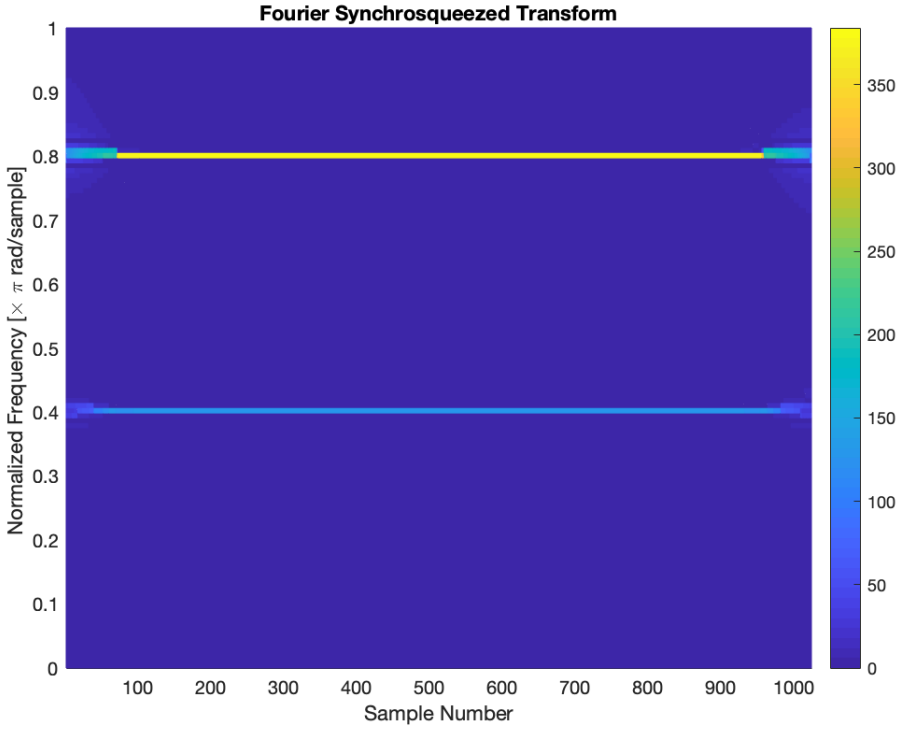


Figura 5.2: FSST de una señal sinusoidal de dos tonos. Generación de una señal suma de dos sinusoidales de 1024 muestras con ruido blanco Gaussiano. La frecuencia más alta presenta una amplitud mayor que la de baja frecuencia.

Observando ambas Figuras 5.1 y 5.2 se nota a simple vista la concentración de energía de ambos tonos en la FSST, mientras que en la STFT no. Esto deja evidenciado en la implementación el concepto de crestas mencionado anteriormente.

Capítulo 6

Deep learning

Deep Learning es un conjunto de métodos de aprendizaje que intentan modelar problemas con arquitecturas complejas combinando diferentes transformaciones no lineales. Los pilares fundamentales son las redes neuronales. Combinadas logran formar las Deep Neural Network (DNN) (*deep learning neural network*). Estas técnicas han logrado un avance significativo en los campos de procesamiento de sonidos y de imágenes, éstos incluyen reconocimiento facial, reconocimiento del habla, visión computadorizada, entre otros.

6.1. Redes neuronales

Una red neuronal artificial es una aplicación, no lineal respecto a su parámetros θ asociado a una entrada x y una salida $y = f(x, \theta)$. Por simplificación $y \in \mathbb{R}$ pero puede ser que y sea multidimensional, $\mathbf{y} \in \mathbb{R}^n$. La red neuronal puede ser utilizada para regresión o clasificación. En aprendizaje estadístico, como es usual, se estima un parámetros θ a partir de una muestra. Generalmente, la función a minimizar no es convexa y resulta en minimizaciones locales. Existe un método muy eficiente para computar el gradiente de una red neuronal. Éste llamado retropropagación del gradiente (*backpropagation of the gradient*), permite obtener una minimización local de un criterio cuadrático fácilmente.

6.2. Perceptrón

Un perceptrón es una función f_j de la entrada $\mathbf{x} = (x_1, ..., x_d)$ pesado por un vector $\mathbf{w} = (w_{j,1}, ..., w_{j,d})$, completado por un sesgo b_j y asociado a una función de activación ϕ.

$$y_j = f_j(x) = \phi(< w_j, x >) + b_j \tag{6.1}$$

Funciones de activación

- Función de identidad:

$$\phi(x) = x, \ -\infty < x < \infty \tag{6.2}$$

- Función sigmoidea:

$$\phi(x) = \frac{1}{1+e^{-x}},\ -\infty < x < \infty \tag{6.3}$$

- Función tangente hiperbólica:

$$\phi(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \frac{e^{2x}-1}{e^{2x}+1},\ -\infty < x < \infty \tag{6.4}$$

- Función de umbral (*hard-threshold*):

$$\phi(x) = \mathbf{1}_{x\geq\beta},\ -\infty < x < \infty \tag{6.5}$$

- Función rectificadora lineal (*rectified linear unit*):

$$\phi(x) = \text{máx}(0, x),\ -\infty < x < \infty \tag{6.6}$$

En la Figura 6.1 se ilustra un esquemático de una neurona donde $\Sigma =< w_j, x > +b_j$.

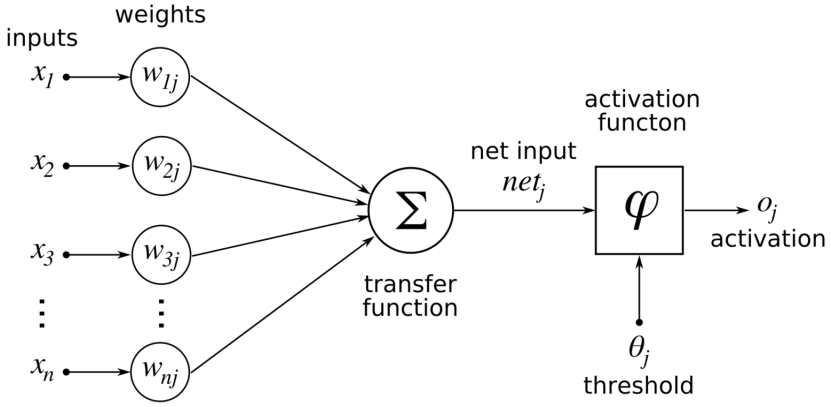


Figura 6.1: Esquemático de una neurona. Imagen obtenida de Wikimedia Commons.

Históricamente, la función sigmoidea fue mayormente utilizada ya que es diferenciable y permite mantener los valores entre [0,1]. Sin embargo, es problemática ya que su gradiente es muy cercano a cero cuando $|x|$ no es se encuentra cerca de 0.

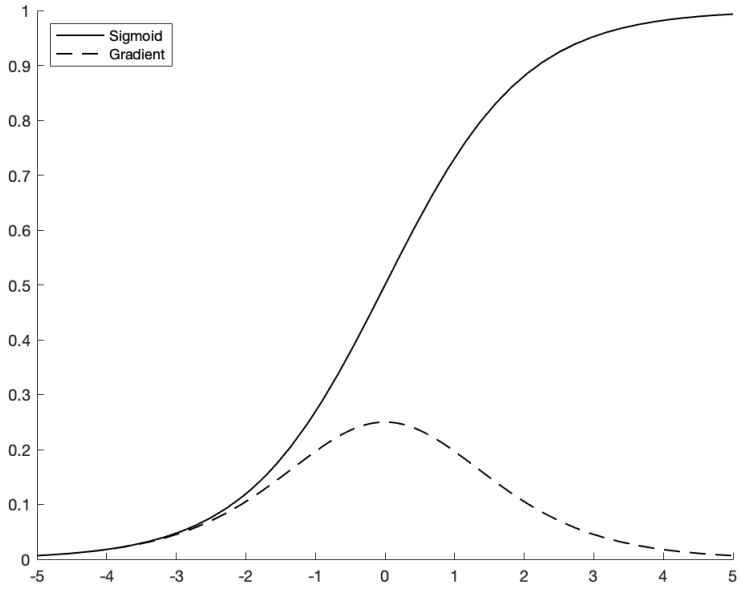


Figura 6.2: Función de activación sigmoidea. El línea sólida se muestra la función de activación y su derivada en línea punteada.

Es el caso de *deep learning* que se utilizan múltiples capas de redes neuronales, lo cual trae problemas con el algoritmo de retropropagación para estimar parámetros. Éste es el por qué la función sigmoidea fue reemplazada por la función Rectified Linear Unit (ReLU) (*Rectified Linear Unit*).

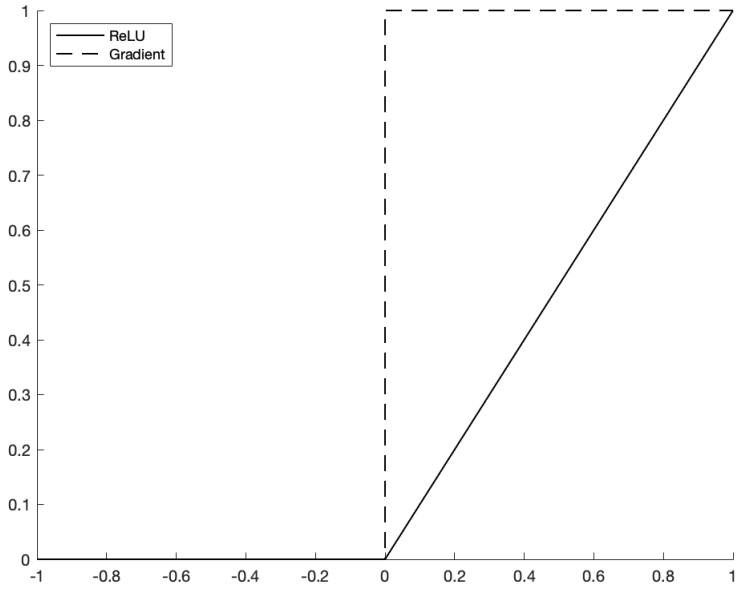


Figura 6.3: Función de activación ReLU. El línea sólida se muestra la función de activación y su derivada en línea punteada.

Esta función no es diferenciable en cero, pero en la práctica esto no es un

problema dado a que la probabilidad de que una entrada sea igual a cero es casi nula. La función ReLU también tiene la propiedad de dispersión (*sparcification*). Ella y su derivada son 0 para valores negativos, y no es posible obtener información en estos casos, pero eso es recomendable agregar un pequeño sesgo para asegurar de que cada unidadad se encuentre activa. Muchas variaciones de la función ReLU consideran mantener a las unidades siempre activas y que sus gradientes para valores negativos no sean 0.

6.3. Perceptrón multicapa

Un perceptrón multicapa (o una red neuronal) es una estructura compuesta por varias capas, las cuales en la literatura se las denomina capas ocultas (*hidden layers*), compuestas por neuronas donde su salida son la entrada de las neuronas de la siguiente capa. Más aún, la salida de una neurona puede ser la entrada de otra neurona de la misma o anterior capa (es el caso de las redes neuronales recurrentes). En la última capa, denominada capa de salida, es posible aplicar una función de activación distinta a las aplicadas en las capas intermedias dependiendo del problema en cuestión: regresión o clasificación.

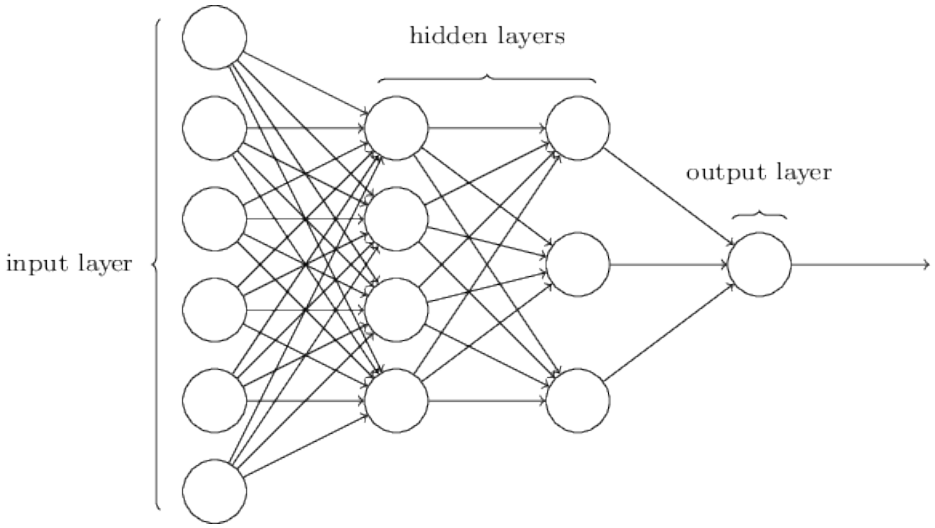


Figura 6.4: Perceptrón multicapa. Imagen obtenida del repositorio de Raymundo Cassani

Multilayer Perceptron (MLP) (Multilayer perceptron) tienen una arquitectura básica. Cada unidad o neurona de una capa se conecta a todas las neuronas de la siguiente capa y no tienen conexión con las neuronas de la misma capa. Los parámetros de la arquitectura son la cantidad de capas y el número de unidades por capa. Generalmente, como se ha dicho antes, la función de activación es diferente a las otras intermedias. En el caso de clasificación binaria, la salida genera una predicción de $\mathbb{P}(Y = 1|X)$ ya que ese valor es [0,1], y la función de activación sigmoidea es utilizada. Para un caso de clasificación multiclase (N es la cantidad de clases), la capa de salida se compone de N neuronas, generando una predicción de $\mathbb{P}(Y = i|X)$, siendo $i = 1, 2, ..., N$. Por supuesto, al ser probabilidades, la suma de todas éstas deben sumar 1.

La mayoría de las veces se usa la función multidimensional *softmax*.

$$\text{softmax}(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}} \tag{6.7}$$

La formulación matemática para el perceptrón multicapa se define con L capas intermedias.

1. La capa de entrada (*input layer*) se define para $k = 0$.

$$h^{(0)}(\boldsymbol{x}) = \boldsymbol{x}$$

2. Para las capas ocultas se define $k = 1, 2, ..., L$.

$$a^{(k)}(\boldsymbol{x}) = \boldsymbol{b}^{(\boldsymbol{x})} + \mathbf{W}^{(k)} h^{(k-1)}(\mathbf{x})$$
$$h^{(k)}(\boldsymbol{x}) = \phi(a^{(k)}(\boldsymbol{x}))$$

3. Para $k = L + 1$, la capa de salida (*output layer*).

$$a^{(L+1)}(\boldsymbol{x}) = b^{(L+1)} + \mathbf{W}^{(L+1)} h^{(L)}(\mathbf{x})$$
$$h^{(L+1)}(\boldsymbol{x}) = \psi(a^{(L+1)}(\boldsymbol{x}))$$

Donde ϕ es la función de activación de las capas intermedias y ψ la función de activación de la capa de salida, este caso la función *softmax*. La matriz $\boldsymbol{W}^{(k)}$ tiene dimensiones M $\times$ N, donde M corresponde a la cantidad de neuronas en la capa k y N a la cantidad de neuronas en la capa $k - 1$.

Estimación de parámetros

Una vez definida la arquitectura de la red neuronal, los parámetros $\mathcal{D} = \{\boldsymbol{W}, \boldsymbol{b_j}\}$ deben ser estimados a partir de muestras. Como es común, estas estimaciones se obtienen a partir de la minimización de lo que se cononce como función de costo (*loss function*)

Existen distintos tipos de funciones de costo. Éstas dependen del problema a resolver. 1) regresión:

- Error cuadrático medio (Mean Square Error (MSE))[1]

$$\textbf{MSE} = \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n} \tag{6.8}$$

- Error absoluto medio (Mean Absolute Error (MAE))[2]

$$\textbf{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{6.9}$$

[1]La función de costo MSE se conoce también como:

- *Quadratic Loss* o *L2 loss*

[2]Se conoce a la función de costo MAE como:

- *L1 loss*

- Error de sesgo medio (Mean Bias Error (MBE))

$$\mathbf{MBE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n} \tag{6.10}$$

2) Clasificación:

- Costo bisagra (*Hinge loss*)

$$\mathbf{L}(\hat{\mathbf{y}}_\mathbf{i}, \mathbf{y}) = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1) \tag{6.11}$$

- Costo de entropía cruzada (*Cross-entropy loss*)

$$\mathbf{L}(\hat{\mathbf{y}}_\mathbf{i}, \mathbf{y}) = -(y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{6.12}$$

6.4. Inicialización

La información a la entrada debe normalizarse y los sesgos pueden ser inicializados en 0. No así, los pesos ya que en el caso de la función de activación *tanh*, su derivada en 0 es 0, el cual es un punto de silla. Tampoco pueden ser inicializados con los mismos valores, de lo contrario todas las neuronas tendrían el mismo comportamiento . Usualmente estos valores se inicializan de form aleatoria: $W_{i,j}^{(k)} \sim \mathcal{U}[-c, c]$, i.i.d con $c = \frac{\sqrt{6}}{N_k + N_{k-1}}$ donde N_k es el tamaño de la capa k. También se suele inicializar los pesos con una distribución normal, $W_{i,j}^{(k)} \sim \mathcal{N}(0, 0.01)$.

6.5. Hiperparámetros

Existen una variadad extensa de algoritmos para minimizar las funciones de costo y todos estos poseen hiperparámetros que deben ser calibrados. Éstos tienen un impacto importante en la convergencia de los mismos. Una herramienta básica en todos estos algoritmos es lo que se conoce como Stochastic Gradient Descent (SGD) (*Stochastic Gradient Descent*). Es la más simple.

$$\theta_i^{new} = \theta_i^{old} - \epsilon \frac{\partial L}{\partial \theta_i}(\theta_i^{old}) \tag{6.13}$$

Donde ϵ es lo que se conoce como tasa de aprendizaje (*learning rate*). Si es muy pequeño, la convergencia es muy lenta y puede quedar bloqueada en un mínimo local. Si es muy grande, la convergencia puede oscilar alrededor de un óptimo sin estabilizarse y converger. Es recomendable comenzar con un ϵ grande e ir iterando.

El algoritmo SGD consiste en computar el gradiente. Para ello se considera la técnica de aprendizaje por lotes (*batch learning*), en el que en cada paso m muestras son elegidas al azar y la media de esas m muestras se utilizara para actualizar los parámetros. Luego, otro concepto es lo que se denomina como *epoch*, proveniente del inglés. Se define *epoch* al paso de todo el set de entrenamiento por la red neuronal una vez y se encuentra íntimamente relacionado con el tamaño del lote (*batch size*). Es decir, si el tamaño es 1/100, significa que un *epoch* contiene 100 batches. La

cantidad de *epochs* es a su vez un hiperparámetro a calibrar.

Existen técnicas para detener la estimación, conocido como parada temprana (*early stopping*) y consiste en considerar un set de validación en el cual se detiene la estimación cuando la función de costo de este set de datos deja de decrecer. El método de aprendizaje por lotes es utilizado por motivos computacionales. El algoritmo de retropropagación mencionado antes necesita guardar todos los valores intermedios y para grandes set de datos como han de ser imágenes resulta prácticamente inviable. Como se ha visto, el tamaño del lote es un hiperparámetro el cual debe ser definido previamente y cuanto más pequeño mejores resultan las generalizaciones de las estimaciones. En el caso de que sea igual a 1, se conoce como gradiente online descendente (*on-line gradient descent*).

Una técnica que hoy en día se utiliza en su mayoría para mitigar el problema de la generalización de estos métodos es la técnica de abandono (*drop-out*). Consiste en definir una probabilidad p, la cual resulta en otro hiperparámetro, algunas unidades de la red se fijan a 0. Tradicionalmente se utiliza 0.5 para las capas intermedias y 0.2 para la capa de entrada. Computacionalmente este método no es costoso dado a que es cambiar los pesos de algunas unidades a 0.

6.6. Redes neuronales recurrentes

A fin de inferir data secuencial, tal como texto o señales en el tiempo, aparecen las redes neuronales recurrentes. La forma más simple de una red neuronal recurrente tiene múltiples copias de la misma red, cada una pasando información a su sucesora. La primer red neuronal recurrente, era una MLP con un bucle hacia si misma. Definiendo $x(t)$, $\hat{y}(t)$, $\hat{z}(t)$ como la entrada, la salida y la capa intermedia en tiempo t respectivamente.

$$\hat{y}^{(k)}(t) = \sum_{i=1}^{I} \boldsymbol{W}_i^{(k)} \hat{z}_i(t) + \boldsymbol{b}^{(k)}$$

$$\hat{z}_i(t) = \sigma\left(\sum_{j=1}^{J} w_{i,j} x_j(t) + \sum_{l=1}^{I} \tilde{w}_{i,l} \hat{z}_l(t-1) + b_i\right)$$

Donde σ es la función de activación. Las neuronas que son retroalimentadas asimismas, se denominan *context units* o unidades contextuales. Estos modelos y variaciones del mismo se han utilizado en el campo del análisis lingüístico. Aunque, nuevas arquitecturas se han desarrollado para abordar estos problemas.

Long Short-Term Memory

Las RNN han sido exitosas en varias aplicaciones como han de ser reconocimiento del habla, traducción, entre otros. Este éxito se debe a la eficiencia de las redes LSTM, que es una especie de red recurrente. Las redes LSTM fueron introducidas por Hochreiter y Schmidhuber (1997) [5] y fueron creadas con el propósito de aprender dependencias a largo plazo. Una celda LSTM comprende, en un instante t, un estado C_t y una salida h_t. Como entrada, la celda requiere x_t, C_{t-1}, h_{t-1}. Dentro de la celda, existen compuertas o *gates* que permiten, o no, el paso de información. Este comportamiento está dado por el siguiente conjunto de ecuaciones.

- *Update gate H*

$$u_t = \sigma(\boldsymbol{W}^u h_{t-1} + \boldsymbol{I}^u x_t + b^u) \tag{6.14}$$

- *Forget gate H*

$$f_t = \sigma(\boldsymbol{W}^f h_{t-1} + \boldsymbol{I}^f x_t + b^f) \tag{6.15}$$

- *Cell candidate H*

$$\tilde{C}_t = \tanh(\boldsymbol{W}^c h_{t-1} + \boldsymbol{I}^c x_t + b^c) \tag{6.16}$$

- *Cell output H*

$$C_t = f_t \odot C_{t-1} + u_t \odot \tilde{C}_t \tag{6.17}$$

- *Output gate H*

$$o_t = \sigma(\boldsymbol{W}^o h_{t-1} + \boldsymbol{I}^o x_t + b^o) \tag{6.18}$$

- *Hidden output H*

$$h_t = o_t \odot \tanh(C_t) \tag{6.19}$$

- *Output K*

$$y_t = \text{softmax}(\boldsymbol{W} \cdot h_t + b) \tag{6.20}$$

- *Recurrent weights $H \times H$*

$$\boldsymbol{W}^u, \boldsymbol{W}^f, \boldsymbol{W}^c, \boldsymbol{W}^o \tag{6.21}$$

- *Input weights $N \times H$*

$$\boldsymbol{I}^u, \boldsymbol{I}^f, \boldsymbol{I}^c, \boldsymbol{I}^o \tag{6.22}$$

- *Biases*

$$b^u, b^f, b^c, b^o \tag{6.23}$$

La Figura 6.5 refleja la diferencia entre una celda RNN tradicional y una celda LSTM. Una celda RNN contiene sólo una capa. En cambio, la celda LSTM contiene 4 capas dadas por los bloques amarillos, interactuando entre ellas como se describen en las ecuaciones anteriores.

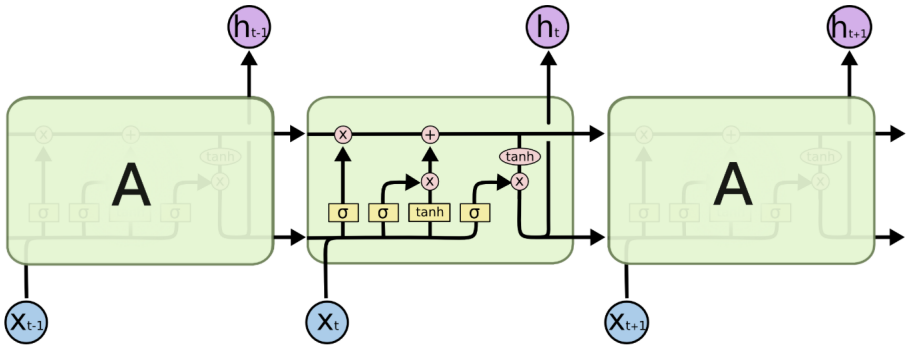


Figura 6.5: Diagrama de una red LSTM. Imagen extraída de *"Understanding LSTM Networks"*

Las redes LSTM son las elegidas en este trabajo. La particularidad de estas redes, como se ha mencionado anteriormente, es la eficiencia para los problemas de señales en el tiempo.

6.7. Sobre-entrenamiento

Estadísticamente hablando sobre-entrenamiento (*overfitting*) se produce cuando un modelo corresponde exactamente al conjunto de datos al que utiliza para estimar sus parámetros. Esto produce que predecir observaciones futuras sea confiable. Un modelo sobre-entrenado generalmente tiene muchos más parámetros de los que se pueden justificar con los datos de entrenamiento. Esencialmente, un modelo de estas características tiene por aprender información descorrelacionada con los datos, por ejemplo ruido, como si fuese información representativa del modelo.

Existe otro fenómeno de los modelos estadísticos, denominado sub-entrenamiento (*underfitting*), que contrario al sobre-entrenamiento ocurre cuando el modelo no es capaz de capturar la estructura de los datos, y sus parámetros quedan en un estado indefinido.

Validación cruzada

Antes de entrar en detalle en la extracción de los atributos tiempo-frecuencia, es necesario definir el método de entrenamiento elegido en este trabajo.

Validación cruzada (*cross-validation*) es un método de validación de un modelo para analizar los resultados estadísticos tal que éstos se encuentren los más generalizado a cualquier set de datos. Es comúnmente usado donde el objetivo es la predicción, y se quiere saber cuán exacto y preciso el modelo predictivo es en la práctica. En un problema de predicción, se define un set de datos de entrenamiento y otro conunto de datos de evaluación. El primero, es un conjunto de datos en el cual el modelo predictivo estimará sus parámetros para luego predecir datos que no hayan sido visto nunca. Aquí es cuando entra el conjunto de datos de evaluación, que va a ser visto por primera vez por el modelo. Este método tiene como objetivo solucionar el problema de sobre-entrenamiento.

Existen varias técnicas dentro de validación cruzada. En este trabajo se hará hincapié en el denominado *K-Folds*. Éste consiste en dividir el set completo de datos

en K grupos (generalmente es común utilizar $K = 10$). El entrenamiento por ende se realizará K veces, eligiendo en cada iteración un grupo de evaluación que no haya sido elegido previamente. El resto de los grupos se utilizarán como entrenamiento. De esta manera se computarán todas las métricas elegidas, K veces y se computará la media, y el desvío de cada una de ellas.

6.8. Métricas

Determinar métricas de error es fundamental a la hora de evaluar cuáles son los próximos pasos a seguir.

Hay que tener en cuenta que en algunas aplicaciones es prácticamente imposible obtener un error nulo. El error de Bayes define el mínimo error que se puede esperar alcanzar, aún obteniendo una cantidad infinita de datos y recuperando la verdadera función de distribución. Este se debe a que los atributos pueden no contener toda la información que se necesita para explicar la variable de salida o porque el sistema a analizar puede ser intrínsecamente estocástico. En la práctica uno siempre va a estar limitad por una cantidad finita de datos.

La cantidad de datos puede estar limitada por varios motivos. Por ejemplo, cuando el objetivo es construir un producto o servicio comercializable, uno puede siempre conseguir más datos pero es necesario evaluar cuál es el costo del mismo comparado a tratar de reducir el error. La recolección de datos implica tiempo, dinero y esfuerzo humano. Un ejemplo típico del esfuerzo humano, cuando se necesita conseguir datos de un paciente clínico por medio de técnicas invasivas. En el ámbito académico o de investigación, cuando el objetivo es responder una pregunta científica sobre qué algoritmo tiene mejor rendimiento en base a un estándar (*benchmark*), no es posible agregar más datos.

En muchas aplicaciones la exactitud (*accuracy*) alcanza para definir el rendimiento de los algoritmos. Sin embargo, en muchas ocasiones esto no es así. Es el ejemplo de la detección de una rara enfermedad, en la que una persona en un millón padecen esta enfermedad. Por ende, una forma fácil de alcanzar 99.9999 % de exactitud es simplemente diciéndole al equipo que reporte que en todos los casos la enfermedad se encuentra ausente. Claramente, esta métrica no es útil y una forma de resolver esto es utilizando otras métricas como son la precisión y sensibilidad (*recall*). La precisión se define como la cantidad de detecciones que el algoritmo definió como correctas y sensibilidad es la cantidad de eventos verdaderos que fueron detectados. Un detector que dice que nadie tiene esta enfermedad, tiene 100 % de precisión pero 0 sensibilidad. Un detector que dice que todos tienen la enfermedad, tiene 100 % de sensibilidad pero precisión igual a la cantidad de las personas que sí la tienen, 0.0001 % en este ejemplo. En muchas aplicaciones, es bueno resumir esta información en una sola métrica. Esta métrica se conoce como métrica F_1.

$$F_1 = 2\frac{PR}{P+R} \tag{6.24}$$

Otra opción es reportar el área bajo la curva P-R, donde en el eje de abscisas se encuentra R y en el eje de ordenadas, P.

Muchas otras métricas son posibles. En distintas aplicaciones especializadas, existen métricas característica de ese campo.

Una manera de mostrar rendimiento de un algoritmo de estimación, es por medio de la matriz de confusión. Esta matriz representa en sus columnas las clases verdaderas y las filas contienen a las clases estimadas. Esta matriz resume distintas métricas, de las cuales es posible obtener resultados. Estas métricas se especifican a continuación.

- Sensibilidad

$$TruePositiveValue(TPV) = \frac{TruePositive(TP)}{TP + FalseNegative(FN)} \tag{6.25}$$

- Especificidad

$$TrueNegativeRate(TNR) = \frac{TrueNegative(TN)}{TN + FalsePositive(FP)} \tag{6.26}$$

- Precisión

$$PositivePredictiveValue(PPV) = \frac{TP}{TP + FP} \tag{6.27}$$

- Valor Predictivo Negativo

$$NegativePredictiveValue(NPV) = \frac{TN}{TN + FN} \tag{6.28}$$

- Tasa de error

$$FalseNegativeRate(FNR) = \frac{FN}{FN + TP} \tag{6.29}$$

- Tasa de falsos positivos

$$FalsePositiveRate(FPR) = \frac{FP}{FP + TN} \tag{6.30}$$

- Tasa de descubrimiento

$$FalseDiscoveryRate(FDR) = \frac{FP}{FP + TP} \tag{6.31}$$

- Tasa de omisión

$$FalseOmissionRate(FOR) = \frac{FN}{FN + TN} \tag{6.32}$$

- Exctitud

$$Accuracy(ACC) = \frac{TP + TN}{TP + TN + FN + FP} \tag{6.33}$$

- Puntaje F_1

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{6.34}$$

Las métricas que aquí se trabajarán son exactitud (ACC), precisión (P_+), sensibilidad (Se) y F_1. Los motivos de esta decisión es la naturaleza de la aplicación y por comparación de desempeño con otros trabajos publicados.

TP define a los verdaderos positivos y TN a los verdaderos negativos. En base a estos dos parámetros se pueden calcular todas las métricas mencionadas. Por otro lado, cabe destacar que dependiendo de la cantidad de clases que el problema posea, es necesario computar TP y TN para cada una de ellas . Más adelante, se dará un ejemplo en el caso de 4 clases.

Capítulo 7

Implementación

Este capítulo tiene como objetivo mostrar y reflejar el trabajo hecho. El preprocesamiento y el procesamiento se ha explicado en capítulos anteriores. Aquí se mencionarán los elementos necesarios para la ejecutar la clasificación, como por ejemplo el *framing*. Se describirá la arquitectura de la red LSTM utilizada, junto a la descripción de sus capas intermedias y los hiperparámetros seleccionados.

Diagrama de flujo del sistema

Antes de abordar cada uno de los distintos pasos, se ilustra una diagrama de flujo de todo el sistema. Empezando desde el filtrado lineal para eliminar ruido que corrompa la señal, pasando por la extra cción de marcas y la extracción de atributos para dar finalmente con la clasificación de los estados de la señal (diagrama que contempla tanto la etapa de entrenamiento como la de evaluación).

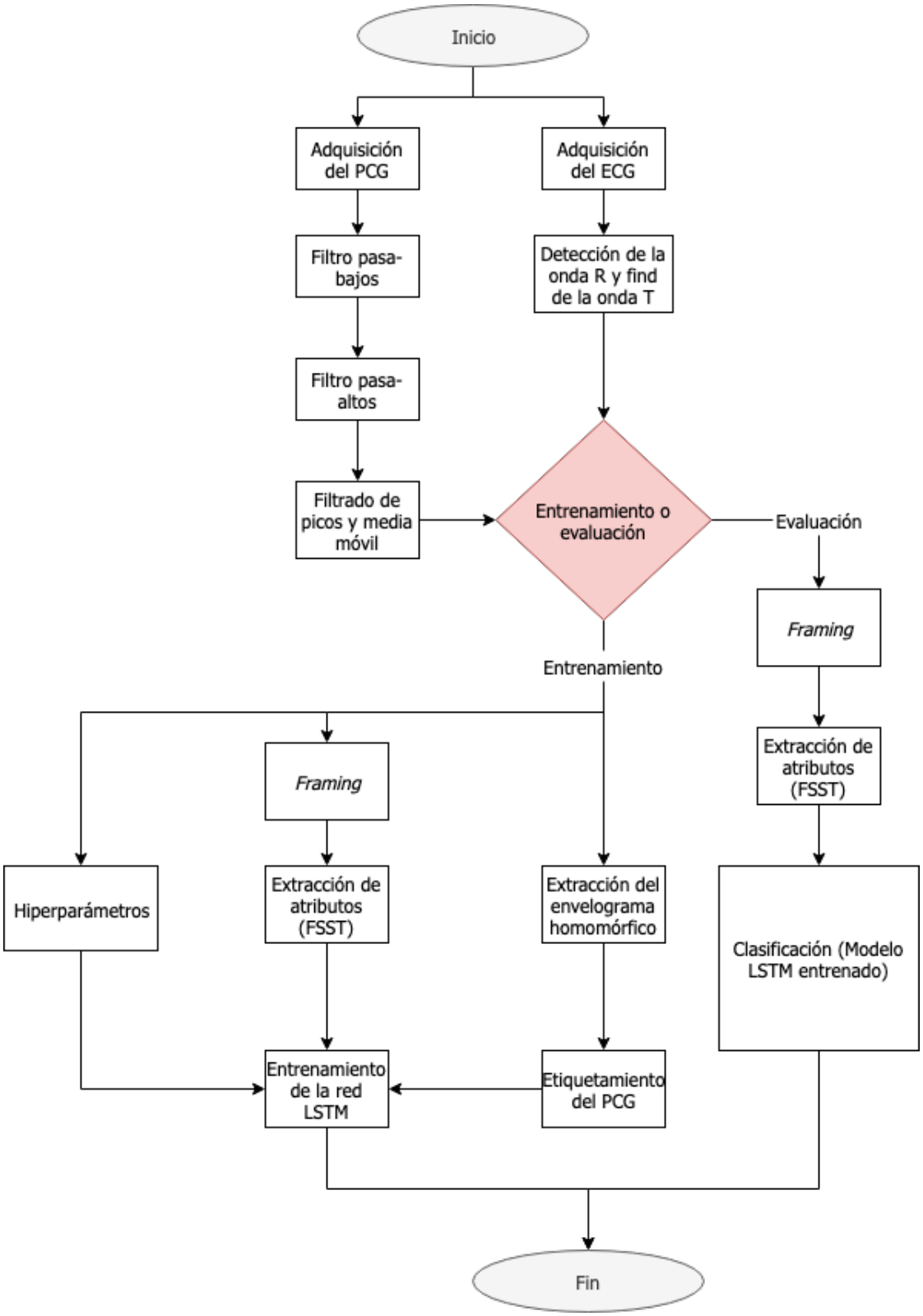


Figura 7.1: Diagrama de flujo del sistema.

7.1. Encuadrado

El proceso de dividir una señal de una dada longitud L en señales de igual longitud se denomina encuadrado (*framing*). Ésto es necesario debido a que las señales de fonocardiograma poseen longitudes diferentes. La base de datos, ya se ha mencionado, tienen adquisiciones de distintas duraciones, y la arquitectura propuesta más adelante, necesita como entrada M señales con una longitud N fija.

El encuadrado es posible realizarlo con una ventana cuadrada, o cualquier otro tipo de ventana. La elección de la ventana esta dada por el problema a resolver. En este caso, se utilizó una ventana cuadrada.

Dada una señal $\mathbf{x} \in \mathbb{R}^T$, se desea obtener una cantidad Q de señales. Este valor está dada por la ecuación 7.1

$$Q := \left[\frac{T-1-N}{\tau}\right] \tag{7.1}$$

La notación [...] implicá el entero más cercano. El deslizamiento de la ventana está dado por τ, dependiendo de N y de τ, se define si existe solapamiento u *overlapping*.

$$O = \begin{cases} 1, \ 0 \leq \tau \leq N \\ 0, \ \tau \geq N \end{cases} \tag{7.2}$$

De esta manera, queda definido el vector, $\tilde{\boldsymbol{x}}_k \in \mathbb{R}^N$ según la siguiente ecuación.

$$\tilde{\boldsymbol{x}}_k = \begin{bmatrix} \boldsymbol{x}_{\tau \cdot k} & \boldsymbol{x}_{\tau \cdot k+1} & \dots & \boldsymbol{x}_{\tau \cdot k+N-1} \end{bmatrix}^\top \tag{7.3}$$

$$\boldsymbol{X}_j = \begin{bmatrix} \tilde{\boldsymbol{x}}_1 & \tilde{\boldsymbol{x}}_2 & \dots & \tilde{\boldsymbol{x}}_Q \end{bmatrix}^\top \tag{7.4}$$

Donde $k = 1, 2, ..., Q$. Por último, una vez obtenidos los Q cuadros para una señal j, donde $j = 1, 2, ..., D$ y D la cantidad total de señales, se itera sobre todo el set de datos y se genera la matriz $\boldsymbol{H} \in \mathbb{R}^{M \times N}$.

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 & \dots & \boldsymbol{X}_D \end{bmatrix}^\top \tag{7.5}$$

Es importante tener en cuenta que en los casos que $\frac{T-1-N}{\tau}$ no sea entero, quedarán muestras (generalmente del final) sin incluir en los datos, las cuales serán descartadas.

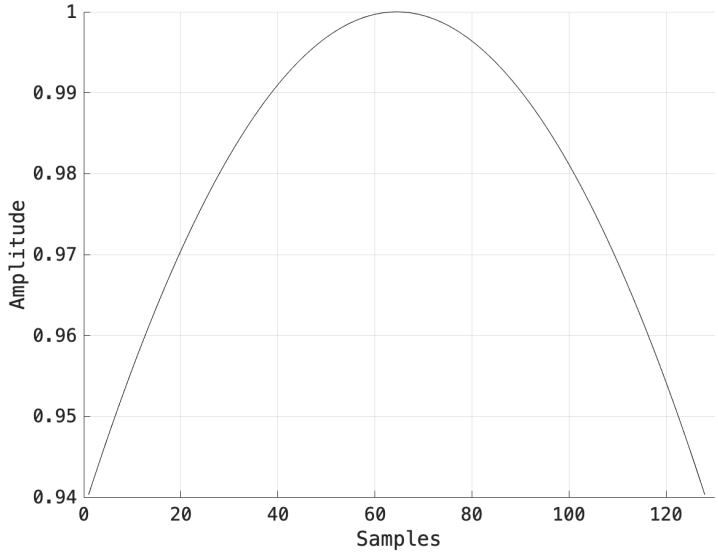
7.2. Extracción de atributos

Una vez hecho el acondicionamiento de la señal (filtrado, normalización en términos de energía) y los cuadros listos, se procede a extraer los features necesarios a introducir al clasificador. Los features extraídos son obtenidos por medio de la FSST, el clasificador es una red diseñada con celdas LSTM.

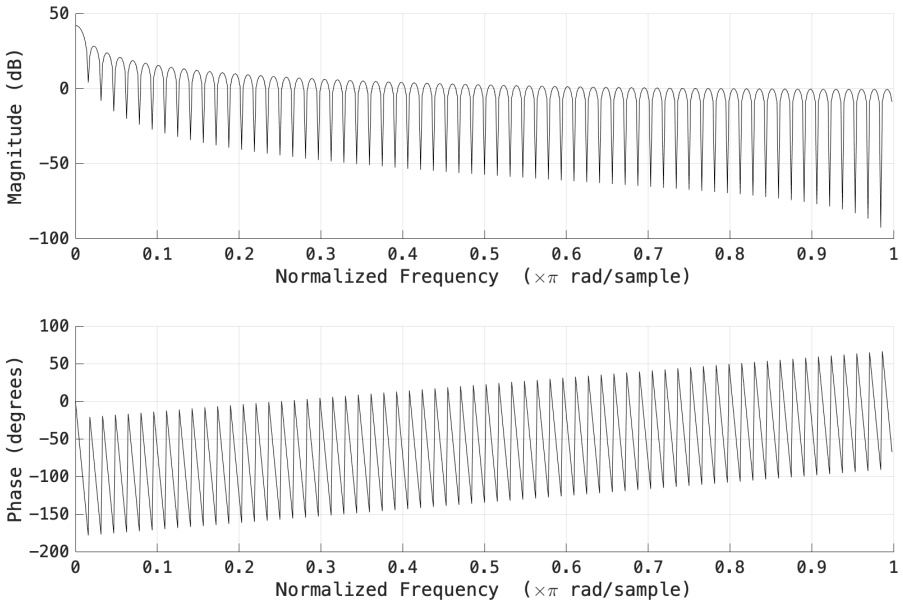
Para la extracción de features en tiempo-frecuencia se utilizó la FSST. Esta se aplica sobre segmentos de señal extraídos del proceso de encuadrado.

En esta ocasión, no se utiliza solapamiento de cuadros (*frames*), con lo cual cada segmento de señal posee información única a excepción de una muestra. Previamente, se eligen las señales mayores o iguales a una longitud N, cuyo valor va a ser el largo de cada segmento.

A cada segmento se le aplica la transformada con una ventana definida. La ventana elegida es la ventana de Kaiser con una longitud $L = 128$ y un $\beta = 0.5$. Esta ventana fue seleccionada dado a que su objetivo es maximizar la relación de energía entre lóbulo principal y sus lóbulos secundarios, reduciendo los efectos del ruido en esas bandas de frecuencia y mejorando la calidad de la transformada.



(a) Ventana de Kaiser. La ventana diseñada con un largo $L = 128$ y un $\beta = 0.5$.



(b) Respuesta en frecuencia de la ventana de Kaiser. La respuesta se encuentra en decibeles y se ilustra tanto la magnitud como la fase.

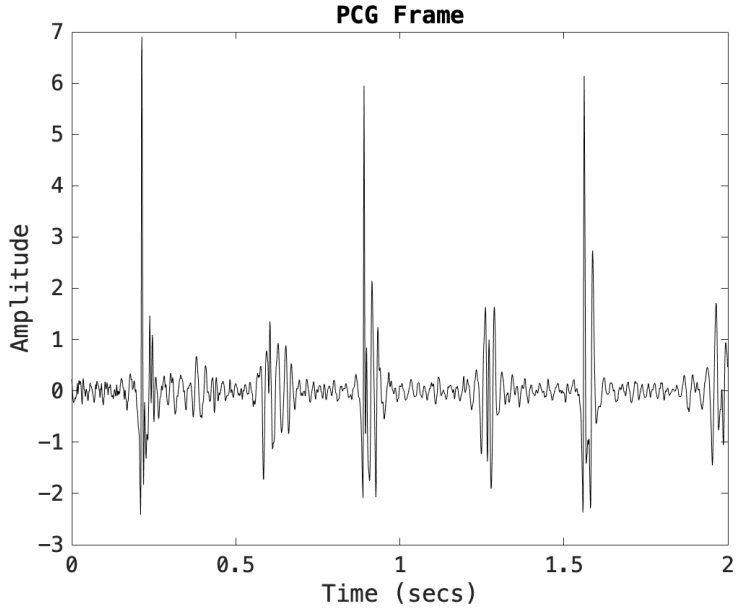
El parámetro β se calcula en base a la ecuación 7.6.

$$\beta = \begin{cases} 0.1102(\alpha - 8.7), \ \alpha > 50 \\ 0.5842(\alpha - 21)^{0.4} + 0.7886(\alpha - 21), \ 50 \geq \alpha \geq 21, \\ 0, \ \alpha \geq < 21 \end{cases} \tag{7.6}$$

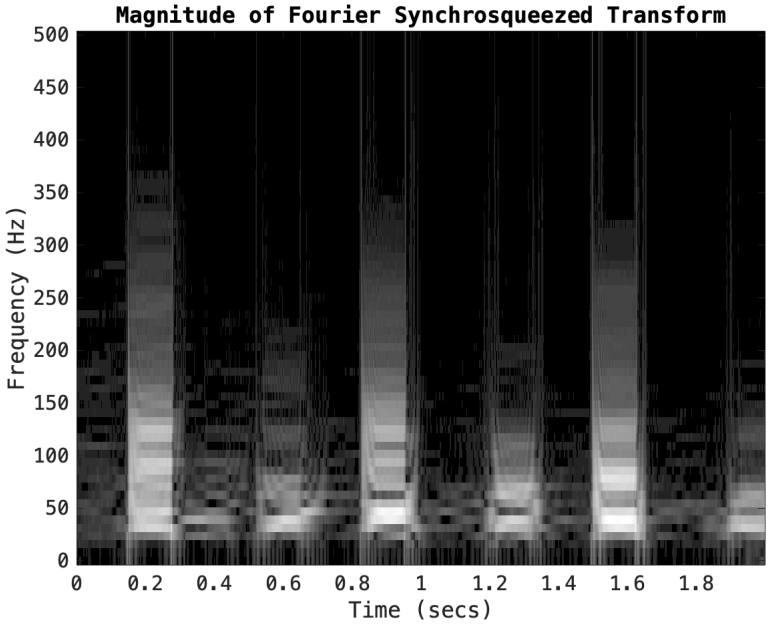
Y la banda de transición, según el largo de la ventana, se calcula despejando $\Delta\omega$ de la ecuación 7.7.

$$L = \frac{\alpha - 8}{2.285\Delta\omega} + 1 \tag{7.7}$$

En la Figura 7.3 se ilustra la transformada extraída de una señal de fonocardiograma. Se visualiza claramente el contenido frecuencia en los distintos sonidos del fonocardiograma y la periodicidad de los mismos. Por otro lado, se muestra que por debajo de frecuencias de los 20 Hz y por encima de los 200 Hz no hay contenido espectral relevante en cuanto a energía. La mayor cantidad de contenido frecuencia se encuentra entre dicho rango. Por ende, se extraen los features entre 20-200 Hz como entrada al clasificador.



(a) Segmento de una señal de fonocardiograma.



(b) Magnitud de la FSST aplicada a un segmento de señal de fonocardiograma.

Figura 7.3: Ejemplo de la transformada FSST a un segmento de señal de PCG.

Se ve claramente como la mayor energía frecuencial corresponde a los instantes donde se producen los sonidos fundamentales. También, en este ejemplo, hay energía proveniente de otras fuentes de ruido, entre los instantes 1-1 .2 segundos.

7.2.1. Modelo

El clasificador es una red neuronal LSTM como se ha mencionado anteriormente. Los parámetros a estimar del modelo pertenecen a la capa LSTM. Por otro lado, es predefinir los hiperparámetros de la red neuronal. Además de la capa recurrente,

a una red neuronal la componen otras capas intermedias y una capa de entrada y salida.

Arquitectura

La arquitectura de la red neuronal consiste en de 5 capas. Una de entrada, una recurrente, una *fully-connected* con otra de softmax y una de salida. La arquitectura se ilustra en la Figura 7.4.

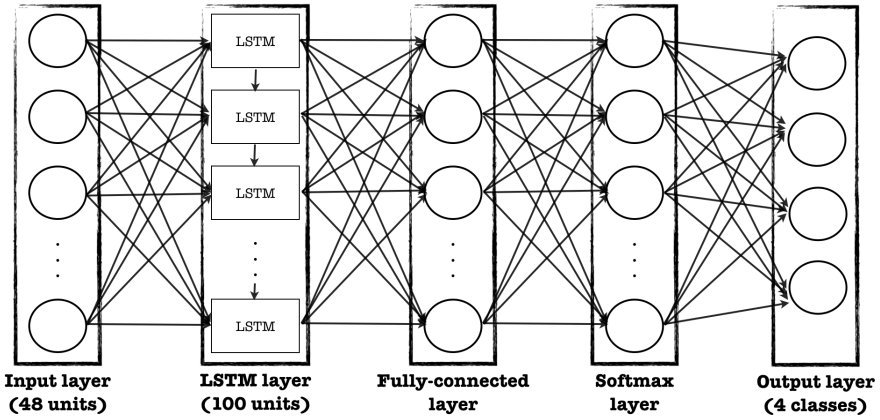


Figura 7.4: Arquitectura de la red neuronal.

Hiperparámetros

Ya se ha visto que es necesario establecer hiperparámetros de la red. Para el entrenamiento se utiliza el algortimo publicado en 2014 en la International Conference on Learning Representations (ICLR) 2015, denominado *Adam* [28]. El algoritmo introduce ciertos hiperparámetros además de los convencionales que es necesario definir a priori.

Los hiperparámetros utilizados en la implementación se listan a continuación [1].

- `MaxEpochs: 10`
- `MiniBatchSize: 50`
- `InitialLearngRate: 0.01`
- `LearnRateDropPeriod: 3`
- `LearnRateDropFactor: 0.1`
- `L2Regularization: 1e-04`
- `GradientDecayFactor: 0.9`
- `SquaredGradientDecayFactor: 0.99`
- `Epsilon: 1e-08`

[1]Los nombres de los hiperparámetros se encuentran en inglés, dado a que no hay literatura en español y muchos de ellos no tienen traducción

- `GradientThreshold: 1`

Generalmente, lo normal es elegir la cantidad de *epochs* alrededor de 10. En disintas aplicaciones se han utilizado un número entre $15 - 30$. Esto hace que el tiempo de entrenamiento sea aún mayor. Recordar que por cada una de ellas todo el set de datos ha pasado por la red. Por temas de cómputo y tiempos, se decidió fijar la cantidad de *epochs* en el valor estándar. Por otro lado, bastante ligado a la cantidad de *epochs* y el tamaño del lote se definió por motivos computacionales, tiempo y desempeño de la red por medio de distintas iteraciones.

Otros parámetros de interés, son *InitialLearnRate*, *LearnRateDropPeriod* y *LearnRateDropFactor*. Éstos son necesarios dado a que el *LearnRate* se modifica a lo largo de las iteraciones, y con estos parámetros se define cada cuánto y por cuánto se reduce el mismo. En esta implementación se definió que cada 3 *epochs* se reduce un 1 %.

Lo que se conoce como *Regularization* intenta resolver el problema de generalización o sobreentrenamiento en lo que respecta a los pesos de la red. También se lo denomina, en inglés, *weight decay*. Estos pesos se encuentran asociados a la función de costo y *L2Regularization* se encuentra representado como un factor. En esta implementación toma el valor de 1e-04. La función de costo, $E_R(\theta)$, con el factor de regularización tiene la forma de la ecuación 7.8.

$$E_R(\theta) = E(\theta) + \lambda\Omega(w) \tag{7.8}$$

Donde,

$$\Omega(w) = \frac{1}{2}w^\top w \tag{7.9}$$

$\Omega(w)$ es la función de regularización y w son los pesos, la cual se encuentra multiplicada por el factor λ correspondiente a *L2Regularization*.

Luego, los factores inherentes al algoritmo Adam, son *GradientDecayFactor* y *SquaredGradientDecayFactor*. Normalmente en la mayoría de los casos toman el valor de 0.9 y 0.99 respectivamente.

Adam utiliza medias móviles para actualizar los parámetros de la red. Estas medias se utilizan para el gradiente y el gradiente al cuadrado, según las ecuaciones 7.10, 7.11.

$$m_l = \beta_1 m_{l-1} + (1 - \beta_1)\nabla E(\theta_l) \tag{7.10}$$

$$v_l = \beta_2 v_{l-1} + (1 - \beta_2)[\nabla E(\theta_l)]^2 \tag{7.11}$$

Donde, β_1 y β_2 son los hiperparámetros *GradientDecayFactor* y *SquaredGradientDecayFactor*.

$$\theta_{l+1} = \theta_l + \frac{\alpha m_l}{\sqrt{v_l} + \epsilon} \tag{7.12}$$

La ecuación 7.12 define cómo se actualizan los parámetros de la red con las medias móviles. Por otro lado, si los gradientes durante varias iteraciones son similares, el uso de las medias móviles de los gradientes permiten a la actualización de los parámetros tomar momento hacia una dirección. Existe la posibilidad de que los gradientes sólo contengan ruido con lo cual, las medias móviles de los gradientes serán pequeñas y así también actualización. Para ellos se elige un valor ϵ tal que la actualización esa no diverja, ya que $\sqrt{v_l}$ puede ser muy pequeño. En muchas ocasiones se utiliza $\epsilon = 0.01$ pero en aplicaciones un valor cercano a 1 funciona mejor. Queda definir el parámetro α como el *LearnRate* mencionado. Es posible que este parámetro tome diferentes valores para distintas capas intermedias y depende del algoritmo en cuestión. Es por eso que no se puede definir un valor estándar de este hiperparámetro.

Por último, queda definir el hiperparámetro *GradientThreshold*. Este umbral intenta acortar el gradiente si los valores lo exceden. El método se basa en utilizar la norma L2 tal que si la norma del gradiente excede el umbral, se escala el gradiente tal que su norma lo igual. En este caso el umbral toma el valor de 1.

7.2.2. K-Folds

Recoradar que la técnica utilizada para resolver el problema de generalidad en el momento del entrenamiento es la ya mencionada en el Capítulo 6, validación cruzada (particularmente *10-Folds*). Para separar los fonocardiogramas en distintos grupos, se define el siguiente algoritmo.

Sea la señal de fonocardiograma $\mathbf{s}_i \in \mathbb{R}^N$, sus etiquetas (previamente calculadas) $\mathbf{l}_i \in \mathbb{R}^N$ y la cantidad de *folds* K. La cantidad de fonocardiogramas está dada por el valor L, por lo tanto $i = 0, 1, 2, \ldots, L-1$.

De esta manera se define M como la cantidad de PCG por *fold*.

$$M = \left\lfloor \frac{L}{K} \right\rfloor \tag{7.13}$$

Por lo tanto la ecuación 7.14 define cómo asignar cada PCG a un *fold*.

$$F_{n,j} = i, \quad (n \cdot M) < i < (n+1) \cdot (M-1) \tag{7.14}$$

Con $i = 0, 1, 2, \ldots, M-1$ y $n = 0, 1, 2, \ldots, K-1$. De esta manera se define la matriz $\boldsymbol{F} \in \mathbb{R}^{K \times M}$, que contiene los indices correspondientes a cada *fold*.

$$\boldsymbol{F} = \begin{bmatrix} F_{0,0} & F_{0,1} & \ldots & F_{0,M} \\ F_{1,0} & F_{1,1} & \ldots & F_{1,M} \\ \vdots & \vdots & \ddots & \vdots \\ F_{K,0} & F_{K,1} & \ldots & F_{K,M} \end{bmatrix} \tag{7.15}$$

Si $\frac{L}{K}$ no fuese un número entero, significaría que no todos los *folds* contendrán la misma cantidad. La cantidad de señales huérfanas se calcula en la ecuación 7.16.

$$L_s = \mathrm{mod}(L, K) \tag{7.16}$$

En este momento es cuando se decide eliminar esas señales, agregar más para que la matriz $\mathbf{F}$ sea consistente o se agregan de manera aleatoria a cualquier *fold*.

Para definir los distintos grupos, los cuales serán K, cada grupo contenerá $K-1$ folds de entrenamiento y 1 *fold* de evaluación.

Se definen la matriz de entrenamiento, $\boldsymbol{T}_p \in \mathbb{R}^{K-1\times M}$ y la matriz de evaluación $\mathbf{E}_p \in \mathbb{R}^M$. Ambos asociados a un grupo $P \in \{0, 1, 2, \ldots, K-1\}$

$$\mathrm{fil}_i(\boldsymbol{T}_p) = \mathrm{fil_j(F)}, \quad j \in \{0, 1, 2, \ldots, K-1\} - \{p\} \tag{7.17}$$

$$\boldsymbol{E}_p = \mathrm{fil_p}(\boldsymbol{F}) \tag{7.18}$$

De esta manera, quedan definidos los grupos para realizar los K entrenamientos y ponderar métricas de performance.

7.3. Clasificación

La clasificación se realiza a partir de las probabilidades de la matriz $\boldsymbol{B} \in \mathbb{R}^{n\times 4}$. Esta matriz es la que la capa de salida genera. En ella la se encuentra la probabilidad de que cada muestra de la señal pertenezca a alguna clase. La forma de elegir la clase se muestra en la ecuación 7.19. Esto es lo que generalmente la mayoría de las redes neuronales a su salida realizan como forma de clasificación, es el caso de la segmentación mediante la una adaptación de la red neuronal *U-Net* de Renna *et al.* [11].

$$C_i = \arg\max_j B_{i,j} \tag{7.19}$$

Es necesario mencionar que en el entrenamiento de la red neuronal no se ha impuesto ninguna restricción de transición de estados, a diferencia de lo que realmente sucede en el fonocardiograma. De esta manera, existen en la predicción y por ende en la clasificación transiciones espurias que no corresponden. En la Figura 7.5 se ilustra este fenómeno.

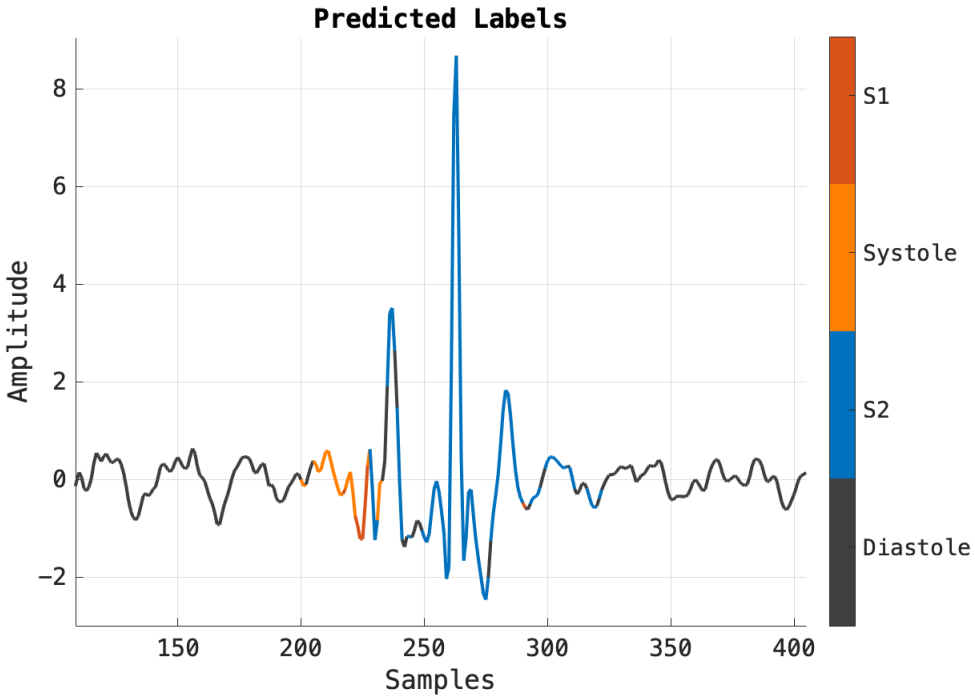


Figura 7.5: Etiquetas predecidas por la red.

Para corregir la transición de estado espúrios los posteriores capítulos se propondrán algunas técnicas. Esto ayudará a la efectividad de la detección y para mantener la consistencia de los estados. Este ruido no es deseado en algoritmos que utilicen las transiciones del fonocardiograma.

Capítulo 8

Discusión

8.1. Resultados

Ya se ha mencionado la necesidad de realizar comparaciones con diferentes algoritmos que tratan de resolver el problema de segmentación de fonocardiogramas a través de un entrenamiento supervisado con diferentes modelos como han de ser redes neuronales convolucionales y cadenas ocultas de Markov, entre otros.

Los resultados se ilustran en los siguientes cuadros donde se ha aplicado la técnica *10-Folds* para cada estado y luego se ha promediado entre todos ellos. La cantidad de atributos extraídos para el entrenamiento ha sido de 44 para un set de datos de 734, 590 y 269 señales tomando segmentos de 2 ms, 3 ms y 5 ms respectivamente. De esta manera la relación de señales-atributos, en el peor caso, es de 6.11 y en el mejor caso, es de 16.68. Esta relación es un índice de generalización del modelo, donde en la literatura y en el campo de la investigación, 10 es el valor mínimo aceptable.

Número de señales	L	F_1	Acc	P_+	Se
269	5 ms	87.3 ± 3.2 81.4 ± 4.0	95.6 ± 2.0 93.5 ± 3.7	87.9 ± 2.2 81.5 ± 2.5	86.6 ± 2.1 81.4 ± 2.8
590	3 ms	88.6 ± 3.4 82.4 ± 3.8	96.0 ± 1.9 93.7 ± 2.1	89.2 ± 2.2 82.6 ± 2.9	88.0 ± 1.8 82.2 ± 2.1
734	2 ms	88.2 ± 2.3 83.7 ± 2.9	95.9 ± 1.7 94.2 ± 3.1	88.1 ± 3.4 83.0 ± 4.1	88.3 ± 2.2 84.5 ± 2.3

Cuadro 8.1: Tabla con las métricas para el sonido 1 (S1). L es el largo de la ventana elegida sin solapamiento. F_1 es el promedio armónico, Acc es la exactitud, P_+ es lo que se conoce como precision y Se es la sensibilidad.

Número de señales	L	F_1	Acc	P_+	Se
269	5 ms	89.6 ± 3.3 81.4 ± 3.5	95.2 ± 3.1 92.9 ± 3.4	89.7 ± 2.3 82.1 ± 2.7	89.6 ± 2.1 80.7 ± 2.7
590	3 ms	90.7 ± 1.4 82.9 ± 2.2	95.7 ± 2.0 92.1 ± 2.4	91.5 ± 2.1 83.7 ± 2.7	89.9 ± 2.9 82.1 ± 3.1
734	2 ms	89.5 ± 2.6 84.0 ± 2.9	95.2 ± 1.0 92.6 ± 1.8	90.8 ± 2.1 85.6 ± 2.5	88.3 ± 3.2 82.5 ± 3.5

Cuadro 8.2: Tabla con las métricas para el sístole isovolumétrico (Sys). L es el largo de la ventana elegida sin solapamiento. F_1 es el promedio armónico, Acc es la exactitud, P_+ es lo que se conoce como precision y Se es la sensibilidad.

Número de señales	L	F_1	Acc	P_+	Se
269	5 ms	87.4 ± 3.6 81.4 ± 3.9	96.8 ± 3.1 95.4 ± 3.5	88.1 ± 2.7 82.1 ± 3.0	86.7 ± 2.2 80.7 ± 2.5
590	3 ms	88.8 ± 1.9 80.1 ± 2.3	97.2 ± 2.0 95.0 ± 2.7	89.5 ± 2.9 82.1 ± 3.1	88.2 ± 3.1 78.2 ± 3.4
734	2 ms	88.8 ± 1.9 83.8 ± 2.1	97.2 ± 2.1 95.9 ± 2.2	89.7 ± 2.5 84.5 ± 3.1	87.9 ± 2.8 83.2 ± 3.0

Cuadro 8.3: Tabla con las métricas para el sonido 2 (S2). L es el largo de la ventana elegida sin solapamiento. F_1 es el promedio armónico, Acc es la exactitud, P_+ es lo que se conoce como precision y Se es la sensibilidad.

Número de señales	L	F_1	Acc	P_+	Se
269	5 ms	95.0 ± 1.7 91.9 ± 2.1	94.9 ± 2.1 91.6 ± 2.5	94.6 ± 1.9 90.9 ± 2.1	95.5 ± 1.5 93.0 ± 2.1
590	3 ms	95.2 ± 1.3 90.7 ± 1.9	95.1 ± 2.0 90.2 ± 2.1	94.5 ± 2.1 89.7 ± 2.3	96.0 ± 1.5 91.6 ± 2.0
734	2 ms	94.7 ± 2.6 92.1 ± 2.9	94.6 ± 1.0 91.8 ± 1.5	93.9 ± 2.1 91.4 ± 2.5	95.5 ± 2.9 92.7 ± 3.2

Cuadro 8.4: Tabla con las métricas para la diástole isovolumétrica (Dias). L es el largo de la ventana elegida sin solapamiento. F_1 es el promedio armónico, Acc es la exactitud, P_+ es lo que se conoce como precision y Se es la sensibilidad.

Número de señales	L	F_1	Acc	P_+	Se
269	5 ms	89.8 ± 3.1 84.9 ± 4.3	95.6 ± 0.7 93.3 ± 1.4	90.1 ± 2.7 85.3 ± 3.8	89.6 ± 3.6 84.5 ± 5.0
590	3 ms	90.8 ± 2.7 84.0 ± 4.0	96.0 ± 0.8 92.8 ± 1.8	91.2 ± 2.1 84.5 ± 3.0	90.5 ± 3.2 83.5 ± 4.9
734	2 ms	90.3 ± 2.6 85.9 ± 3.6	95.7 ± 1.0 93.6 ± 1.6	90.6 ± 2.1 86.1 ± 3.2	90.0 ± 3.2 85.7 ± 4.1

Cuadro 8.5: Tabla con las métricas promediadas del sistema. L es el largo de la ventana elegida sin solapamiento. F_1 es el promedio armónico, Acc es la exactitud, P_+ es lo que se conoce como precision y Se es la sensibilidad.

Los resultados muestran que para cada uno de los sonidos las mejores métricas se alcanzan para las señales divididas en cuadros de 3 ms. Sin embargo, dada al desvío que presentan todas ellas, no se puede afirmar definitivamente que aumentando la cantidad de señales y al mismo tiempo reduciendo la longitud de los cuadros se mejore la segmentación.

Por otro lado, haciendo una comparación de las métricas inter-sonidos, es posible afirmar que el mejor desempeño se obtiene para el cuarto estado o diástole silenciosa. Diferente son los otros tres estados que tienen similares métricas. Uno de los potenciales motivos que hace al cuarto estado facilmente clasificable, es la diferencia de duración temporal claramente marcada frente a los demás, y a esto se le suma el bajo contenido de frecuencias altas. Esto parece permitirle a la red detectarlo con facilidad. Contrariamente, el sístole silencioso muchas veces contiene ruidos asociados a los dos sonidos fundamentales y su duración es mucho más corta. Asimismo, las duraciones entre los tres estados son similares. Esto da la iniciativa que la noción temporal de los estados es un factor muy importante a la hora de la segmentación. Ya hemos visto que los algoritmos [7] y [10] donde aplican algoritmos más tradicionales de estimación utilizan el concepto temporal de los distintos estados alcanzando métricas impresionantes. No es así el caso de técnicas más modernas como es *Deep learning* en [11] que explícitamente no queda definida la duración de los estados. De todos modos superan los algoritmos mencionados anteriormente.

Comparación de algoritmos

Los algoritmos comparados para mostrar en dónde se encuentra el presente trabajo se ilustran en el Cuadro 8.6. Estos algoritmo se han probado con el mismo set de datos y diferentes técnicas de procesamiento.

Algoritmos \ Métricas	F_1	Acc	P_+	Se
Schmidt [7]	93.0 ± 3.2	87.4 ± 2.6	93.3 ± 2.8	92.7 ± 3.8
Springer [10]	94.5 ± 1.8	89.8 ± 1.2	94.8 ± 1.8	94.3 ± 1.8
CNN+max [11]	$\mathbf{95.7 \pm 1.3}$	$\mathbf{93.7 \pm 1.0}$	$\mathbf{95.7 \pm 1.4}$	$\mathbf{95.7 \pm 1.2}$
LSTM	84.9 ± 4.3	93.3 ± 1.4	85.3 ± 3.8	84.5 ± 5.0

Cuadro 8.6: Tabla comparativa de las métricas entre diferentes arquitecturas y técnicas. Las señales son *frames* de 5 ms. La arquitectura de Renna *et. al* logra la mayor performance en términos de segmentación.

8.2. Limitaciones

Por otro lado, este trabajo no ha abordado por completo todos los aspectos de la implementación. La optimización del algoritmo, la arquitectura, el postprocesamiento de la clasificación y la segmentación en tiempo real son temas que quedan por resolver para lograr mejorar esta técnica.

Post-procesamiento

En el caso del post-procesamiento pueden aplicar varias técnicas. Éstas tienen relación con dar una restricción de transición de estados. Es decir, que sólo luego de un sonido S_1 proceda una sístole silenciosa, y así con el resto de los estados.

Modelo temporal secuencial máximo

Para restringir la transición de los estados se consigue aplicando la ecuación 8.1.

$$\hat{s}(t) = \begin{cases} \tilde{s}(t), & \tilde{s}(t) = \text{mod}(\hat{s}(t-1)+1, 4) \\ \hat{s}(t-1), & \text{en otro caso} \end{cases} \tag{8.1}$$

Esta ecuación necesita una semilla donde $\hat{s}(0) = \tilde{s}(0)$. La principal ventaja de este método es la baja complejidad algorítmica y la posibilidad de aplicarlo en una segmentación en tiempo real.

Modelado basado en Cadenas Ocultas de Markov (Hidden Markov Model (HMM))

Estrategias más complejas se pueden aplicar para restringir las posibles transiciones. La idea es utilizar las probabilidades a posteriori que la red provee en la matriz **B**. En particular, es posible utilizar esas probabilidades en un modelo HMM que restringue estas transiciones. Este modelo se describe en función de los siguientes parámetros: probabilidad a priori de los estados π, la probabilidad de transiciones de los estados dada por $\gamma_{i,j} = p(s(t) = j|s(t-1) = i)$ y las probabilidades de emisión $e_{t,j} = p(\mathbf{x}(t)|s(t) = j)$. En el caso de π y $\gamma_{i,j}$ se pueden estimar mediante el método de Máxima Verosimilitud (*Maximum Likelihood*) a partir de los datos de entrenamiento y sus etiquetas.

Por otro lado, las emisiones pueden ser calculadas a partir de la matriz **B**. Asumiendo que representan una buena aproximación de la probabilidad a posteriori en el tiempo t dada la observación del vector de emisión $\mathbf{x}(t)$.

$$p(s(t) = j|\mathbf{x}(t)) \sim B_{t,j} \tag{8.2}$$

Y a través del Teorema de Bayes, es posible computar dichas probabilidades.

$$e_{t,j} = p(\mathbf{x}(t)|s(t) = j) = \frac{p(s(t) = j|\mathbf{x}(t)) \cdot p(\mathbf{x}(t))}{p(s(t) = j)} \tag{8.3}$$

En este caso la distribución $p(\mathbf{x}(t))$ es aproximada por una gaussiana multivariable cuya media y matriz de covarianza son estimados a partir de los datos de entrenamiento usando ML.

Luego, el modelo HMM se encarga de determinar la sequencia de estados asociada que maximimiza la función de verosimilitud mediante el algoritmo de Viterbi.

$$\mathcal{L}(s, \mathbf{x}) = p(s(0), \ldots, s(n-1), \mathbf{x}(0), \ldots, \mathbf{x}(n-1)) \tag{8.4}$$

De esta manera la sequencia $\hat{s}(t)$ se consigue a partir de la secuencia que maximiza la función de verosimilitud.

Modelado basado en Cadenas Ocultas de Markov dependientes del tiempo (DHMM)

Las probabilidades de emisiones del método anterior, también son utilizadas en este. Además, se introduce la dependencia del tiempo en donde se modela al tiempo en un estado como una distribución gaussiana donde su media y varianza es estimada a partir de un análisis de autocorrelación explicado en [7]. Una vez más a partir del algoritmo de Viterbi se decodifica la secuencia $\hat{s}(t)$.

Modelado adaptativo basado en Cadenas Ocultas de Markov dependientes del tiempo

Nuevamente las probabilidades de emisión se utilizan en este método. La distribución de los tiempos en un estado siguen siendo gaussianas, sin embargo las medias y varianzas se computan con el método explicado en [29] a partir de la información del PCG. Así se estiman los parámetros que mejor se adaptan al PCG a partir de la maximización de una función de verosimilitud incompleta asociada a la secuencia $\mathbf{x}(t)$, $t = 0, 1, \ldots, T-1$. Por supuesto, la secuencia $\hat{s}(t)$ es estimada por Viterbi.

Capítulo 9

Conclusiones generales

Este trabajo muestra que la correcta selección de atributos adecuado, acompañado de un modelo relativamente simple en la etapa de clasificación, alcanza para obtener una efectividad muy cercana a la del estado del arte. En el cuadro 8.6 se ve que, entre los cuatro algoritmos más famosos en la segmentación de PCG, el de F. Renna y M. Coimbra [11] obtiene las mejores métricas. Éste se basa en el mismo preprocesamiento (acondicionamiento de la señal) y extracción de atributos de los PCG que Springer *et al.* [10], el cual utiliza los métodos propuestos por Schmidt [7] mejorando algunos de los algoritmos.

El presente trabajo utiliza, ya explicado en capítulos anteriores, una DNN con una arquitectura muy simple. Ésta, se ha visto, que contiene sólo una capa recurrente LSTM y diferentes capas intermedias que ayudan a la clasificación. De esta manera, se alcanzan métricas muy cercanas al estado del arte con una buena elección atributos. En cuanto a la exactitud (Acc) queda en segundo lugar pero carece de eficiencia en el resto de las métricas, precisión (P_+) y sensibilidad Se.

La arquitectura de la red LSTM es relativamente simple en comparación con otras arquitecturas, por ejemplo U-Net, de caracter convolucional, donde se aplican mayores grados de profundidad en cuanto a las capas. Por otro lado, posee antecedentes en cuanto a la aplicación de distintos problemas de segmentación. Un ejemplo es el conteo de células, detección y morfometría por medio de imágenes.

9.1. Tiempo de procesamiento

Dado que *Deep learning* se encuentra basado en conexionismo: cuando una neurona o unidad en un modelo de *Machine learning* no es inteligente, un conjunto de neuronas pueden demostrar un comportamiento lo suficientemente inteligente para ciertas aplicaciones. Es importante enfatizar que el número de neuronas debe ser grande para realizar tareas complejas. El tamaño de las redes neuronales han crecido exponencialmente, aunque se comparan con el tamaño del sistema nervioso central de insectos. Por lo tanto, debido a que el tamaño de las redes neuronales es crítico, requiere que una alto desempeño en cuanto a hardware y software.

En definitiva la implementación de los algoritmos, el hardware y el contexto en el que se desea llevar la aplicación imponen restricciones en el tiempo de procesamiento.

Implementaciones en CPU

En un principio el entrenamiento y la evaluación de las redes neuronales se diseñaban para un único CPU. Hoy en día esto no es suficiente. En este trabajo no se prioriza el tiempo de procesamiento, con lo cual la etapa de entrenamiento que depende de muchos factores, como han de ser la cantidad de datos de entrada y la arquitectura de la red, entre otros. Generalmete, la mayoría de estos modelos son utilizados en GPU. Sin embargo, con un cuidado desarrollo e implementación se pueden lograr mejores tiempos de ejecución con implementaciones en CPU.

Implementaciones en GPU

Para el desarrollo de los algoritmos de clasificación, por ejemplo de este trabajo y para la implementación en tiempo real, utilizar GPU acelera el tiempo de procesamiento. Las GPU al ser hardware especializado que permite realizar multiplicación de matrices y división en paralelo, a diferencia de las CPU que para realizar tareas en paralelo es necesario utilizar lo que se conoce como *branching*.

Compresión

La compresión del modelo es una ventaja ante la limitante de memoria y tiempos de lectura y evaluación. Es el caso de utilizar la idea de este trabajo en una suerte de producto en tiempo real donde se considera el tiempo de evaluación mucho más importante que el tiempo de entrenamiento. Esto se debe a que el desarrollador tiene mucho más recursos para realizar las etapas de entrenamiento y evaluación. Asimismmo, aunque sólo sea necesario entrenar el modelo una única vez e implementarlo para su uso, el usuario cuenta con cómputo más barato y menos performante. Aquí, es cuando la compresión del modelo juega un papel muy importante a la hora de llevar a producción una implementación.

9.2. Futuras líneas de trabajo

Nuevas bases de datos

La base de datos es un asunto bastante crítico en cuanto a la generalización de los modelos. Aumentar la cantidad de registros de fonocardiograma con sus anotaciones asociadas es un factor clave para desarrollar aún más la propuesta de este trabajo y muchos otros también. Junto al dispositivo que el IAM del CONICET se encuentra desarrollando, será posible acceder a adquisiciones de nuevas señales. Por otro lado, ligado con la arquitectura de la red, sería una buena línea de trabajo obtener una base de datos basadas en imágenes tiempo-frecuencia [1].

Mejoras en la arquitectura

La arquitectura es una de las principales limitaciones que resultan de este trabajo. En la arquitectura presentada sólo consta con un único nivel de profundidad, lo que hace una primera versión de arquitectura muy simple pero con un gran desempeño.

[1] Algunos ejemplos de transformaciones que logran esto son la STFT, FSST, CWT/DWT

Esta genera la pregunta de cuál sería la eficiencia de un modelo recurrente con una profundidad mucho mayor.

Una buena línea de investigación, para obtener mejores métricas, es la arquitectura donde sería bueno aumentar la profundidad del modelo con otras capas recurrentes o convolucionales. Para lo último es necesario aplicar transformaciones a los tensores para que sean compatibles entre sí. Distintos trabajos han demostrado conseguir eficiencias altas gracias a la complejidad de los modelos. Casos como las arquitecturas de AlexNet [30] y GoogleNet [31] han mejorado distintas aplicaciones como son aplicaciones de *Computer Vision* y reconocimiento de imágenes, entre otras.

Complejidad algorítmica

La complejidad algorítmica de las redes neuronales es un tema de discusión que se encuentra ligada generalmente al algoritmo utilizado para resolver la estimación de los parámetros de la arquitectura. Algunos ejemplos de estos algoritmos son *Adam* y *RMSProp*. Por supuesto que además depende de la profundidad y de la naturaleza de las redes. Por ende, ésta es una línea de trabajo importante para definir los tiempos de entrenamiento y evaluación del modelo.

Segmentación en tiempo real

Después de tener definido varios de los problemas mencionados anteriormente, es factible pensar en una implementación de segmentación en tiempo real. Los temas más críticos para llevar a cabo esto son la elección de la implementación de la red (bajo CPU o GPU), la arquitectura y la complejidad algorítmica y los tiempos asociados.

La segmentación en tiempo real requiere definir lo que se conoce como *pipeline* donde a partir de un *stream* de datos que ingresa al mismo y se realizan distintas etapas de procesamiento (acondicionamiento, preprocesamiento, procesamiento, post-procesamiento). Para ellos es necesario bajo ciertos requerimientos del segmentador online definir los tiempos alcanzables en cada instancia del *pipeline*. Esta no es una tarea trivial y requiere una investigación previa o acompañada de las distintas líneas de trabajo antes mencionada.

Índice de figuras

Índice de cuadros

Bibliografía

[1] A. Keith and M. Flack. *"The form and nature of the muscular connections between the primary divisions of the vertebrate"*. Apr, 4. 1907.

[2] H. Liang, S. Lukkarinen, and I. Hartimo, *"Heart Sound Segmentation Algorithm based on heart sound Envelogram"*, in Computers in Cardiology, vol. 24, Lund, Swed, 1997, pp. 105–108.

[3] H. Liang, L. Sakari, and H. Iiro, *"A Heart Sound Segmentation algorithm using Wavelet Decomposition and Reconstruction"* in Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 4, Chicago, IL, USA, 1997, pp. 1630–1633.

[4] Boron, W. F. and Boulpaep E. L. *Fisiología Médica*, 3.ª ed. 2017 Elsevier España, S.L.U.

[5] S. Hochreiter and J. Schmidhuber. *"Long short-term memory. Neural Computation"*, 9(8):1735–1780, 1997.

[6] J. P. Martínez, R. Almeida, S. Olmos, A. P. Rocha, P. Laguna. *"Wavelet-Based ECG Delineator: Evaluation on Standard Databases"*. Apr, 2. 2004.

[7] S. E Schmidt, C. Holst-Hansen, C. Graff, E. Toft and JJ. Struijk. *"Segmentation of Heart Sound recordings by a Duration-dependent Hidden Markov model"*. Jan, 22. 2010.

[8] X. J. Hu, J. W Zhang G. T Cao, H.H Zhu ; H. Li. *"Feature Extraction and Choice in PCG based on Hilbert Transfer"*. Dec, 12. 2011.

[9] A. Abbas, R. Bassam and R. Mazin. *.Automated Pattern Classification for PCG Signal based on Adaptive Spectral K-means Clustering Algorithm"*. Mar, 15. 2014.

[10] D. Springer, L. Tarassenki and G. Clifford. *Logistic Regression-HSMM-based Heart Sound Segmentation.* Sep, 1 . 2015.

[11] F. Renna and M. Coimbra. *Deep Convolution Neural Netwok for Heart Sound Segmentation.* Jan, 24. 2019.

[12] D. Springer,*"Logistic Regression-HSMM-based Heart Sound Example Code"*, Physionet. 2016.

[13] A. Illanes-Manriquez and Q. Zhang, "An algorithm for QRS onset and offset detection in single lead electrocardiogram records," in 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France, 2007, pp. 541 – 544.

[14] J. Behar, et al., *"A comparison of single channel fetal ecg extraction methods"*, Annals of Biomedical Engineering, vol. 42, no. 6, pp. 1340–1353, June 2014.

[15] J. Behar, J. Oster, and G. D. Clifford, *"Combining and Benchmarking Methods of Foetal ECG Extraction Without Maternal or Scalp Electrode Data"*, Physioogical Measurement, vol. 35, no. 8, pp. 1569–1589, Aug. 2014.

[16] Q. Zhang, et al., *"An algorithm for robust and efficient location of T wave ends in electrocardiograms"*. IEEE Transactions on Biomedical Engineering, vol. 53, no. 12 (Pt 1), pp. 2544–52, Dec. 2006.

[17] C. R. Vázquez-Seisdedos, et al., *"New approach for T-wave end detection on electrocardiogram: performance in noisy conditions"*. Biomedical Engineering Online, vol. 10, no. 1, p. 77, Jan. 2011.

[18] S. R. Messer, J. Agzarian, and D. Abbott, "Optimal wavelet denoising for phonocardiograms," Microelectronics Journal, vol. 32, no. 12, pp. 931–941, Dec. 2001.

[19] D. Kumar, et al., "Noise detection during heart sound recording using periodicity signatures." Physiological Measurement, vol. 32, no. 5, pp. 599–618, May 2011.

[20] T. Oskiper and R. Watrous, "Detection of the first heart sound using a time-delay neural network," in Computers in Cardiology. Memphis, TN, USA: IEEE, 2002, pp. 537–540.

[21] B. Ergen, Y. Tatar, and H. O. H. Gulcur, "Time-frequency analysis of phonocardiogram signals using wavelet transform: a comparative study," Computer Methods in Biomechanics and Biomedical Engineering, no. October, pp. 37–41, Jan. 2011.

[22] H. Liang, L. Sakari, and H. Iiro, "A heart sound segmentation algorithm using wavelet decomposition and reconstruction," in Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, vol. 4, Chicago, IL, USA, 1997, pp. 1630–1633.

[23] C. Gupta, et al., "Neural network classification of homomorphic segmented heart sounds," Applied Soft Computing, vol. 7, no. 1, pp. 286–297, Jan. 2007.

[24] I. Daubechies and S. Maes, "A nonlinear squeezing of the continuous wavelet transform based on auditory nerve models," Wavelets in Medicine and Biology, pp. 527–546, 1996.

[25] D. Gabor, "Theory of communication. part 1: The analysis of information," Journal of I.E.E., vol. 93, no. 26, pp. 429–441, 1946.

[26] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," SIAM journal on mathematical analysis, vol. 15, no. 4, pp. 723–736, 1984.

[27] D. Gill, N. Gavrieli, and N. Intrator, "Detection and identification of heart sounds using homomorphic envelogram and self-organizing probabilistic model," in Computers in Cardiology, Lyon, France, 2005, pp. 957–960.

[28] Kingma, Diederik, and Jimmy Ba. *Ädam: A method for stochastic optimization."*, 3rd International Conference for Learning Representations, San Diego, 2015.

[29] J. Oliveira, F. Renna, and M. T. Coimbra, "Adaptive sojourn time HSMM for heart sound segmentation," IEEE J. Biomed. Health Informatics, 2018, early access.

[30] A. Krizhevsky, I. Sutskever, G. E. Hinton, *"ImageNet Classification with Deep Convolutional Neural Networks"*. ImageNet LSVRC-2010.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, *"Going Deeper with Convolutions"*. CVPR 2015.