



Universidad de Concepción
Dirección de Postgrado
Facultad de Ingeniería
Programa de Magíster en Ciencias de la Ingeniería con mención en Ingeniería Eléctrica

RED NEURONAL PROFUNDA PARA DETECCIÓN DE ARTEFACTOS POR MOVIMIENTO EN SEÑALES DE BALISTOCARDIOGRAFÍA



Tesis para optar al grado de Magíster en Ciencias de la Ingeniería con mención en
Ingeniería Eléctrica.

JUAN PABLO MORENO DEL SOLAR
CONCEPCIÓN - CHILE
2019

Profesor Guía: Esteban Pino Quiroga
Dpto. de Ingeniería Eléctrica, Facultad de Ingeniería
Universidad de Concepción

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.



Resumen

La presencia de artefactos por movimiento (*Motion Artifacts, MA*) en señales de electrocardiograma (ECG) y balistocardiograma (BCG) disminuye la eficacia de los algoritmos de detección de latidos, y con ello la confiabilidad de métricas cardíacas como el período cardíaco instantáneo, útiles para la detección de arritmias. El presente trabajo propone un sistema de detección de MA basado en redes neuronales convolucionales unidimensionales (*1D Convolutional Neural Networks, 1D CNNs*). El reconocimiento de MAs permitiría a los sistemas tomar decisiones de alto nivel, como descartar los segmentos de señal que presenten anomalías, con el fin de evitar falsos positivos. El clasificador propuesto consiste en una 1D CNN de 6 capas (8 en el caso del BCG) de extracción de características (Convolucional + *Max-pooling*), con 32 filtros de 9 muestras de campo receptivo, más una capa densa de 512 neuronas (*fully-connected*). El modelo se validó con la base de datos MIT-BIH de PhysioNet (Exactitud promedio de 95.9 %), y luego con una base de datos de BCG semi-sintético creada mediante la concatenación aleatoria de latidos de BCG obtenidos en entornos reales (Exactitud promedio de 91.1 %). Al incorporar el detector en un sistema de detección de latidos, la sensibilidad de los algoritmos mejoró de 75 % a 98.5 % en el caso del ECG, y de 72.1 % a 94.5 % en el caso del BCG, para señales contaminadas a $\text{SNR} = 0$ [dB]. Se concluye que la incorporación de una etapa de detección de MA basada en el clasificador propuesto, mejora la confiabilidad de las métricas de interés. De acuerdo a los resultados obtenidos, se puede vislumbrar la aplicación de la metodología presentada en este trabajo a otros tipos de señales cardíacas, así como su implementación en dispositivos de monitoreo portables donde las señales suelen ser baja calidad.

Agradecimientos

Quisiera agradecer, en primer lugar, a mi familia y a Loreto, por su constante e incondicional apoyo en los momentos de estrés y dificultad. Gracias por estar ahí siempre.

También a mis compañeros de grupo en las distintas asignaturas cursadas, a mis amigos ingenieros de Zambia, y a Jaime Jiménez y Javier Chávez por su ayuda y disposición a resolver dudas.

En especial, quisiera dar las gracias al profesor Esteban Pino, por la dirección de este trabajo y por su apoyo y ayuda durante el desarrollo de esta tesis. De la misma forma, quisiera agradecer a los profesores Miguel Figueroa y Pablo Aqueveque por los consejos clave al momento de postular al programa y a la beca. Sin ello, empezar este desafío no habría sido posible.

Esta tesis fue apoyada por CONICYT-PCHA/Magíster Nacional/2016-22161867.

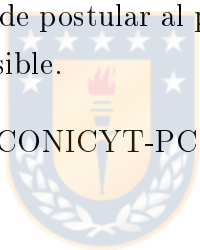


Tabla de Contenidos

Resumen	III
Agradecimientos	IV
Lista de Tablas	VIII
Lista de Figuras	X
1. Introducción	1
1.1. Hipótesis	3
1.2. Objetivos	3
1.2.1. Objetivo general	3
1.2.2. Objetivos específicos	4
1.3. Alcances y limitaciones	4
1.4. Temario	5
2. Antecedentes	6
2.1. Estado del arte	6
2.2. Discusión	11
3. Marco teórico	13
3.1. Electrocardiograma	13
3.1.1. Historia	13
3.1.2. Forma de onda	14
3.1.3. Interpretación del ECG	16
3.1.4. Ruidos típicos en señales de ECG	16

3.2. Balistocardiograma	17
3.2.1. Historia	18
3.2.2. Forma de onda	19
3.2.3. Interpretación del BCG	19
3.2.4. Ruidos típicos en la señal	21
3.3. Redes neuronales y aprendizaje profundo	22
3.3.1. Redes neuronales artificiales	22
3.3.2. Aprendizaje profundo	29
3.3.3. Redes neuronales convolucionales	31
4. Materiales y métodos	39
4.1. Elección de hardware y software	41
4.1.1. Hardware	41
4.1.2. Software	41
4.2. Creación del conjunto de datos para CNN	42
4.2.1. Base de datos de ECG	42
4.2.2. Creación del conjunto de datos ECG-MA	49
4.2.3. Base de datos de BCG	51
4.3. Diseño de la red convolucional	54
4.4. Evaluación de desempeño de la CNN	57
4.4.1. Métricas de desempeño	57
4.4.2. Validación del modelo	58
4.5. Evaluación de impacto en la confiabilidad de métricas cardíacas . . .	60
4.6. Evaluación de implementación en dispositivo portable	61
4.7. Discusión	63
5. Resultados	65

5.1. Detector de artefactos en ECG	65
5.1.1. Exploración de hiperparámetros	65
5.1.2. Validación del modelo	66
5.1.3. Evaluación de confiabilidad de métricas cardíacas	68
5.2. Detector de artefactos en BCG	71
5.2.1. Exploración de hiperparámetros	71
5.2.2. Validación del modelo	75
5.2.3. Evaluación de confiabilidad de métricas cardíacas	79
5.2.4. Evaluación de implementación en dispositivo portable	81
5.3. Discusión	82
6. Conclusiones	85
Bibliografía	86



Lista de Tablas

4.1. Características del sistema computacional a utilizar	41
4.2. Valores a explorar de los hiperparámetros del modelo	55
4.3. Características del sistema portable	62
5.1. Valores iniciales y finales de los hiperparámetros mediante búsqueda iterativa para ECG	66
5.2. Resultados de Validación Cruzada K-fold en ECG	70
5.3. Desempeño del clasificador de ECG a distintos niveles de SNR	70
5.4. Impacto del clasificador en el desempeño del algoritmo de detección de latidos en ECG	72
5.5. Valores iniciales y finales de los hiperparámetros obtenidos mediante primera búsqueda iterativa en BCG	72
5.6. Valores encontrados de los hiperparámetros para clasificador de BCG en segunda exploración	73
5.7. Valores iniciales y finales de los hiperparámetros obtenidos mediante segunda búsqueda iterativa en BCG	74
5.8. Resultados de Validación Cruzada K-fold en BCG.	78
5.9. Desempeño del clasificador de BCG a distintos niveles de SNR	78
5.10. Impacto del clasificador en desempeño del algoritmo de detección de latidos en BCG	79

Lista de Figuras

2.1. Señal de BCG	7
2.2. Artefacto por movimiento en señales de BCG	8
3.1. Señal teórica y real de electrocardiografía	15
3.2. Señales teórica y real de balistocardiografía	20
3.3. Ruidos en señal de BCG real	22
3.4. Redes neuronales básicas	23
3.5. Esquema de una red neuronal profunda	30
3.6. Red neuronal profunda	33
3.7. Esquema general de una red neuronal convolucional unidimensional. .	35
4.1. Metodología general de esta investigación	40
4.2. Herramienta <i>nst</i>	44
4.3. Registros de artefactos por movimiento EM1 y EM2	46
4.4. Extracto del registro 100 contaminado a distintos niveles de SNR . .	47
4.5. Sensibilidad v/s SNR del algoritmo SQRS para la base de datos de ECG contaminado	48
4.6. Proceso de contaminación con <i>nst</i>	48
4.7. Creación del conjunto de datos ECG-MA	50
4.8. Creación de BCG semi-sintético	53

4.9. Arquitectura de una CNN básica	54
4.10. Método propuesto para exploración de hiperparámetros	57
4.11. Método de validación K-fold	59
4.12. Sistema de detección en línea de latidos y cálculo período cardíaco . .	61
4.13. Raspberry Pi 3, modelo B. Fuente: Página web de Rasperry Pi [1]. . .	62
5.1. Arquitectura final de CNN para detección de artefactos en ECG . . .	67
5.2. Costo y Ac_{val} durante el entrenamiento del modelo para ECG (Fold 1)	68
5.3. Detección de artefactos en el registro de ECG 213	69
5.4. Desempeño del detector de latidos de ECG en función del SNR	71
5.5. Arquitectura final de CNN para detección de artefactos en BCG . . .	75
5.6. Costo y Ac_{val} durante el entrenamiento del modelo para BCG (Fold 1)	76
5.7. Detección de artefactos en el registro de BCG 21	77
5.8. Desempeño del detector de latidos de BCG en función del SNR . . .	80
5.9. Frecuencia cardíaca instantánea del registro de BCG 21	81

Abreviaciones

ANN	Red neuronal artificial (<i>Artificial Neural Network</i>)
BCG	Balistocardiograma
BIH	Hospital Beth Israel (<i>Beth Israel Hospital</i>)
CNN	Red neuronal convolucional (<i>Convolutional Neural Network</i>)
DL	Aprendizaje profundo (<i>Deep Learning</i>)
DNN	Red neuronal profunda (<i>Deep Neural Network</i>)
ECG	Electrocardiograma
EMFi	Film electromecánico (<i>Electromechanical Film</i>)
EMG	Electromiograma
FC	Totalmente conectada (<i>Fully Connected</i>)
FE	Extracción de características (<i>Feature Extraction</i>)
FSR	Razón de señal filtrada (<i>Filtered Signal Ratio</i>)
HR	Período cardíaco (<i>Heart Rate</i>)
ICA	Análisis de componentes independientes (<i>Independent Component Analysis</i>)
ILSVRC	Desafío de ImageNet sobre reconocimiento visual de gran escala (<i>ImageNet Large Scale Visual Recognition Challenge</i>)
MA	Artefacto por movimiento (<i>Motion Artifact</i>)
MIT	Instituto Tecnológico de Massashusetts (<i>Massachusetts Institute of Technology</i>)
MP	<i>Max Pooling</i>

MLP	Perceptrón multi-capa (<i>Multi Layer Perceptron</i>)
PCA	Análisis de componentes principales (<i>Principal Component Analysis</i>)
ReLU	Unidad lineal rectificada (<i>Rectified Linear Unit</i>)
SNR	Razón señal a ruido (<i>Signal to Noise Ratio</i>)
SVM	Máquina de soporte vectorial (<i>Support Vector Machine</i>)
WFDB	Base de datos de formas de onda (<i>Waveform Database</i>)



Capítulo 1. Introducción

Los fallos cardíacos son la principal causa de muerte en el mundo desarrollado y en vías de desarrollo [2]. Sólo en Estados Unidos, se proyecta que para el 2030 el 40.5 % de la población podría sufrir algún tipo de desorden cardíaco [3]. En Chile, el panorama no es muy distinto: el progresivo envejecimiento de la población (16.7 % de adultos mayores en el 2013 [4]), y la alta prevalencia de factores de riesgo como el consumo de tabaco (40.6 %), la obesidad (64.5 %), la hipercolesterolemia (38.5 %) y la diabetes (9.4 %) [5], explican de alguna forma que en el año 2014 el 27.53 % de las defunciones fueran debido a problemas del corazón [6].

Si bien la prevalencia de factores de riesgo puede disminuirse mediante la adopción de estilos de vida más saludables, el problema del envejecimiento paulatino de la población es más complejo. La creciente cantidad de adultos mayores conlleva que en el futuro los controles médicos y visitas a clínicas y hospitales también aumentarán, con la consiguiente sobrecarga al sistema de salud y la demanda de mayor presupuesto. Ante este escenario, resulta urgente desarrollar nuevas tecnologías que permitan disminuir el costo de la medición de variables cardíacas, aumentando su accesibilidad a los pacientes, y permitiendo el traslado del monitoreo y diagnóstico preliminar (*screening*) desde los establecimientos de salud hacia los hogares.

A través de los años se han desarrollado diversos métodos para registrar señales cardíacas. En entornos no hospitalarios, y sólo por mencionar algunos, han surgido la fonocardiografía, apexcardiografía, sismocardiografía, kinetocardiografía, electrocardiografía (ECG) y la balistocardiografía (BCG). Para cada uno de estos métodos, se pueden encontrar en la literatura variadas implementaciones, de distinta complejidad, costo y efectividad. En particular, las implementaciones para balistocardiografía destacan por ser no-obtrusivas: balanzas caseras modificadas [7, 8], sillas de rueda [9, 10], sillas normales [11], y camas [12]. Alternativas así resultan llamativas pues permiten la adquisición de señales mientras el paciente descansa o realiza otras actividades en reposo, tales como dormir, ver televisión o leer.

Desgraciadamente, puesto que las implementaciones en ambientes no controlados buscan la comodidad del paciente y la no-obtrusividad, estos sistemas tienden a

ser propensos a la aparición de ruidos, particularmente artefactos por movimiento (*Motion Artifacts, MA*). Los MAs ocurren por movimientos abruptos del paciente o del sensor, y a diferencia de otros ruidos como la interferencia electromagnética (EMI) o la variación de línea base debido a la respiración, son repentinos por naturaleza. Más aún, su espectro en frecuencia se traslapa con el de las señales cardíacas, por lo que es casi imposible filtrarlos con filtros lineales sin eliminar características importantes de la señal original [13].

La presencia de artefactos dificulta a los médicos la correcta identificación de los intervalos P-QRS-T de las señales de ECG, lo que resulta en diagnósticos erróneos de patologías y tratamientos inapropiados [14]. De forma similar, las señales cardíacas contaminadas alteran el funcionamiento de los algoritmos de detección automática de latidos, generalmente porque los artefactos son confundidos con los picos R del ECG (o J en el caso del BCG) [15]. Se ha reportado que los MAs son la causa más frecuente de falsas alarmas en dispositivos ambulatorios [16], entre otras cosas debido a que aumentan artificialmente la variabilidad de la frecuencia cardíaca, que a su vez se asocia a la existencia de arritmias. En otras palabras, la presencia de MAs reduce significativamente la confiabilidad de las métricas calculadas y el subsiguiente diagnóstico.

Es de suma importancia entonces contar con mecanismos que permitan identificar automáticamente la ocurrencia de artefactos, con el fin de reportar que la confiabilidad de las señales adquiridas es baja. De esta forma, los análisis y cálculos posteriores podrían ser realizados sólo sobre los segmentos de señal libres de MAs, lo cual es la práctica común de los médicos [17]. Esta estrategia, en definitiva, aumenta la confiabilidad de las métricas calculadas automáticamente, ya sea la variabilidad del intervalo R-R o la elevación del segmento S-T en el ECG, o el intervalo J-J y la amplitud I-J en el caso del BCG. Asegurar que las métricas reportadas son confiables permitirá la adopción de sistemas portables para *homecare* y ambientes laborales de forma más amplia, donde las señales adquiridas suelen ser de baja calidad.

El problema de la detección automática de MAs nos lleva a pensar en aplicar métodos de aprendizaje automático y reconocimiento de patrones, un área de investigación popular en estos tiempos. En particular, los algoritmos de aprendizaje profundo (*Deep Learning, DL*) basados en redes neuronales convolucionales (*Convolutional*

Neural Networks, CNN) presentan resultados sobresalientes en el reconocimiento de imágenes y otros problemas complejos. En nuestro caso, en vez de caracterizar las señales cardíacas en términos de su información estadística o espectral, podríamos resolver el problema de reconocimiento de MAs a través de sus características morfológicas.

Por todo lo anterior, la presente tesis propone un sistema basado en CNN para el reconocimiento de MAs, cuyo fin es asistir a los algoritmos de detección automática de latidos. Puesto que la aparición de MAs es un problema transversal a los distintos métodos de registro de señales cardíacas, y considerando que no existen bases de datos estándar de BCG disponibles de forma gratuita, la solución propuesta se aplicará primero en señales de ECG, más extensamente estudiadas y de las cuales si existen bases de datos disponibles. Una vez se haya diseñado la arquitectura del sistema, se aplicará a señales de BCG semi-sintéticas creadas a partir de registros obtenidos en entornos reales, haciendo los ajustes necesarios. En ambos casos, se espera aumentar la confiabilidad de las métricas cardíacas de interés calculadas.

1.1. Hipótesis

Es posible utilizar una red neuronal convolucional profunda para detectar artefactos por movimiento en señales de ECG y BCG, y con ello mejorar la sensibilidad y valor predictivo positivo de algoritmos de detección de latidos.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar un sistema basado en *Deep Learning* para la detección de artefactos por movimiento en señales de ECG y BCG, con el fin de asistir a los algoritmos de detección de latidos y cálculo de métricas cardíacas de interés, y mejorar de esta forma la confiabilidad de los resultados en sistemas de medición portables.

1.2.2. Objetivos específicos

1. Investigar sobre las distintas topologías de redes neuronales profundas y elegir una acorde al problema planteado.
2. Diseñar un programa computacional que implemente una CNN en Python, para la detección de artefactos presentes señales biomédicas.
3. Entrenar una red utilizando señales de ECG de la base de datos de arritmia de PhysioNet (MIT-BIH), contaminadas con artefactos de movimiento.
4. Entrenar una red utilizando señales de BCG sintéticas, creadas a partir de mediciones en entornos reales.
5. Medir el desempeño del clasificador en ambos casos.
6. Determinar un tamaño mínimo de la red que permita obtener buenos resultados de clasificación, y que al mismo tiempo sea implementable en un dispositivo portable.



1.3. Alcances y limitaciones

Debido al gran número de herramientas de programación y *frameworks* disponibles actualmente para la implementación y sistemas basados en *Deep Learning*, su elección no se realizará en base a una evaluación cuantitativa, sino más bien por conveniencia, popularidad y simpleza.

Se compilarán dos set de datos, uno de ECG y otro de BCG. Para el primero, se utilizarán las muestras de la base de datos de arritmia MIT-BIH, disponible en PhysioNet. Para el set de BCG, se utilizarán las señales obtenidas por Javier Chávez en [18] y descritas en el Capítulo 8. No se tomarán nuevas muestras.

Por limitaciones del sistema computacional disponible, al momento de diseñar la CNN no se realizará una búsqueda exhaustiva de hiperparámetros (búsqueda en grilla). En vez de lo anterior, se llevará a cabo una búsqueda iterativa que garantiza encontrar sólo óptimos locales.

1.4. Temario

El presente trabajo se divide en 6 capítulos. El capítulo 1 revisa a modo general el problema que motiva esta investigación, y define la hipótesis y los objetivos de trabajo. En el capítulo 2 se revisa el Estado del Arte en la detección de MAs en señales de ECG, BCG y otras similares. El capítulo 3 detalla los fundamentos teóricos relacionados a las señales de ECG y BCG, como también las bases matemáticas de las CNNs. Por su parte, en el capítulo 6 se describen los sistemas de hardware y software a utilizar, además de los métodos propuestos para la creación de las bases de datos, diseño de la red, y medición de desempeño del sistema. El capítulo 5 muestra los resultados obtenidos, y una discusión a partir de los mismos. Finalmente, en el capítulo 6 se sintetizan las conclusiones de la presente investigación.



Capítulo 2. Antecedentes

En el presente capítulo se lleva a cabo una revisión bibliográfica de los distintos temas que atañen la presente investigación. En primer lugar, estudian los sistemas de adquisición de BCG, desde las primeras implementaciones hasta los actuales sistemas portables y de *homecare*. Se pone especial atención en los métodos de detección y filtrado de MAs y otros tipos de ruido, y las ventajas y desventajas de las distintas alternativas. Posteriormente, se analiza la aparición de artefactos y ruidos repentinos en otros tipos de señales similares, y las estrategias de mitigación aplicadas por los investigadores. A continuación, se revisa el uso de CNNs en la detección de complejos QRS en señales de ECG, pues este problema comparte algunas similitudes con el asunto central de esta tesis. Finalmente, se lleva a cabo una breve discusión acerca de los tópicos revisados, y se exponen algunos fundamentos de la solución propuesta.



2.1. Estado del arte

El BCG es un método que mide la reacción mecánica del cuerpo ante la expulsión de la sangre en cada ciclo cardíaco [19]. Es una señal muy tenue, que resulta de la suma de distintas fuerzas balísticas relacionadas al movimiento cardíaco (Figura 2.1). Una particularidad del BCG es que, dependiendo del método de medición usado, la morfología de las señales varía sustancialmente [20]. Se han determinado también diferencias significativas entre registros de sujetos distintos, como también entre grupos de personas sanas y enfermas. Todo lo anterior ha dificultado históricamente la interpretación de la señal para el diagnóstico clínico, tanto por expertos como de forma automática mediante computador.

Si bien para la forma típica de diagnóstico, donde un experto analiza una señal única tomada en un momento determinado y la contrasta con la norma, el BCG no ha sido demasiado útil por lo comentado anteriormente, el *homecare* abre nuevas posibilidades debido a que en este caso el diagnóstico se hace a partir de los cambios relativos de la señal de un sujeto determinado, en largos periodos de tiempo [7]. Algunos ejemplos clásicos de casos de diagnóstico y pronóstico de enfermedades

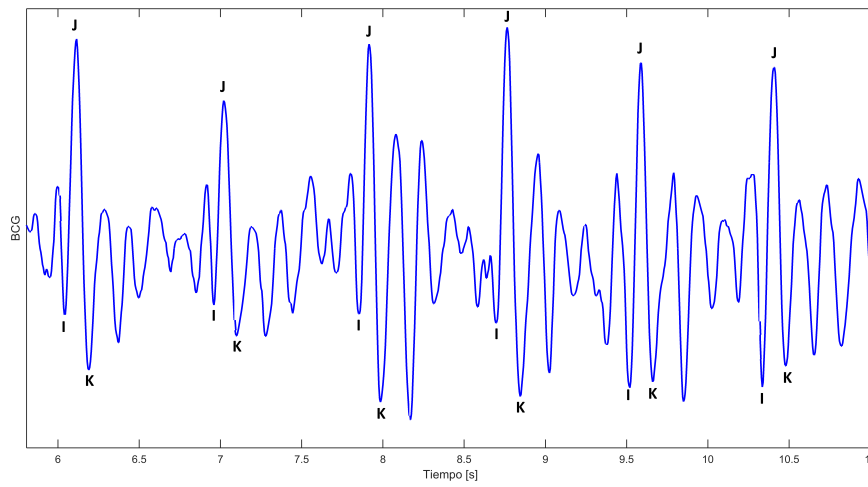


Figura 2.1: Señal de BCG. Se distinguen los segmentos más relevantes, rotulados con las letras I, J y K. Fuente: Elaboración Propia.

donde el BCG ha sido útil son:

- Mandelbaum y Mandelbaum [21] realizaron mediciones durante 18 meses en pacientes que se recuperaron de ataques cardíacos. Se concluyó que el BCG era un indicador valioso para evaluar la recuperación funcional.
- Starr and Wood [22] llevaron a cabo un estudio donde determinaron que la disminución de la amplitud IJ en el tiempo era un indicador de la degradación de la salud cardíaca.

Por su parte, estudios recientes han demostrado que el BCG puede ser usado para el estudio de la capacidad cardíaca a través del cálculo de la potencia RMS de la señal [23], y la contractibilidad del miocardio mediante la medición del intervalo R-J (picos del ECG y BCG, respectivamente) [8]. Otra aplicación es el diagnóstico de arritmias mediante el cálculo de la frecuencia cardíaca instantánea o *beat-to-beat*. Al igual que en el caso del ECG, donde el periodo cardíaco se obtiene a partir de la diferencia temporal de dos picos R consecutivos, en el balistocardiograma el ritmo cardíaco se puede medir mediante la distancia entre los picos J. De esta forma, el cálculo del intervalo J-J en señales adquiridas con sistemas electrónicos no invasivos de bajo costo ha sido foco de importante estudio en el último tiempo.

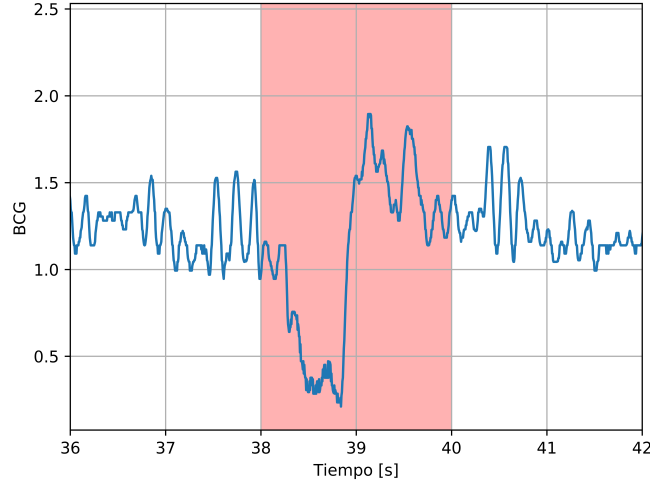


Figura 2.2: Artefacto por movimiento en señales de BCG. Fuente: Elaboración Propia.

Como se mencionó anteriormente, las mediciones de BCG se componen no sólo de la reacción del cuerpo ante la expulsión de la sangre, sino también de movimientos respiratorios, ruidos de alta frecuencia y artefactos producidos por movimientos musculares del paciente o movimientos del sensor mismo (Figura 2.2). Por lo mismo, en estas circunstancias es estrictamente necesario pre-procesar la señal antes de realizar cualquier cálculo.

Algunas investigaciones han sugerido la utilización de filtros digitales clásicos como FIR e IIR (estáticos o adaptativos) o *wavelets*, para eliminar el ruido y posteriormente detectar los picos J o formas más complejas de la señal [24, 25, 26]. Si bien estas estrategias ayudan a eliminar ruidos de banda acotada o con algún patrón conocido, ninguno es suficientemente bueno para lidiar con los MAs. Es crucial contar con algún método que considere la aparición de dichos artefactos, pues éstos suelen afectar tanto a la señal, que el cálculo de índices de interés se vuelve dificultoso y poco confiable.

Para este fin, algunos acercamientos se han valido de algoritmos de *clustering* (previa extracción de características relevantes de los segmentos de la señal) para encontrar el periodo cardíaco [27, 28]. También se ha ahondado en estrategias que,

en vez de detectar eventos específicos (como los picos J) para luego calcular el periodo cardíaco, simplemente estiman una “frecuencia fundamental” de la señal [29, 30]. Lo anterior tiene la desventaja de que hace difícil (si no imposible) hacer diagnósticos de patologías relacionadas a la variación de periodos cardíacos contiguos, como lo son las arritmias.

Finalmente, otra forma de atacar el problema es identificar automáticamente la ocurrencia de artefactos para tomar decisiones de alto nivel, como aplicar algún algoritmo complejo de filtrado, o descartar por completo los segmentos contaminados. En [9], la señal de BCG es procesada por un filtro basado en Análisis de Componentes Independientes (ICA, por su sigla en inglés) para descomponerla en tres partes: señal de interés, ruido continuo (movimientos musculares) o cambios abruptos. Luego, calculan la Curtosis de las últimas dos componentes y las comparan con valores de umbral para determinar si existe ruido o no. De existir, dichas componentes son excluidas para la reconstrucción de la señal final. En [11], los artefactos provocados por el movimiento del paciente se detectan simplemente comparando a señal en el tiempo contra un umbral de intensidad, basándose en la suposición que los artefactos siempre tienen mayor potencia que la señal de BCG misma. En [31], se utiliza una ventana deslizante sobre la señal para obtener los máximos y mínimos, y si el rango es mayor que un umbral, la ventana se descarta. En [32] se propone un sistema de medición redundante consistente en 4 sensores, realizando el cálculo del periodo cardíaco en paralelo para cada sensor, y descartando aquellas mediciones cuyo valor DC se escapa de la media de las cuatro. Por último, en [18] se calculan índices estadísticos para cada segmento de la señal, como la varianza y la media, y si estos escapan de algún umbral empíricamente definido, se descartan. Todos estos métodos, como puede apreciarse, se basan en la comparación de ciertos aspectos de la señal contra umbrales pre-definidos, lo que los hace altamente dependientes del sujeto, sistema y condiciones externas.

Es interesante tener en cuenta que el problema aquí explicado tiene similitudes con otros encontrados en áreas relacionadas. Un campo que comparte desafíos parecidos es la fonocardiografía, en particular las implementaciones de bajo costo en forma de *Apps* para teléfonos inteligentes, que obtienen la señal de interés a través del micrófono del teléfono. Aquí, la habilidad de los algoritmos para calcular frecuen-

cia cardíaca u otros índices se ve deteriorada por el ruido ambiente, el movimiento del aparato o del usuario, etc. Para atacar estos problemas, en la literatura es posible encontrar métodos basados en *Support Vector Machines* (SVM) previa extracción de características relevantes de la señal [33], o métodos basados en *Random Forests* (árboles de decisión complejos) sobre características estadísticas extraídas previamente [34].

En el ECG, que es probablemente el método más ampliamente utilizado actualmente para registrar señales cardíacas, también es posible encontrar el problema de los MAs. En general, el ECG se obtiene en entornos de medición clínicos, que son propicios para capturar una señal libre de ruido, o mediante instrumentos portátiles como el Holter, donde la finalidad es hacer un registro extenso de la actividad cardíaca (al menos por 24 horas) para su posterior análisis médico. En ambos casos, la aparición de MAs no resulta un problema de gran envergadura. Sin embargo, en la literatura es posible encontrar trabajos donde se implementan sistemas portátiles de ECG para monitoreo y cálculo en línea de métricas de interés. En esta situación, sí resulta importante tener en cuenta los artefactos en las señales, por lo que se han propuesto algunos métodos para lidiar con ellos.

En [35], los autores descomponen la señal con *wavelets* y comparan los coeficientes contra umbrales, que se adaptan en tiempo real a partir de las mismas señales, para así descartar esas componentes en la reconstrucción de la señal. Otros acercamientos, como en [36], se valen de mediciones adicionales de impedancia y de un acelerómetro, que son comparadas contra umbrales previamente definidos para marcar los segmentos de señal como artefactos. Por su parte, en [37], se propone utilizar un máquina de soporte vectorial para clasificación automática de segmentos de señales obtenidas del monitoreo de ECG de pacientes transportados en helicóptero. Para este fin, se extraen características en el dominio del tiempo y la frecuencia (energía de alta frecuencia y de baja frecuencia, razón señal a ruido, entre otras).

Relacionado al ECG, otra área de estudio que resulta interesante es la detección y clasificación automática de complejos QRS, y el diagnóstico de patologías cardíacas. Si bien no todos los trabajos contemplan la detección de MAs, existen similitudes en la forma de abordar el problema pues por lo general la detección y clasificación es basada en la morfología del complejo QRS. En [38], los autores proponen un sistema

para clasificar señales cardíacas en normal, fibrilación atrial, otro ritmo, o ruido. El sistema se basa en una combinación de CNNs, redes neuronales recurrentes, y SVM, y utiliza la base de datos MIT-BIH como *benchmark*. En [39], y nuevamente haciendo uso de la base de datos MIT-BIH, se clasifican latidos de ECG de un mismo paciente en normal, supraventricular, ventricular, fusión e inclasificable, utilizando una red neuronal convolucional unidimensional de tres capas. Distintas topologías de redes recurrentes y residuales son evaluadas en [40] para clasificar registros de ECG multi-canal en seis clases distintas, incluyendo ritmo normal, cuatro tipos de arritmias, y artefactos. Una aplicación distinta se persigue en [41], donde los autores, a partir de una señal de ECG obtenida mediante un dispositivo portable, proponen clasificar los artefactos por movimiento en distintos tipos de movimientos corporales (levantar el brazo derecho, el izquierdo, etc.). Para ello, extraen las características de las señales usando *wavelets*, e implementan un clasificador basado en un perceptrón multicapa (*Multi-Layer Perceptron*, MLP).

Por último, en el campo del reconocimiento del habla, la aparición de ruidos repentinos (ruido ambiente, ruidos de teclas de computador, etc.) también es un problema que afecta enormemente la capacidad de los sistemas para identificar palabras, cuando se implementan en entornos reales como *call centers* u otros. En ésta línea, algunas soluciones propuestas se basan en *clustering* con *k-means* e identificación de ruido con AdaBoost [42] y detección por linealidad de fase entre segmentos contiguos de señal [43, 44].

2.2. Discusión

Abordar el problema de los MAs resulta interesante pues, como se dijo antes, su presencia resta confiabilidad a los cálculos de métricas cardíacas de interés. En particular, su estudio en señales de BCG es una buena oportunidad para desarrollar y ejercitar métodos de detección, pues en general los sistemas de bajo costo para obtención del balistocardiograma, por sus implementaciones no-obstrusivas, tienden a ser propensos a la aparición de artefactos (por ejemplo, un sensor puesto bajo el cojín de una silla). En este sentido, si los detectores desarrollados funcionan bien en dicho entorno, es probable que también lo hagan en otros donde las señales son más

robustas.

De acuerdo a lo revisado en la literatura, las metodologías que abordan el problema de la detección de artefactos en señales biomédicas se pueden resumir en los siguientes pasos:

1. Pre-procesamiento las señales: división en ventanas y aplicación de filtros para reducir ruidos de banda conocida
2. Extracción de características: Aplicación de algoritmos como Wavelets, ICA, obtención de métricas estadísticas, entre otros.
3. Clasificación de patrones: Diversos algoritmos de clasificación se encuentran en la literatura, desde la simple comparación contra umbrales pre-definidos, hasta SVMs, árboles de decisión de distinta complejidad, etc.

Como se verá más adelante en detalle, el proceso de extracción de características es un paso crítico, pues el buen desempeño del clasificador depende enormemente de las características elegidas para representar los datos de entrada. Resultaría interesante entonces proponer un sistema que sea capaz de descubrir por si solo una representación de las señales de BCG, discriminando de forma precisa los segmentos que presentan artefactos de aquellos limpios. Los algoritmos de *Deep Learning*, y en particular las *Deep Neural Networks* (DDNs), se presentan como una buena alternativa para resolver dicho problema.

Capítulo 3. Marco teórico

Este capítulo presenta los fundamentos teóricos necesarios para entender la investigación llevada a cabo en la presente tesis. Primero, una breve descripción e historia de la electrocardiografía y la balistocardiografía son detalladas. A continuación, se introducen conceptos relativos a las redes neuronales artificiales, el aprendizaje profundo y las CNNs, y se exponen algunos fundamentos matemáticos que explican los algoritmos a implementar.

3.1. Electrocardiograma

El ciclo cardíaco consiste en la repetición rítmica de eventos de contracción y relajación de las aurículas y ventrículos. Estos movimientos musculares son originados por impulsos eléctricos iniciados y conducidos por conjuntos de células especiales similares a las neuronas. El ECG es un método que mide las diferencias de potencial en la superficie del cuerpo, provocadas por dichos impulsos eléctricos. En nuestros días, el ECG es parte esencial del diagnóstico de problemas cardíacos.

3.1.1. Historia

La existencia de fenómenos eléctricos asociados al movimiento muscular fue descrita por primera vez por el Dr. Luigi Galvani en 1786, quien registró actividad eléctrica en músculos disecados [45]. La presencia de corriente eléctrica asociada al movimiento del corazón de una rana fue demostrada en 1842 por el Dr. Carlo Matteucci, en la Universidad de Pisa [46]. Treinta y cinco años más tarde, el primer ECG humano fue publicado por Augustus Waller, fisiólogo británico, demostrando que cierta actividad eléctrica precedía a la contracción ventricular [47].

Basándose en el trabajo de Waller, el Dr. Willem Einthoven introdujo nuevas mejoras al método y demostró la obtención de cinco puntos fiduciaros, rotulados con las letras PQRST [45, 48]. La medición se obtenía mediante el uso de cinco electrodos

conectados a las cuatro extremidades y a la boca. Adicionalmente, desarrolló un galvanómetro de alta sensibilidad y bautizó el método tal como lo conocemos hoy. La implementación de Einthoven era voluminosa, pesaba alrededor de 300 [Kg], y los electrodos que usaba eran cubos con solución electrolítica donde las extremidades del paciente debían ser sumergidas.

Para hacerlo más práctico, redujo el número de electrodos a tres (brazo derecho, brazo izquierdo y pierna izquierda), construyendo el concepto del Triángulo de Einthoven, todavía usado en nuestros días. En esta configuración, las distintas derivaciones consistían en la diferencia de potencial entre brazo derecho e izquierdo (D1),; entre brazo derecho y pierna izquierda (D2); y entre brazo izquierdo y pierna izquierda (D3).

Durante las primeras décadas del siglo XX, el instrumento empezó a ser adoptado en ambientes hospitalarios, y aparecieron los primeros diagnósticos de arritmias e infartos cardíacos. Sin embargo, se descubrió la existencia de áreas del miocardio silenciosas, por lo que nuevas modificaciones fueron introducidas. Las mejoras en la construcción de los electrodos y el uso de contactos unipolares propició el desarrollo del ECG de 12 derivaciones, obtenido a través de 10 electrodos, y actual estándar de la medicina moderna.

3.1.2. Forma de onda

El ECG mide la actividad eléctrica del corazón producida por la propagación de los potenciales de acción cada ciclo cardíaco. No es la medición directa de la depolarización y repolarización de las células del corazón, si no el efecto relativo resultante de este proceso, manifestado en la superficie de la piel [49]. Una forma de onda de la derivación D2 se muestra en la Figura 3.1(a), donde es posible ver distintos picos y ondas rotuladas con las letras P, Q, R, S y T.

El ciclo comienza con el impulso del nodo sinoauricular en la aurícula derecha, mas este evento no alcanza a ser registrado por el ECG debido a la baja amplitud del potencial desarrollado. Luego, las aurícula derecha e izquierda se depolarizan sincrónicamente, dando origen a la onda P. Cuando la depolarización se completa,

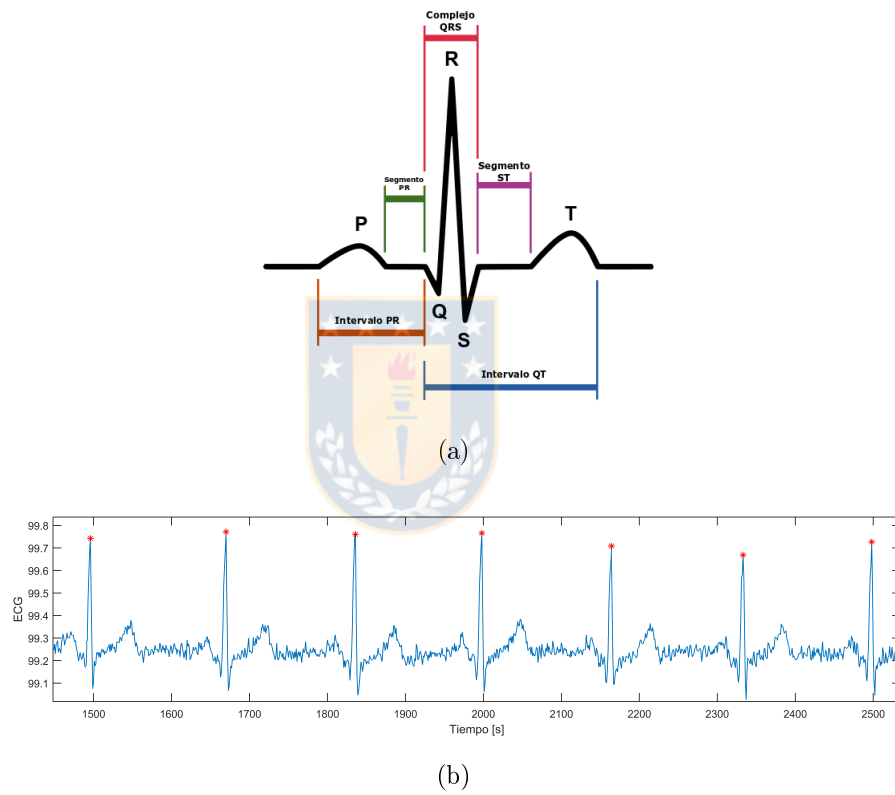


Figura 3.1: Señal teórica y real de electrocardiografía. (a) Forma de onda teórica de electrocardiografía. Fuente: Wikipedia [50]. (b) Señal real de ECG. Las marcas corresponden a los picos R. Fuente: Elaboración Propia.

comienza la contracción. El potencial de acción luego se propaga al nodo atrioventricular y al haz e His, lo que provoca la depolarización de los ventrículos derecho e izquierdo, que resultan en el complejo QRS. Finalmente, los ventrículos se repolarizan, marcando la ocurrencia de la onda T para completar el ciclo y dar inicio a uno nuevo. Cabe destacar que la repolarización de las aurículas rara vez se detecta en el ECG, pues se ve enmascarada por la aparición del complejo QRS. Una señal real de ECG se puede apreciar en la Figura 3.1(b).

3.1.3. Interpretación del ECG

El ECG es considerado como el mejor método para analizar perturbaciones en el ritmo cardíaco y el estado del corazón [51]. La forma más ampliamente usada es para la medición del periodo cardíaco, obtenido mediante la diferencia temporal entre dos picos R consecutivos. El análisis del ritmo cardíaco instantáneo sirve a su vez para la detección de arritmias, es decir, un periodo cardíaco no estable. Los distintos tipos de arritmia se pueden interpretar a través de la morfología del complejo QRS y de las ondas P y T [51]. El intervalo PR, por su parte, proporciona información acerca de bloqueos cardíacos. La presencia de una onda P sin un complejo QRS puede significar un fallo en la conducción del nodo atrioventricular. Un aumento del segmento S-T puede indicar una isquemia ventricular, mientras que su disminución podría ser síntoma de infarto al miocardio, problemas en las arterias coronarias o pericarditis. Estos son sólo algunos ejemplos en el uso del ECG para el diagnóstico de problemas cardíacos.

3.1.4. Ruidos típicos en señales de ECG

Los ruidos presentes en el ECG se pueden dividir en dos grupos: ruidos persistentes de alta y baja frecuencia, y ruidos repentinos. Dentro del primero, podemos mencionar el ruido por interferencia electromagnética de la red eléctrica y equipos electrónicos cercanos, el ruido de electromiografía (ECG), y la variación de línea base. Dentro del segundo, encontramos el ruido de contacto y los MAs del paciente o el electrodo .

Los ruidos persistentes suelen ser atacados mediante filtros analógicos o digitales, pues poseen un ancho de banda más o menos acotado y conocido. La desviación de línea base, debida principalmente al movimiento muscular respiratorio, suele encontrarse en la zona de frecuencia bajo los 1 [Hz], por lo que un filtro pasa altos es comúnmente usado para atenuar su efecto. El ruido por movimientos musculares o EMG suele estar sobre los 20 [Hz] y, aunque existe un traslape con el ancho de banda del ECG, a veces se usan filtros pasabajo a 20 [Hz] para tratar de eliminar el EMG. Por su parte, la interferencia de la red eléctrica tiene una componente fundamental en 50 o 60 [Hz], por lo que un filtro *notch* es útil en esos casos [13].

A diferencia de lo anterior, los MAs son mucho más difíciles de atacar, debido a su naturaleza aleatoria tanto en amplitud, duración y composición espectral. Estos tienen origen en variaciones en la impedancia electrodo-piel, debida a vibraciones, golpes y movimientos del electrodo o de la zona de contacto. Las características del artefacto dependen de factores no conocidos asociados a las propiedades del electrodo, del electrolítico, la impedancia de la piel y el tipo de movimiento [52].

Para la obtención de ECGs de buena calidad, se recomienda que el paciente se mantenga en posición de descanso durante el registro. Esto es posible en ambientes hospitalarios y en general da buenos resultados, no así cuando la medición se hace mediante dispositivos portátiles como el *holter*. En estos casos, es necesario implementar métodos más avanzados de detección y eliminación de ruido, tales como los revisados en la Sección 2.1.

3.2. Balistocardiograma

El BCG es un método que mide la reacción mecánica del cuerpo ante la expulsión de la sangre en cada ciclo cardíaco [19]. Es por ello la suma de las fuerzas de acción y reacción relacionadas al movimiento de la sangre en el corazón, en las arterias, y al movimiento del corazón mismo.

3.2.1. Historia

Los movimientos balísticos del cuerpo fueron estudiados por primera vez a fines del Siglo XIX por Gordon [53], midiendo el desplazamiento de una cama péndulo (una cama colgada con cuerdas desde el techo) al situar a una persona sobre ella. El sistema no era ni muy práctico ni muy exacto, y los movimientos causados por la respiración eran siempre mayores a aquellos debidos a la actividad cardíaca.

Durante los años siguientes, la mayoría de las implementaciones de balistocardiógrafos seguían basándose en la medición del movimiento de una cama. Por ejemplo, el balistocardiógrafo de Starr [54], inventado casi medio siglo después, mantenía la idea de la cama péndulo, pero esta vez se usaban cables para suspenderla del techo, se restringieron los movimientos laterales, y se ideó un sistema para minimizar el efecto de la respiración. Si bien la calidad de las mediciones mejoraba con este tipo de modificaciones, el aparato en sí seguía siendo voluminoso y no portable.

El primer balistocardiógrafo de contacto directo con el cuerpo data de 1949 [55]. En éste, el paciente se recostaba sobre una superficie de madera dura, y se le amarraba con bandas elásticas una barra de madera a las pantorrillas. La barra, a su vez, llevaba adosada un imán. El movimiento balístico del cuerpo era registrado por un transductor electromagnético montado en la superficie de madera, que a su vez se conectaba a un galvanómetro. Si bien esta técnica era una mejora significativa, seguía siendo un instrumento poco práctico, por lo que su adopción en ambientes hospitalarios fue escasa.

Debido a lo anterior, y sumado a falta de conocimiento para interpretar la señal y extraer información relevante para el diagnóstico certero de patologías, como también a la aparición de nuevos métodos no invasivos como el ultrasonido y el ECG, la investigación relativa al BCG fue mermando considerablemente [56] en los años subsiguientes. No obstante, con la aparición de nuevas tecnologías de medición de bajo costo (sensores de carga estática, *films* piezoeléctricos, placas de esfuerzo o *force-plates*, etc.) y el desarrollo de sistemas computacionales portables más poderosos, el BCG ha resurgido en las últimas décadas como un método digno de ser reevaluado, especialmente para el sistemas de *homecare*.

3.2.2. Forma de onda

Como se explicó antes, el BCG mide la reacción del cuerpo ante la actividad mecánica del corazón. El ciclo empieza con la contracción del miocardio, expulsando la sangre en la fase llamada sístole. Ahí, el centro de gravedad de la sangre se desplaza hacia los extremos del cuerpo, primero hacia la cabeza, y luego en dirección a los pies. Este desplazamiento del centro de masa durante el desarrollo del ciclo cardíaco, afectado también por el movimiento respiratorio (y eventualmente también por otros movimientos musculares), es lo que origina la forma de onda del BCG [57].

Una forma de onda teórica de BCG se muestra en la Figura 3.2(a). En ella, se pueden distinguir puntos extremos rotulados con las letras F, G, H, I, J, K, L, M y N. Existe consenso en dividir la señal en tres grupos: pre-expulsión de la sangre (FGH), expulsión (IJK) y diástole (LMN) [58]. La contraparte real de esta señal se puede apreciar en la Figura 3.2(b), donde también puede apreciarse una señal de ECG obtenida simultáneamente. En la interpretación del BCG, suele usarse como referencia el pico R del ECG, pues el impulso eléctrico siempre antecede a la reacción mecánica [7].

3.2.3. Interpretación del BCG

A través de los años, formas cualitativas y cuantitativas de interpretar la señal han sido presentadas en la literatura. La primera forma consiste en dividir la onda correspondiente a un ciclo cardíaco en cuatro grupos distintos, dependiendo del grado de anormalidad de su morfología. Los expertos entonces pueden detectar patologías examinando el registro visualmente y encontrando patrones particulares, por ejemplo, la estenosis subaórtica hipertrófica idiopática [58].

En cuanto a medidas cuantitativas, la amplitud I-J ha sido la más estudiada, pues las amplitudes de las demás ondas son pequeñas y más propensas a variaciones. Se ha encontrado que la anterior puede usarse para describir el efecto de algunas drogas, en la evaluación de problemas en la válvula aorta y en las arterias coronarias, y para predecir la esperanza de vida. Otra medición que ha cobrado importancia en el

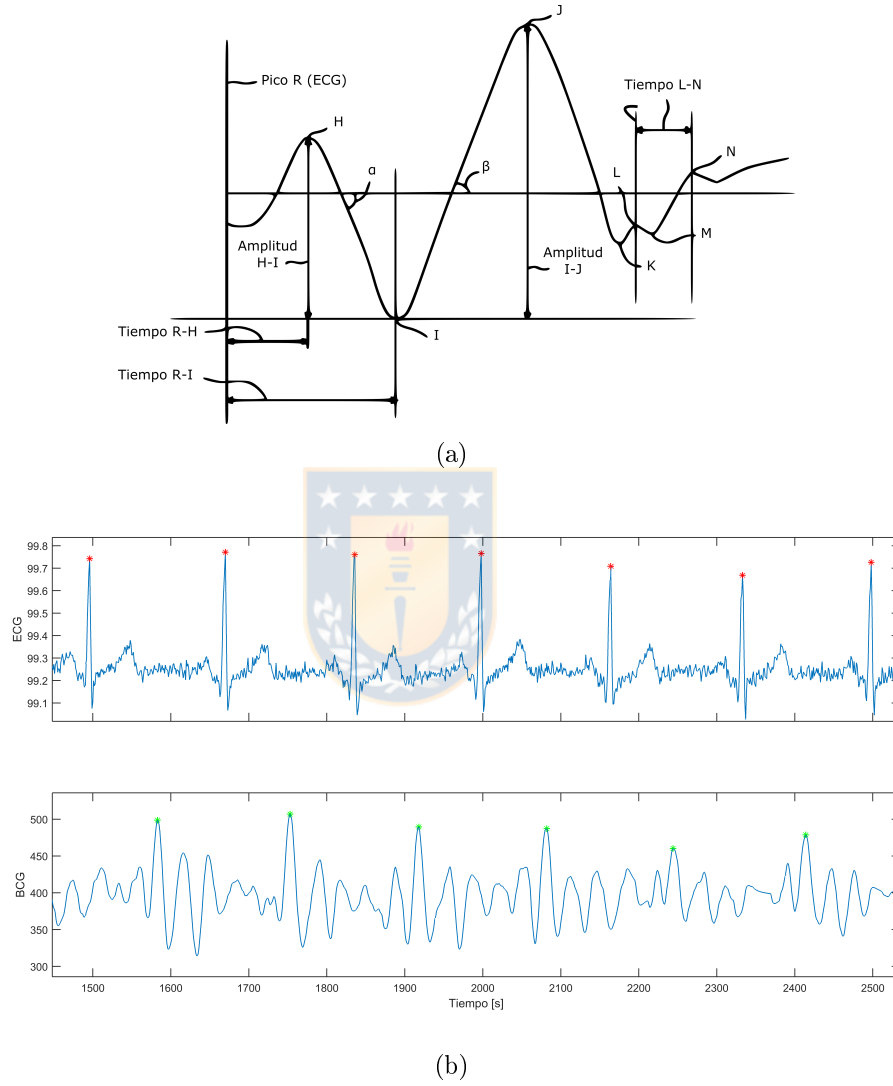


Figura 3.2: Señales teórica y real de balistocardiografía. (a) Forma de onda teórica de balistocardiografía. Fuente: Varri [58]. (b) Señales reales de ECG y BCG. Las marcas corresponden respectivamente a los picos R del ECG, y los picos J del BCG. Fuente: Elaboración Propia.

último tiempo es el intervalo J-J, pues con él se puede calcular el período cardíaco instantáneo, que a su vez es útil en el diagnóstico de arritmias. No obstante, la detección de picos J es en general más difícil que la detección de picos R en el ECG, principalmente porque el BCG tiene una morfología más suave, con picos menos pronunciados, además de presentar más variabilidad entre latidos [59].

3.2.4. Ruidos típicos en la señal

Puesto que el BCG se mide la reacción del cuerpo ante una suma de distintos vectores de fuerza, además de la actividad cardíaca suelen sumarse los movimientos respiratorios, movimientos musculares del paciente o del sensor mismo. La señal resultante también puede contener ruidos de alta frecuencia que interfieren en el circuito electrónico del instrumento. Para obtener una señal pura de BCG, es importante entonces aplicar filtros analógicos o digitales a la señal adquirida. De acuerdo a la literatura, la mayor parte de la información importante se encuentra en el rango 0.3-20 [Hz], por lo que se recomienda filtrarla apropiadamente. Bajo este rango se encuentra el movimiento respiratorio, y sobre él, principalmente ruido eléctrico 3.3.

A pesar de que los nuevos instrumentos para la medición de BCG han mejorado la calidad de los registros, aún siguen siendo propensos a la aparición de ruido por movimientos musculares o del sensor. A diferencia de los antiguos balistocardiógrafos, cuya intención de uso eran ambientes hospitalarios controlados y operados por un experto, los nuevos dispositivos de medición han sido diseñados para el cuidado en casa y para que sean operados por el mismo paciente. Por ejemplo, en un sistema de medición integrado a una cama para el monitoreo nocturno, es esperable que aparezcan interferencias por el movimiento natural de la persona.

Estos ruidos de naturaleza repentina debidos al movimiento del paciente o del sensor, se conocen como artefactos por movimiento y, como se explicó en el apartado 2.1, afectan la señal al punto de hacerla irreconocible. En particular, son un problema para los algoritmos de cálculo de periodo cardíaco instantáneo, que se basan en la detección de máximos locales para encontrar los picos J, pues los picos producidos por el ruido suelen ser confundidos por ondas J, alterando artificialmente el periodo cardíaco y, por ende, haciendo menos confiable esta métrica y el subsiguiente

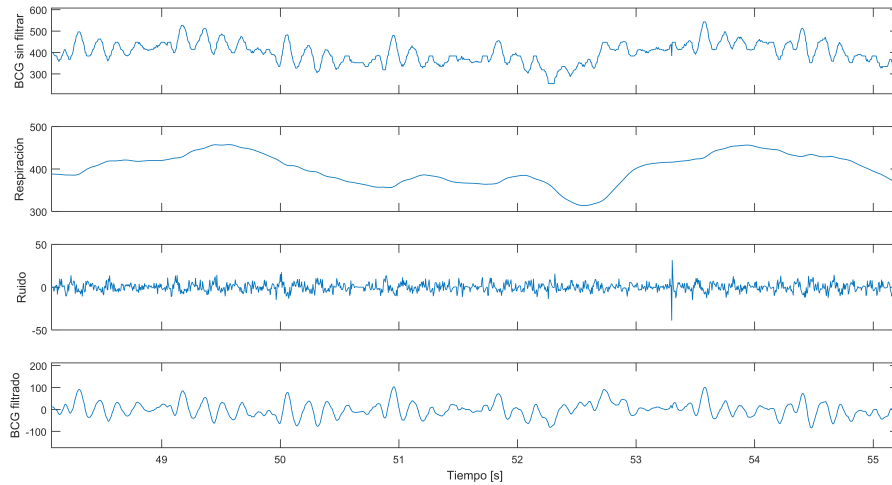


Figura 3.3: Ruidos en señal de BCG real. Señal de BCG real. BCG sin filtrar, movimientos respiratorios, ruido eléctrico de alta frecuencia, y BCG filtrado en banda 1 [Hz] - 20 [Hz]. Fuente: Elaboración Propia.

diagnóstico de arritmias.

3.3. Redes neuronales y aprendizaje profundo

3.3.1. Redes neuronales artificiales

Una red neuronal artificial (*Artificial Neural Network*, ANNs) es un sistema basado en un gran número de unidades simples de cómputo llamadas *perceptrones*, altamente conectadas entre sí [60] (Figura 3.4(a)). Algunas ventajas de las ANNs es que son no-lineales, adaptativas y además altamente paralelizables.

El desarrollo de las ANNs se remonta a la década del 1950, y tuvo periodos sucesivos de auge y declive hasta fines de la década de 1990, principalmente debido a la falta de poder computacional necesaria para la realización de experimentos y mejoras. Sin embargo, desde mediados de los años 2000 ha habido un resurgimiento de la investigación en este tema, atribuido principalmente a nuevos descubrimientos

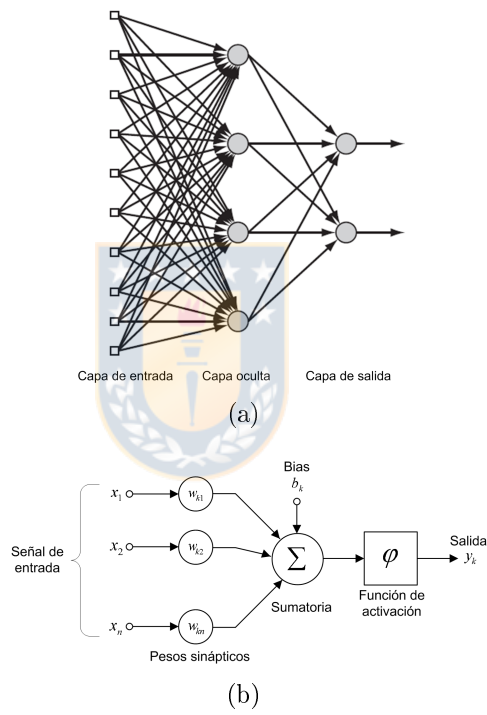


Figura 3.4: Redes neuronales básicas. (a) Red neuronal simple. Fuente: Nielsen [61].
 (b) Perceptrón, modelo no lineal de una neurona. Fuente: Haykin, 2009 [60].

matemáticos útiles para el entrenamiento de estos sistemas, avances en la capacidad del hardware, y también a la cantidad de datos disponibles para el entrenamiento de los sistemas producto de la alta digitalización de la sociedad.

Fundamentos matemáticos

Como se dijo antes, el eslabón primordial de cualquier red neuronal es el perceptrón, una unidad de cómputo que modela el funcionamiento básico de una neurona, y que tiene una componente lineal a la cual se le aplica una función de activación no lineal (Figura 3.4(b)).

$$z = \mathbf{w} \cdot \mathbf{x} + b \quad (3.1)$$

$$a = \phi(z) \quad (3.2)$$

Donde $\mathbf{x}$ es el vector de entrada, $\mathbf{w}$ es el vector de pesos de la neurona, b es el sesgo o *bias*, z es la salida de la parte lineal, ϕ es la función de activación, a es la salida de la neurona y $\cdot$ denota el producto punto entre vectores.

La red se forma por la interconexión de un gran número de perceptrones, agrupados en capas. La primera capa posee la misma dimensionalidad que el vector de entrada. Las salidas de cada elemento de la primera capa se conectan en paralelo a todas las neuronas de la segunda capa, y a su vez las salidas de ésta a las neuronas de la capa siguiente., hasta llegar a la última, que sólo tiene una neurona. Las capas intermedias (entre la primera y la última) son llamadas *capas ocultas*. En la literatura, se recomienda escoger la función ReLU (*Rectified Linear Unit*) como función de activación para las neuronas de todas las capas, excepto la última, donde se aconseja usar la función *sigmoidal* [62].

$$\phi^l(x) = \begin{cases} ReLU(x), & \text{si } l < L. \\ sigmoid(x), & \text{si } l = L. \end{cases} \quad (3.3)$$

Donde L es el número de capas de la red, y

$$ReLU(x) = \begin{cases} 0, & \text{si } x < 0. \\ x, & \text{si } x \geq 0. \end{cases} \quad (3.4)$$

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (3.5)$$

La función de activación ReLU, sin embargo, produce un problema conocido como ReLU moribunda"(*dying ReLU*). Una neurona cuya parte lineal es siempre negativa, origina una salida nula. Puesto que la pendiente de la función ReLU en esa zona también es 0, es muy probable que dicha neurona se atasque en ese valor. En consecuencia, estas neuronas no aportan al cómputo de la clasificación y se vuelven inútiles. Para atacar este problema, se propuso modificar esta función de activación [63], añadiendo una pequeña pendiente a_{lr} (por lo general 0.01) en la zona negativa. Esta función se conoce como *Leaky ReLU*, definida en la Ecuación 3.6.

$$ReLU(x) = \begin{cases} a_{lr}x, & \text{si } x < 0. \\ x, & \text{si } x \geq 0. \end{cases} \quad (3.6)$$

Existen en la literatura diversos algoritmos para hallar los valores óptimos de $\mathbf{w}$ y b , siendo el Descenso de gradiente el más común, ya sea en su versión estocástica ó por lotes. Este algoritmo consiste en buscar el mínimo de una función mediante un proceso de ajuste iterativo, restando al valor actual un valor proporcional al gradiente de la función en el punto actual.

$$a_{m+1} = a_m - \gamma \nabla F(a_m) \quad (3.7)$$

Aquí, F es una función definida y diferenciable en torno al punto a , a_m es el punto actual, a_{m+1} es el punto siguiente, γ es un escalar, y ∇ representa la operación gradiente. Para un γ suficientemente pequeño, $F(a_m) \geq F(a_{m+1})$, lo que con suerte llevará a un mínimo local.

En el último tiempo han aparecido nuevos algoritmos que añaden mejoras al

Descenso de gradiente, acelerando la convergencia y reduciendo el riesgo de quedarse atascados en mínimos locales o mesetas. Algunos de ellos son:

- Descenso de gradiente con moméntum
- AdaGrad
- RMSProp
- Adam

Varios trabajos señalan al algoritmo Adam como la mejor alternativa para redes complejas, por su facilidad de implementación, eficiencia computacional, requerimientos de memoria pequeños, e hiperparámetros de fácil interpretación y que casi no requieren ajustes. El proceso de entrenamiento de la red tiene como objetivo el ajuste de los pesos, es pos de lograr un valor mínimo de una función de costo. Se divide en dos fases: propagación hacia adelante (*forward propagation*) y propagación hacia atrás (*backward propagation*). La primera fase consiste en calcular la predicción del clasificador a partir de una muestra de entrada.

$$\mathbf{a}^0 = \mathbf{x} \quad (3.8)$$

$$\mathbf{z}_i^l = \mathbf{w}_i^l \cdot \mathbf{a}^{l-1} + b_i^l \quad (3.9)$$

$$\mathbf{a}_i^l = \phi(\mathbf{z}_i^l) \quad (3.10)$$

$$\hat{y} = \mathbf{a}^L \quad (3.11)$$

Donde $\mathbf{z}_i^l$ es la parte lineal de la neurona i -ésima de la capa l , $\mathbf{a}_i^l$ es la salida de la neurona i -ésima de la capa l , $\mathbf{a}^l$ es el vector de todas las salidas de la capa l , ϕ^l es la función de activación para la capa l , y $\hat{y}$ corresponde a la salida de la red, es decir, la predicción del clasificador.

Posteriormente, se calcula una función de costo que mide el error de la salida de la red con respecto a un valor esperado. En la literatura se mencionan diversas funciones de costo, pero en general se sugiere utilizar la Entropía Cruzada, pues se ha

visto que otras funciones (como el Error Absoluto Medio) producen gradientes muy pequeños cuando las salidas de algunas neuronas se saturan, dificultando el proceso de aprendizaje. La Entropía Cruzada se define como sigue:

$$C_{ce} = -\frac{1}{m} \sum_{m=0}^M y \ln \hat{y} + (1 - y) \ln (1 - \hat{y}) \quad (3.12)$$

Donde $\hat{y}$ es la salida de la red, y es la salida esperada, y M es el número de ejemplos del set de entrenamiento o del lote. En el campo del aprendizaje profundo, es común añadir un término de *regularización*, que ayuda a evitar el sobre-ajuste del modelo sobre el set de datos de entrenamiento, mejorando así el desempeño sobre el set de prueba.

$$C_{ce} = -\sum_{m=0}^M y \ln \hat{y} + (1 - y) \ln (1 - \hat{y}) + \frac{\lambda_r}{2m} \sum_{l=0}^L \|\mathbf{W}^l\|_F^2 \quad (3.13)$$

Aquí, λ_r es un escalar que controla el impacto del término de regularización sobre el total, y $\|\mathbf{W}^l\|_F^2$ es la norma Frobenius de $\mathbf{W}^l$, que corresponde a la matriz conformada por los pesos de todas las neuronas de la capa l . Este tipo de regularización se conoce como *regularización L2*. Sin embargo, existen otro tipo de métodos alternativos, como la regularización *dropout*, que consiste en seleccionar aleatoriamente ciertas neuronas de cada capa durante el entrenamiento y “apagarlas”, es decir, ignorar su contribución a la capa siguiente [64]. Si bien el funcionamiento de este método no está totalmente explicado, se cree que tiene el efecto de generar múltiples modelos internos independientes, por lo que la red total es capaz de generalizar mejor.

La segunda fase el entrenamiento utiliza el método de *backpropagation* (basado en la regla de la cadena) para calcular los gradientes de $\mathbf{w}$ y b de cada neurona de cada capa. Esto se lleva a cabo desde la última capa de la red hacia atrás, derivando primero C_{ce} con respecto a $\hat{y}$. La nomenclatura comúnmente usada designa de forma genérica a ∂p como la derivada de la función de costo con respecto a p . De esta forma, $\partial \mathbf{a}^l$ representa a la derivada de la función de costo con respecto a $\mathbf{z}^l$. En particular $\partial \mathbf{a}^L$ (ó $\partial \hat{y}$) se calcula como sigue

$$\partial \mathbf{a}^L = \frac{y}{\mathbf{a}^L} + \frac{(1-y)}{1-\mathbf{a}^L} \quad (3.14)$$

Luego, los gradientes de la capa L se utilizan para calcular los de la capa $L-1$, y así hasta llegar a la primera capa. Las expresiones que permiten calcular los gradientes son conocidas:

$$\partial \mathbf{z}^l = \partial \mathbf{a}^l \phi'^l(\mathbf{z}^l) \quad (3.15)$$

$$\partial \mathbf{w}^l = \partial \mathbf{z}^l \mathbf{a}^{l-1} \quad (3.16)$$

$$\partial \mathbf{b}^l = \partial \mathbf{z}^l \quad (3.17)$$

$$\partial \mathbf{a}^{l-1} = (\mathbf{w}^l)^T * \partial \mathbf{z}^l \quad (3.18)$$

$$\phi'^l(x) = \begin{cases} x, & \text{si } l < L. \\ \frac{1}{1+e^{-x}}(1 - \frac{1}{1+e^{-x}}), & \text{si } l = L. \end{cases} \quad (3.19)$$

Donde ϕ'^l es la derivada de la función de activación de la capa l , y $*$ denota el producto interno.

Una vez obtenidos todos los gradientes, se actualizan el peso y *bias* de cada neurona mediante la regla de actualización del método usado, en este caso el método Adam [65]. Es importante notar que la actualización puede realizarse luego de cada predicción (descenso estocástico), luego de presentar todo el set de datos (descenso por lote o *batch*), o luego de un subconjunto del set completo (descenso por mini-lotes o *mini-batches*). Para el método Adam, se recomienda esta última forma. La regla de actualización es la siguiente:

$$\mathbf{w}_i^l[k] = \mathbf{w}_i^l[k-1] - \alpha \frac{\hat{V}_{\partial \mathbf{w}_i^l}}{\sqrt{\hat{S}_{\partial \mathbf{w}_i^l} + \epsilon}} \quad (3.20)$$

$$b_i^l[k] = b_i^l[k-1] - \alpha \frac{\hat{V}_{\partial b_i^l}}{\sqrt{\hat{S}_{\partial b_i^l}} + \epsilon} \quad (3.21)$$

Donde:

$$\begin{aligned} \hat{V}_{\partial \mathbf{w}_i^l}[k] &= \frac{\beta_1 \hat{V}_{\partial \mathbf{w}_i^l}[k-1] + (1-\beta_1) \partial \mathbf{w}_i^l[k]}{1-\beta_1^k} \\ \hat{V}_{\partial b_i^l}[k] &= \frac{\beta_1 \hat{V}_{\partial b_i^l}[k-1] + (1-\beta_1) \partial b_i^l[k]}{1-\beta_1^k} \\ \hat{S}_{\partial \mathbf{w}_i^l}[k] &= \frac{\beta_2 \hat{S}_{\partial \mathbf{w}_i^l}[k-1] + (1-\beta_2) (\partial \mathbf{w}_i^l[k])^2}{1-\beta_2^k} \\ \hat{S}_{\partial b_i^l}[k] &= \frac{\beta_2 \hat{S}_{\partial b_i^l}[k-1] + (1-\beta_2) (\partial b_i^l[k])^2}{1-\beta_2^k} \end{aligned}$$

Aquí, $\mathbf{w}_i^l[k]$ y $b_i^l[k]$ representan respectivamente al vector de pesos y el *bias* de la neurona i -ésima de la capa l -ésima, para la iteración k , $\hat{V}_{\partial b_i^l}[k]$ y $\hat{V}_{\partial \mathbf{w}_i^l}[k]$ son los términos de momento del método para la iteración k , β_1 y β_2 son escalares que controlan el “largo” de la media móvil calculada en los denominadores de los términos anteriores, y $\hat{S}_{\partial \mathbf{w}_i^l}[k]$ y $\hat{S}_{\partial b_i^l}[k]$ son términos adaptativos de ajuste de la tasa de aprendizaje α .

3.3.2. Aprendizaje profundo

Deep Learning (DL) es un nombre acuñado hace alrededor de diez años atrás a una clase particular de métodos de aprendizaje automático que poseen dos características importantes [66]:

1. Son modelos que consisten múltiples capas o etapas de procesamiento no-lineal de información y
2. Son métodos de aprendizaje supervisado, no-supervisado, o para representación de características, cuyas capas son sucesivamente más abstractas.

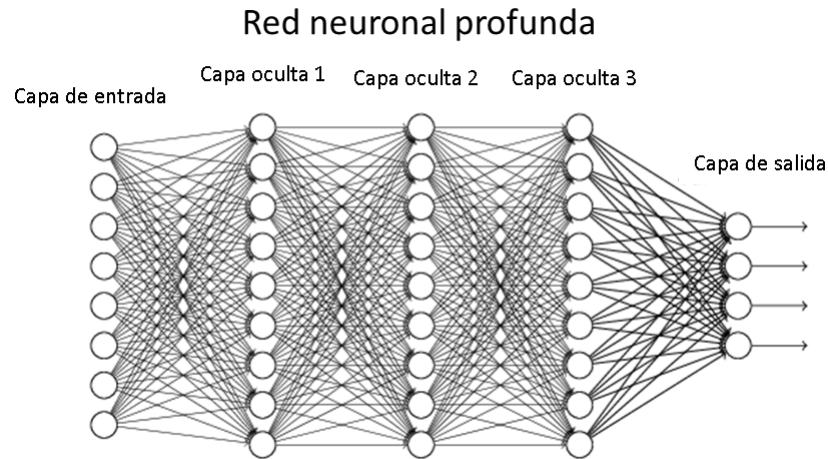


Figura 3.5: Esquema de una red neuronal profunda. Fuente: Nielsen [61]

Existen variadas arquitecturas de DL, pero una de las más usadas son aquellas basadas en ANNs. Las arquitecturas clásicas de ANNs poseían a lo más una capa oculta, con sólo cientos de unidades de cómputo. La diferencia sustantiva de las redes profundas (*Deep Neural Networks*, DNNs) con respecto a las primeras, es que tienen un número mucho mayor de unidades (10^5 o más), agrupadas en una gran cantidad de capas intermedias u ocultas (Figura 3.5). Estas arquitecturas suelen entrenarse con una cantidad muy grande de datos, aspecto clave para lograr un buen desempeño. Adicionalmente, los métodos de entrenamiento incluyen etapas que reducen el sobreajuste del modelo a los datos [67], lo cual es importante si se trabaja con bases de datos extensas. Con todo lo anterior, se han logrado sistemas que, al tener un gran número de parámetros, son mucho más flexibles y capaces de modelar relaciones complejas y altamente no lineales entre entradas y salidas [68].

En general, la mayoría de los sistemas de aprendizaje automático no procesan los datos tal cual como fueron obtenidos, si no que trabajan con una representación de estos mismos. Estas representaciones son conocidas en el ámbito del aprendizaje automático como *características* o *features*, y pueden pensarse como una transformación o proyección de los datos del espacio original a un nuevo *espacio de características*. El objetivo de esta transformación es extraer la información más relevante de los datos originales, y al mismo tiempo disminuir su dimensionalidad, con el consiguiente

ahorro computacional. Definir cuántas y cuáles características son seleccionadas es tarea del diseñador del sistema y, en consecuencia, los algoritmos tienden a depender fuertemente de aquella elección. Técnicas clásicas de extracción de características son el análisis de componentes principales (PCA) y el análisis de componentes independientes (ICA).

Uno de los aspectos relevantes de los algoritmos de aprendizaje representacional (y en particular aquellos de DL es que, además de aprender funciones complejas para *mapear* la representación de los datos a una valor de salida específico, también son capaces de descubrir la representación en sí. En este ámbito, las CNNs han destacado por sus buenos resultados en diversas aplicaciones.

3.3.3. Redes neuronales convolucionales

Las CNNs son un tipo especial de redes neuronales, usadas típicamente para procesar datos que poseen una topología conocida y en forma de grilla. Ejemplos clásicos son las imágenes, que pueden ser pensadas como una grilla 2D de píxeles, y la series de tiempo, que pueden considerarse grillas unidimensionales. Fueron presentadas por primera vez por LeCun et al. en 1989 [69], y durante las últimas dos décadas han sido foco de extenso estudio y desarrollo.

A grandes rasgos, una CNN se parece en muchos aspectos a una red profunda común: se componen de neuronas con parámetros entrenables agrupadas en capas, que reciben entradas y realizan operaciones de suma y multiplicación y, opcionalmente, aplican una función no lineal para obtener una salida. Sin embargo, y a diferencia de las primeras, poseen tres tipos distintos de capas que desempeñan tareas particulares: las capas convolucionales, que se desempeñan como filtros de parámetros ajustables; las capas de *pooling*, que reducen la dimensionalidad de los datos al escoger la media o el máximo dentro de una vecindad; y las capas totalmente conectadas o *fully-connected*, que corresponden a las capas típicas de las redes neuronales clásicas [70].

Una propiedad relevante de las CNN es que son capaces de extraer las características esenciales de los datos de entrada crudos, y generar representaciones com-

plejas que son expresadas en términos de otras representaciones más simples, como si fueran conceptos relacionados jerárquicamente [71]. Estructuras de este tipo se asemejan mucho a aquellas encontradas en los cerebros de los mamíferos, donde los múltiples niveles de abstracción corresponden a distintas áreas de la corteza cerebral. Un ejemplo de lo anterior es el sistema de visión en los primates (Figura 3.6), donde la información se procesa en una secuencia de etapas: detección de bordes, detección de formas primitivas, y así sucesivamente hasta lograr detectar patrones más complejos [67, 72].

Esta característica de las CNN es de suma importancia pues los problemas más complejos de resolver computacionalmente suelen ser, irónicamente, aquellos que parecen más simples para los humanos. En este sentido, la dificultad radica, por lo general, en que no somos capaces de definir dichos problemas formalmente, o especificarlos explícitamente a partir de datos de entrada crudos. En nuestro caso, diferenciar un segmento de BCG limpio de otro con artefactos resulta sencillo a simple vista, pero escoger qué características son útiles para representar y distinguir uno de otro parece ser una tarea mucho más compleja. Es llamativo entonces utilizar una CNN para atacar el problema central de este trabajo.

Algunos casos exitosos de aplicación de sistemas basados en DL se presentan a continuación:

1. Clasificación de objetos en imágenes: En el Desafío de ImageNet sobre Reconocimiento Visual de Gran Escala (ILSVRC, por su sigla en inglés), los participantes compiten por reconocer objetos de acuerdo a clases definidas previamente. En el trabajo de Krizhevsky [73], se utiliza una CNN que se entrena con 1.2 millones de muestras para clasificar objetos en 1000 clases distintas. Los resultados se consideran sorprendentes al tener una tasa de error de sólo 15.3 %.
2. Generación automática de títulos de imágenes: Karpathy [74] desarrolla un sistema para generar descripciones de imágenes en lenguaje natural, utilizando una combinación de CNNs y redes neuronales recurrentes (RNN).
3. Traducción automática de texto: Algunos sistemas como Google Translator utilizan CNNs para traducir automáticamente el texto presente en imágenes,

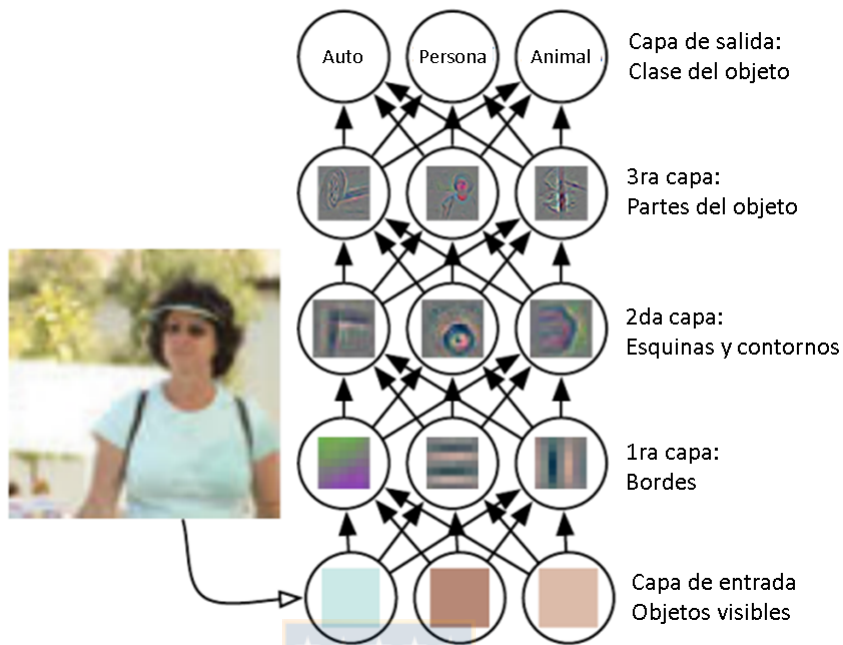


Figura 3.6: Red neuronal profunda. Las capas ocultas representan la imagen de entrada en conceptos que van desde lo simple a lo más complejo. Fuente: Goodfellow, 2016 [76].

que pueden ser obtenidas de cualquier parte mediante, por ejemplo, la cámara de un celular. Un resumen de los avances más recientes se presenta en [75].

Si bien las aplicaciones en procesamiento de imágenes son las más numerosas en la literatura, el uso de las CNNs se ha expandido también al procesamiento de señales o series de tiempo, naciendo así el término *redes neuronales convolucionales unidimensionales*, o 1D CNNs. Teniendo esto en cuenta, podemos plantear la detección de MAs en señales de ECG y BCG como una aplicación de detección de patrones morfológicos en señales unidimensionales, por lo cual resulta razonable utilizar 1D CNNs para este fin.

Fundamentos matemáticos

Debido a sus orígenes en el procesamiento de imágenes, los datos de entrada de las CNNs suelen ser considerados "volúmenes", que tienen alto, ancho y profundidad, siendo ésta última el número de canales de la imagen (tres, en las imágenes RGB). Puesto que las señales de ECG y BCG que se utilizarán son unidimensionales y de un sólo canal, para efectos del desarrollo de las ecuaciones consideraremos que la señal de entrada no es un "volumen", sino un vector unidimensional denotado con la letra $\mathbf{x}$ y de dimensión N .

Las arquitecturas de CNN se diseñan conectando sucesivamente capas de tipo convolucional, *max-pooling* y totalmente conectadas, donde cada una aplica una transformación no lineal a los datos mediante una función diferenciable. Una estructura típica se muestra en la Figura 3.7, compuesta por una serie de capas convolucionales y *max-pooling* conectadas de forma alternada, a la que finalmente se añade una o más capas totalmente conectadas.

Como se mencionó, las capas convolucionales consisten en un conjunto de filtros entrenables o *kernels*. Éstos son vectores cuyo alcance temporal o campo receptivo es usualmente pequeño (ej. 3, 5, o 7 muestras). Puede haber más de un filtro por capa, siendo el número de filtros por capa un hiperparámetro del modelo. La idea es que cada filtro aprenda a detectar un patron particular de la entrada, independiente de su ubicación temporal. Así, podemos pensar cada capa convolucional como un conjunto de Q^l *kernels* de igual dimensión, cuyo número es igual al número de vectores de salida.

Durante el paso de propagación hacia delante, la primera capa convolucional recibe un vector $\mathbf{s}_1^0 = \mathbf{x}$, que corresponde a una de las muestras del conjunto de entrenamiento. A modo general, cada capa convolucional realiza la convolución $(*)$ entre el vector $\mathbf{s}_p^{l-1}$ de entrada y cada uno de los *kernels* $\mathbf{f}_{p,q}$ de dimensión K_c de la capa l , operación definida en la Ec. 3.22.

$$\mathbf{c}_{p,q}^l = \mathbf{s}_p^{l-1} * \mathbf{f}_q^l \quad (3.22)$$

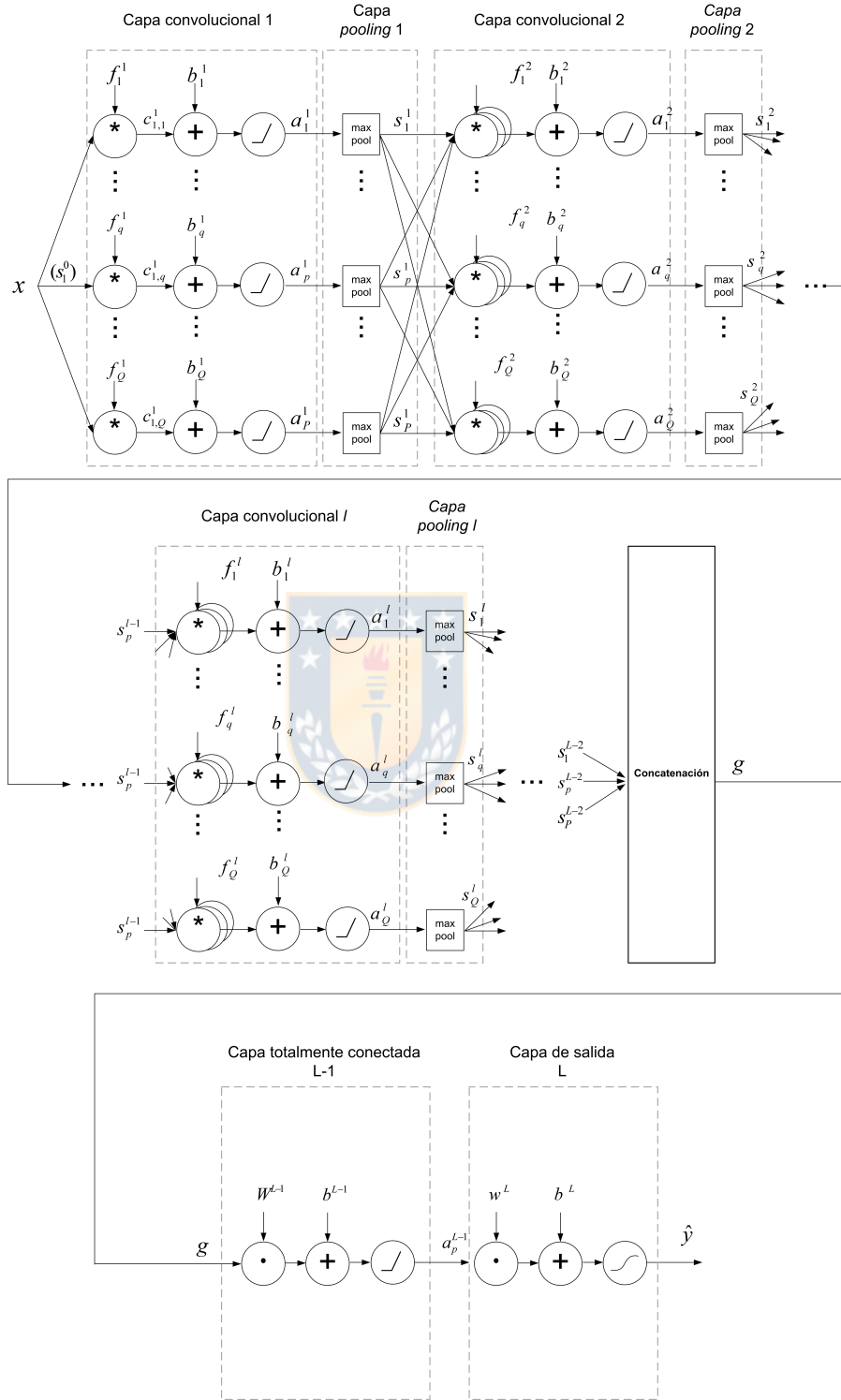


Figura 3.7: Esquema general de una red neuronal convolucional unidimensional.
Fuente: Elaboración Propia.

$$\mathbf{c}_{p,q}^l(j) = \sum_{k=0}^{K_c} \mathbf{s}_p^{l-1}(k) \cdot \mathbf{f}_q(j-k) \quad (3.23)$$

Aquí, p es el índice del conjunto de vectores de salida de la capa anterior $l-1$, y q es el índice del conjunto de filtros de la capa actual l . Es importante hacer hincapié en que, en vez de existir un filtro especializado en cada capa convolucional para ser operado con cada vector de entrada $\mathbf{s}_p^{l-1}$, en realidad sólo hay conjunto de Q^l *kernels* $\mathbf{f}_q$, que se convolucionan con cada uno de los vectores de entrada. Esta estrategia se conoce como reutilización de parámetros, y permite reducir la utilización de memoria de los algoritmos.

Una técnica común es usar *zero-padding* sobre la entrada, si las dimensiones no son múltiplos del tamaño del filtro. Lo anterior consiste en rellenar los extremos de la señal con ceros, para que la operación quede totalmente definida. La convolución discreta clásica se basa “recorrer” la señal en desplazamientos de una muestra. Sin embargo, no es extraño encontrar implementaciones de operaciones de convolución modificadas, donde el *kernel* es desplazado en más de una muestra. La cantidad de desplazamientos S_c se conoce como paso o *stride*, siendo $S_c = 1$ el caso típico. Esta operación equivale a realizar la operación de convolución normal, y luego aplicar *downsampling* por descarte a la salida.

$$\mathbf{c}_{p,q}^l(j) = \sum_{k=0}^{K_c} \mathbf{s}_p^{l-1}(k) \cdot \mathbf{f}_q(jS_c - k) \quad (3.24)$$

De esta forma, para un vector $\mathbf{s}_p^{l-1}$ de entrada de dimensión S^{l-1} y *padding* P_c , convolucionado con un filtro $\mathbf{f}_q$ de dimensión K_c con paso S_c , la dimensión del vector de salida está dada por

$$C = \frac{S^{l-1} - K_c + 2P_c}{S_c} + 1 \quad (3.25)$$

De aquí es posible ver que tanto S_c como P pueden ser elegidos convenientemente para obtener un vector de salida de igual dimensión que el de entrada.

El resultado final de la capa convolucional es un conjunto de vectores $\mathbf{a}_q^l$, donde cada uno se calcula como

$$\mathbf{a}_q^l = \phi\left(\sum_{p=0}^{Q^{l-1}} \mathbf{s}_p^{l-1} * f_q^l + b_q^l\right) \quad (3.26)$$

Donde b_q^l es un escalar (*bias*), y ϕ es una función no lineal, típicamente ReLU o *Leaky* ReLU.

Luego de una o más capas convolucionales, es común insertar periódicamente capas de *pooling* con el fin de aplicar *downsampling* a la entrada, y de esta forma reducir progresivamente la dimensionalidad de los datos. Este tipo de capas consiste en un filtro de dimensiones fijas, mediante el cual se aplica una operación de promedio (*average pooling*) o máximo (*max-pooling*) sobre una vecindad espacial del vector de entrada. Si consideramos K_s la extensión espacial del filtro de *pooling*, y S_s el paso o *stride* con que se aplica, la operación *max-pooling* en la capa l y sobre la salida del filtro convolucional $\mathbf{a}_q$ queda definida como sigue

$$\mathbf{s}_q^l(i) = \max_{S_s i \leq j \leq S_s i + K_s} \mathbf{a}_q^l(j) \quad (3.27)$$

Por su parte, la dimensión S^l de cada uno de los p vectores de salida $\mathbf{s}_p^l$ de la capa l queda dada por

$$S^l = \frac{C - K_s}{S_s} + 1 \quad (3.28)$$

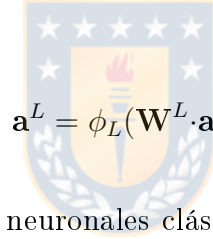
El conjunto capa convolucional más capa de *max-pooling* suele ser llamado *capa de extracción de características* (*Feature Extraction*, FE), pues su salida es una representación de la entrada, de menor dimensionalidad, y que muestra la presencia de ciertos patrones en ella.

Por su parte, las capas totalmente conectadas operan igual que las capas de las redes neuronales clásicas, donde cada neurona de la capa se conecta a la totalidad de

las activaciones de la capa anterior. Una o más capas de este tipo suelen insertarse sólo al final de la arquitectura, actuando en si misma como una red neuronal multi-capas. Para efectos de la presente explicación, consideraremos la utilización sólo dos capas totalmente conectadas $L - 1$ y L . La primera recibe los Q^{L-2} vectores de características $\mathbf{s}_p^{L-2}$ obtenidos en la capa $L - 2$, concatenados en un gran vector de dimensiones $G = Q^{L-2} * S^{L-2}$ que denominaremos $\mathbf{g}$. Así,

$$\mathbf{a}^{L-1} = \phi_{L-1}(\mathbf{W}^{L-1} \cdot \mathbf{g} + b^{L-1}) \quad (3.29)$$

La capa L se conoce como capa de salida, y es la encargada de entregar la predicción del clasificador $\hat{y} = \mathbf{a}^L$ ante la entrada $\mathbf{x}$. En esta capa, y al igual que en el caso de las redes *feed-forward*, la función de activación más utilizada es la función sigmoideal.



$$\mathbf{a}^L = \phi_L(\mathbf{W}^L \cdot \mathbf{a}^{L-1} + b^L) \quad (3.30)$$

Al igual que en las redes neuronales clásicas, una vez completado el proceso de propagación hacia delante, es necesario calcular el error de clasificación entre $\hat{y}$ e y mediante una función de costo, usualmente la Entropía Cruzada (Ec. 3.12). El paso siguiente es retro-propagar el error mediante el método de *backpropagation*, que permite calcular los gradientes que se utilizarán para actualizar los pesos de cada neurona de acuerdo a las reglas del algoritmo de optimización elegido. Como se explicó en la Sección 3.3.1, utilizaremos el método Adam, cuyas reglas quedaron definidas por las Ecuaciones 3.20 y 3.21.

La aplicación del método de *backpropagation* para la obtención de las de las ecuaciones de los gradientes no es una tarea simple en el caso de las CNNs, razón por la cual no serán expuestas en este trabajo. Sin embargo, éstas pueden ser consultadas en detalle en [74, 77, 78]. Para efectos de programar la red propuesta, una buena noticia es que la mayoría de los *frameworks* modernos de DL (Tensorflow, Keras, Caffe, etc.) poseen funciones ya implementadas para ejecutar este paso.

Capítulo 4. Materiales y métodos

De acuerdo a los objetivos planteados, se debe diseñar un clasificador binario basado en CNNs para detectar presencia de artefactos por movimiento en señales de ECG y BCG. La Figura 4.1 muestra un esquema general de los pasos a seguir para lograr este cometido.

Como se mencionó en el Capítulo 1, debido a que el problema de aparición de artefactos es transversal a los distintos métodos de adquisición de señales cardíacas, y considerando que el volumen de investigación en ECG es mayor y que existen bases de datos estándar, se propone diseñar primero un clasificador basado en CNNs para artefactos por movimiento en señales de ECG. Una vez se haya escogido una arquitectura apropiada, se procederá a implementar una similar para la detección de artefactos en BCG. Se separará entonces el problema en dos grandes etapas

A Diseño de CNN para detección de MAs en ECG y posterior validación

B Diseño de CNN para detección de MAs en BCG y posterior validación

La metodología a seguir en cada una de estas etapas es muy similar, y se dividirá en los siguientes pasos

1. Elección de hardware y software

Se escogerá el sistema de hardware y las herramientas de software a utilizar, de acuerdo a los recursos disponibles.

2. Creación del conjunto de datos

El conjunto de datos debe estar compuesto por muestras de señales limpias (clase negativa) y de señales contaminadas con artefactos (clase positiva), ojalá en una proporción similar.

3. Diseño de la arquitectura de la CNN

A partir de una arquitectura básica propuesta, se explorarán distintas variaciones de los hiperparámetros de la red.

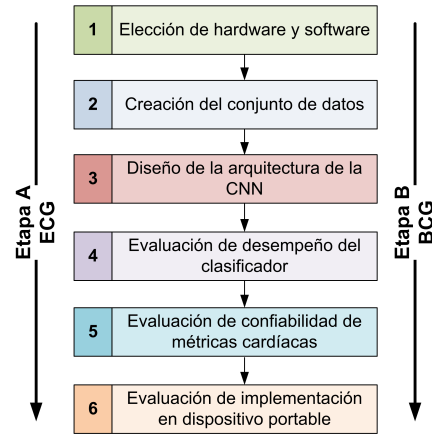


Figura 4.1: Metodología general de esta investigación. Dividida en pasos consecutivos, se aplicará primero para el diseño y evaluación del clasificador de ECG, y luego para el clasificador de BCG. Fuente: Elaboración Propia.

Esta exploración implica múltiples procesos de entrenamiento de la red, dividiendo la base de datos antes creada en subconjuntos de entrenamiento (80 %), validación (10 %) y prueba (10 %).

4. Evaluación de desempeño del clasificador

Con la arquitectura final, se seleccionarán métricas para medir el desempeño y se entrenará el modelo múltiples veces mediante una estrategia de validación cruzada *k-fold*.

5. Evaluación de la confiabilidad de métricas cardíacas

Se integrará el clasificador diseñado en un sistema de detección de detección de latidos y cálculo de período cardíaco instantáneo, y se evaluará su impacto en la confiabilidad de los resultados, mediante métricas seleccionadas apropiadamente.

6. Evaluación de implementación en dispositivo portable

A partir del número de operaciones del algoritmo, el tiempo de cálculo de una predicción y las restricciones técnicas propias de un sistema de medición de BCG portable, se evaluará la factibilidad de implementar el método propuesto en un sistema portable.

Tabla 4.1: Características del sistema computacional a utilizar. Fuente: Elaboración Propia.

Procesador	Intel Core i7-4790
Frecuencia de reloj	3.60 [GHz]
Núcleos	8
Memoria Caché	8 [MB]
Memoria RAM	16 [GB]
Tarjeta de video dedicada	No
Sistema operativo	openSUSE Leap 42.3

El presente capítulo ahondará en cada uno de estos pasos, terminando con una breve discusión de la metodología propuesta.



4.1. Elección de hardware y software

4.1.1. Hardware

Para el diseño de la arquitectura de la red, su entrenamiento y las pruebas experimentales a desarrollar, se utilizará un computador disponible en el Laboratorio de Ingeniería Civil Biomédica de la Universidad de Concepción, cuyas características se detallan en la Tabla 4.1, y al cual accederá mediante protocolo SSH.

4.1.2. Software

La implementación de la red neuronal y su entrenamiento se realizará en Python v. 3.4.6, utilizando el *framework* Keras v. 2.2.2, con TensorFlow v. 1.8.0 como *backend*.

Keras, de acuerdo a su página oficial, es *una API de alto nivel, escrita en Python, y capaz de correr sobre Tensorflow, CNTK o Theano*. Fue desarrollada *con el foco puesto en permitir la experimentación rápida*, y permitiendo así mejores investiga-

ciones.

Para la evaluación de desempeño se hará uso del módulo Scikit-Learn v. 0.19.1, que contiene implementaciones de métricas de desempeño, entre muchas otras capacidad. Se utilizarán también las herramientas de la biblioteca WFDB (*WaveForm DataBase*) de PhysioNet, tanto en su versión nativa como su traducción a Python (wfdb-python).

4.2. Creación del conjunto de datos para CNN

Para poder entrenar el clasificador basado en CNNs, es necesario contar con un conjunto de datos que conste de muchos ejemplos señales de ECG (y BCG) limpias y contaminadas, de igual tamaño o largo. Cada muestra de señal limpia se rotulará como clase negativa (0), y las muestras contaminadas se rotularán como clase positiva (1). El largo de cada una de estas muestras o segmentos se considerará como un hiperparámetro del modelo, pues se intuye que esto puede tener implicancias en el desempeño del clasificador, y a priori no se sabe qué opción dará mejores resultados.

Las muestras del conjunto de datos de ECG se obtendrán dividiendo los registros de una base de datos estándar de ECG de forma consecutiva. Por su parte, los ejemplos del conjunto de datos de BCG se conseguirán de la división de una serie de registros de BCG sintetizados a partir de las mediciones obtenidas por Javier Chávez en [18].

4.2.1. Base de datos de ECG

Existen diversas bases de datos disponibles para el estudio del ECG. En particular, una de las más utilizadas como *benchmark* para la evaluación de algoritmos de detección de latido, es la base de datos MIT-BIH proporcionada por el Instituto de Tecnología de Massachusetts y el Hospital Beth Israel [79]. Ésta es un conjunto de diez bases de datos destinadas a distintos propósitos [80], siendo la Base de Datos de Arritmias la más frecuentemente usada.

Base de Datos de Arritmias

La Base de Datos de Arritmias consiste de 48 registros de ECG de alta calidad, de media hora de duración, muestreados a 360 [Hz] y 11 bits de resolución, y en un rango de 10 [mV]. Contiene además un archivo de anotaciones manuales para cada registro, que indican la ocurrencia de diversos eventos y su momento de aparición, como latidos normales, latidos prematuros y otros tipos de anomalías, cambios de ritmo y presencia de ruidos o latidos inclasificables [81]. En total, la base de datos cuenta con 116137 complejos QRS.

Veinticinco de los cuarenta y ocho registros fueron escogidos por presentar arritmias poco frecuentes, y los veintitrés restantes corresponden a registros ambulatorios elegidos de forma aleatoria. La mayoría de ellos contiene complejos QRS de morfología claramente reconocible, pero algunos (como el 108 o el 207) presentan latidos con formas poco comunes, además de ruido y artefactos.

Base de Datos para Pruebas de Estrés de Ruido

Adicionalmente se hará uso de la Base de Datos para Pruebas de Estrés de Ruido (MIT-BIH NST), que contiene tres registros de media hora de duración. Cada uno de ellos corresponde a un tipo distinto de ruido: variación de línea base, artefactos musculares, y artefactos por movimiento del electrodo (EM). Éste último es de interés en el presente trabajo.

Contaminación con ruido

De lo anterior, es fácil deducir que la Base de Datos de Arritmia proveerá las muestras de la clase negativa, y el registro EM, por su parte, las muestras de la clase positiva. Sin embargo, esta estrategia nos llevaría a obtener una base de datos totalmente desbalanceada. Se propone entonces utilizar la herramienta *nst* de la suite WFDB, que permite contaminar registros de ECG con ruido de forma controlada. El diagrama de la Figura 4.2(a) muestra gráficamente el funcionamiento de esta herramienta.

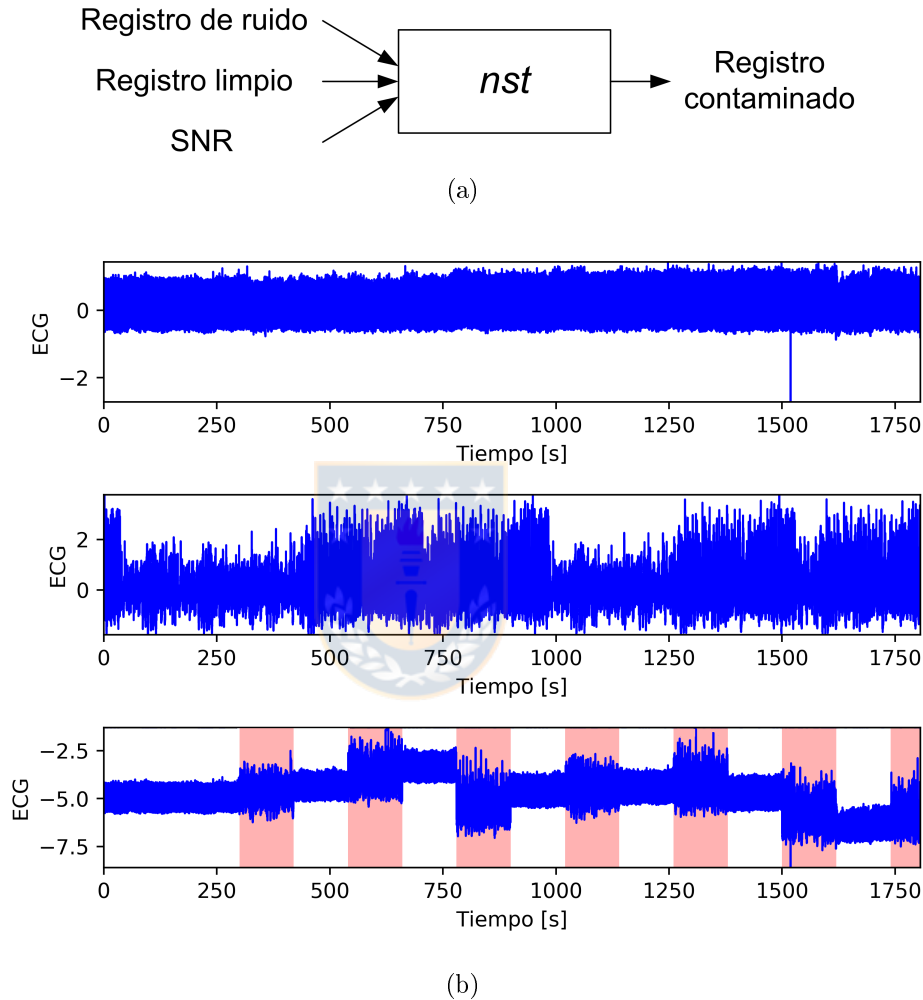


Figura 4.2: Herramienta *nst*. (a) Diagrama explicativo de la herramienta. Fuente: Elaboración Propia. (b) Registro de ECG limpio, registro EM, y registro contaminado con *nst*. Las zonas destacadas corresponden a los segmentos contaminados, de acuerdo al protocolo estándar. Se aprecia que la adición de ruido es temporalmente correlativa, quedando zonas del registro EM que no son utilizadas para la contaminación. Fuente: Elaboración Propia.

nst recibe como entradas un registro de ECG limpio, un registro de ruido y un nivel de Razón Señal a Ruido (SNR) deseado. De acuerdo al manual [82], *nst* por defecto genera una señal contaminada de acuerdo a un protocolo estándar 4.2(b) que consiste en

- Cinco minutos de señal limpia ("periodo de entrenamiento")
- Segmentos sucesivos de dos minutos de señal contaminada seguidos de dos minutos de señal limpia, hasta terminar el registro.

Los segmentos contaminados se obtienen añadiendo el ruido de acuerdo a la siguiente ecuación

$$o = s + a * n + b \quad (4.1)$$

Donde o es la señal contaminada de salida, s es la señal limpia de entrada, n es el registro de ruido, y a y b son dos escalares calculados de forma de obtener el SNR deseado. La Razón Señal a Ruido se define como

$$SNR = 10 \log\left(\frac{S}{N}\right) \quad (4.2)$$

Aquí, S es la potencia de la señal limpia y N la potencia de la señal de ruido. Como la cantidad de ruido a añadir se controla mediante el factor multiplicativo a , la Ecuación 4.2 se puede reescribir como

$$SNR = 10 \log\left(\frac{S}{Na^2}\right) \quad (4.3)$$

La ganancia a se calcula entonces despejando aquél término de la Ecuación 4.3.

Un defecto de *nst* es que, por la forma temporalmente correlativa en que se adiciona el ruido a la señal de salida, aproximadamente la mitad del registro de ruido queda sin utilizarse (Figura 4.2(b)). Para hacer mejor uso de los datos disponibles,

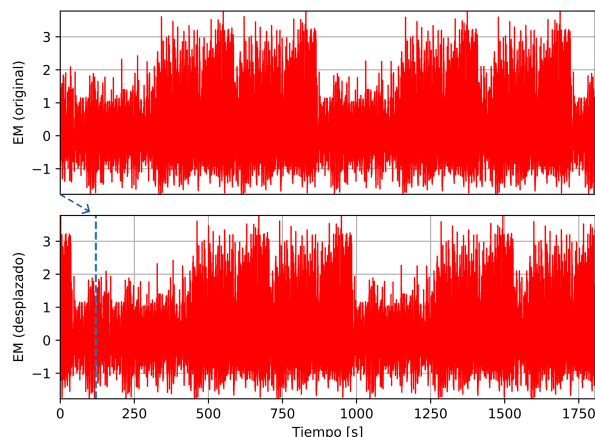


Figura 4.3: Registros de artefactos por movimiento EM1 y EM2. Este último ha sido obtenido por desplazamiento circular de 2 [min], que corresponde al largo de los segmentos contaminados de acuerdo al protocolo estándar de *nst*. Fuente: Elaboración Propia.

se creará un segundo registro de artefactos, nombrado EM2, obtenido mediante el desplazamiento circular del registro EM en 2 minutos 4.3.

Se propone entonces crear dos conjuntos de registros contaminados. En el primero, los 48 registros limpios serán contaminados con el registro EM (que llamaremos de ahora en adelante EM1), a seis niveles de SNR (24 [dB], 18 [dB], 12 [dB], 6 [dB], 0 [dB] y -6 [dB]), obteniéndose 288 registros contaminados. El segundo conjunto será similar, pero contaminado con EM2. A modo de ejemplo, en la Figura 4.4 se puede ver un extracto temporal del registro 100, contaminado a distintos niveles de SNR.

Determinación de niveles de SNR

Como es de imaginar, el nivel de SNR de un registro particular tiene impacto en el desempeño de los algoritmos de detección de latidos. Por ejemplo, un algoritmo de detección de latidos aplicado a un registro contaminado a $\text{SNR} = 24$ [dB], donde la presencia de ruido es casi imperceptible, tendrá mejor desempeño de el mismo algoritmo aplicado a un registro contaminado a $\text{SNR} = -6$ [dB], donde casi no se distingue la señal original. De aquí, surge el problema de determinar qué es lo que

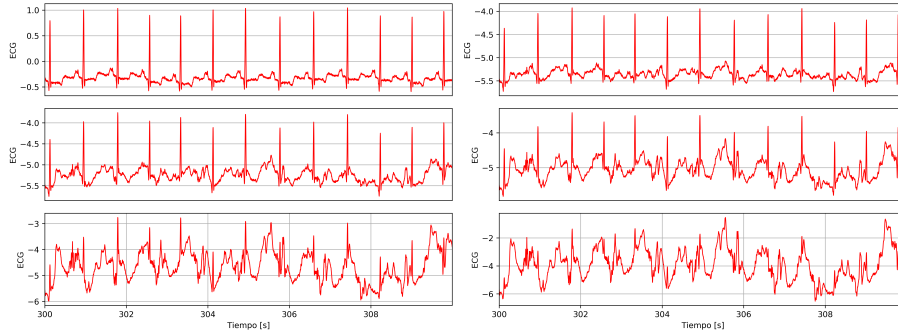


Figura 4.4: Extracto del registro 100 contaminado a distintos niveles de SNR, 24 [dB], 18 [dB], 12 [dB], 6 [dB], 0 [dB] y -6 [dB] (de izquierda a derecha y de arriba a abajo). Fuente: Elaboración Propia.

se considerará como una señal contaminada.

Para resolver el problema anterior de una forma cuantitativa, se aplicó el algoritmo de detección de latidos SQRS, disponible en la suite WFDB, a cada uno de los 288 registros. Luego, se compararon latido a latido las anotaciones generadas por el algoritmo contra las anotaciones de referencia con la herramienta *bxb*, obteniéndose la Sensibilidad y el Valor Predictivo Positivo, dos métricas estadísticas típicas para la evaluación de algoritmos de detección de latidos, y definidas en la Sección 4.5. Se agruparon las estadísticas por nivel de SNR, obteniéndose el gráfico de la Figura 4.5. Se puede apreciar que a partir de 12 [dB] ocurre una caída drástica en la Sensibilidad.

De acuerdo a lo anterior, se define que todas las muestras obtenidas de los segmentos contaminados de los registros de $\text{SNR} \leq 12$ serán rotuladas como clase positiva. Así, los registros contaminados a 24 y 18 [dB] no se considerarán para la etapa de entrenamiento, puesto que no poseen zonas contaminadas de forma importante. Luego, la base de datos de ECG contaminado que se utilizará para la creación del conjunto de datos corresponde a los dos grupos de 192 registros contaminados con EM1 y EM2 respectivamente, a 12 [dB], 6 [dB], 0 [dB] y -6 [dB] de SNR. El proceso de contaminación se resume en la Figura 4.6.

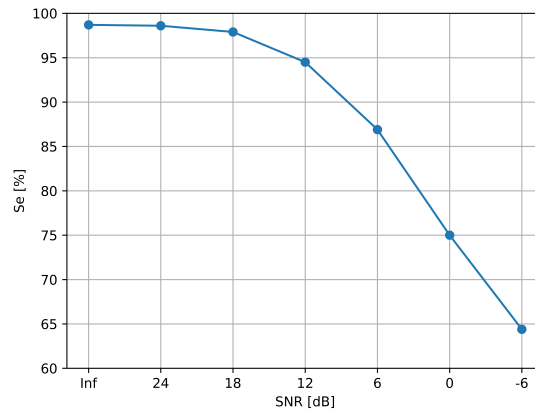


Figura 4.5: Sensibilidad v/s SNR del algoritmo SQRS para la base de datos de ECG contaminado. Fuente: Elaboración Propia.

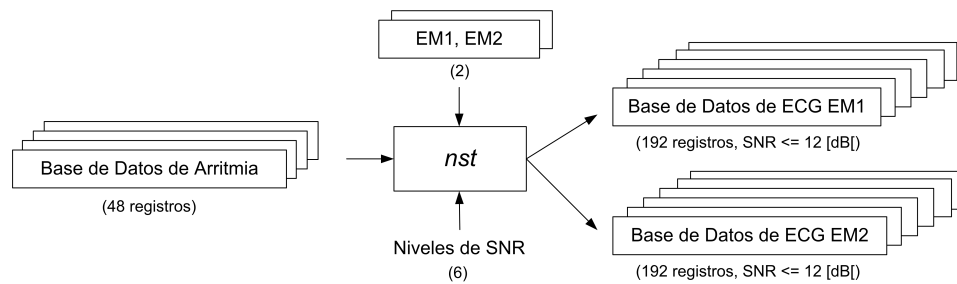


Figura 4.6: Proceso de contaminación con *nst*. Fuente: Elaboración Propia.

4.2.2. Creación del conjunto de datos ECG-MA

Como se dijo anteriormente, para el entrenamiento de la CNN se necesita un conjunto de datos compuesto de muestras de señal de ECG, que serán utilizadas para ajustar los parámetros del modelo. Estas muestras se obtendrán dividiendo cada uno de los registros contaminados en ventanas consecutivas, sin traslape, cuyo largo se considerará como un hiperparámetro. Dichas ventanas corresponden a las muestras del conjunto de datos, al que llamaremos ECG-MA.

Una práctica común en el ámbito del aprendizaje automático es dividir el conjunto de datos en conjuntos de entrenamiento, validación y prueba. El conjunto de entrenamiento se utiliza para ajustar iterativamente los parámetros del modelo. El conjunto de validación se utiliza para evaluar el desempeño del modelo durante el proceso de entrenamiento, y con esto ajustar sus hiperparámetros o tomar decisiones como dejar de entrenar, entre otras. Finalmente, el conjunto de prueba se utiliza para la evaluación final del modelo, ya que son datos no vistos por éste durante el entrenamiento.

La Figura 4.7 muestra un esquema resumen del proceso de creación del conjunto de datos ECG-MA. Es importante que exista una separación entre los datos que se usan para entrenar el modelo, y los datos que se usan para evaluar su desempeño, lo que aplica tanto a las señales contaminadas como a las limpias. Para garantizar aquello, se proponen dos estrategias

- Dividir la lista original de los 48 registros en tres grupos de 38, 5 y 5 registros, correspondientes a entrenamiento, validación y prueba. Esto equivale a 79.2 %, 10.4 % y 10.4 %, respectivamente. Así, se asegura que las muestras limpias usadas en la validación y prueba, obtenidas de las cuatro versiones de cada registro original a diferentes SNR, no hayan sido usadas durante el entrenamiento.
- Las muestras del conjunto de entrenamiento se obtendrán de a partir de la base de datos de registros de ECG contaminados con EM1, y los conjuntos de validación y prueba se obtendrán de la base de datos de registros de ECG contaminados con EM2. De esta forma, tampoco habrá traslape de muestras contaminadas entre los conjuntos de entrenamiento, y validación y prueba.

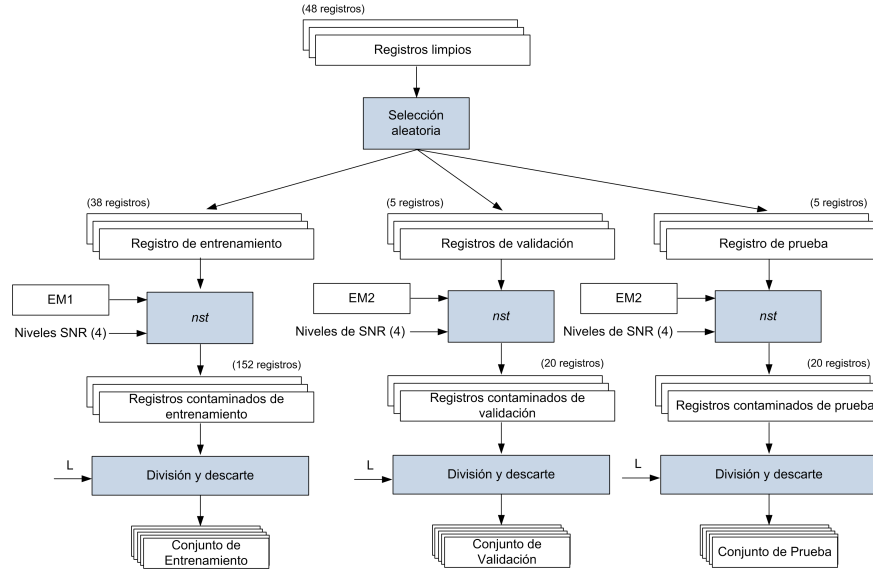


Figura 4.7: Creación del conjunto de datos ECG-MA. El tamaño del conjunto de datos depende del largo de ventana a escoger, denominado L . Fuente: Elaboración Propia.

Un punto importante a tener en cuenta es que, tal como se mencionó, el largo de ventana es un hiperparámetro de la red. Como la cantidad total de registros de la base de datos de ECG contaminado se mantiene constante, la cantidad de muestras del conjunto de datos ECG-MA dependerá del largo de cada ventana, en una relación de proporcionalidad inversa. Sin embargo, a pesar que la cantidad de muestras puede variar, la cantidad total de *latidos* con los que se entrenará cada versión del modelo se mantendrá más o menos constante.

Luego de obtener el conjunto de muestras, se descartarán todas aquellas que presenten contaminación parcial. Éstas provienen de los lugares de transición entre segmentos limpios y contaminados, de acuerdo al protocolo de contaminación de *nst*. Adicionalmente, se eliminarán las ventanas que, conforme al archivo de anotaciones de referencia de cada registro, contengan alguna anotación de ruido. Luego, se contará el número de muestras limpias y contaminadas, y se igualarán ambos grupos descartando aleatoriamente muestras del grupo de mayor tamaño. Esto garantiza un conjunto de datos balanceado.

4.2.3. Base de datos de BCG

A diferencia del ECG, y de acuerdo a la información recopilada en la Revisión Bibliográfica de esta Tesis, no existen bases de datos estándar de BCG. Las investigaciones publicadas usualmente utilizan bases de datos propia, con datos tomados experimentalmente pero no estandarizados por la comunidad científica. Ante esta situación se decidió utilizar un conjunto de registros de BCG capturados por Javier Chávez en [18, 83].

Base de datos de Javier Chávez

Esta base de datos corresponde a 35 registros de señales de BCG, obtenidas mediante un sistema de adquisición basado en sensores de film electro-mecánico (EMFi), de distintas personas en la sala de espera del área de Cardiología del Hospital Regional de Concepción. Cada registro tiene una duración de 2 [min], y consta de una señal de BCG y una de ECG registrada simultáneamente a 200 [Hz], y un vector auxiliar de valores binarios que marca la ocurrencia de artefactos por movimiento. En total, el conjunto de registros contienen aproximadamente 4490 latidos.

El pequeño tamaño de esta base de datos es un problema al momento de querer entrenar una red neuronal profunda, pues probablemente su capacidad de generalizar será muy pobre. Como no existe otra base de datos gratuita o de fácil acceso, se propone crear una base de datos de BCG semi-sintético.

Base de datos de BCG semi-sintético

De los 35 registros anteriores, se eligieron 27 y se descartaron los demás por presentar señales ruidosas o gran cantidad de latidos de morfología extraña. Para cada uno de los registros seleccionados, se aplicó el procedimiento descrito a continuación:

1. Detección de picos J

Los picos J se detectaron tomando como referencia los picos R (más fáciles de detectar) del ECG correspondiente a cada registro.

2. Obtención de valores estadísticos

Se calculó la media y la desviación estándar de las amplitudes de los picos J.

3. División del registro y creación de colección de latidos

Puesto que las ondas F y N que marcan el inicio y el final del latido en el BCG son difícil de detectar mediante algoritmos computacionales, se propuso dividir la señal en segmentos J-J.

Adicionalmente, y basándose en el vector auxiliar, se descartaron todos los segmentos que presentaran artefactos por movimiento.

4. Procesamiento de latidos y normalización

Se aplicó a cada latido una ganancia variable (en forma de rampa) para nivelar el valor inicial y final de cada segmento.

Luego, se dividió cada segmento por su valor máximo, y se multiplicó por la media de las amplitudes de los picos J del registro completo.

5. Concatenación de segmentos

Los latidos se concatenaron en una secuencia aleatoria, para formar un registro sintético de 30 [min] de duración.

6. Post-procesamiento del registro semi-sintético

Se aplicó un factor de ganancia aleatoria a cada latido, para simular la variabilidad presente en las señales de BCG reales.

La Figura 4.8(a) muestra gráficamente el proceso descrito, mientras que en la Figura 4.8(b) se puede apreciar un extracto de del registro de BCG sintetizado a partir del registro 24 de la base de datos de Chávez.

Contaminación con ruido y creación del conjunto de datos BCG-MA

En los registros originales, el total de segmentos de MAs suma aproximadamente 4 [min], lo cual es extremadamente poco. Por lo anterior, se propuso contaminar los

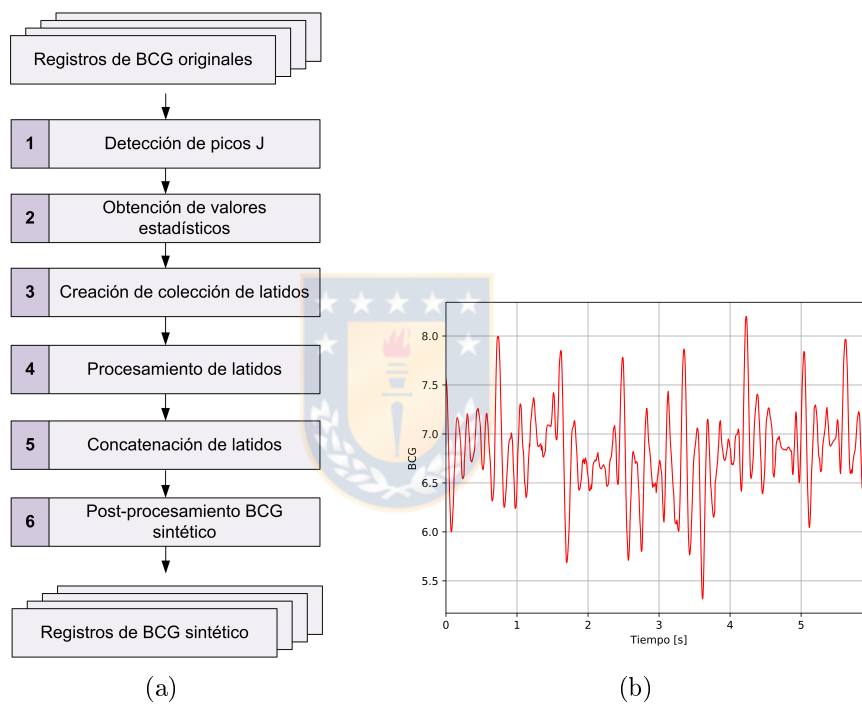


Figura 4.8: Creación de BCG semi-sintético. Para ello, se utilizan los registros de la base de datos de [18]. (a) Proceso de creación de BCG semi-sintético. Fuente: Elaboración Propia. (b) Extracto de registro de BCG sintético. Fuente: Elaboración Propia.

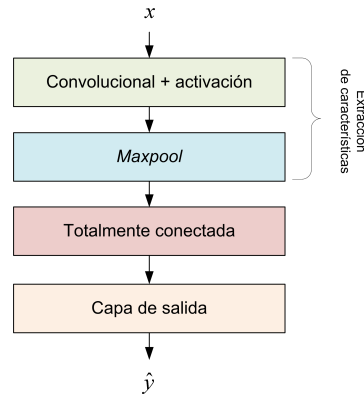


Figura 4.9: Arquitectura de una CNN básica. Fuente: Elaboración Propia.

registros de BCG semi-sintéticos con los registros EM1 y EM2 de la base de datos de ECG. El proceso de contaminación es idéntico al explicado la Sección 4.2.1. Así, se obtuvieron dos conjuntos de 162 registros cada uno, contaminados con EM1 y EM2 respectivamente, y que serán considerados como la base de datos BCG. Por su parte, el conjunto de datos BCG-MA también será creada a partir de la metodología utilizada para el ECG.

4.3. Diseño de la red convolucional

El diseño de la CNN se realizará a partir de una topología básica, mostrada en la Figura 4.9. Para conseguir un modelo óptimo en términos de desempeño, se definirán conjuntos de valores discretos para cada hiperparámetro, que originarán distintos posibles modelos (Tabla 4.2). Luego, cada modelo se entrenará con los conjuntos de datos ECG-MA y BCG-MA y, adicionalmente, para disminuir el tiempo de entrenamiento y evitar el sobre-ajuste, se utilizará un criterio de parada temprana o *early-stopping*. Éste consiste en calcular la Exactitud sobre el conjunto de validación (Eq. 4.4), almacenar los pesos del modelo luego de cada época, y detener el entrenamiento si la Exactitud no ha mejorado luego de un número definido de épocas de *paciencia*.

De acuerdo a la literatura, la mejor forma de llevar a cabo la exploración de

hiperparámetros es mediante una búsqueda de grilla. Esta grilla debe pensarse en realidad como una hipergrilla, de dimensión igual al número de hiperparámetros, donde cada punto corresponde a una combinación única. Luego, el método consiste en entrenar el modelo con todas estas combinaciones, y posteriormente escoger aquella con la que se obtenga el mejor desempeño.

Tabla 4.2: Valores a explorar de los hiperparámetros del modelo. Fuente: Elaboración Propia.

Hiperparámetro	Valores posibles
Tamaño vector de entrada	0.5, 1, 2, 4, 6, 8 [s]
Función de activación	Linear, ReLU, Leaky ReLU
Número de capas FE	1, 2, 3, 4, 5, 6, 7, 8, 9
Tamaño capa totalmente conectada	32, 64, 128, 256, 512, 1024
Campo receptivo del <i>kernel</i>	3, 5, 7, 9
Número de filtros	16, 32, 64, 128
<i>Stride</i> de la capa <i>maxpool</i>	2
Tamaño de lote	1024
Número de épocas	1000
α	10^{-3}
β_1	0.9
β_2	0.999
Regularización en capa totalmente conectada	<i>Dropout</i> , $P=0.5$
Paciencia	15 épocas

Si bien esta estrategia garantiza encontrar un óptimo global entre todos los valores posibles, en nuestro caso implica entrenar la red 16200 veces. Un análisis preliminar arroja que, con el sistema de hardware a utilizar, el tiempo de entrenamiento promedio de cada iteración es de 1.2 [h] aproximadamente. De acuerdo a esto, el tiempo total del proceso de exploración sería 810 [días], lo cual evidentemente es poco factible dentro del contexto de esta investigación.

Se propone entonces realizar una *búsqueda iterativa*, partiendo de un punto “medio” de la hipergrilla, y explorando los posibles valores de un hiperparámetro mientras

se mantiene el resto constante. Esto permite encontrar el óptimo en ese eje. Luego, dicho óptimo local es el punto de partida en la iteración siguiente, donde se exploran los valores posibles de un nuevo hiperparámetro (Figura 4.10). Una implementación en pseudo-código basado en Python se presenta a continuación.

```

1  # Crear base de datos
2  train_set, validation_set, test_set = create_database(...)
3
4  # Definir listas de valores para cada hiperparametro
5  hyperparam1 = [value1, value2, ..., valueN]
6  ...
7  hyperparamH = [value1, value2, ..., valueN]
8  # Lista de listas
9  hyperparams = [hyperparam1, ..., hyperparamH]
10
11 # Escoger punto de partida arbitrario
12 # En este caso, punto medio de la hipergrilla
13 start_point = [hyperparam1[n/2], ..., hyperparamH[n/2]]
14 current_point = start_point
15
16 # Iterar sobre cada hiperparametro
17 for i, hp_list in enumerate(hyperparams):
18     # Iterar sobre valores posibles del i-esimo hiperparametro
19     for j, hp in enumerate(hp_list):
20         current_point[i] = hp
21         # Crear modelo a partir de la combinacion actual
22         # de hiperparametros
23         model = create_model(current_point)
24         model.train(train_set.x, train_set.y)
25         y_hat_validation = model.predict(validation_set.x)
26         Accuracy_validation[j] = accuracy_score(y_hat_validation,
27             validation_set.y)
28
29     # Escoger el valor que dio la mejor Exactitud
30     # sobre el set de validacion
31     aux_j = argmax(Accuracy_validation)
32     current_point[i] = hp_list[aux_j]
33
34 end_point = current_point

```

Aunque este método no asegura encontrar el óptimo global, tomará menos tiempo en total, pues son sólo 32 iteraciones (38 [h]). Esto nos permitirá tener de forma rápida una idea general de cuán buena es la solución propuesta.

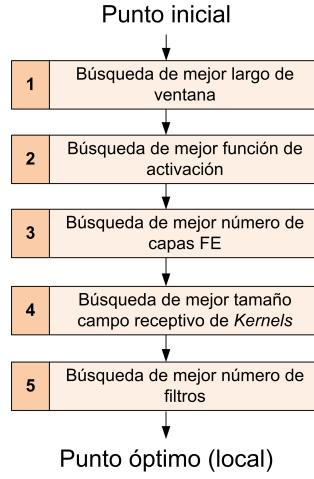


Figura 4.10: Método propuesto para exploración de hiperparámetros. Búsqueda iterativa, donde en cada paso se varía uno de los hiperparámetros y se mantiene el resto constante. Luego, se elige el valor del hiperparámetro que proporciona la mejor Exactitud de validación, y se avanza al paso siguiente. Fuente: Elaboración Propia.

4.4. Evaluación de desempeño de la CNN

4.4.1. Métricas de desempeño

El desempeño de la red neuronal se medirá en términos de Exactitud (Eq. 4.4), Precisión (Eq. 4.5), Exhaustividad (Eq. 4.6) y Valor-F1 (Eq. 4.7), que corresponden a algunas de las métricas más usadas en ML:

$$Ac = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.4)$$

$$P = \frac{TP}{TP + FP} \quad (4.5)$$

$$R = \frac{TP}{TP + FN} \quad (4.6)$$

$$F1 = 2 \frac{P \cdot R}{P + R} \quad (4.7)$$

Donde:

TP: verdaderos positivos, muestras de la clase positiva clasificadas correctamente.

TN: verdaderos negativos, muestras de la clase negativa clasificadas correctamente.

FP: falsos positivos, muestras de la clase negativa clasificadas incorrectamente como clase positiva.

FN: falsos negativos, muestras de la clase positiva clasificadas incorrectamente como clase negativa.



4.4.2. Validación del modelo

Una vez definida la arquitectura final del modelo, se aplicará el método de Validación Cruzada K-fold. Éste consiste en dividir la base de datos en K grupos, para luego utilizar uno de ellos como conjunto de prueba, otro como conjunto de validación, y el resto como conjunto de entrenamiento. Luego, se entrena el modelo y se evalúa su desempeño mediante las métricas elegidas, a partir de los resultados del set de prueba. El procedimiento se repite K veces, de forma que cada grupo sea conjunto de prueba una vez. El desempeño final del modelo se calcula promediando los K valores de cada una de las métricas. El diagrama de la Figura 4.11 muestra de forma gráfica el método de Validación Cruzada K-fold.

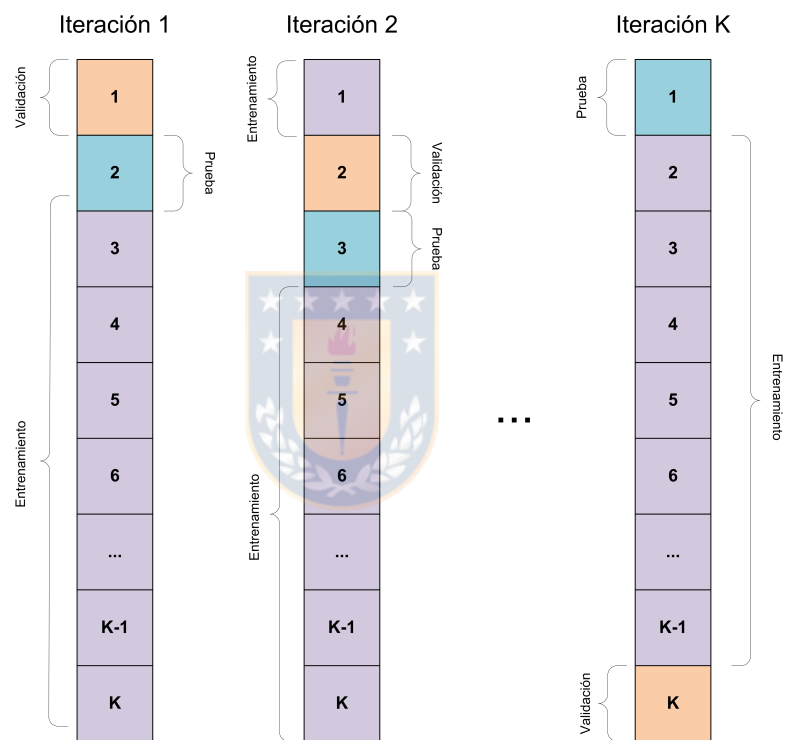


Figura 4.11: Método de validación K-fold. Cada cuadrado representa uno de los K grupos. Fuente: Elaboración Propia.

4.5. Evaluación de impacto en la confiabilidad de métricas cardíacas

El clasificador diseñado se integrará en el sistema de detección de latidos y cálculo de período cardíaco instantáneo mostrado en la Figura 4.12. Este sistema recibe uno de los registros de la base de datos de ECG contaminado, y su vector de anotaciones de referencia correspondiente. El registro se divide en segmentos de largo igual al tamaño óptimo encontrado durante la etapa de diseño, para posteriormente ser rotulados como segmentos limpios o contaminados por la CNN. Luego, un registro filtrado (es decir, sin artefactos) se obtendrá a partir de la concatenación de todos los segmentos limpios, en orden temporal. En este paso, también se generará un nuevo vector de anotaciones de referencia.

Se aplicará el algoritmo de detección de latidos SQRS en las señales de ECG, y el algoritmo de detección de picos J presentado en [18] a las señales de BCG, tanto al registro original como al registro filtrado, y se compararán latido a latido las anotaciones generadas contra las anotaciones de referencia. A partir de lo anterior, se calculará la Sensibilidad (Eq. 4.8) y el Valor Predictivo Positivo (Eq. 4.9), dos métricas estándar para la evaluación de algoritmos de detección de latidos.

$$Se = \frac{TP}{TP + FN} \quad (4.8)$$

$$PPV = \frac{TP}{TP + FP} \quad (4.9)$$

Finalmente, se calculará el Período Cardíaco Instantáneo (*Heart Rate*, HR) como la diferencia temporal entre dos latidos consecutivos. Esta métrica cardíaca es comúnmente usada para predecir la mortalidad luego de un infarto al miocardio, y para el diagnóstico de otras enfermedades. En general, una alta variabilidad es signo de anomalías.

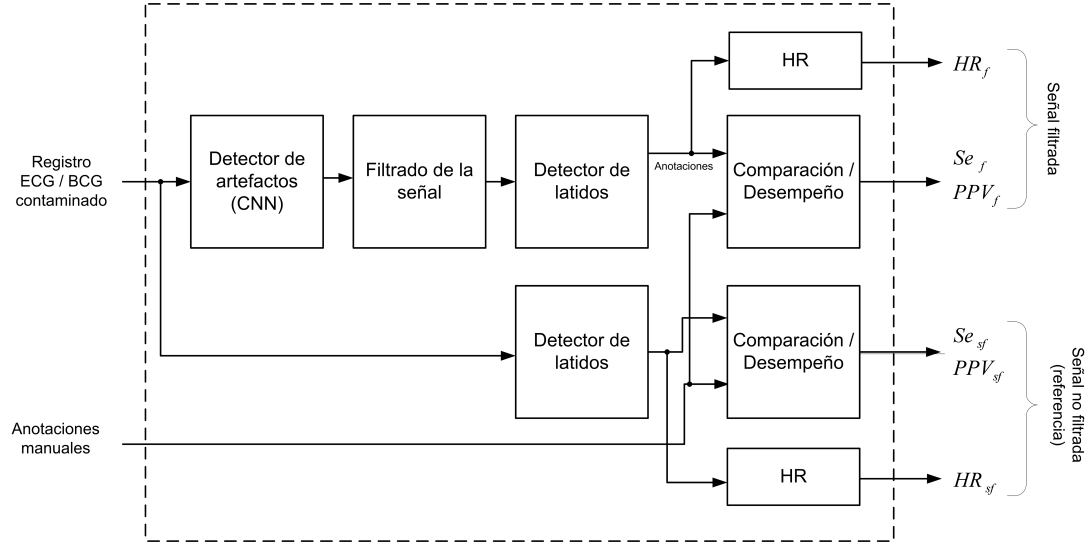


Figura 4.12: Sistema de detección en línea de latidos y cálculo período cardíaco.
Fuente: Elaboración Propia.

4.6. Evaluación de implementación en dispositivo portable

Por último, se evaluará la factibilidad de implementación del clasificador en un dispositivo portable, para su eventual uso en detección de artefactos en señales de BCG en tiempo real. De esta forma, se implementará la CNN en un sistema basado en Raspberry Pi 3 (Figura 4.13), cargando los parámetros resultantes del proceso de entrenamiento, y se medirá el tiempo de predicción de un segmento de señal. Este sistema de hardware es el mismo utilizado en [83] para la obtención de BCG con sensores EMFi, con el objetivo de ser usado en aplicaciones de *home-care* y de medición en terreno (salas de espera, ambulancias, entre otras). Las características de dicho sistema se resumen en la Tabla 4.3).

Es importante señalar que no se entrenará la red en este dispositivo, pues dicho proceso es costoso computacionalmente y los recursos disponibles son muy limitados. Esta práctica se conoce como entrenamiento *offline* y es común al momento de implementar redes neuronales en sistemas de hardware dedicados.

Tabla 4.3: Características del sistema portable. Fuente: Elaboración Propia, basado en [1].

Procesador	ARM Cortex-A53
Frecuencia de reloj	1.2 [GHz]
Núcleos	4
Memoria RAM	1 [GB]
Tarjeta de video dedicada	No
Sistema operativo	Raspbian
Frecuencia de muestreo ADC	100 [S/s]
Resolución del ADC	12 [bits]

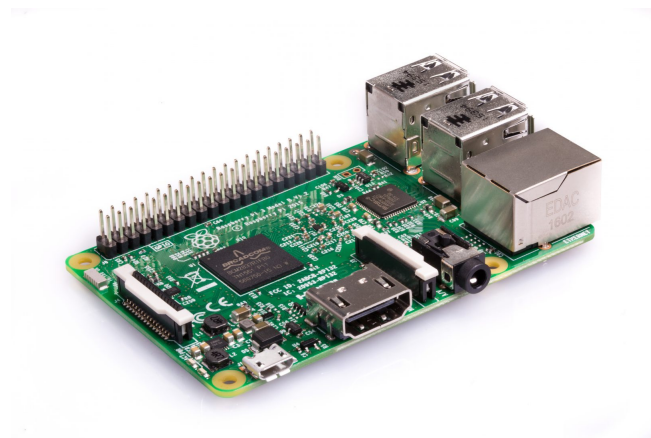


Figura 4.13: Raspberry Pi 3, modelo B. Fuente: Página web de Raspberry Pi [1].

4.7. Discusión

Una metodología para la creación de los conjuntos de datos ECG-MA y BCG-MA, y para el diseño de la CNN fueron presentados. La escasez de datos de BCG, que lleva inevitablemente al sobre-ajuste del modelo, fue resuelta mediante la creación de señales semi-sintéticas. El método de síntesis ideado es básico, y se fundamenta en crear secuencias aleatorias de latidos ya existentes. En otras palabras, no se crean nuevos latidos, a pesar de incluir ciertas estrategias para agregar variabilidad a los mismos. Por lo mismo, este método tiene sentido en la medida de que el largo de ventana del conjunto de datos BCG-MA, aún no definido, sea mayor la duración de un latido. De otra forma, se terminaría entrenando el modelo con información cuasi redundante.

Por su parte, la escasez de muestras de artefactos por movimiento de BCG reales también se presentó como un problema grave en el desarrollo de esta investigación. La alternativa planteada consistió en contaminar los registros de BCG con artefactos por movimiento de los registros de ECG, lo cual se basa en la hipótesis de un supuesto parecido entre los artefactos de ambas señales. La justificación de lo anterior es que, a simple vista, los artefactos de ECG y BCG son similares en morfología. Además, se sabe que en ambos casos los artefactos aparecen por movimientos bruscos o golpes en los transductores, aunque de acuerdo a la literatura revisada no se tiene completa certeza de los mecanismos físicos que los originan.

Sin embargo, en el futuro y a partir de lo anterior, podría plantarse un problema de *aprendizaje por transferencia*, donde el proceso de entrenamiento antes descrito nos entregaría un modelo *pre-entrenado*, cuyas capacidades podrían ser mejoradas mediante un segundo entrenamiento con una base de datos de BCG real más extensa. Este tipo de prácticas son relativamente comunes en el área del aprendizaje automático. Sólo por mostrar un ejemplo acorde a nuestra área de estudio, el trabajo presentado en [84] propone la utilización de la famosa red AlexNet, pre-entrenada para clasificación de imágenes, en una aplicación de detección de arritmias en señales de ECG.

Por último, y tal como se discutió en la Sección 4.3, el método propuesto para

encontrar la combinación óptima de hiperparámetros, sin ser el mejor, es el más factible de implementar de acuerdo a las restricciones impuestas por el sistema de hardware a usar. Cabe recordar que el propósito de esta investigación no es encontrar un algoritmo de clasificación óptimo, sino una solución de ingeniería que mejore la confiabilidad de las métricas cardíacas.



Capítulo 5. Resultados

Los resultados obtenidos de aplicar la metodología propuesta se detallan en este capítulo. Éste se dividirá en tres grandes secciones. Primero, se expondrán los resultados del modelo entrenado con señales de ECG, mostrando la arquitectura final de la CNN encontrada mediante el proceso de búsqueda iterativa de hiperparámetros, para luego evaluar el desempeño de clasificación del modelo y el impacto de incluir el clasificador en un sistema de detección de latidos. Posteriormente, se realizará el mismo análisis para el modelo entrenado con señales de BCG, incluyendo un breve estudio de factibilidad técnica para la implementación del modelo en un sistema de medición portable. Finalmente, se elabora una breve discusión de lo aquí expuesto.

5.1. Detector de artefactos en ECG

5.1.1. Exploración de hiperparámetros

Aplicando la metodología presentada en la Sección 4.3 sobre el conjunto de hiperparámetros del modelo, se obtuvieron los resultados resumidos en la Tabla 5.1. El criterio para seleccionar el valor óptimo del hiperparámetro en cada iteración de la búsqueda, como se explicó antes, se basó en la Exactitud calculada sobre el conjunto de validación. Sin embargo, en algunos casos, elegir el valor óptimo implicaba una pequeña mejora en la Exactitud, pero un gran impacto en la cantidad de parámetros entrenables (pesos de las neuronas) del modelo. En aquella situación, se eligieron los dos valores con mejor Exactitud, y entre ellos se escogió el que generara un modelo de menor tamaño.

La arquitectura final del modelo puede verse de forma gráfica en la Figura 5.1.1. Se aprecia que el tamaño de la red asciende a 130000 parámetros entrenables, lo cual no es tan grande si se compara con AlexNet (60 millones de parámetros) u otros modelos famosos en el área del *Deep Learning*. Almacenados en un archivo HDF5, éste sólo pesa 1.6 [MB].

Tabla 5.1: Valores iniciales y finales de los hiperparámetros mediante búsqueda iterativa para ECG. La columna “Valor inicial” indica el punto inicial de la búsqueda, escogida arbitrariamente. La columna “Valor final” denota el punto encontrado tras la exploración. Por su parte, Ac_{val} simboliza la Exactitud de validación, y Ac_{train} , la Exactitud de entrenamiento. Fuente. Elaboración Propia.

Hiperparámetro	Valor inicial	Valor final
Tamaño vector de entrada	2 [s]	1 [s]
Función de activación	ReLU	Leaky ReLU
Número de capas FE	4	6
Tamaño capa t. conectada	256	512
Campo receptivo del <i>kernel</i>	5	9
Número de filtros	32	32
Parámetros entrenables	106273	129505
Ac_{train}	0.97	0.99
Ac_{val}	0.951	0.979

Por su parte, las curvas de Costo y Exactitud de entrenamiento y validación del modelo final se aprecian en la Figuras 5.2(a) y 5.2(b). El entrenamiento se llevó a cabo en al rededor de 28 épocas, en un tiempo promedio de 2 [h]. Por su parte, el tiempo promedio de predicción de una muestra corresponde a 0.5 [ms]. Cabe recordar, sin embargo, que estos tiempos dependen de las características particulares del sistema computacional usado, por lo que son sólo referenciales.

A modo de ejemplo, la Figura 5.3 muestra gráficamente la detección de MAs en el registro 213 contaminado a $SNR = 0$ [dB]. Se aprecia que existe una gran cantidad de falsos positivos (segmentos limpios detectados como artefactos), no así falsos negativos, lo cual mantiene las predicciones en el lado seguro.

5.1.2. Validación del modelo

El modelo diseñado fue posteriormente evaluado mediante Validación Cruzada K-fold, tal como se describió en la Sección 4.4.2. En cada iteración (*Fold*) se midió el desempeño de la CNN, calculando Ac (Eq. 4.4), P (Eq. 4.5), R (Eq. 4.6) y $F-1$ (Eq.

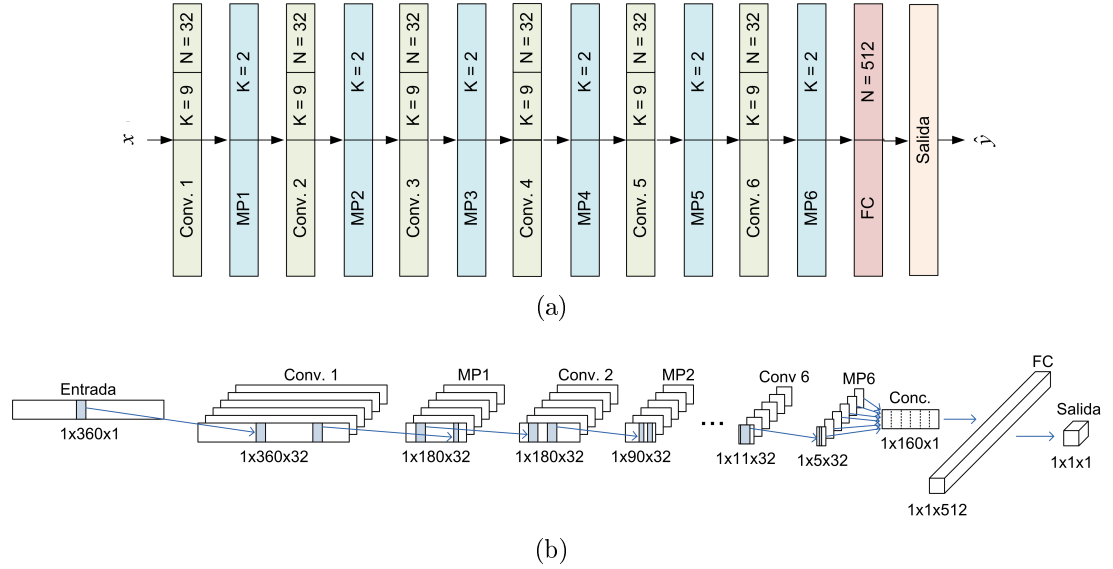
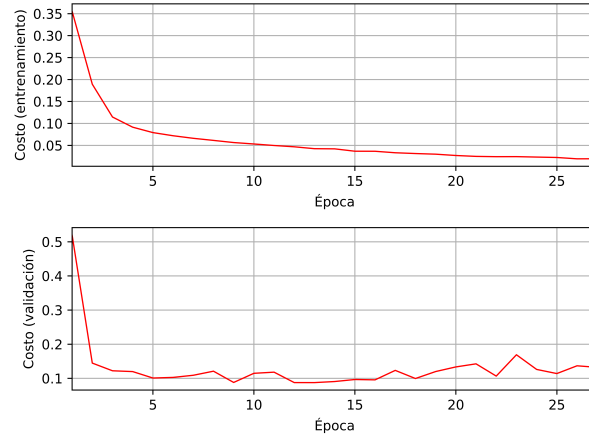


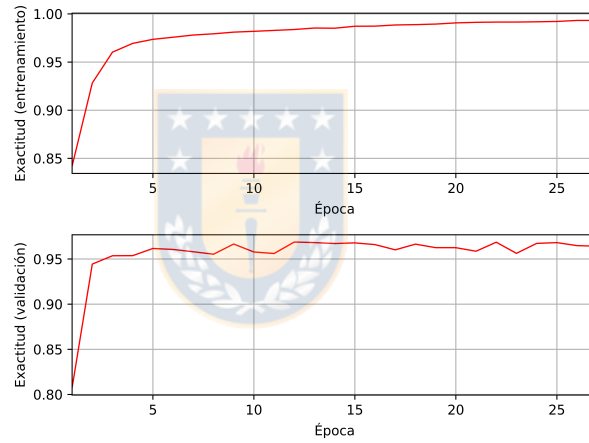
Figura 5.1: Arquitectura final de CNN para detección de artefactos en ECG. (a) Arquitectura final. Fuente: Elaboración Propia. (b) Flujo de datos. *Conv.* es la abreviación de la capa convolucional, MP representa la capa de *Max pooling*, *Conc.* representa la operación de concatenación de todos los vectores de características de la penúltima capa, y FC denota la capa totalmente conectada. Fuente: Elaboración Propia.

4.7). Los resultados de cada *Fold*, así como los valores máximo, mínimo y medio, se muestran en la Tabla 5.2.

Para tener una noción de la calidad de los resultados obtenidos, suele hacerse una comparación contra un estándar que se considera una línea base o *baseline*. Un clasificador típico para estos fines es el regresor logístico o *logit*, que corresponde a un modelo estadístico básico y ampliamente utilizado. El desempeño promedio de este clasificador luego de ajustar sus parámetros con los mismos datos, es cercano al 52 %. Luego, y en comparación con lo anterior, podemos considerar que el 95 % obtenido con la CNN es bastante bueno. Es posible notar también que la *P* promedio del clasificador es menor que la *Ac* promedio, de lo cual se infiere que el comportamiento visto en la Figura 5.3 es común en los demás registros también. Resulta interesante, además, conocer el desempeño del clasificador a distintos niveles de SNR (Tabla 5.3). Como es de esperar, se aprecia que entre mayor es el nivel de contaminación, el desempeño del clasificador es mejor. Esto se justifica porque las muestras más



(a)



(b)

Figura 5.2: Costo y Ac_{val} durante el entrenamiento del modelo para ECG (Fold 1). (a) Evolución del Costo durante el entrenamiento (Fold 1). Fuente: Elaboración Propia. (b) Evolución de Ac_{val} durante el entrenamiento (Fold 1). Fuente: Elaboración Propia.

contaminadas (SNR menor) presentan una morfología más claramente distinguible.

5.1.3. Evaluación de confiabilidad de métricas cardíacas

Utilizando los parámetros (valores de los pesos de la red) obtenidos en cada Fold, se procedió a integrar el modelo en el sistema de detección de latidos (Sección

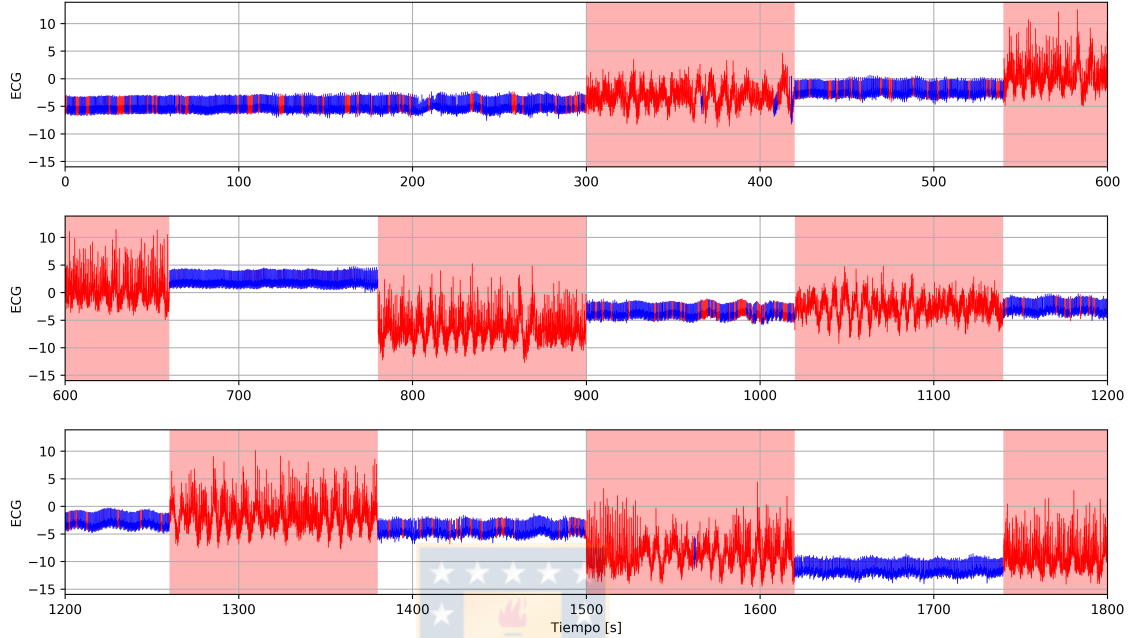


Figura 5.3: Detección de artefactos en el registro de ECG 213. Representación gráfica de la detección de artefactos en el registro de ECG 213 contaminado a 0 [dB] de SNR. Los segmentos de ECG detectados como artefactos se grafican en rojo, mientras que los segmentos limpios se muestran en azul. Por su parte, las zonas efectivamente contaminadas con *nst* han sido destacadas con fondo rojo transparente. Fuente: Elaboración Propia.

4.5). Un resumen de los resultados, agrupados por nivel de SNR, se pueden ver en la Tabla 5.4, y de forma gráfica en la Figuras 5.4(a) y 5.4(b). Si comparamos Se_{sf} contra S_f , podemos apreciar que existe una mejora de 12 %, 23 % y 34 % para registros contaminados a 6 [dB], 0 [dB] y -6 [dB], respectivamente. De las figuras, se aprecia que tanto la Se como PPV calculados sobre los registros filtrados, no se mantienen constantes ante todos los valores de SNR, bajando levemente en el rango [24, 12] [dB]. Esto es esperable de acuerdo a los resultado de la Tabla 5.3, pues la tarea de clasificar señales con poco ruido es más difícil que cuando los artefactos son evidentes, como en el caso de SNR = -6 [dB]. De todas formas, la variación de la Se calculada sobre los registros filtrados, a distintos niveles de SNR, es sólo de un 1 %,

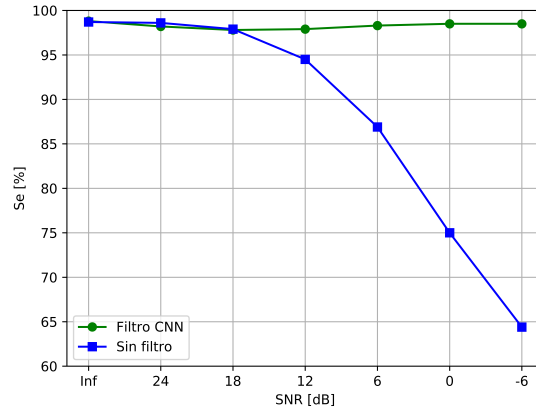
Tabla 5.2: Resultados de Validación Cruzada K-fold en ECG. Métricas calculadas con respecto al conjunto de prueba. Fuente: Elaboración Propia.

Fold	Ac	P	R	$F1$
1	0.958	0.957	0.960	0.959
2	0.972	0.964	0.981	0.973
3	0.974	0.983	0.966	0.974
4	0.944	0.945	0.942	0.944
5	0.944	0.917	0.976	0.946
6	0.939	0.903	0.984	0.942
7	0.982	0.992	0.971	0.982
8	0.964	0.971	0.957	0.964
9	0.956	0.942	0.971	0.956
Promedio	0.959	0.952	0.968	0.960
Máximo	0.982	0.992	0.984	0.982
Mínimo	0.939	0.903	0.942	0.942
Logit (promedio)	0.522	0.533	0.371	0.437

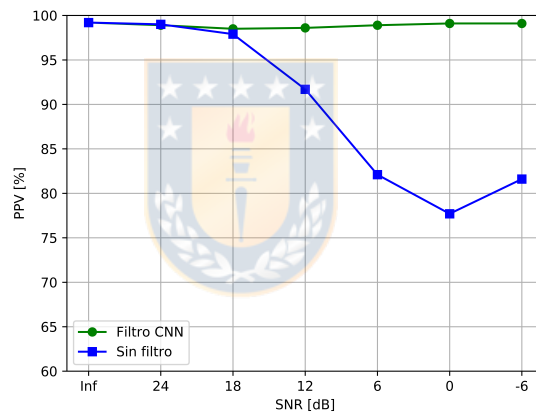
Tabla 5.3: Desempeño del clasificador de ECG a distintos niveles de SNR. Fuente: Elaboración Propia.

SNR [dB]	Ac	P	R	$F1$
12	0.916	0.935	0.896	0.914
6	0.963	0.948	0.981	0.964
0	0.976	0.958	0.995	0.976
-6	0.982	0.969	0.998	0.983

aproximadamente.



(a)



(b)

Figura 5.4: Desempeño del detector de latidos de ECG en función del SNR. Comparación de las versiones original y filtrada. (a) Sensibilidad del algoritmo en función del SNR. Fuente: Elaboración Propia. (b) Valor predictivo positivo del algoritmo en función del SNR. Fuente: Elaboración Propia.

5.2. Detector de artefactos en BCG

5.2.1. Exploración de hiperparámetros

Se aplicó la misma metodología que el caso de ECG, consiguiéndose los resultados resumidos de la Tabla 5.5. En general, éstos distan bastante de los obtenidos para las

Tabla 5.4: Impacto del clasificador en el desempeño del algoritmo de detección de latidos en ECG. Comparación del desempeño del algoritmo de detección de latidos SQRS ante registros contaminados a distintos niveles de SNR. Los sub-índices sf y f denotan las métricas calculadas sobre los registros sin filtrar y filtrados, respectivamente, y FSR corresponde al porcentaje de la señal que fue eliminada por el filtro. Fuente: Elaboración Propia.

SNR	Se_{sf}	PPV_{sf}	Se_f	PPV_f	$FSR[\%]$
Inf.	0.987	0.992	0.988	0.992	0.191
24	0.986	0.99	0.982	0.989	14.790
18	0.979	0.979	0.978	0.985	30.134
12	0.945	0.917	0.979	0.986	42.617
6	0.869	0.821	0.983	0.989	45.889
0	0.75	0.777	0.985	0.991	45.826
-6	0.644	0.816	0.985	0.991	45.253

señales de ECG, tanto en los valores de los hiperparámetros como el desempeño del clasificador sobre el conjunto de validación. Se optó entonces por tomar como punto inicial el conjunto de hiperparámetros obtenidos del proceso de exploración en ECG (Valor final de la Tabla 5.1), y realizar una búsqueda iterativa a partir de ahí.

Tabla 5.5: Valores iniciales y finales de los hiperparámetros obtenidos mediante primera búsqueda iterativa en BCG. La columna “Valor inicial” indica el punto inicial de la búsqueda, escogida arbitrariamente. La columna “Valor final” denota el punto encontrado tras la exploración. Por su parte, Ac_{val} simboliza la Exactitud de validación. Fuente: Elaboración Propia.

Hiperparámetro	Valor inicial	Valor final
Tamaño vector de entrada	2 [s]	4 [s]
Función de activación	ReLU	ReLU
Número de capas FE	4	9
Tamaño capa t. conectada	256	256
Campo receptivo del <i>kernel</i>	5	5
Número de filtros	32	32
Ac_{val}	0.850	0.891

Como se explicó en la Sección 3.3.3, las capas de extracción características (con-

volucional + *max-pooling*) generan una representación de los datos de entrada, que son clasificados en distintas clases por la capa totalmente conectada. En la arquitectura planteada, puesto que la cantidad de filtros en cada capa convolucional es la misma, el tamaño del vector de características final se reduce aproximadamente a la mitad por cada capa de FE adicional, debido al *downsampling* llevado a cabo por las capas de *pooling* ($Stride = 2$).

Por su parte, el desempeño de clasificación de la capa totalmente conectada, que puede pensarse como una red neuronal clásica, depende del tamaño de los datos de entrada a esa capa (tamaño y cantidad de vectores de características finales) y de la cantidad de parámetros entrenables de la misma (cantidad de neuronas). Por decirlo de una forma simple, entre más grande es cada vector de entrada, más neuronas se necesitan para aprender de la información disponible.

Se propuso entonces hacer una segunda exploración, dejando todos los hiperparámetros fijos (incluido el tamaño de la capa totalmente conectada), y variar el tamaño del vector de entrada (largo de cada muestra del conjunto de datos BCG-MA) y el número de capas FE sincrónicamente, de forma de obtener un tamaño de vector de características cercano al del clasificador de ECG (160 *samples*), y más o menos igual en todas las combinaciones. Los resultados de esta nueva exploración se muestran en la Tabla 5.6.

Tabla 5.6: Valores encontrados de los hiperparámetros para clasificador de BCG en segunda exploración. Aquí, $\#FE$ es el número de capas de extracción de características, N_{FV} es el tamaño en *samples* del vector de características final, Ac_{val} es la Exactitud sobre el conjunto de validación, y $\#TrP$ es el número de parámetros entrenables. Fuente: Elaboración Propia.

Tamaño de x [s]	$\#FE$	N_{FV}	Ac_{val}	$\#TrP$
0.5	4	192	0.808	127393
1	5	192	0.817	136641
2	6	192	0.873	145889
4	7	192	0.908	155137
6	8	128	0.928	131617
8	8	192	0.928	164385

Se observa que la mejor Ac se obtiene con un tamaño de ventana de 6 [s] y 8 capas

FE. Con ventanas de 8 [s], el Ac es el mismo, pero con un impacto en la cantidad de parámetros entrenables es de 25 %. Por lo anterior, se eligió la combinación de tamaño de ventana 6 [s] y 8 capas FE.

Tabla 5.7: Valores iniciales y finales de los hiperparámetros obtenidos mediante búsqueda iterativa en BCG. La columna “Valor inicial” indica el punto inicial de la búsqueda, que coincide con el punto final encontrado en la búsqueda en ECG. La columna “Valor final” denota el punto encontrado tras la segunda exploración. Por su parte, Ac_{val} simboliza la Exactitud de validación, y Ac_{train} , la Exactitud de entrenamiento. Fuente: Elaboración Propia.

Hiperparámetro	Valor inicial	Valor final
Tamaño vector de entrada	1 [s]	6 [s]
Función de activación	Leaky Relu	Leaky ReLU
Número de capas FE	6	8
Tamaño capa t. conectada	512	512
Campo receptivo del <i>kernel</i>	9	9
Número de filtros	32	32
Ac_{train}	0.941	0.994
Ac_{val}	0.817	0.928

La arquitectura final se resume en la Tabla 5.7, y puede verse de forma gráfica en la Figura 5.5(a). Por su parte, las curvas de Costo y Ac de entrenamiento y validación se aprecian en la Figuras 5.6(a) y 5.6(b).

La Figura 5.7 muestra gráficamente la detección de MAs en el registro sintético 21, contaminado a $SNR = 0$ [dB]. Aquí, y a diferencia que en clasificador para ECG, se aprecian pocos falsos positivos. Sin embargo, hay una porción significativa de la última zona contaminada (destacada con fondo rojo) que no fue detectada como artefactos, originando falsos negativos. Este comportamiento no parece ser la norma, pues P promedio sigue siendo mayor que R (Tabla 5.8).

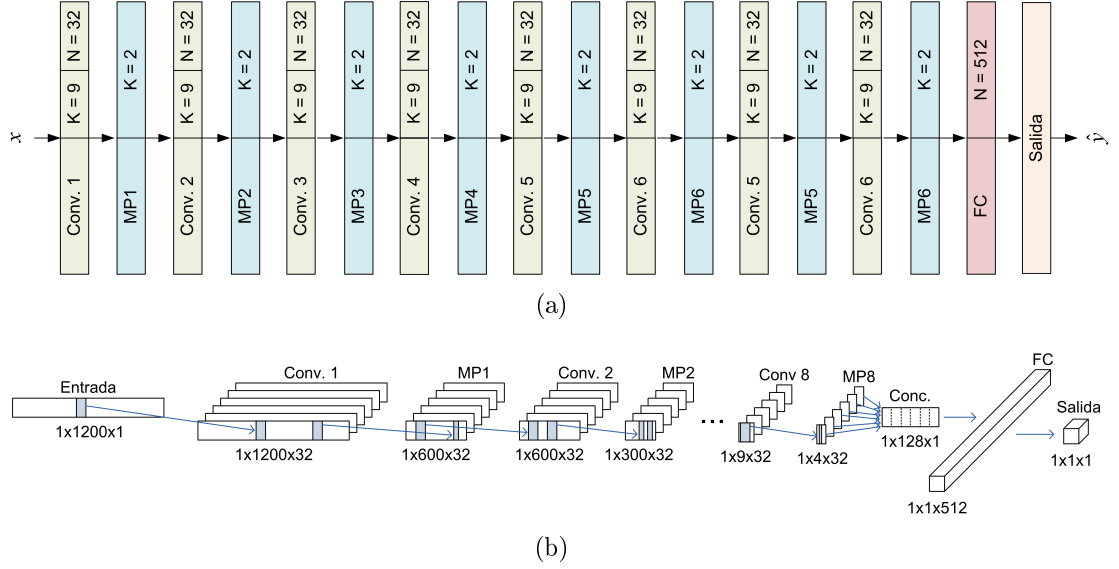
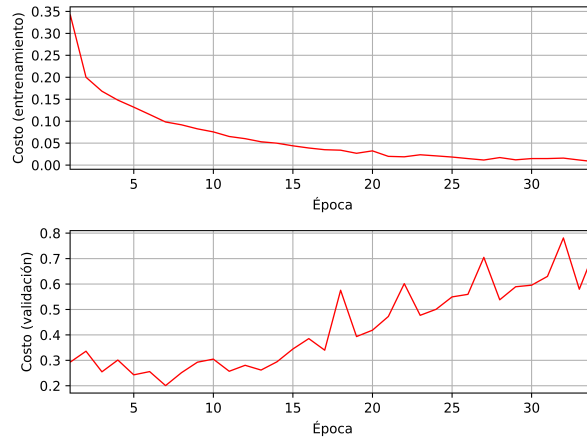


Figura 5.5: Arquitectura final de CNN para detección de artefactos en BCG. (a) Arquitectura final. Fuente: Elaboración Propia. (b) Flujo de datos. *Conv.* es la abreviación de la capa convolucional, *MP* representa la capa de *Max pooling*, *Conc.* representa la operación de concatenación de todos los vectores de características de la penúltima capa, y *FC* denota la capa totalmente conectada. Fuente: Elaboración Propia.

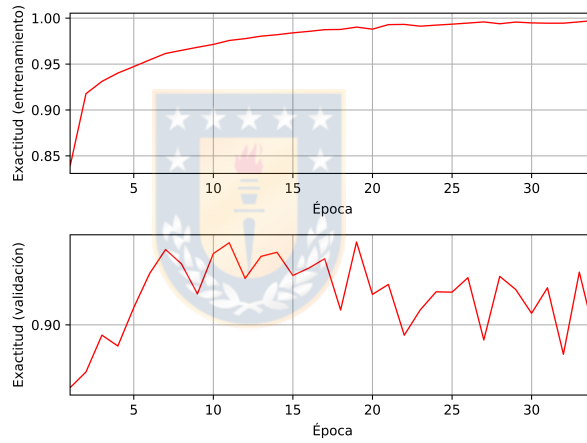
5.2.2. Validación del modelo

Utilizando Validación Cruzada K-fold, se evaluó el desempeño del modelo. En pos de obtener mejores resultados, se decidió generar la Base de Datos de Entrenamiento mediante la división de los registros, pero utilizando traslape de 2 [s]. Los resultados de cada *fold*, así como los valores máximo, mínimo y promedio, se resumen en la Tabla 5.8. En este caso, el tiempo promedio de entrenamiento corresponde a 3.5 [h], y el tiempo de predicción de una muestra a 1.1 [ms].

Los valores promedio de estas métricas, cercanos al 91%, son menores que los del clasificador entrenado con señales de ECG. Sin embargo, y como se mencionó antes, el detector tiene un comportamiento conservador, pues pareciera que hubiese una tendencia hacia los falsos positivos de acuerdo a valores promedio de *P* y *R*. El comportamiento del clasificador ante distintos niveles de SNR es similar al del clasificador de ECG, notándose un aumento en los valores de las métricas de desempeño



(a)



(b)

Figura 5.6: Costo y Ac_{val} durante el entrenamiento del modelo para BCG (Fold 1). (a) Evolución del Costo durante el entrenamiento (Fold 1). Fuente: Elaboración Propia. (b) Evolución de la Exactitud de Validación durante el entrenamiento (Fold 1). Fuente: Elaboración Propia.

a medida que el SNR es mayor. No obstante, existe una diferencia mayor entre el máximo y el mínimo de cada métrica.

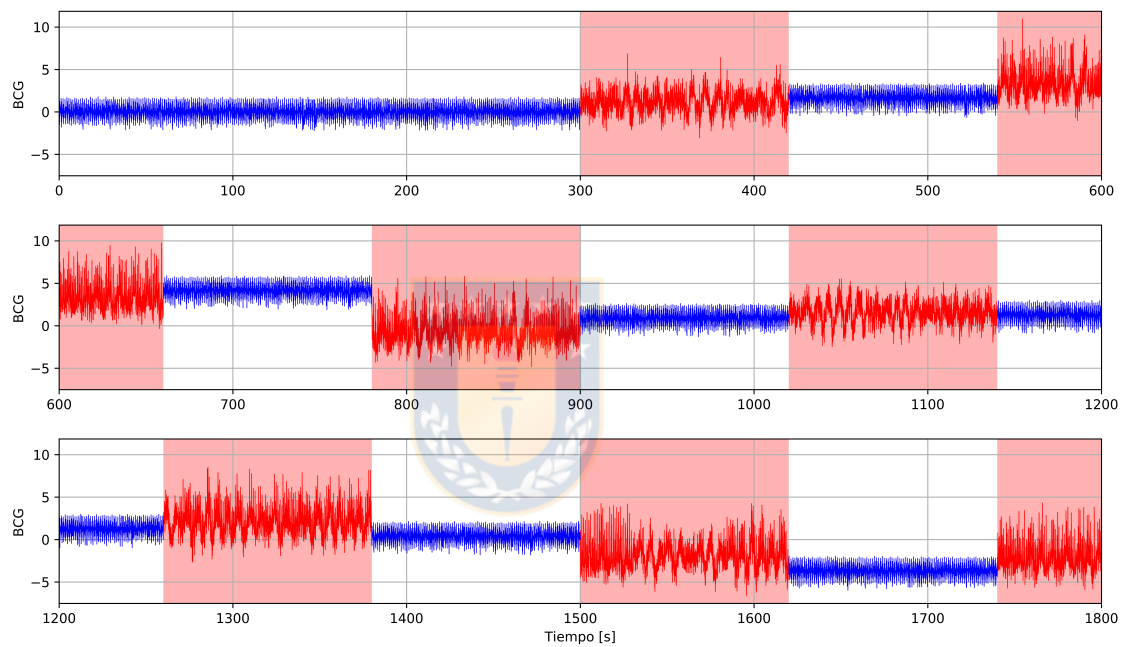


Figura 5.7: Detección de artefactos en el registro de BCG 21. Representación gráfica de la detección de artefactos en el registro 21 contaminado a 0 [dB] de SNR. Los segmentos de BCG detectados como artefactos se grafican en rojo, mientras que los segmentos limpios se muestran en azul. Por su parte, las zonas efectivamente contaminadas con *nst* han sido destacadas con fondo rojo transparente. Fuente: Elaboración Propia.

Tabla 5.8: Resultados de Validación Cruzada K-fold en BCG. Métricas calculadas con respecto al conjunto de prueba. Fuente: Elaboración Propia.

Fold	Ac	P	R	$F - 1$
1	0.967	0.985	0.949	0.967
2	0.918	0.901	0.940	0.920
3	0.960	0.988	0.932	0.959
4	0.768	0.693	0.963	0.806
5	0.914	0.895	0.939	0.916
6	0.935	0.961	0.906	0.933
7	0.931	0.938	0.924	0.931
8	0.932	0.957	0.904	0.930
9	0.929	0.969	0.887	0.926
Promedio	0.911	0.913	0.924	0.915
Máximo	0.967	0.988	0.963	0.967
Mínimo	0.768	0.693	0.887	0.806

Tabla 5.9: Desempeño del clasificador de BCG a distintos niveles de SNR. Fuente: Elaboración Propia.

SNR [dB]	Ac	P	R	$F1$
12	0.815	0.889	0.744	0.803
6	0.931	0.916	0.966	0.937
0	0.956	0.929	0.998	0.960
-6	0.967	0.945	1.000	0.970

5.2.3. Evaluación de confiabilidad de métricas cardíacas

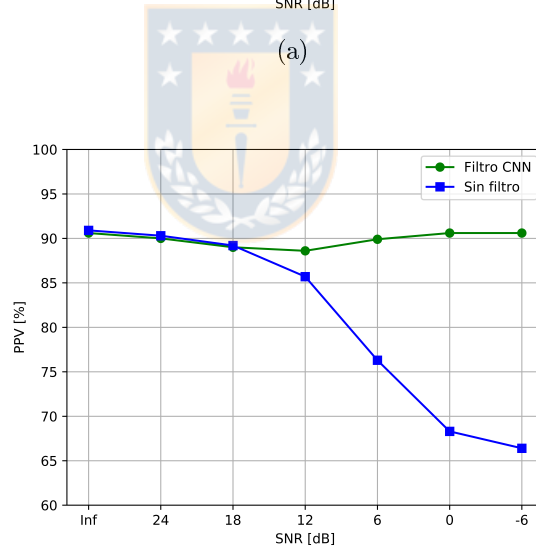
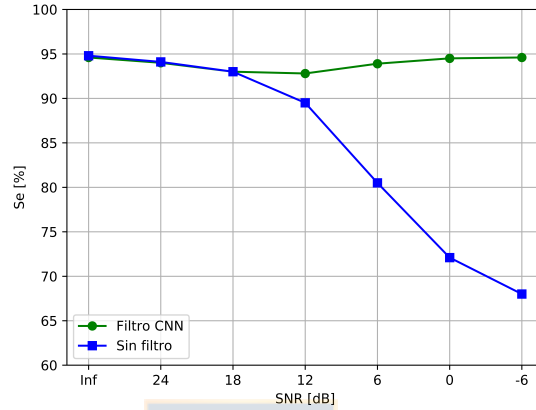
Al igual que en el caso del ECG, se integró el modelo final del clasificador en el sistema de detección de latidos. Este sistema se alimentó con registros de BCG sintético contaminados a distintos niveles de SNR. Los resultados agrupados por nivel de SNR se pueden apreciar en la Tabla 5.10, y de forma gráfica en la Figuras 5.8(a) y 5.8(b). Nuevamente, si comparamos Se_{sf} contra Se_f , podemos apreciar que existe una mejora de 13.4 %, 22.4 % y 26.6 % para registros contaminados a 6 [dB], 0 [dB] y -6 [dB], respectivamente.

Tabla 5.10: Impacto del clasificador en desempeño del algoritmo de detección de latidos en BCG. Comparación del desempeño del algoritmo de detección de latidos de[18] ante registros contaminados a distintos niveles de SNR. Los sub-índices sf y f denotan las métricas calculadas sobre los registros sin filtrar y filtrados, respectivamente, y FSR corresponde al porcentaje de la señal que fue filtrada. Fuente: Elaboración Propia.

SNR [dB]	Se_{sf}	PPV_{sf}	Se_f	PPV_f	FSR [%]
inf	0.948	0.909	0.946	0.906	7.397
24	0.941	0.903	0.94	0.9	8.889
18	0.93	0.892	0.93	0.89	16.698
12	0.895	0.857	0.928	0.886	35.302
6	0.805	0.763	0.939	0.899	45.191
0	0.721	0.683	0.945	0.906	45.905
-6	0.68	0.664	0.946	0.906	45.111

Impacto en la Variabilidad de la Frecuencia Cardíaca

El sistema de la Figura 4.12 cuenta con un módulo de cálculo de frecuencia cardíaca instantánea que, como se explicó anteriormente, se utiliza para el diagnóstico de diversas enfermedades como las arritmias. La introducción de un detector de artefactos en dicho sistema tiene por objetivo rotular o marcar los segmentos de la señal que se consideren contaminados, de forma de alertar al usuario que las métricas cardíacas calculadas en aquellos instantes no son confiables.



(b)

Figura 5.8: Desempeño del detector de latidos de BCG en función del SNR. Comparación de las versiones original y filtrada. (a) Sensibilidad del algoritmo en función del SNR. Fuente: Elaboración Propia. (b) Valor predictivo positivo del algoritmo en función del SNR. Fuente: Elaboración Propia.

Para ver el impacto del detector de artefactos, se graficó la frecuencia cardíaca instantánea del registro 102 a $\text{SNR} = 0$ [dB], para la versión filtrada y original (Figura 5.9). Se puede apreciar en la imagen superior que, en las zonas contaminadas, la frecuencia tiene una alta variabilidad, lo cual podría ser interpretado erróneamente por el usuario como signo de alguna anomalía cardíaca. Por el contrario, en la imagen inferior, los puntos en esas zonas no se muestran por ser poco confiables, de acuerdo al clasificador.

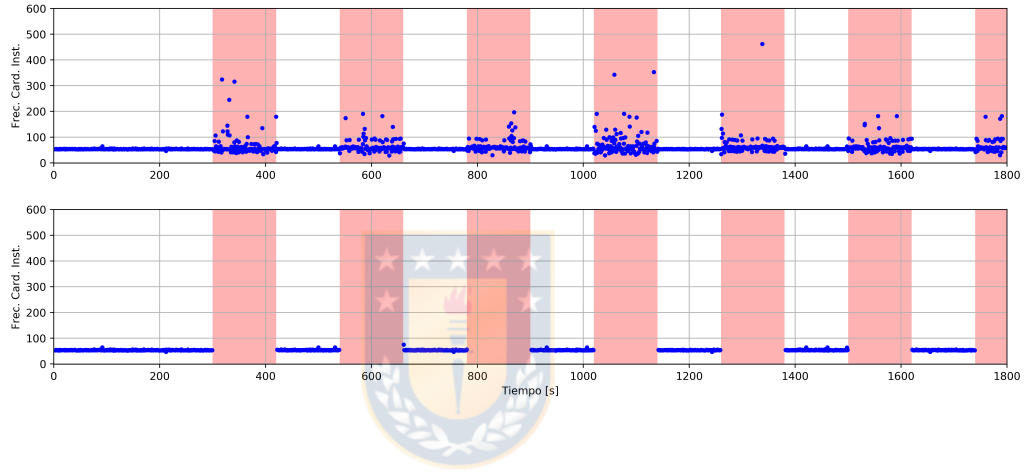


Figura 5.9: Frecuencia cardíaca instantánea del registro de BCG 21. Representación gráfica de la frecuencia cardíaca instantánea en el registro 21 contaminado a 0 [dB] de SNR, en sus versiones sin filtrar (imagen superior) y filtrada de acuerdo al detector de artefactos (imagen inferior). Las zonas efectivamente contaminadas con *nst* se destacan con fondo rojo transparente. Fuente: Elaboración Propia.

5.2.4. Evaluación de implementación en dispositivo portable

Se implementó el modelo del clasificador de BCG en el sistema portable descrito en la Sección 4.6, utilizando los parámetros obtenidos mediante el proceso de entrenamiento (*Fold* 1). El tiempo de predicción promedio calculado corresponde a 178 [ms], que es bastante más que el obtenido con el sistema computacional usado en el entrenamiento (1.1 [ms]). Sin embargo, en el contexto de una detección en línea de MAs, no es un resultado restrictivo, pues representa sólo 3 % del tamaño de la ventana. Además, si considera un sistema en línea que procesa una ventana de 6 [s]

de largo cada un segundo (ventanas traslapadas), el tiempo de predicción es sólo el 17.8 % de cada ciclo, quedando aproximadamente 802 [ms] para realizar otras tareas.

5.3. Discusión

El resultado del diseño de la arquitectura de los clasificadores, su desempeño de clasificación y el impacto en un sistema de detección de latidos, fueron expuestos en el presente capítulo. La metodología propuesta para la elección de los hiperparámetros del modelo funcionó bien en el caso del clasificador de ECG, mas no tanto en el clasificador de BCG. Esto era una posibilidad que se tenía en cuenta, pues la estrategia de búsqueda iterativa no explora todas las combinaciones de valores posibles, y por ende no es capaz de encontrar un óptimo global. Como solución a este problema, en el caso el BCG se propuso realizar una segunda búsqueda iterativa, partiendo esta vez desde el modelo encontrado en la exploración del clasificador de ECG, con lo cual se obtuvieron mejores resultados que los hallados inicialmente. Lo anterior da cuenta de lo sensible que es la elección del punto inicial en el proceso de búsqueda. Sin embargo, esto al mismo tiempo abre la posibilidad de mejorar la metodología mediante la realización de múltiples búsquedas iterativas con distintos puntos iniciales, explorando así una mayor cantidad de combinaciones. De cualquier forma, queda pendiente llevar a cabo una exploración de hiperparámetros más exhaustiva, para lo cual sería ideal contar con un sistema de hardware más potente basado en GPUs, o un *cluster* de computadores dedicado a este tipo de tareas.

La diferencia de las arquitecturas de los clasificadores radica básicamente en el tamaño del vector de entrada. En general, la frecuencia cardíaca de una persona en reposo varía entre 60 y 100 [bpm], es decir, entre 1 y 1.66 [latidos/s]. No sorprende entonces que el largo de ventana óptimo encontrado en el caso del ECG sea de 1 [s]. De acuerdo a esto, pareciera ser que la morfología de un complejo S-QRS-T fuera suficientemente distinta de la morfología de los MAs. Por su parte, el largo de ventana utilizado para el clasificador de BCG es de 6 [s], lo cual no debería atribuirse sólo a las diferencias de morfología de las señales de ECG y BCG, sino más bien al hecho de que los registros de BCG utilizados son señales semi-sintéticas, compuestas de una concatenación aleatoria de latidos de BCG reales. Usar señales semi-sintéticas

es, en efecto, una forma artificial de aumentar el tamaño del conjunto de datos, ya que son básicamente los mismos latidos pero ordenados de formas distintas. Por lo mismo, pareciera ser que el clasificador encuentra ciertos patrones en los conjuntos de latidos sucesivos, más que en cada latido por separado. La decisión de contaminar las señales de BCG con MAs de ECG es motivada, de forma similar, por la escasez de registros de MAs en BCG, y se funda en el parecido morfológico de dichos artefactos. Resultaría interesante en el futuro entrenar el clasificador con una base de datos de BCG real más extensa, y así confirmar la anteriores hipótesis.

El desempeño de clasificación promedio del detector de ECG, cercano al 96 %, es bueno si se compara con otros trabajos disponibles en la literatura. A modo de ejemplo, en [37] se presenta un clasificador basado en SVM, cuya exactitud alcanza sólo el 93 %. Nuestro clasificador de BCG, si bien no logra el mismo nivel de exactitud, aún así muestra un desempeño cercano al 91 %. Una inspección simple de señales de BCG con presencia natural de artefactos, muestra que en general su aparición no es sutil. Por esto, podríamos aventurarnos y decir que el desempeño de clasificación del detector de ECG y BCG en ambientes reales sería más cercano al 97 % y 95 % respectivamente (de acuerdo el promedio aproximado de la Exactitud de los registros contaminados a 6, 0 y -6 [dB], Tabla 5.9). Por su parte, los resultados de Precisión y Exhaustividad nos muestran que los clasificadores diseñados son conservadores con respecto a la detección de artefactos, presentando más falsos positivos que falsos negativos. Ambos clasificadores, además, poseen una Exactitud de entrenamiento elevada, cercana al 99 % en ambos casos. Esto nos dice que los modelos propuestos no presentan *underfitting*, siendo capaces de aprender de los datos de entrada de forma eficaz. Sin embargo, y a pesar de haber incluido una estrategia de regularización agresiva basada en *Dropout*, el modelo no es capaz de generalizar tan bien, lo que se explica por los valores de Exactitud calculados sobre el conjunto de validación y prueba. De acuerdo a la literatura, contar con un conjunto de datos de entrenamiento más grande podría ayudar a aumentar el desempeño de la red.

Al evaluar el impacto del clasificador en el sistema de detección de latidos, se encuentra que la Sensibilidad y el Valor Predictivo Positivo mejoran de 75 % a 98.5 % y de 77 % a 99 % respectivamente, para el caso de señales de ECG contaminadas a $SNR = 0$ [dB]. En el caso del BCG, las métricas mejoran de 72.1 % a 94.5 %, y de

68 % a 90 %. En otras palabras, la inclusión del clasificador ayuda a disminuir la cantidad de falsos positivos del detector de latidos, causa principal del aumento no fisiológico de la variabilidad de la frecuencia cardíaca, como quedó demostrado en la Sección 5.2.3. No obstante, la variabilidad de la frecuencia cardíaca es sólo una de las métricas utilizadas para el diagnóstico de enfermedades. En este sentido, en el futuro podrían realizarse análisis más elaborados e implementarse, por ejemplo, algoritmos para la medición de la elevación del segmento S-T en el ECG, o la amplitud I-J en el BCG. Detectar automáticamente la aparición de MAs, en definitiva, permite a los sistemas trabajar con señales de mejor calidad, de manera confiable. Esto es crucial en dispositivos de medición portables para *homecare* y ambientes laborales, donde la calidad de las señales adquiridas se ve altamente comprometida. Mejorar la robustez de estos sistemas, a su vez, facilitará la adopción de los mismos para el monitoreo y diagnóstico preliminar de enfermedades.

Finalmente, y de acuerdo a los resultados de tiempo de predicción obtenidos al implementar la red en un sistema basado en Raspberry Pi 3, podemos concluir que es factible utilizar el clasificador para la detección en línea de artefactos. Si consideramos que el *throughput* de un sistema portable de cálculo de frecuencia cardíaca es de una medición por segundo (una medición por latido), el sistema de detección propuesto podría ser incorporado sin mayores problemas, quedando tiempo suficiente para llevar a cabo otras tareas.

Capítulo 6. Conclusiones

Las redes neuronales convolucionales unidimensionales se presentan como una buena alternativa para la clasificación de señales por su morfología, y en particular para la detección de artefactos por movimiento en ECG y BCG.

La metodología propuesta para el diseño de un clasificador basado en 1D CNN, y su correspondiente implementación en software utilizando Python, Tensorflow, Keras y otros módulos disponibles de forma gratuita, permiten la obtención de modelos con buen desempeño de clasificación, en un tiempo relativamente corto.

Los modelos entrenados con señales de ECG y BCG semi-sintético son capaces de detectar la presencia de MAs con una alta Exactitud de clasificación. En el caso del ECG, se logra un 96 %, y en el caso del BCG, un 91 %.

La inclusión de estos clasificadores en sistemas de detección de latidos disminuye sustantivamente los falsos positivos, con la correspondiente mejora de la Sensibilidad y el Valor Predictivo Positivo de los algoritmos. En el caso del ECG, y para señales contaminadas a $\text{SNR} = 0$ [dB], se logra una mejora de 75 % a 98.5 % y de 77 % a 99 %, respectivamente. Por su parte, en el caso del BCG las métricas mejoran de 72.1 % a 94.5 % y de 68 % a 90 %.

Rotular segmentos de señal como zonas contaminadas permite, además, aumentar la confiabilidad de la frecuencia cardíaca instantánea, evitando de esta forma falsas alarmas y el diagnóstico erróneo de patologías.

Al implementar el clasificador de BCG en un sistema basado en Raspberry Pi 3, la detección de artefactos en ventanas de 6 [s] sólo toma 178 [ms], lo que equivale al 2 % del tiempo disponible. Por ende, es posible incluirlo en sistemas portables de adquisición de señal y cálculo en línea de variables cardíacas, mejorando la robustez de los mismos.

Bibliografía

- [1] Raspberry pi 3 model b+. [Online; accessed 23-January-2018].
- [2] World Health Organization. The top 10 causes of death. [Online; accessed 26-December-2018].
- [3] Paul A. Heidenreich, Justin G. Trogdon, Olga A. Khavjou, Javed Butler, Kathleen Dracup, Michael D. Ezekowitz, Eric Andrew Finkelstein, Yuling Hong, S. Claiborne Johnston, Amit Khera, Donald M. Lloyd-Jones, Sue A. Nelson, Graham Nichol, Diane Orenstein, Peter W.F. Wilson, and Y. Joseph Woo. Forecasting the future of cardiovascular disease in the united states. *Circulation*, 2011.
- [4] Ministerio de desarrollo social. Encuesta de caracterización socioeconómica nacional (CASEN) 2013. Technical report, Gobierno de Chile, 2015.
- [5] Ministerio de salud. Encuesta nacional de salud. Technical report, Gobierno de Chile, 2010.
- [6] Departamento de Estadísticas e Información en salud. Indicadores básicos de salud. Technical report, Gobierno de Chile, Ministerio de Salud., 2014. <http://www.deis.cl/wp-content/uploads/2015/08/IBS-2014.pdf>.
- [7] O T Inan, M Etemadi, R M Wiard, L Giovangrandi, and G T A Kovacs. Robust ballistocardiogram acquisition for home monitoring. *Physiological Measurement*, 30(2):169, 2009.
- [8] M. Etemadi, O. T. Inan, L. Giovangrandi, and G. T. A. Kovacs. Rapid assessment of cardiac contractility on a home bathroom scale. *IEEE Transactions on Information Technology in Biomedicine*, 15(6):864–869, Nov 2011.
- [9] Octavian Postolache, Pedro Silva Girão, Joaquim Mendes, Eduardo C. Pinheiro, and Gabriela Postolache. Physiological parameters measurement based on wheelchair embedded sensors and advanced signal processing. *IEEE Trans. Instrumentation and Measurement*, 59:2564–2574, 2010.
- [10] D.E. Arias, E.J. Pino, P. Aqueveque, and D.W. Curtis. Unobtrusive support system for prevention of dangerous health conditions in wheelchair users. *Mobile Information Systems*, 2016:1–14, 2016.
- [11] S. Junnila, A. Akhbardeh, A. Varri, and T. Koivistoinen. An EMFi-film sensor based ballistocardiographic chair: performance and cycle extraction method. In *IEEE Workshop on Signal Processing Systems Design and Implementation, 2005.*, pages 373–377, Nov 2005.
- [12] A. Lindqvist, K. Pihlajamäki, J. Jalonen, V. Laaksonen, and J. Alihanka. Static-charge-sensitive bed ballistocardiography in cardiovascular monitoring. *Clinical Physiology*, 16(1):23–30, 1996.

- [13] M. M. Butt, U. Akram, and S. A. Khan. Denoising practices for electrocardiographic (ECG) signals: A survey. In *2015 International Conference on Computer, Communications, and Control Technology (I4CT)*, pages 264–268, April 2015.
- [14] Y. Liu and M. G. Pecht. Reduction of skin stretch induced motion artifacts in electrocardiogram monitoring using adaptive filtering. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6045–6048, Aug 2006.
- [15] J. Lee, D. D. McManus, S. Merchant, and K. H. Chon. Automatic motion and noise artifact detection in holter ecg data using empirical mode decomposition and statistical approaches. *IEEE Transactions on Biomedical Engineering*, 59(6):1499–1506, June 2012.
- [16] S. Lawless. Crying wolf: False alarms in a pediatric intensive care unit. *Critical Care Medicine*, 22(6), 1994.
- [17] O. Such. Motion tolerance in wearable sensors - the challenge of motion artifact. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1542–1545, Aug 2007.
- [18] E. J. Pino, J. A. P. Chávez, and P. Aqueveque. Noninvasive ambulatory measurement system of cardiac activity. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7622–7625, Aug 2015.
- [19] Andrew D. Wiens, Mozziyar Etemadi, Shuvo Roy, Liviu Klein, and Omer T. Inan. Toward continuous, noninvasive assessment of ventricular function and hemodynamics: Wearable ballistocardiography. *IEEE J. Biomedical and Health Informatics*, 19(4):1435–1442, 2015.
- [20] William R. Scarborough, Samuel A. Talbot, John R. Braunstein, Maurice B. Rappaport, William Dock, William R. Scarborough, W. F. Hamilton, John E. Smith, John L. Nickerson, Samuel A. Talbot, and Isaac Starr. Proposals for ballistocardiographic nomenclature and conventions: Revised and extended. *Circulation*, 14(3):435–450, 1956.
- [21] Harry Mandelbaum and Robert A. Mandelbaum. Studies utilizing the portable electromagnetic ballistocardiograph. *Circulation*, 6(4):578–585, 1952.
- [22] Isaac Starr and Francis C. Wood. Twenty-year studies with the ballistocardiograph. *Circulation*, 23(5):714–732, 1961.
- [23] O. T. Inan, M. Etemadi, A. Paloma, L. Giovangrandi, and G. T A Kovacs. Non-invasive cardiac output trending during exercise recovery on a bathroom-scale-based ballistocardiograph. *Physiological Measurement*, 30(3):261–274, 2009.
- [24] R González-Landaeta, O Casas, and R Pallés-Areny. Heart rate detection from an electronic weighing scale. *Physiological Measurement*, 29(8):979, 2008.

- [25] A. Akhbardeh, B. Kaminska, and K. Tavakolian. Bseg++: A modified blind segmentation method for ballistocardiogram cycle extraction. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1896–1899, Aug 2007.
- [26] D. C. Mack, J. T. Patrie, P. M. Suratt, R. A. Felder, and M. Alwan. Development and preliminary validation of heart rate and breathing rate detection using a passive, ballistocardiography-based sleep monitoring system. *IEEE Transactions on Information Technology in Biomedicine*, 13(1):111–120, Jan 2009.
- [27] Joonas Paalasmaa. Detecting heartbeats in the ballistocardiogram with clustering. *Proceedings of the ICML/UAI/COLT 2008 Workshop on Machine Learning for Health-Care Applications, Helsinki, Finland, 2008.*, 2008.
- [28] L. Rosales, M. Skubic, D. Heise, M. J. Devaney, and M. Schaumburg. Heartbeat detection from a hydraulic bed sensor using a clustering approach. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 2383–2387, Aug 2012.
- [29] J. M. Kortelainen and J. Virkkala. FFT averaging of multichannel BCG signals from bed mattress sensor to improve estimation of heart beat interval. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6685–6688, Aug 2007.
- [30] C Brüser, S Winter, and S Leonhardt. Robust inter-beat interval estimation in cardiac vibration signals. *Physiological Measurement*, 34(2):123, 2013.
- [31] I. Sadek, J. Biswas, B. Abdulrazak, Z. Haihong, and M. Mokhtari. Continuous and unconstrained vital signs monitoring with ballistocardiogram sensors in headrest position. In *2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pages 289–292, Feb 2017.
- [32] K. Lydon, B. Y. Su, L. Rosales, M. Enayati, K. C. Ho, M. Rantz, and M. Skubic. Robust heartbeat detection from in-home ballistocardiogram signals of older adults using a bed sensor. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7175–7179, Aug 2015.
- [33] D. B. Springer, T. Brennan, L. J. Zühlke, H. Y. Abdelrahman, N. Ntusi, G. D. Clifford, B. M. Mayosi, and L. Tarassenko. Signal quality classification of mobile phone-recorded phonocardiogram signals. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1335–1339, May 2014.
- [34] D. Das, R. Banerjee, A. Dutta Choudhury, S. Bhattacharya, P. Deshpande, A. Pal, and K. M. Mandana. Novel features from autocorrelation and spectrum to classify phonocardiogram quality. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4516–4520, July 2017.

- [35] F. Shi and X. Li. Development of automatic motion artifact detection in mobile ECG monitor based on wavelet transform. In *2012 IEEE International Conference on Intelligent Control, Automatic Detection and High-End Equipment*, pages 166–170, July 2012.
- [36] J. Ottenbacher, M. Kirst, L. Jatoba, M. Huflejt, U. Grossmann, and W. Stork. Reliable motion artifact detection for ECG monitoring systems with dry electrodes. *Conf Proc IEEE Eng Med Biol Soc*, 2008:1695–1698, 2008.
- [37] Chenggang Yu, Zhenqiu Liu, and Andrew T. Reisner. A method for automatic identification of reliable heart rates calculated from ECG and PPG waveforms. *Journal of the American Medical Informatics Association*, 2006.
- [38] M. Limam and Frédéric Precioso. Atrial fibrillation detection and ECG classification based on convolutional recurrent neural network. *2017 Computing in Cardiology (CinC)*, pages 1–4, 2017.
- [39] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Biomedical Engineering*, 63:664–675, 2016.
- [40] Kyungna Kim. Arrhythmia classification in multi-channel ECG signals using deep neural networks. Master’s thesis, EECS Department, University of California, Berkeley, May 2018.
- [41] Rahul Kher, Tanmay Pawar, and Vishvjit Thakar. Classification of body movements in wearable ECG (W-ECG) signals using artificial neural networks. *American Journal of Biomedical Engineering*, 3(6):132–142, 2013.
- [42] Nobuyuki Miyake, Tetsuya Takiguchi, and Yasuo Ariki. Sudden noise reduction based on GMM with noise power estimation. *Journal of Software Engineering and Applications, Vol.3 No.4, 2010*, 2010.
- [43] A. Sugiyama, R. Miyahara, and Kwangsoo Park. Impact-noise suppression with phase-based detection. In *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5, Sept 2013.
- [44] T. Umematsu, S. Komeiji, M. Tsujikawa, and R. Isotani. Sudden-noise suppression with strike-portion detection based on phase linearity for speech recognition. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–4, Dec 2016.
- [45] Majd AlGhatrif and Joseph Lindsay. A brief review: history to understand fundamentals of electrocardiography. *Journal of community hospital internal medicine perspectives*, 2(1):10.3402/jchimp.v2i1.14383, April 2012.

- [46] Carlo Matteucci. Sur un phenomene physiologique produit par les muscles en contraction. *Ann Chim Phys* 6:339–41, 1842.
- [47] A. D. Waller. A demonstration on man of electromotive changes accompanying the heart's beat. *The Journal of physiology*, 8(5):229–234, October 1887.
- [48] W. Bruce Fye. A history of the origin, evolution, and impact of electrocardiography. *The American Journal of Cardiology*, 73(13):937 – 949, 1994.
- [49] Paul A. Iaizzo, editor. *Handbook of Cardiac Anatomy, Physiology and Devices*, chapter Basic ECG Theory, Recordings, and Interpretation, page 191. Humana Press, 2005.
- [50] Wikipedia contributors. Electrocardiography - Wikipedia, the free encyclopedia. [Online; accessed 26-December-2018].
- [51] JS; Rothbaum DA. Noble, RJ; Hillis. *Clinical Methods: The History, Physical, and Laboratory Examinations*. Butterworths, 1990.
- [52] V.V. Limaye, Hrishikesh; Deshmukh. ECG noise sources and various noise removal techniques: A survey. *International Journal of Application or Innovation in Engineering & Management (IJAIEEM)*, 5(2), February 2016.
- [53] J. W. Gordon. Certain molar movements of the human body produced by the circulation of the blood. *Journal of anatomy and physiology*, 11(Pt 3):533–536, April 1877.
- [54] Isaac Starr, A. J. Rawson, H. A. Schroeder, and N. R. Joseph. Studies on the estimation of cardiac ouptut in man, and of abnormalities in cardiac function, from the heart's recoil and the blood's impacts; the ballistocardiogram. *American Journal of Physiology-Legacy Content*, 127(1):1–28, 1939.
- [55] William Dock and Felix Taubman. Some technics for recording the ballistocardiogram directly from the body. *The American Journal of Medicine*, 7(6):751–755, December 1949.
- [56] L. Giovangrandi, O. T. Inan, R. M. Wiard, M. Etemadi, and G. T. A. Kovacs. Ballistocardiography; a method worth revisiting. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4279–4282, Aug 2011.
- [57] Emelie Vogt, David MacQuarrie, and John Patrick Neary. Using ballistocardiography to measure cardiac performance: a brief review of its history and future significance. *Clinical Physiology and Functional Imaging*, 32(6):415–420, 2012.
- [58] Alpo Varri. Ballistocardiogram (BCG), August 2008. [Online; accessed 26-December-2018].
- [59] J. Paalasmaa, H. Toivonen, and M. Partinen. Adaptive heartbeat modeling for beat-to-beat heart rate measurement in ballistocardiograms. *IEEE Journal of Biomedical and Health Informatics*, 19(6):1945–1952, Nov 2015.

- [60] Simon S. Haykin. *Neural Networks and Learning Machines*. Pearson Education, Upper Saddle River, NJ, third edition, 2009.
- [61] Michael Nielsen. Why are deep neural networks hard to train? [Online; accessed 23-January-2019].
- [62] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.
- [63] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.
- [64] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
- [65] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [66] Li Deng and Dong Yu. *Deep Learning: Methods and Applications*. Now Publishers Inc., Hanover, MA, USA, 2014.
- [67] Yoshua Bengio. Learning deep architectures for AI. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.
- [68] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012.
- [69] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, Dec 1989.
- [70] Fei Fei Li, Justin Johnson, and Serena Yeung. Convolutional neural networks. [Online; accessed 26-December-2018].
- [71] Goodfellow, I., Y. Bengio, and A. Courville. *Deep Learning*, chapter Representation Learning, page 555. Adaptive computation and machine learning. MIT Press, 2016.
- [72] Thomas Serre, Gabriel Kreiman, Minjoon Kouh, Charles Cadieu, Ulf Knoblich, and Tomaso Poggio. A quantitative theory of immediate visual recognition. *Progress in Brain Research*, 165:33–56, 2007 2007.
- [73] Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks. In *NIPS proceedings*, 2012.

- [74] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):664–676, April 2017.
- [75] J. Zhang and C. Zong. Deep neural networks in machine translation: An overview. *IEEE Intelligent Systems*, 30(5):16–25, Sept 2015.
- [76] Goodfellow, I., Y. Bengio, and A. Courville. *Deep Learning*, chapter Introduction, page 6. Adaptive computation and machine learning. MIT Press, 2016.
- [77] Bottou L. Orr G. LeCun, Y. and K. Muller. *Neural Networks: Tricks of the trade*, chapter Efficient backprop. Springer, 1998b.
- [78] Jianxin Wu. CNN for dummies. Technical report, National Key Lab for Novel Software Technology. Nanjing University, China., 2015.
- [79] L. Goldberger Ary, N. Amaral Luis A., Glass Leon, M. Hausdorff Jeffrey, Ivanov Plamen Ch., G. Mark Roger, E. Mietus Joseph, B. Moody George, Peng Chung-Kang, and Stanley H. Eugene. Physiobank, physiotoolkit, and physionet. *Circulation*, 101(23):e215–e220, June 2000.
- [80] B. . Kohler, C. Hennig, and R. Orglmeister. The principles of software QRS detection. *IEEE Engineering in Medicine and Biology Magazine*, 21(1):42–57, Jan 2002.
- [81] PhysioNet. Physiobank annotations. [Online; accessed 26-December-2018].
- [82] PhysioNet. nst - noise stress test for ECG analysis programs. [Online; accessed 26-December-2018].
- [83] Javier Chávez. Non-invasive monitoring system of physiological variables. Master’s thesis, Universidad de Concepción, 2017.
- [84] Ali Isin and Selen Ozdalili. Cardiac arrhythmia detection using deep learning. *Procedia Computer Science*, 120:268 – 275, 2017. 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017, 22-23 August 2017, Budapest, Hungary.