

Anexos Trabajo Fin de Grado

Diseño, optimización y validación de un delineador de electrocardiogramas basado en redes neuronales profundas U-NET

Design, optimization and validation of an Electrocardiogram delineator based on deep neural networks U-NET

Autor

Rubén Fernández Rodrigo

Directores

Juan Pablo Martínez Cortés
Antonio Miguel Artiaga

Titulación del autor

Ingeniería de Tecnologías y Servicios de Telecomunicación

Escuela de Ingeniería y Arquitectura
2024

Índice

ANEXO I: Características del ECG	2
1.1 Ciclo cardíaco y ECG	2
1.1.1 Ciclo cardíaco	2
1.1.2 Relación ciclo cardíaco con ECG	3
1.2 Afecciones cardiacas de dataset sintético	4
1.2.1 Sr y sr_ecto	4
1.2.2 Arritmia	5
1.2.3 Bigeminy	6
1.3 Ruido.....	6
ANEXO II: Redes neuronales	9
2.1 Fundamentos de redes neuronales.....	9
2.1.1 Unidad básica	9
2.1.2 Función de activación	9
2.1.3 Hiperparametros.....	11
2.1.4 Funciones de pérdida	13
ANEXO III: Lectura y escritura	14
3.1 Lectura de la base de datos QTDB	14
3.2 Lectura de la base de datos LUDB	14
3.4 Normalización de datos	17
ANEXO IV: Pruebas entrenamiento.....	17
4.1 Generador de datos y cargador de datos.....	17
4.2 Robustez U-NET	19
4.3 Niveles y capas U-NET	25
4.4 Arquitectura U-NET con LSTM	28
4.5 Arquitectura U-NET con conexiones residuales.....	30
4.6 Arquitectura U-NET con mecanismos de atención	31
ANEXO V: Evaluación	31
5.1 Postprocesado	31
5.2 Evaluación single-lead y multilead	34
Bibliografía	35

ANEXO I: Características del ECG

1.1 Ciclo cardíaco y ECG

1.1.1 Ciclo cardíaco

El ciclo cardíaco está compuesto por una serie de eventos que ocurren desde el inicio de un latido cardíaco hasta el comienzo del siguiente. El ciclo se divide en dos partes, la contracción ventricular llamada sístole, y la relajación ventricular llamada diástole.

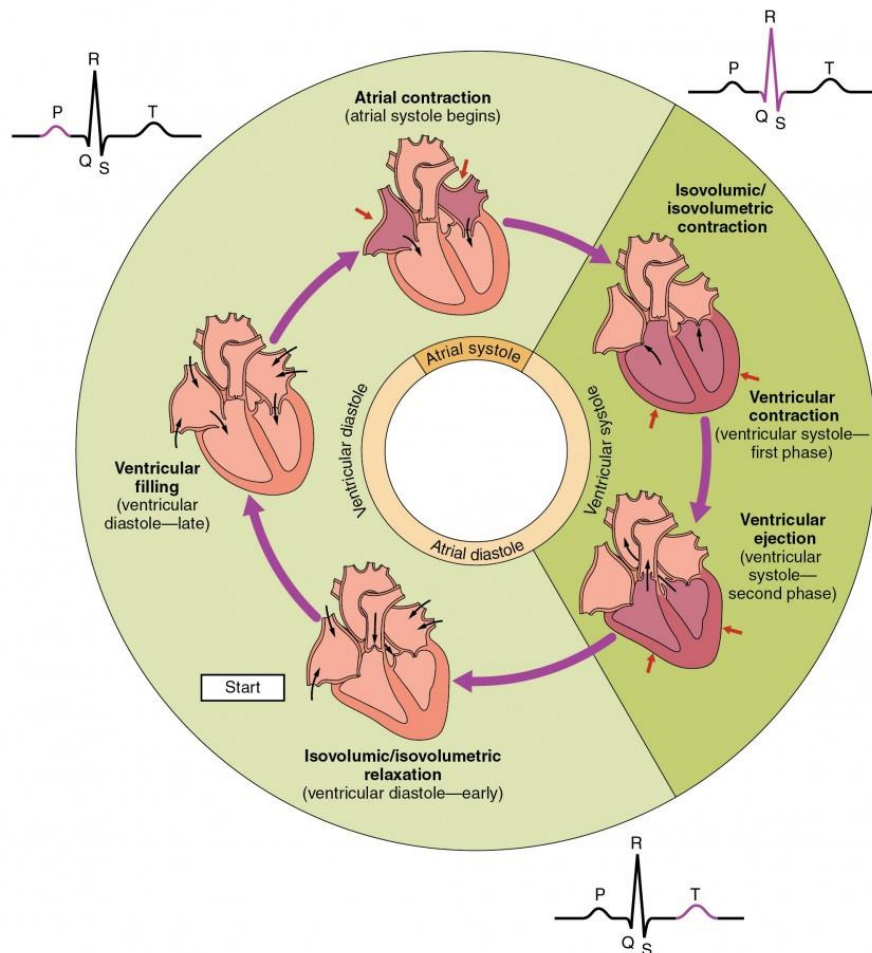


Figura 1: Representación de las fases del ciclo cardíaco junto con su onda correspondiente.

Disponible en: <https://courses.lumenlearning.com/suny-ap2/chapter/cardiac-cycle/>

La contracción ventricular, al igual que la relajación, consta de varias fases. Inicialmente, en la fase denominada contracción isovolumétrica, los ventrículos se contraen debido a la despolarización ventricular, aumentando la presión dentro de la cavidad. Durante esta fase, todas las válvulas permanecen cerradas (auriculoventriculares y semilunares), debido a que la presión en los ventrículos supera la de las aurículas y es menor que la de la aorta y la arteria pulmonar.

Seguidamente, en una fase conocida como eyección, la presión en los ventrículos excede la de la aorta y la arteria pulmonar, lo que provoca la apertura de las válvulas semilunares. Esto permite la expulsión de sangre desde el ventrículo izquierdo hacia la aorta y desde el ventrículo derecho hacia la arteria pulmonar.

La relajación ventricular se divide en cuatro fases. Al final de la sístole, en la fase de relajación isovolumétrica, los ventrículos se relajan y la presión disminuye rápidamente. Durante esta fase, las presiones elevadas en la aorta y la arteria pulmonar hacen que la sangre regrese a los ventrículos, cerrando las válvulas semilunares, mientras las aurículas se llenan de sangre.

A continuación, cuando la presión de los ventrículos cae por debajo de la de las aurículas, las válvulas auriculoventriculares se abren, permitiendo un flujo rápido de sangre hacia los ventrículos. Esta fase se conoce como llenado ventricular.

Finalmente, las aurículas se contraen para impulsar la sangre restante hacia los ventrículos (contracción auricular), completando así el ciclo cardíaco [18].

1.1.2 Relación ciclo cardíaco con ECG

La relación entre el ciclo cardíaco y el electrocardiograma (ECG) es fundamental para comprender la actividad eléctrica del corazón y cómo se refleja en un registro gráfico.

El ECG es una representación gráfica de la actividad eléctrica del corazón durante el ciclo cardíaco, capturando eventos como la despolarización y repolarización de las cámaras cardíacas, lo cual es esencial para el diagnóstico y la evaluación de la salud cardiovascular.

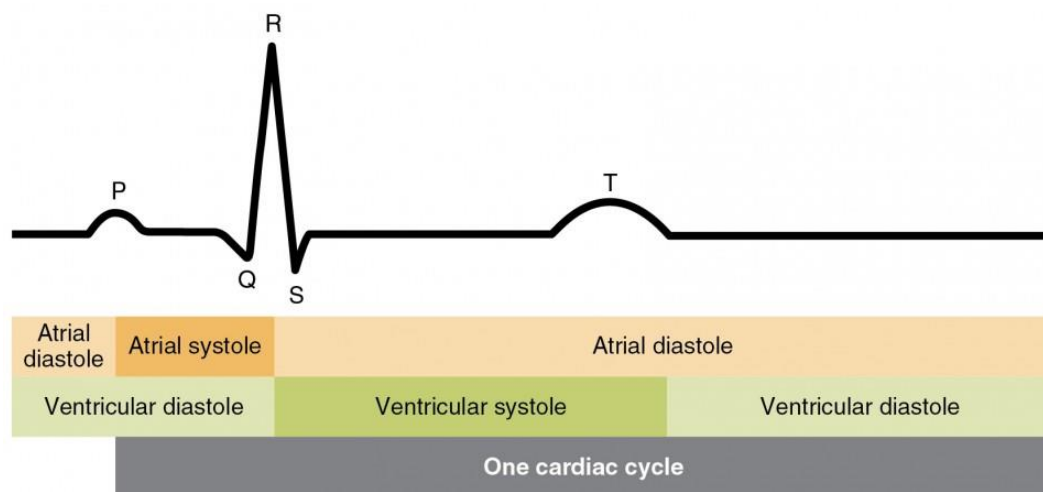


Figura 2: Correspondencia entre las fases del ciclo cardíaco y las ondas de un electrocardiograma “normal”. Disponible en: [Cardiac Cycle | Anatomy and Physiology II \(lumenlearning.com\)](https://lumenlearning.com)

En la Figura 2 se muestran las ondas presentes en un ciclo cardíaco completo. La onda P aparece inicialmente, marcando la transición de la diástole auricular a la sistole auricular durante la diástole ventricular. Esta onda representa la propagación del impulso eléctrico desde el nodo sinusal a través de las aurículas, indicando su contracción mientras los ventrículos permanecen relajados y llenos, como se observa en la Figura 1 en el paso de llenado ventricular a contracción auricular.

El complejo QRS está compuesto por dos señales eléctricas. La primera señal corresponde a la despolarización auricular, marcando el cambio de la sistole auricular a la diástole auricular, que coincide con la relajación de las aurículas. Sin embargo, esta señal se enmascara durante la contracción ventricular, que marca el cambio de la diástole ventricular a la sistole ventricular.

Desde la perspectiva de las fases cardíacas, esta onda ocurre durante la transición de la contracción auricular a la contracción isovolumétrica.

Finalmente, la onda T corresponde a la relajación de los ventrículos mientras las aurículas permanecen relajadas. Esta onda representa el cambio de la eyección ventricular a la relajación isovolumétrica.

1.2 Afecciones cardíacas de dataset sintético

La base de datos sintéticas contiene 5 afecciones cardíacas diferentes creadas a través de un simulador basado en ECGs reales.

1.2.1 Sr y sr_ecto

Las siglas SR hacen referencia a ritmo sinusal, que no se considera una afección, sino que esta caracterizado por un ciclo cardíaco normal donde los impulsos se originan en el nodo sinusal (“marcapasos natural”) y se propagan de manera ordenada a través de las aurículas y ventrículos.

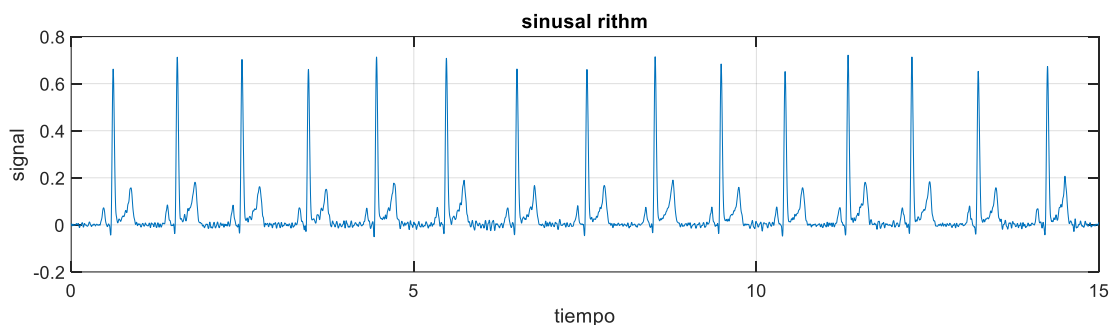


Figura 3: Ritmo sinusal obtenido del paciente 1 con ruido muscular de la base de datos sintética

Durante un ritmo sinusal pueden aparecer los denominados latidos ectópicos (sr_ecto). Un latido ectópico es un latido cardíaco que se origina en una parte del corazón distinta del nodo sinusal, que es el marcapasos natural del corazón. En condiciones normales, el nodo sinusal genera impulsos eléctricos que inician cada latido del corazón y aseguran un ritmo regular. Sin embargo, a veces otras áreas del corazón pueden generar impulsos eléctricos prematuros, lo que causa latidos adicionales [13], como se observa en la figura 4.

Pueden existir latidos auriculares prematuros, los cuales se originan en las aurículas (cámaras superiores del corazón) fuera del nodo sinusal. A su vez, los latidos ventriculares prematuros se originan en los ventrículos (las cámaras inferiores del corazón) [13].

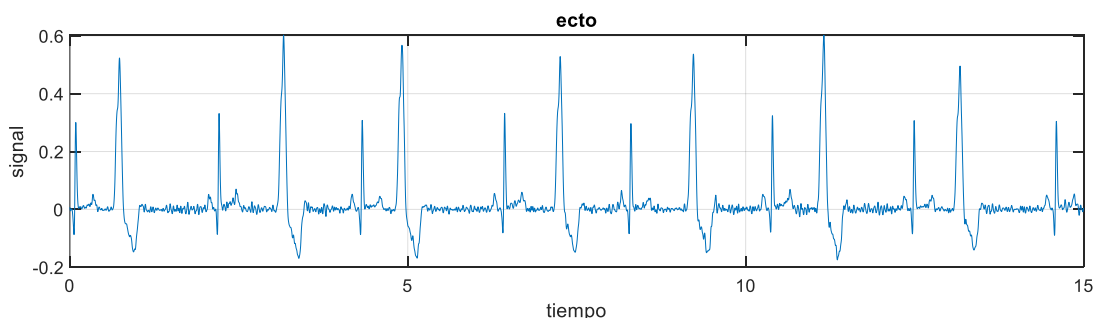


Figura 4: Ritmo sinusal ectópico obtenido del paciente 1 con ruido muscular de la base de datos sintética

1.2.2 Arritmia

La arritmia es una alteración del ritmo cardíaco normal, que puede manifestarse de diversas maneras, como latidos irregulares, demasiado rápidos (taquicardia) o demasiado lentos (bradicardia). Las arritmias pueden variar en severidad, desde ser inofensivas hasta potencialmente mortales [11]. A continuación se explican los diferentes tipos [11]:

Bradicardia sinusal: Ritmo sinusal normal pero con una frecuencia cardíaca más lenta de lo normal (menos de 60 latidos por minuto).

Bloqueos cardíacos: Alteración en la transmisión de los impulsos eléctricos a través del sistema de conducción del corazón. Ejemplos incluyen bloqueo AV (auriculoventricular) de primer, segundo o tercer grado.

Taquicardia sinusal: Ritmo sinusal normal pero con una frecuencia cardíaca más rápida de lo normal (más de 100 latidos por minuto).

Taquicardia supraventricular (TSV): Ritmos rápidos que se originan en las aurículas o en la unión auriculoventricular. Incluye taquicardia paroxística supraventricular (TPSV), aleteo y fibrilación auriculares.

Taquicardia ventricular (TV): Ritmo rápido que se origina en los ventrículos. Puede ser potencialmente mortal si no se trata.

Fibrilación auricular (FA): Latidos rápidos y descoordinados en las aurículas, que pueden llevar a una disminución en la eficiencia del bombeo del corazón y aumentar el riesgo de formación de coágulos. Aur_tach son las siglas usadas en esta base de datos para referirse a la fibrilación auricular

Fib_aur_ecto hace referencia a la fibrilación auricular con latidos ectópicos, como se ve en la figura 5.

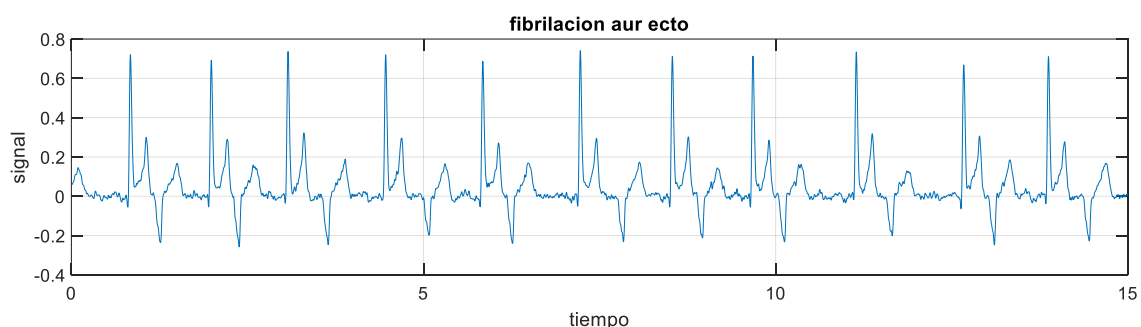


Figura 5: Fibrilación auricular ectópica obtenida del paciente 1 con ruido muscular de la base de datos sintética

Fibrilación ventricular (FV): Latidos rápidos y descoordinados en los ventrículos, lo que puede causar colapso circulatorio y muerte si no se trata inmediatamente con desfibrilación.

Extrasístoles auriculares: Latidos prematuros que se originan en las aurículas.

Extrasístoles ventriculares: Latidos prematuros que se originan en los ventrículos.

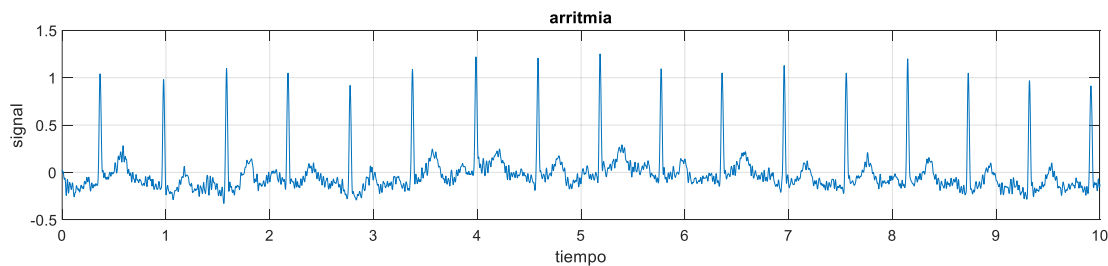


Figura 6: Arritmia obtenida del paciente 1 con ruido muscular de la base de datos sintética

1.2.3 Bigeminy

El término "bigeminy" (bigeminismo en español) se refiere a un patrón específico de arritmia cardíaca en el cual los latidos del corazón ocurren en pares: un latido normal es seguido por un latido ectópico. Este ciclo se repite continuamente, resultando en un patrón regular de un latido normal seguido por un latido prematuro [12].

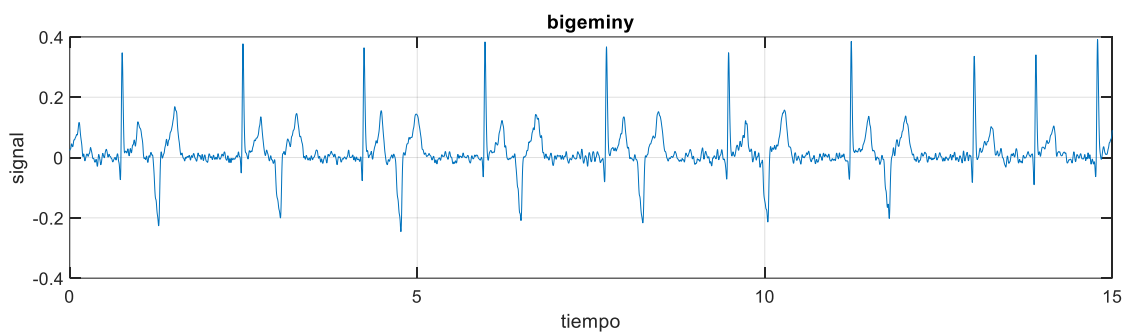


Figura 7: Bigeminismo obtenido del paciente 1 con ruido muscular de la base de datos sintética

1.3 Ruido

El cuerpo humano contiene diversas señales eléctricas originadas en el sistema nervioso. En un electrocardiograma, las señales de interés son aquellas que regulan la actividad cardíaca; no obstante, al efectuarse una medición mediante electrodos colocados en distintas partes del cuerpo, es inevitable que la señal bruta obtenida de estos electrodos no contenga exclusivamente la señal proveniente del corazón. Además, durante la realización de un electrocardiograma, los electrodos no permanecen estáticos en una posición fija, sino que es probable que se encuentren en movimiento debido a los procesos fisiológicos inherentes a cualquier ser humano. Por consiguiente, se observa la presencia de diversos ruidos al efectuar un ECG.

Uno de los ruidos más comunes presentes en el electrocardiograma (ECG) es la deriva de la línea de base. La línea de base se refiere al nivel de voltaje de referencia que se considera como el estado "en reposo" del sistema medido. Esta deriva surge principalmente debido a la respiración. Los electrodos V1-V6, en particular, se aproximan y alejan del corazón durante la respiración, lo

que produce una mayor amplitud de la señal durante la exhalación y una menor amplitud durante la inhalación.

Si se observa un periodo relativamente largo de la señal con deriva de línea de base, como se muestra en la figura 6, se puede apreciar la periodicidad de la respiración, manifestándose en oscilaciones con una frecuencia inferior a 0.5 Hz. Además, los movimientos corporales y los contactos débiles de los electrodos también contribuyen a la aparición de este tipo de ruido [1] [2].

Este ruido puede ser mitigado prácticamente con filtros paso bajo de frecuencia de corte $f_c = 0.5 \text{ Hz}$.

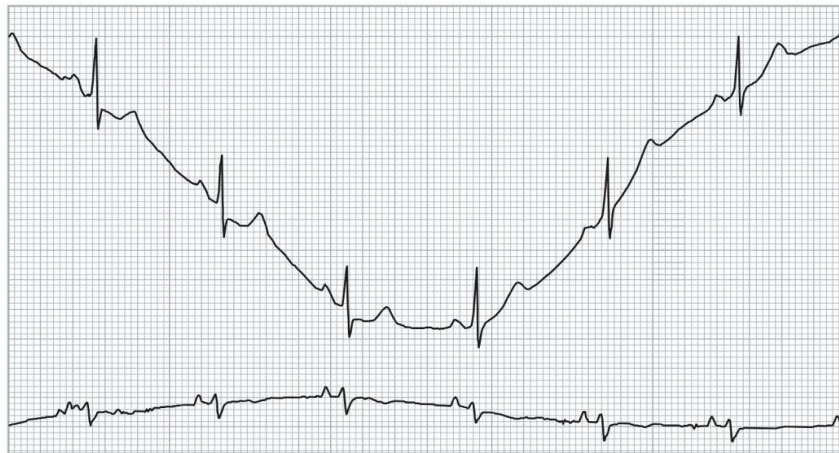


Figura 8: Sección de un electrocardiograma con ruido de deriva de la línea de base.

Disponible en: <https://www.gehealthcare.com/insights/article/a-guide-to-ecg-signal-filtering>

Otro de los ruidos existentes en la señal de un ECG en bruto es la interferencia de la línea eléctrica. Esta se produce debido a los campos magnéticos de los instrumentos acoplados a la red eléctrica. Produce una oscilación de entre 50 y 60 Hz, enmascarando la señal ECG, como se puede ver en la figura 7 [1] [2].

Durante la adquisición de registros de ECG, se pueden tomar diversas precauciones para minimizar el nivel de interferencia, tales como el blindaje de los cables, la correcta puesta a tierra del sistema de ECG y la reducción de la impedancia piel-electrodo [3]. Sin embargo, puede ser necesario aplicar algún procesamiento de señal para eliminar la interferencia residual de la línea eléctrica en los registros de ECG como el filtrado lineal de banda de rechazo o el filtrado no lineal [4].

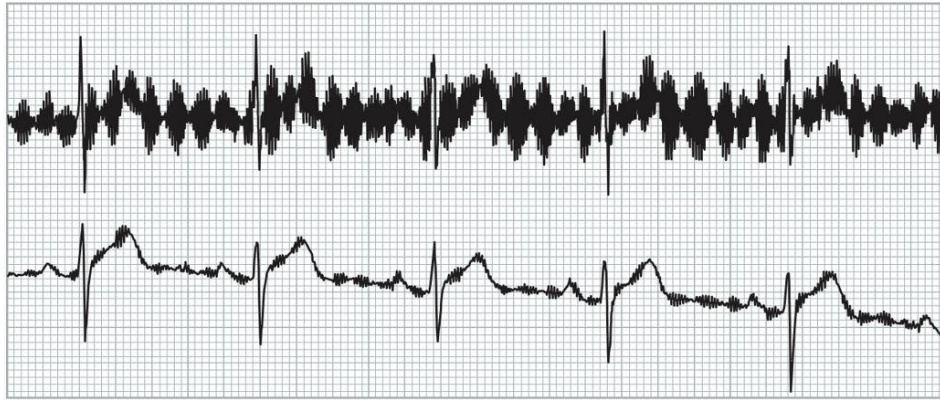


Figura 9: Sección de un electrocardiograma con ruido debido a interferencias con la línea eléctrica. Disponible en: <https://www.gehealthcare.com/insights/article/a-guide-to-ecg-signal-filtering>

Por último, en la figura 7 aparece ruido electromiográfico, ruido causado por actividades eléctricas en los músculos, que surgen de los movimientos oculares, movimientos musculares y el latido del corazón. Las fuentes típicas de ruido MA son los movimientos musculares cerca de la región de la cabeza, como los movimientos del cuello, la deglución, entre otros [1].

Las frecuencias de este ruido oscilan desde la continua hasta los 10000Hz, produciéndose un solape entre el espectro de la ECG y del ruido, provocando que la eliminación de este sea compleja.

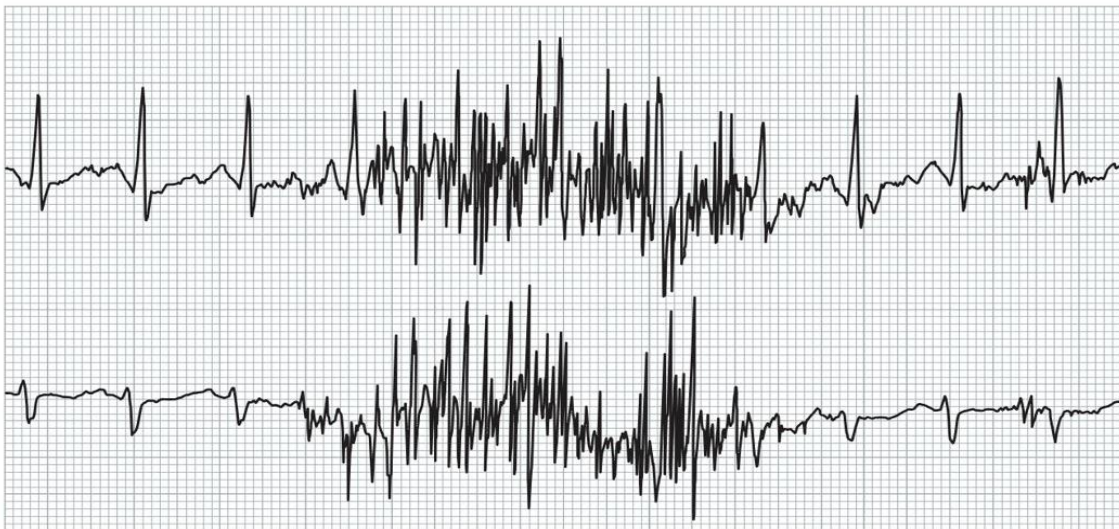


Figura 10: Sección de un electrocardiograma con ruido electromiográfico. Disponible en: <https://www.gehealthcare.com/insights/article/a-guide-to-ecg-signal-filtering>

ANEXO II: Redes neuronales

2.1 Fundamentos de redes neuronales

2.1.1 Unidad básica

Las redes neuronales están basadas en el funcionamiento del cerebro humano. Por ello, se creó una unidad básica que, mediante agrupaciones considerables de esta, pudiese conformar una red que fuese capaz de aprender tareas concretas.

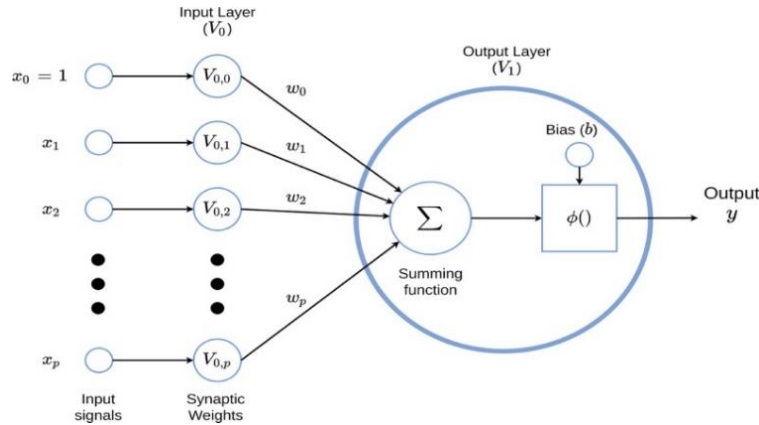


Figura 11: Esquema general del perceptrón. Disponible en:
<https://www.ncbi.nlm.nih.gov/books/NBK583971/>

Esta unidad básica de las redes neuronales se conoce como perceptrón. El perceptrón está compuesto por varias entradas, representadas como x_1, \dots, x_p en la Figura 9, que corresponden a la información recibida de un sistema externo u otros perceptrones dentro de la red. Cada entrada está asociada con un vector de pesos $w = (w_1, \dots, w_p)$, los cuales son ajustados durante el entrenamiento de la red para que esta pueda aprender las tareas específicas. Estos pesos determinan la importancia relativa de cada entrada en la salida del perceptrón [5].

De esta manera, v_j representa una combinación lineal de las entradas, calculada como la suma ponderada de x_j multiplicadas por sus respectivos pesos $w_{i,j}$:

$$v_j = \sum_{j=1}^p w_{i,j} * x_j \quad (1)$$

Adicionalmente, la activación del perceptrón depende de la función de activación elegida, la cual evalúa el valor v_j para determinar si el perceptrón se activa o no [5].

2.1.2 Función de activación

La función de activación en una red neuronal es crucial, ya que determina la salida de una neurona o de un conjunto de neuronas en función de la entrada que recibe. Esta función introduce no linealidades en el modelo, permitiendo a la red neuronal aprender y modelar relaciones complejas en los datos. Sin no linealidades, la combinación de perceptrones siempre sería una función lineal, lo que equivale a un modelo de DSP u no de red neuronal.

$$y_j = g(v_j) \quad (2)$$

Existen distintos tipos de funciones de activación dependiendo del problema abordado, en este caso se van a comentar las funciones de activación utilizadas en este trabajo.

La función ReLU (Rectified Linear Unit) es una función de activación que se ha vuelto extremadamente popular en el campo del aprendizaje profundo debido a sus propiedades beneficiosas y su eficiencia computacional. Su representación matemática es la siguiente:

$$RELU(v_j) = \max(0, v_j) \quad (3)$$

Y su representación gráfica es la siguiente:

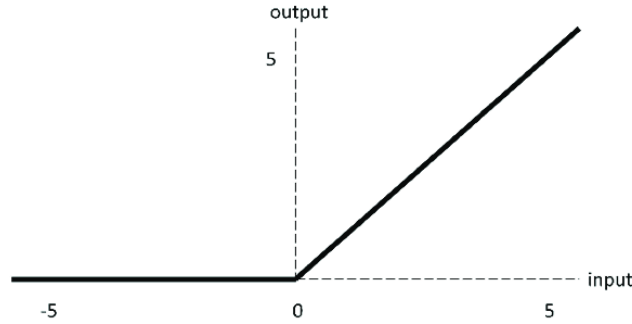


Figura 12: Gráfica de la función de activación ReLU. Disponible en:
<https://vidyasheela.com/post/relu-activation-function-with-python-code>

El problema que presentaba esta función era su falta de diferenciabilidad en el punto 0. Sin embargo, se ha adoptado convencionalmente asignarle un valor de 0 a su derivada en $v_j = 0$. Es una función fácil de calcular y altamente eficiente computacionalmente, lo cual facilita una convergencia más rápida en comparación con otras funciones como la tangente hiperbólica o la sigmoide. Además, al tener la capacidad de activar celdas con valor 0, promueve la activación esparcida (sparsity), lo que puede ser beneficioso para la regularización y la eficiencia computacional del modelo [6].

La segunda función de activación utilizada en este trabajo es la softmax. La función softmax es una función de activación que se utiliza comúnmente en la capa de salida de redes neuronales para problemas de clasificación multiclase. Esta función convierte un vector de números reales en un vector de probabilidades, donde cada componente del vector de salida representa la probabilidad de pertenencia a una clase específica [7].

Su representación matemática es la siguiente:

$$softmax(v_j)_i = \frac{e^{(v_j)_i}}{\sum_{l=1}^p e^{(v_j)_l}} \quad (3)$$

Esta función es suave y diferenciable en todo su dominio. Esto es crucial para el entrenamiento de redes neuronales utilizando métodos de optimización basados en gradientes, como el descenso de gradiente estocástico. Además, por su naturaleza suave y continua, la función softmax permite evaluar de manera más precisa el error de la red utilizando las probabilidades de cada clase como porcentajes. Esto facilita el análisis detallado de la confianza del modelo en sus predicciones y mejora la capacidad de ajuste durante el entrenamiento de redes neuronales para tareas de clasificación multiclase.

2.1.3 Hiperparametros

2.1.3.1 Batch, epcho y profundidad

Un batch (o lote) es un subconjunto del conjunto de datos de entrenamiento que se utiliza para una sola iteración del entrenamiento del modelo. En lugar de procesar todo el conjunto de datos a la vez, el conjunto de datos se divide en varios lotes más pequeños. Permiten una mayor eficiencia que el procesamiento completo de datos y mayor estabilidad que el procesamiento de ejemplos individuales.

Para elegir el tamaño del batch hay que tener en cuenta el hardware utilizado. Si permite procesar tamaños grandes de datos en paralelo lo más eficiente es establecer un batch alto, sin embargo, si esto no es posible, establecer un batch alto podría realentizar el entrenamiento.

Una epoch (o época) se refiere a una pasada completa a través de todo el conjunto de datos de entrenamiento.

La profundidad de una red neuronal se refiere al número de capas que componen la red. Esto incluye capas de entrada, capas ocultas y la capa de salida.

2.1.3.2 Learning rate

El learning rate o ratio de aprendizaje es el hiperparámetro que define el cambio que experimenta el modelo en respuesta al error que se estima cada vez que los pesos son actualizados.

$$w_{i,j}^{(k+1)} = w_{i,j}^{(k)} - \alpha * \frac{\partial J(x_n, y_n, w^k)}{\partial w_{i,j}} \quad (4)$$

En la fórmula 4 el valor $w_{i,j}^{(k+1)}$ representa el peso resultante de una entrada de un perceptrón concreto al aplicar el algoritmo de propagación, $w_{i,j}^{(k)}$ es el peso actual y $\frac{\partial J(x_n, y_n, w^k)}{\partial w_{i,j}}$ es la derivada de la función de coste con respecto a los pesos la cual informará de la dirección de actualización de dicho peso [8].

Por ende, ratios de aprendizaje elevados indica una convergencia más rápida (necesitando menos epochs) pero posiblemente un mayor error por no alcanzar nunca el resultado óptimo, provocando una oscilación notable de la función de coste alrededor de dicho valor.

Por el contrario, si se elige un ratio de aprendizaje bajo el número de epochs para llegar al óptimo será mayor pero obtendremos un valor mucho más próximo al óptimo y con menor oscilación de la función de coste.

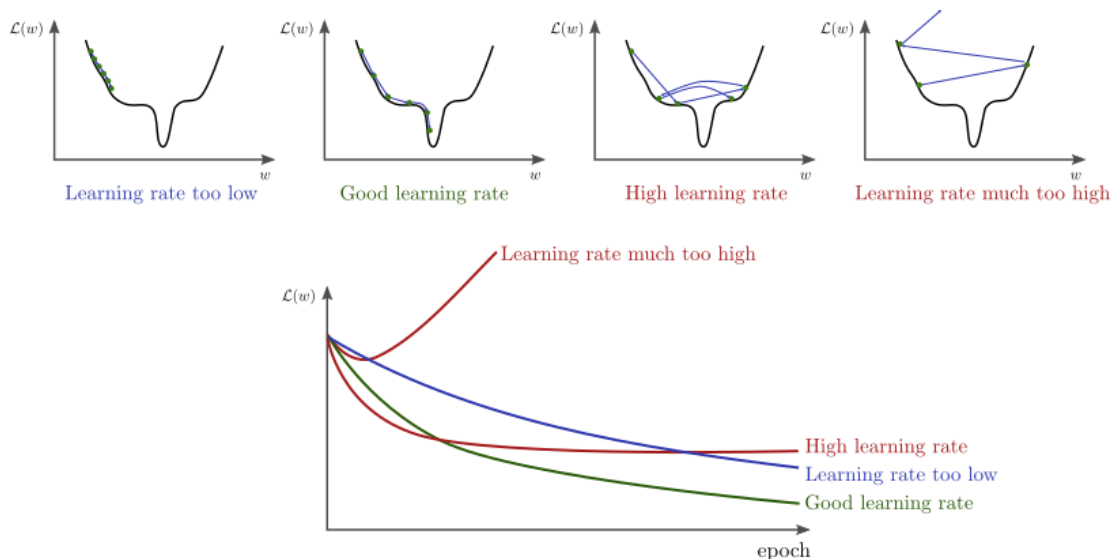


Figura 11: Representación de la función de pérdidas de una red neuronal durante el entrenamiento con diferentes ratios de aprendizaje. Disponible en:

<https://www.bdhammel.com/learning-rates/>

Por tanto, existe un compromiso entre rapidez de convergencia y precisión, como se observa en la figura 11. Adicionalmente, se debe de tener especial cuidado al elegir learning rates altos puede provocar que el modelo salte de mínimos locales de la función de coste a globales resultando en una falta de convergencia o convergencia inestable.

2.1.3.3 Dropout

Uno de los problemas más habituales durante la fase de entrenamiento de modelos de aprendizaje automático es el fenómeno conocido como overfitting o sobreajuste. El sobreajuste se produce cuando un modelo se ajusta excesivamente a los datos de entrenamiento, capturando no solo los patrones subyacentes sino también el ruido y las peculiaridades específicas de estos datos. Como consecuencia, el modelo logra un rendimiento excelente en el conjunto de datos de entrenamiento, pero presenta un rendimiento deficiente al enfrentarse a nuevos datos o entradas no vistas previamente. Esto indica una falta de capacidad de generalización del modelo.

Para evitar el sobreajuste se utiliza dropout. Durante cada iteración, en el entrenamiento, cada neurona tiene asignada una probabilidad p (hiperparámetro a elegir) de ser “apagada”, es decir, de no contribuir a la propagación. Por lo tanto, en cada iteración, se crea una red neuronal ligeramente diferente que reduce la co-adaptación de las neuronas [9].

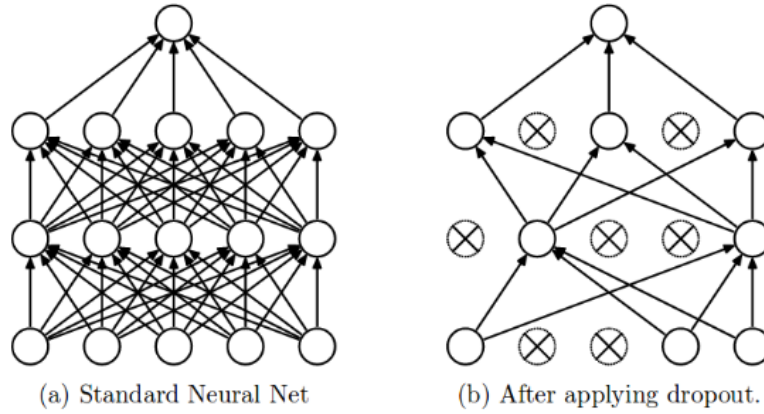


Figura 13: Representación de un conjunto de neuronas (perceptrones) de una red estándar (a) y la misma red aplicando dropout (b). Disponible en:
[https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer,](https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer)

2.1.4 Funciones de pérdida

La función de pérdida es una herramienta fundamental en el ámbito de las redes neuronales, encargada de comparar los objetivos con los resultados generados por la red, evaluando el grado de similitud entre ellos. En el presente trabajo, se utiliza una red U-Net 1D cuya salida comprende 4 clases: onda P, complejo QRS, onda T y nulo. Cada una de estas 4 salidas representa la probabilidad de que una muestra específica de la señal pertenezca a una de estas clases, facilitado por la implementación de la función de activación softmax.

Para la evaluación de las predicciones, se ha optado por la función de pérdida de entropía cruzada, reconocida por su efectividad en problemas de clasificación de múltiples clases, tal como se presenta en este estudio.

La entropía cruzada cuantifica la discrepancia entre dos distribuciones de probabilidad: la distribución verdadera (etiquetas reales) y la distribución predicha (probabilidades de clase generadas por el modelo) [10]. Una predicción precisa, donde la distribución predicha se asemeja a la verdadera, resulta en una baja entropía cruzada. En contraste, diferencias significativas entre ambas distribuciones incrementan la entropía cruzada. La expresión matemática de la entropía cruzada es la siguiente:

$$\ell(x, y) = \{l_1, \dots, l_N\}^T, l_n = -w_{y_n} * \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^C \exp(x_{n,c})} * 1 \quad (5)$$

Donde x es la entrada, y el objetivo, w los pesos, C el número de clases, y N es la dimensión del minibatch y d_1, \dots, d_k las dimensiones de ventana [10].

Por tanto, la salida de la red, en el caso general, tendrá (N, d_1, \dots, d_k) , y el objetivo tendrá dimensionalidad (N, d_1, \dots, d_k) , ya que son vectores cuyos términos indican la clase correcta (0-3) [10].

ANEXO III: Lectura y escritura

3.1 Lectura de la base de datos QTDB

En esta librería existen varios ficheros con diferente información. Sin embargo, los realmente necesarios para la delineación son los archivos con las extensiones .hea, .dat, .q1c, .q2c, .pu0 y .pu1.

Para comenzar la extracción de datos, es necesario leer primero el archivo de cabecera (.hea). Este archivo contiene información sobre el registro, como la que se muestra en la figura 14.

```
sel31 2 250 224993
sel31.dat 212 200 12 0 -22 -23899 0 ECG1
sel31.dat 212 200 12 0 26 -30651 0 ECG2
#Produced by xform_new from record 31, beginning at 13:20.000
#Produced by xform_new from record 31, beginning at 12:57:00.000
#-----
#Produced by "/files2/scheherazade/plaguna/database/delay" from record sel31
#The signal 0 was delayed with a delay=7 samples
```

Figura 14: Fichero .hea con información de cabecera de la grabación sel31 de la base de datos QT.

En la primera línea del archivo se especifica el nombre del archivo a analizar (sel31), junto con las derivaciones de dicha grabación (2), la frecuencia de muestreo (250 Hz) y el número total de muestras que contiene el registro de ECG (224,993 muestras).

En la segunda y tercera línea se describen las características de las dos derivaciones contenidas en el archivo .dat, que almacenará la señal ECG. La información relevante incluye el valor correspondiente a la línea de base (0) y la ganancia (200), parámetros utilizados para asegurar que la señal extraída se exprese en milivoltios.

Como se mencionó anteriormente, los archivos .dat contienen la señal sin normalizar, obtenida mediante un holter durante un periodo de 15 minutos.

Los archivos .q1c y .q2c contienen anotaciones realizadas por dos cardiólogos, indicando los inicios, finales y picos de las ondas P, QRS, T, y en algunos casos, la onda U (aunque esta última no se considera en el presente trabajo). Estas anotaciones se registran mediante números que identifican cada inicio, final y pico de cada tipo de onda, facilitando así su extracción.

Finalmente, los archivos .pu0 y .pu1 contienen las marcas generadas por un delineador automático basado en procesamiento digital de señales (DSP), ofreciendo la posibilidad de realizar un preprocesamiento con delineación automática. En estos archivos, se encuentran delineados los 15 minutos completos del ECG.

3.2 Lectura de la base de datos LUDB

Al igual que en la QT, es necesario leer la cabecera antes de la extracción de los datos. En este caso la estructura es la mostrada en la figura 15.


```

1 12 500 5000
1.dat 16 1716(6)/mV 0 0 -120 -32198 0 i
1.dat 16 1206(2)/mV 0 0 25 12402 0 ii
1.dat 16 1229(-5)/mV 0 0 145 -20936 0 iii
1.dat 16 1368(-5)/mV 0 0 47 -23456 0 avr
1.dat 16 1368(5)/mV 0 0 -132 26680 0 avl
1.dat 16 698(-1)/mV 0 0 85 -4930 0 avf
1.dat 16 1372(-1)/mV 0 0 150 -1105 0 v1
1.dat 16 1572(2)/mV 0 0 62 10564 0 v2
1.dat 16 2259(3)/mV 0 0 65 15400 0 v3
1.dat 16 2317(4)/mV 0 0 145 20455 0 v4
1.dat 16 2074(4)/mV 0 0 105 24979 0 v5
1.dat 16 1457(1)/mV 0 0 -25 7482 0 v6
#<age>: 51
#<sex>: F
#<diagnoses>:
#Rhythm: Sinus bradycardia.
#Electric axis of the heart: left axis deviation.
#Left ventricular hypertrophy.
#Left ventricular overload.
#Non-specific repolarization abnormalities: posterior wall.

```

Figura 15: Fichero .hea con información de cabecera de la grabación 1 de la base de datos Lobachevsky.

La estructura de la cabecera es prácticamente igual. La única diferencia existente con respecto a la QT es que existen 12 derivaciones en vez de 2, por lo que aparece la información de cada derivación.

Asimismo, los archivos .dat contienen la señal ECG medida en las 12 derivaciones sin normalizar.

Los archivos (.i, .ii, .iii, .avr, .avl, .avf, .v1, .v2, .v3, .v4, .v5, .v6) contienen las marcas de cada una de las 12 derivaciones. En esta base de datos no existen marcas al comienzo y al final de la ECG debido a que al iniciar y acabar la medida pueden aparecer artefactos haciendo que esa parte de la señal no sea representativa.

3.3 Escritura de datos

Una vez leídos todos los archivos, se almacenan en variables, añadiendo una terminación _x al nombre de cada archivo, donde x será el número que diferencie cada derivación.

La señal del ECG se guardará en un archivo .CSV, donde la primera fila corresponderá al nombre de cada archivo y derivación, separados por comas. El resto de las filas contendrán las muestras de cada una de las derivaciones de las grabaciones, también separadas por comas, como muestra la figura 16.

	A
1	1_1 0_1 1_1 2_1 3_1 4_1 5_1 6_1 7_1 8_1 9_1 10_1 11_10 0_10 1_10 2_10 3_10
2	0,-0.07342657342657342,0.019071310116086235,0.12205044751830757
3	765957447,-0.06387832699619772,-0.062135922330097085,0.04871323
4	1,-0.028554778554778556,0.07877280265339967,0.11798209926769732
5	00082,0.5259259259259259,-0.007909604519774011,-0.1505145797598

Figura 16: Archivo .csv contenedor de la base de datos Lobachevsky.

Las marcas de cada ECG se guardan en otro archivo .CSV, de manera que el primer dato de cada fila será el nombre del archivo y la derivación correspondiente. El formato del archivo CSV donde se guardan las muestras se muestra en la figura 17.

	A	B	C
1	1_0,1304, 1963, 2605, 3272, 3930		
2	1_1,1304, 1963, 2605, 3272, 3930		
3	1_2,1304, 1963, 2605, 3272, 3930		
4	1_3,1304, 1963, 2605, 3272, 3930		

Figura 17: Archivo .csv contenedor del offset de las ondas P de la base de datos Lobachevsky.

Finalmente, se creará un vector de 1s y 0s para cada onda (máscara binaria) [14]. Inicialmente, se generará un vector compuesto por 0s cuya duración coincidirá con la duración de la señal ECG. Posteriormente, se leerán los inicios y finales de cada una de las ondas, y se asignarán 1s a las componentes del vector que se encuentra entre el inicio y final de cada onda concreta. De esta forma, se obtendrán vectores para cada una de las ondas (onda P, complejo QRS, onda T), indicando las muestras en las que estas están presentes, como se muestra en la figura 18.

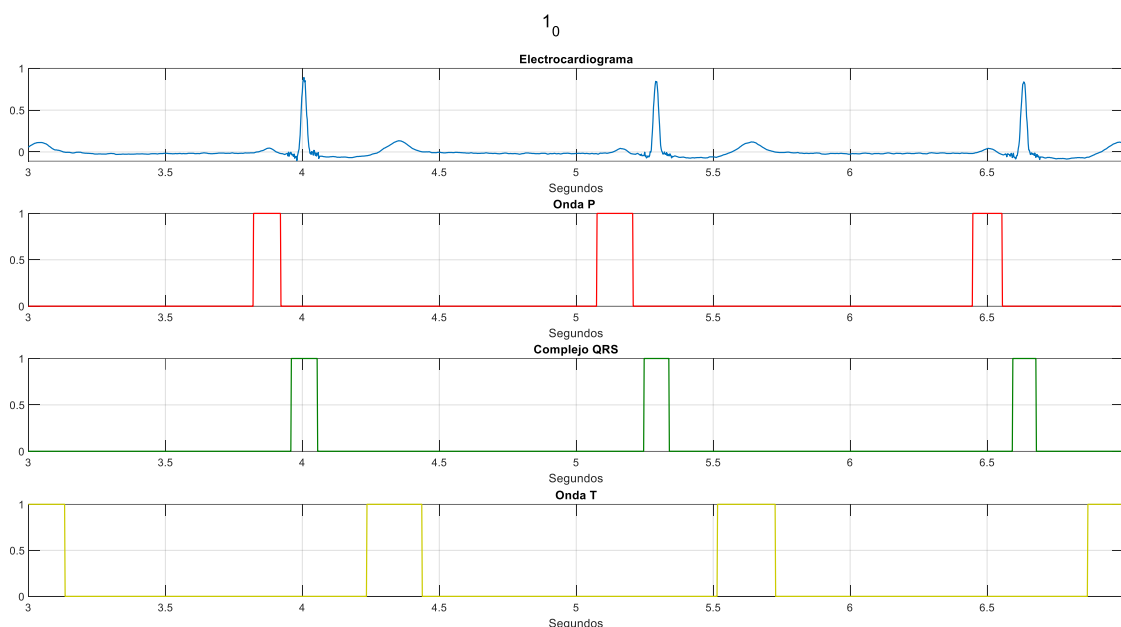


Figura 18: Ejemplo de máscaras binarias creadas a partir de las marcas de cardiólogos en la primera derivación del registro 1

Este archivo se guardará en formato .CSV, de igual forma a como se guardan las muestras de los ECGs. En la figura 19 se muestra las 5 primeras columnas (del total de 5000) de algunas grabaciones de la base de datos LUDB,

En segundo lugar, el generador de datos incorpora la función “getitem”, encargada de generar el batch de datos. Esta función lee el número de componentes del tensor de datos correspondiente al tamaño del batch. Utilizando la clave (nombre del archivo), el onset (inicio de la ventana a procesar) y el tamaño de la ventana, esta función almacena tres tensores: uno con las señales a procesar, otro con las etiquetas a comparar y el último con la máscara para aplicar en el cálculo de la pérdida.

Cabe destacar que el generador tiene una mayor complejidad de la descrita hasta ahora, ya que es capaz de procesar múltiples bases de datos simultáneamente. Adicionalmente, es necesaria una distinción entre las diferentes bases de datos; por ejemplo, al procesar datos generados con ruido de la base de datos Lobachevsky, las etiquetas a comparar son las de los archivos que no contienen ruido.

Finalmente, este generador se integra en el dataloader, encargado de enviar los datos al bucle de entrenamiento para su procesamiento.

En las figuras 20, 21 se representa el esquema de funcionamiento del dataset.

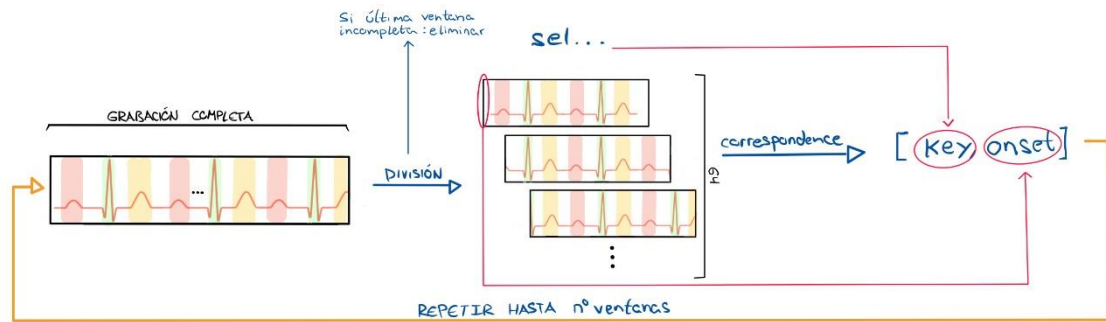


Figura 20: Esquema de funcionamiento de escritura del dataset

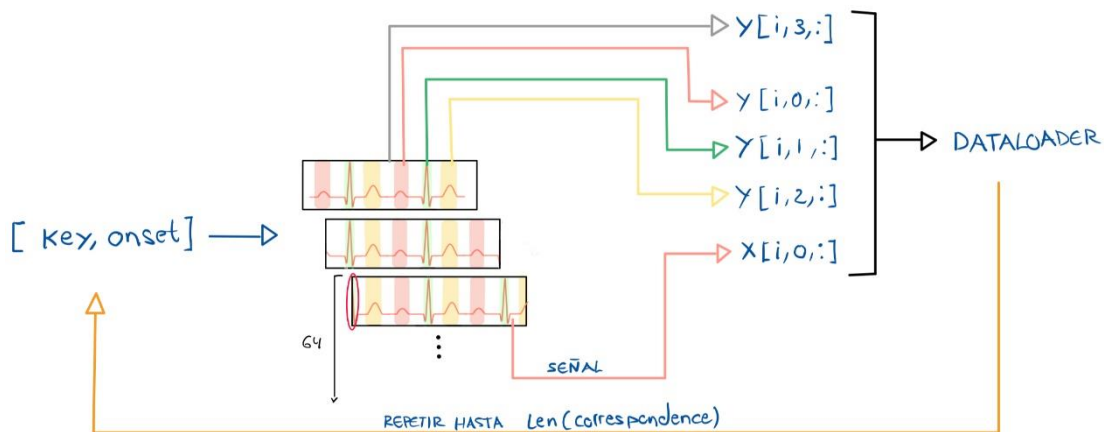


Figura 21: Esquema de funcionamiento de lectura del dataset

4.2 Robustez U-NET

Se van a añadir a continuación todas las gráficas de las medias y varianzas de las diferentes métricas del entrenamiento de la red U-NET con 4 niveles y 4 capas. Se ha ejecutado el mismo proceso 4 veces para obtener estas gráficas.

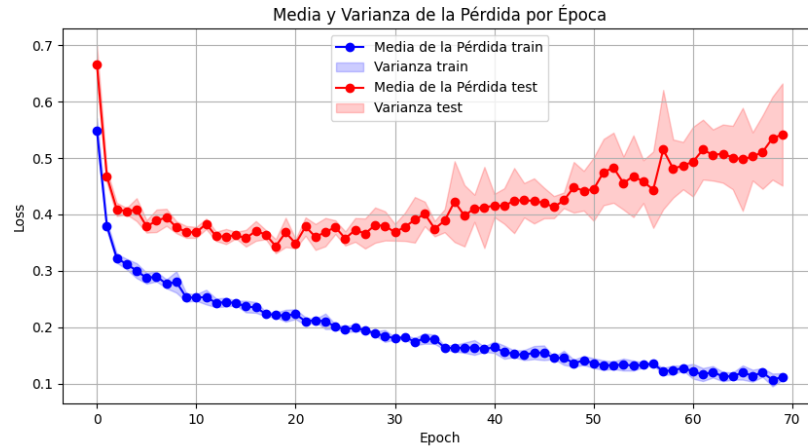


Figura 22: Media y varianza de función de pérdida (cross entropy) durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

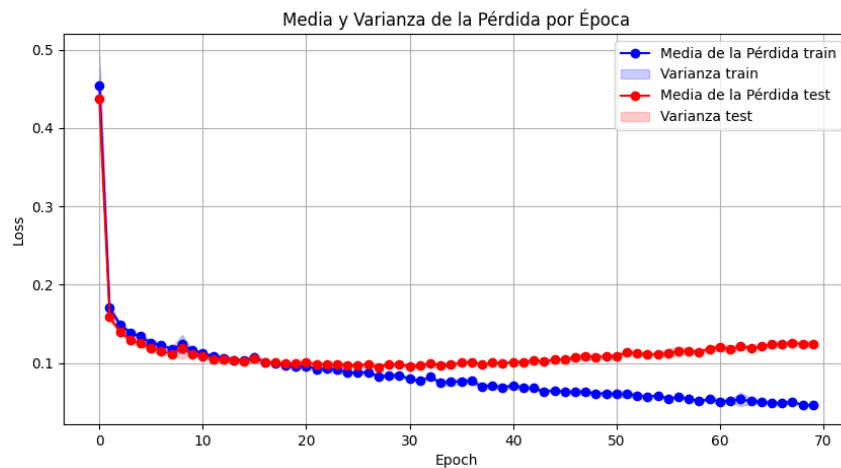


Figura 23: Media y varianza de función de pérdida (cross entropy) durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

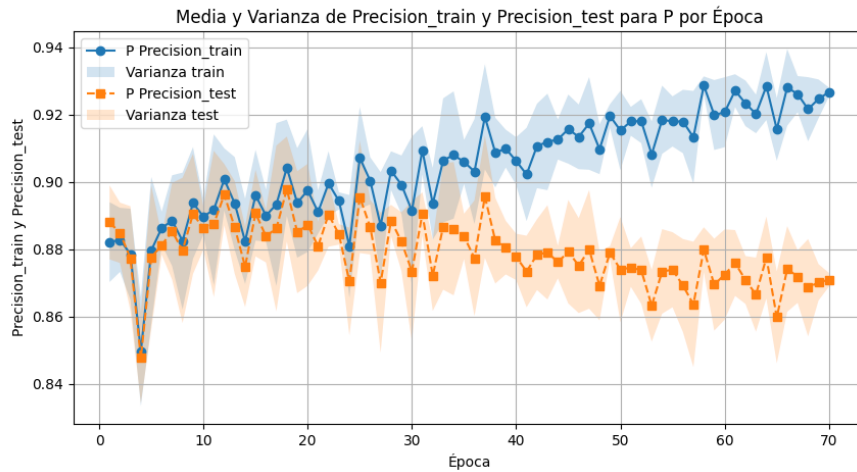


Figura 24: Media y varianza de la precisión de la onda P durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

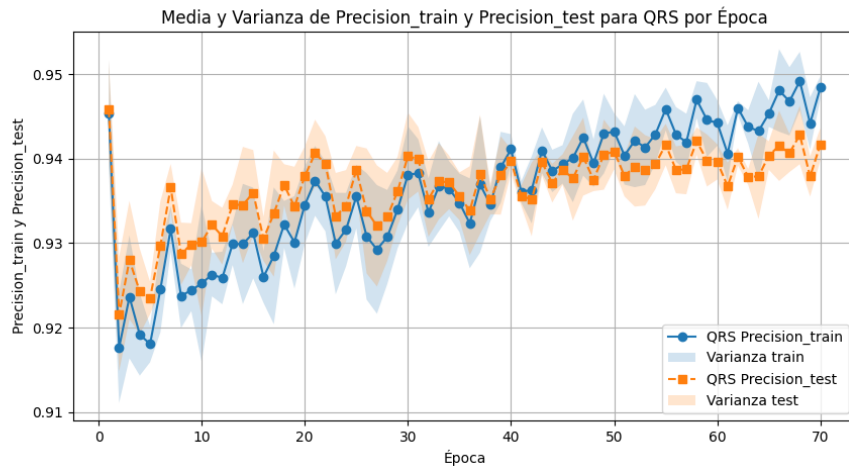


Figura 25: Media y varianza de la precisión del complejo QRS durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

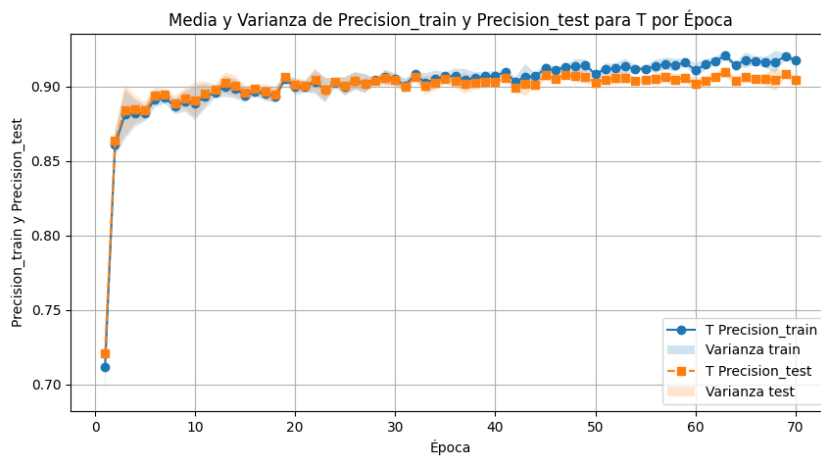


Figura 26: Media y varianza de la precisión de la onda T durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

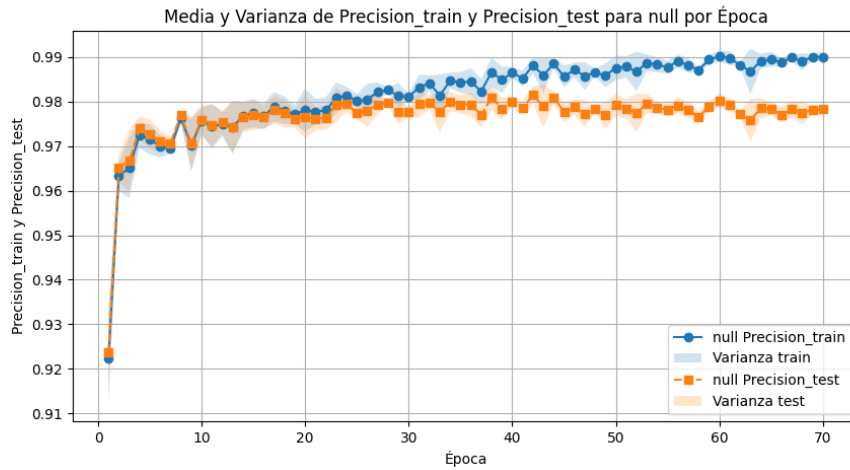


Figura 27: Media y varianza de la precisión de null durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

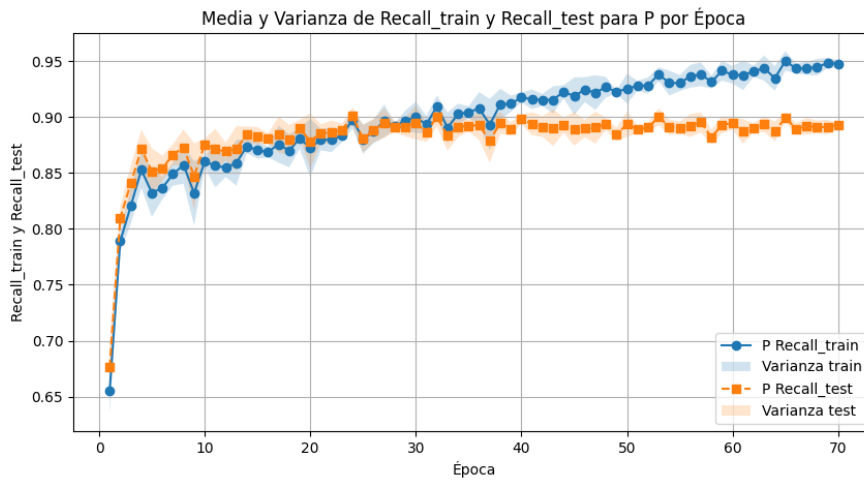


Figura 28: Media y varianza del recall de la onda P durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

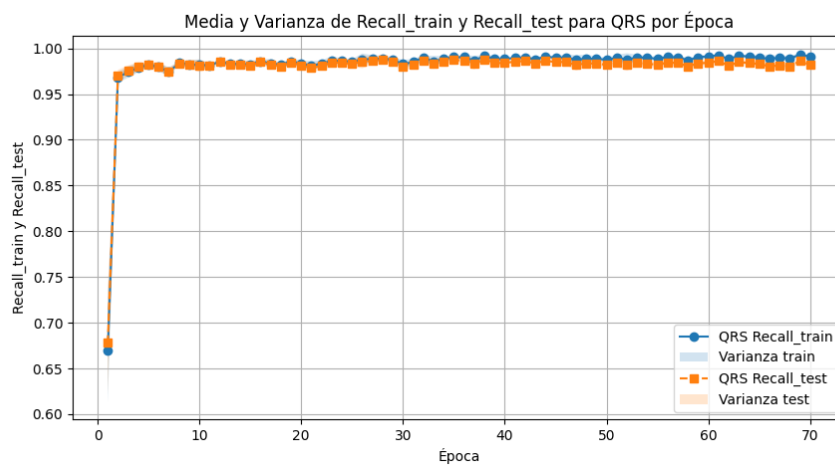


Figura 29: Media y varianza del recall del complejo QRS durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

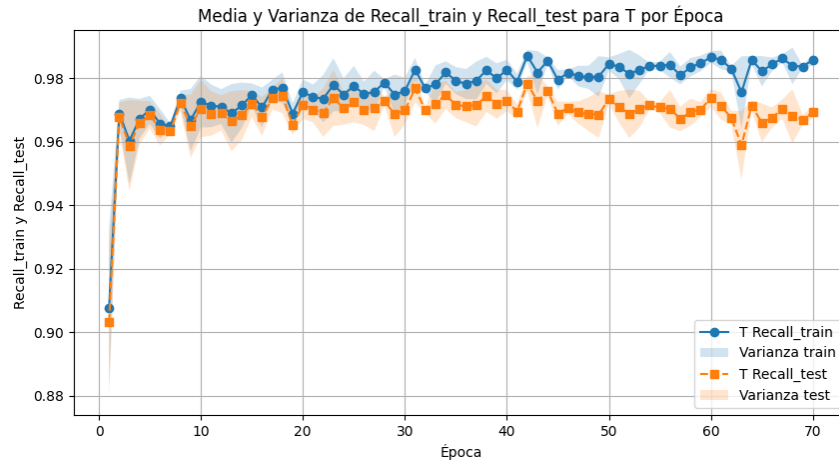


Figura 30: Media y varianza del recall de la onda T durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

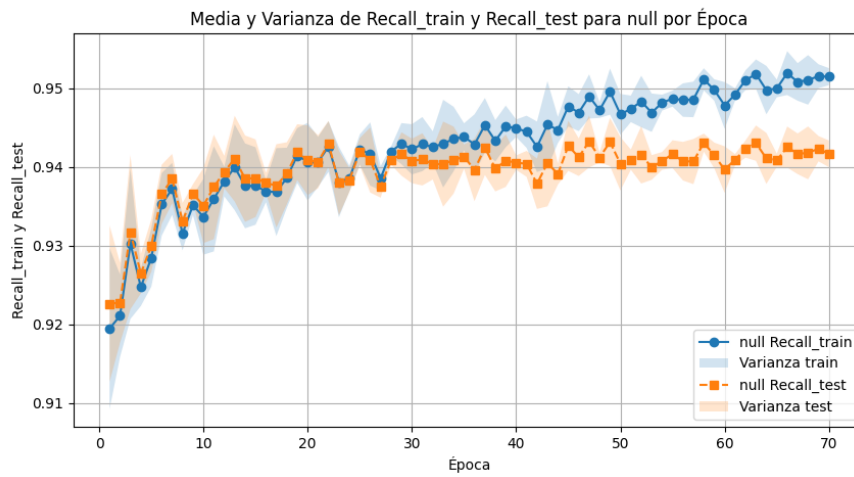


Figura 31: Media y varianza del recall de null durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de preentrenamiento.

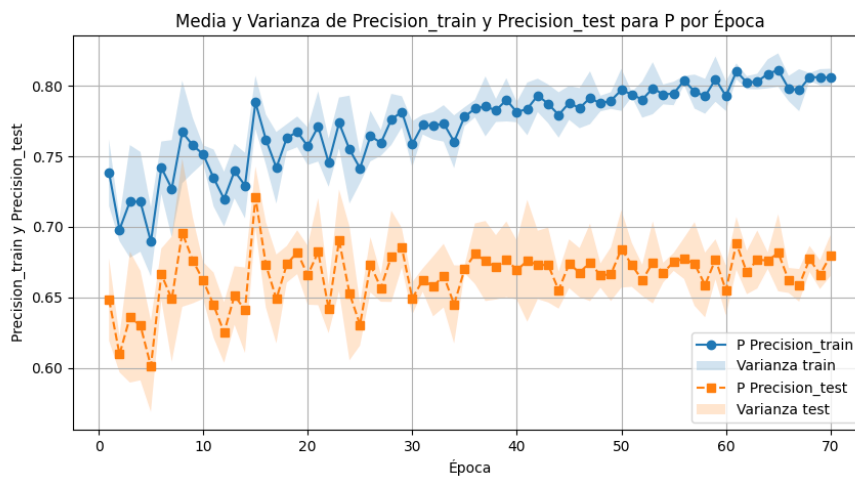


Figura 32: Media y varianza de la precisión de la onda P durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

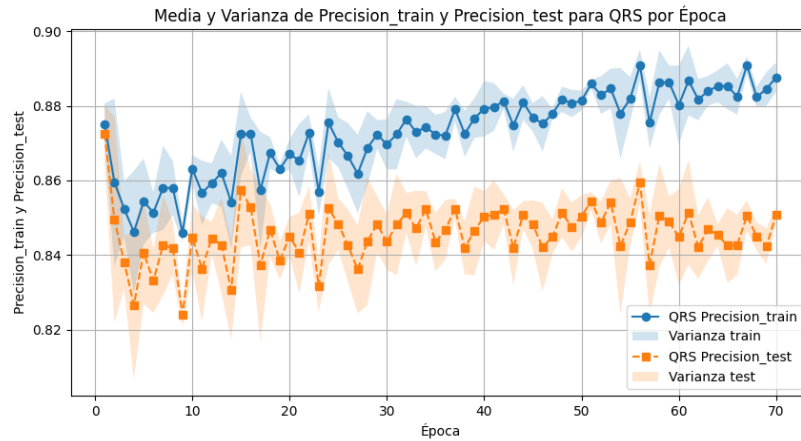


Figura 33: Media y varianza de la precisión del complejo QRS durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

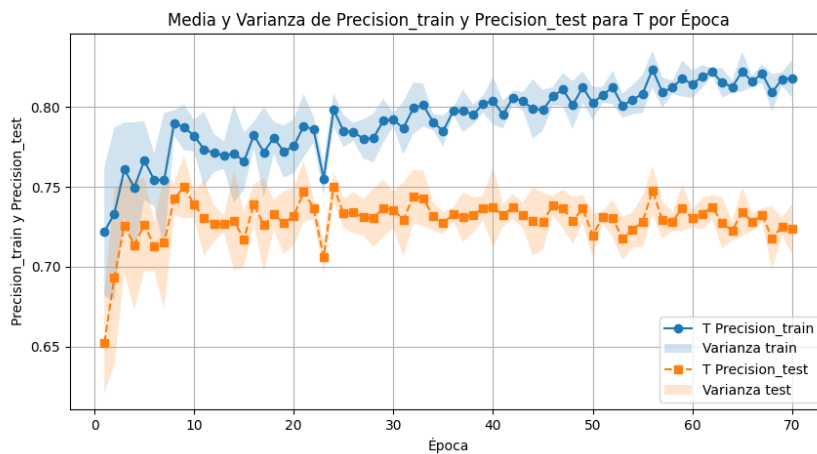


Figura 34: Media y varianza de la precisión de la onda T durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

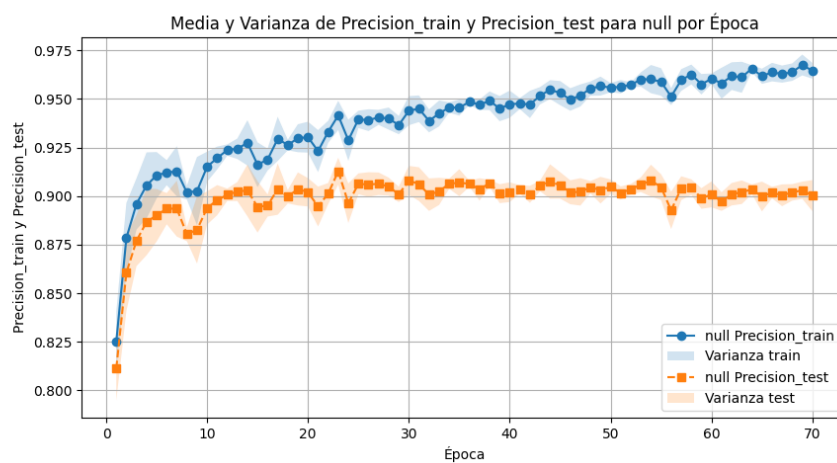


Figura 35: Media y varianza de la precisión de null durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento

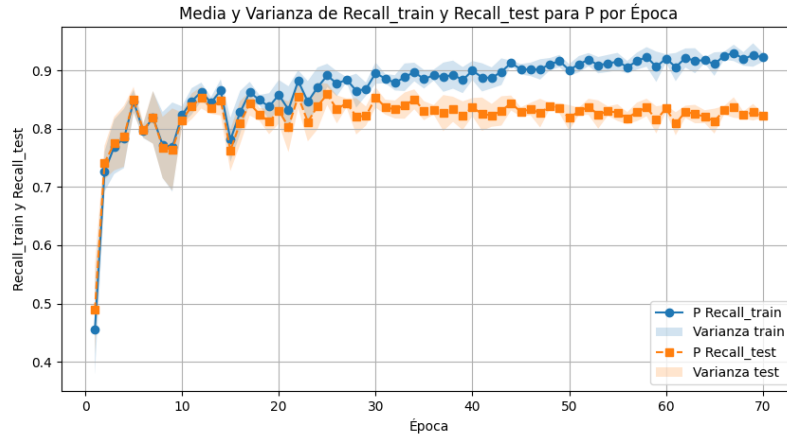


Figura 36: Media y varianza del recall de la onda P durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

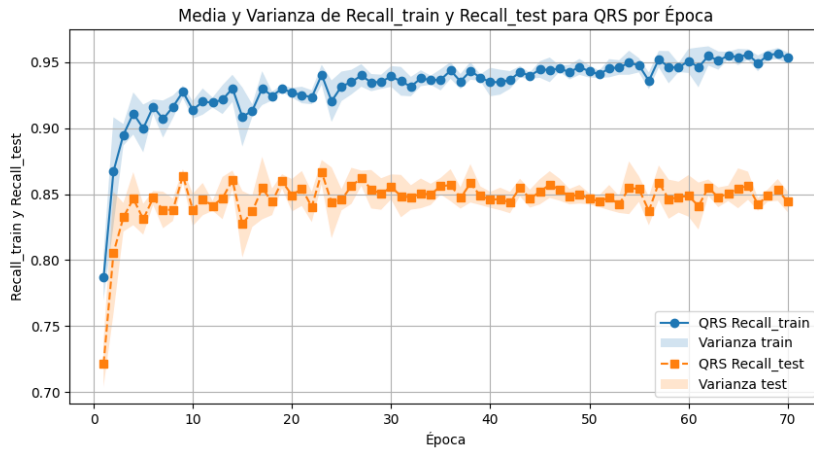


Figura 37: Media y varianza del recall del complejo QRS durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

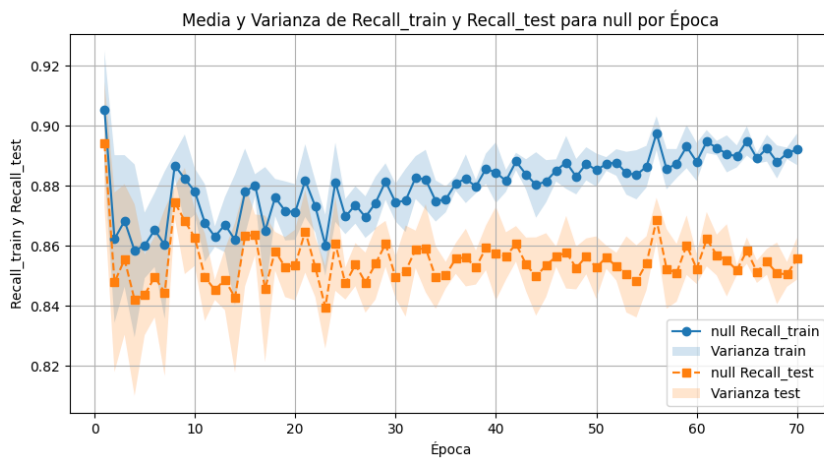


Figura 38: Media y varianza del recall de la onda T durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

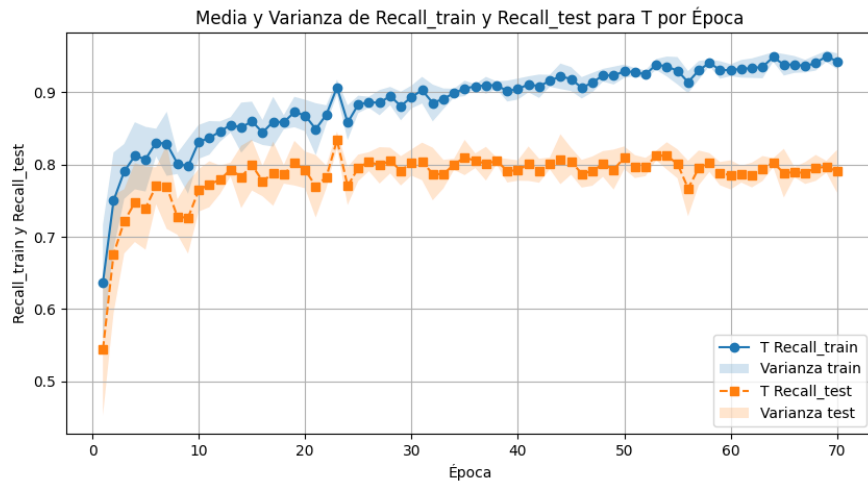


Figura 39: Media y varianza del recall de null durante el entrenamiento de la red U-NET de 4 niveles y 4 capas con datos de entrenamiento.

4.3 Niveles y capas U-NET

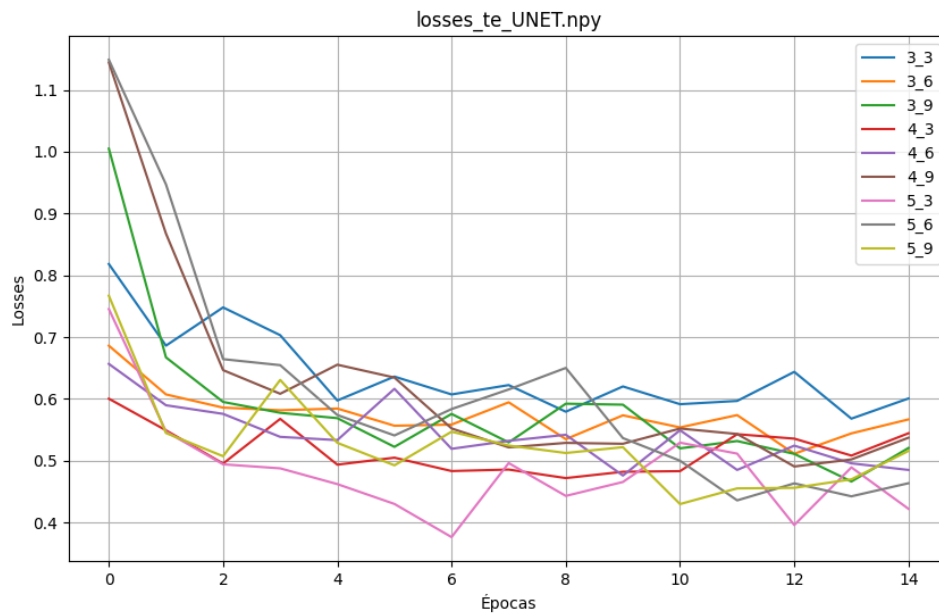


Figura 40: Función de pérdida durante el entrenamiento con las base de datos QT y Lobachevsky obtenida con los datos de entrenamiento para cada configuración.

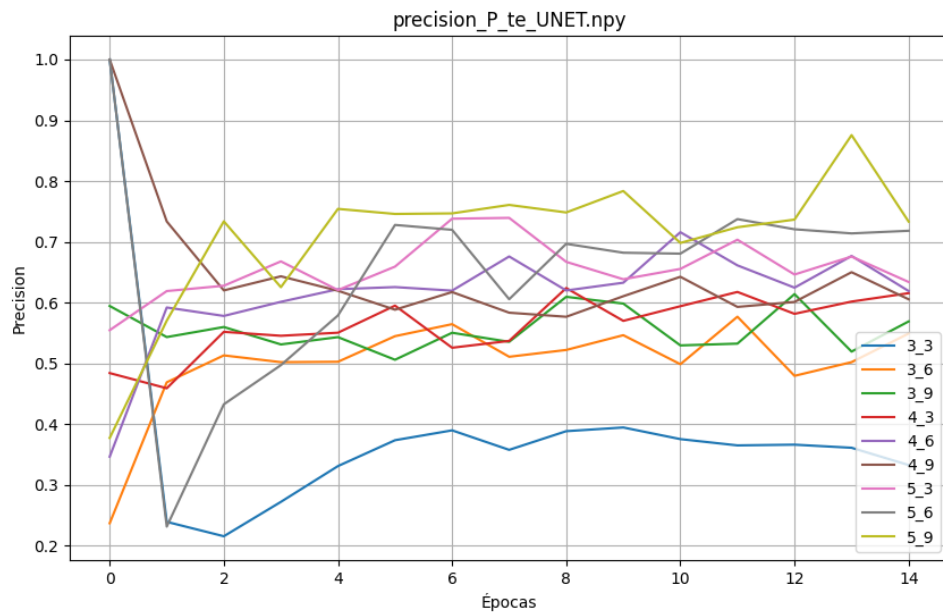


Figura 41: Precisión de la onda P durante el entrenamiento con las bases de datos QT y Lobachevsky obtenida con los datos de entrenamiento para cada configuración.

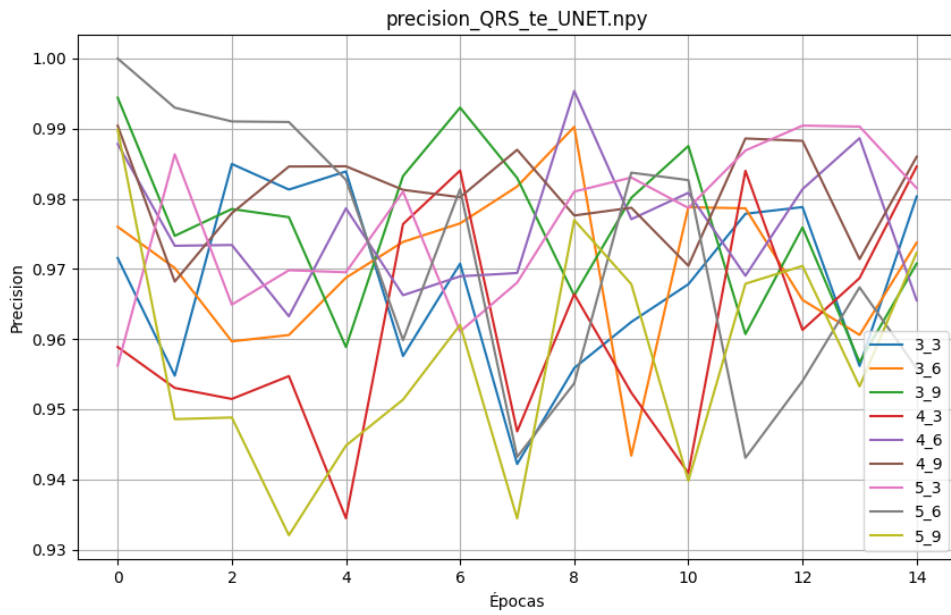


Figura 42: Precisión del complejo QRS durante el entrenamiento con las bases de datos QT y Lobachevsky obtenida con los datos de entrenamiento para cada configuración.

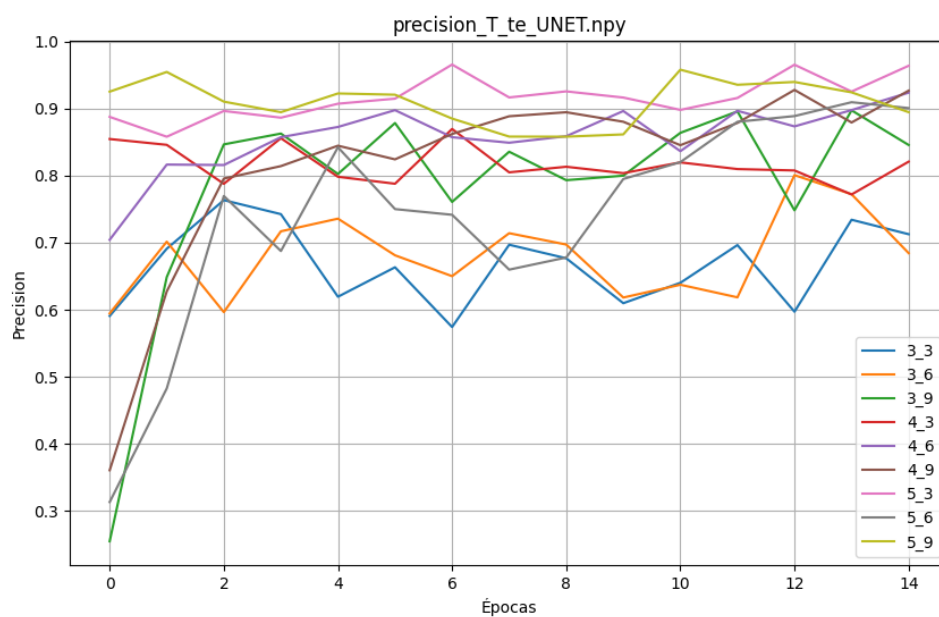


Figura 43: Precisión de la onda T durante el entrenamiento con las bases de datos QT y Lobachevsky obtenida con los datos de entrenamiento para cada configuración.

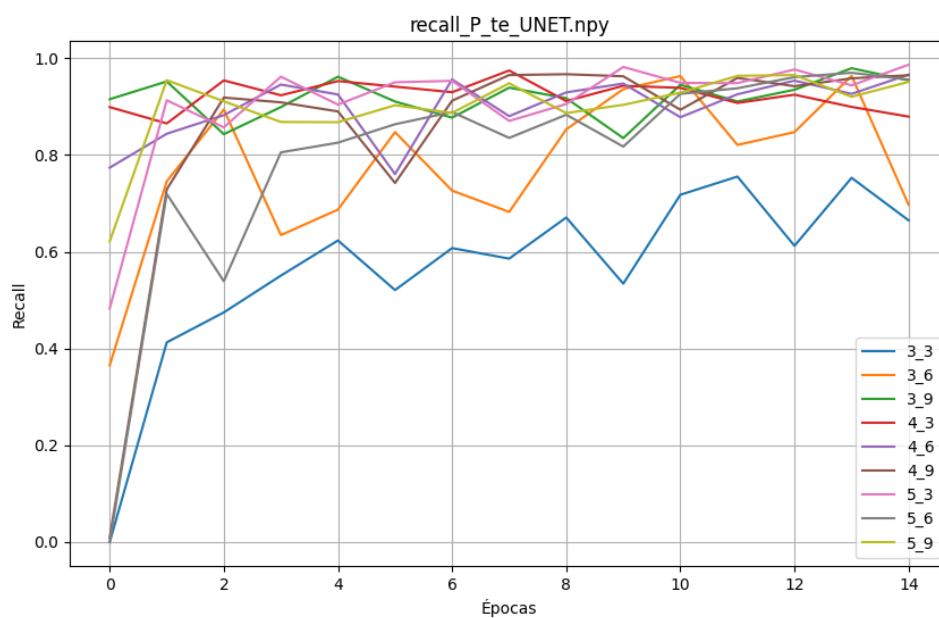


Figura 44: Recall de la onda P durante el entrenamiento con las bases de datos QT y Lobachevsky obtenida con los datos de entrenamiento para cada configuración.

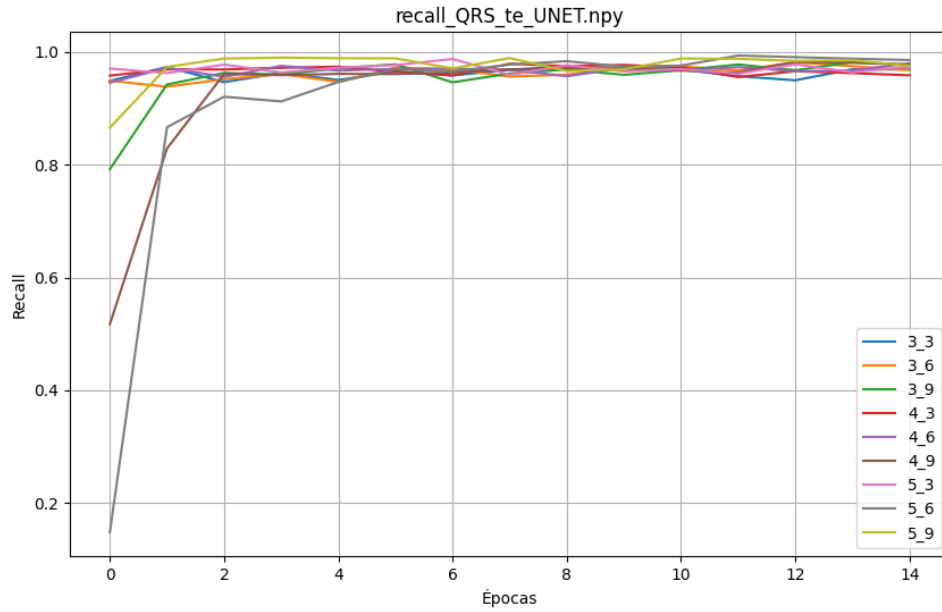


Figura 45: Recall del complejo QRS durante el entrenamiento con las bases de datos QT y Lobachevsky obtenida con los datos de entrenamiento para cada configuración.

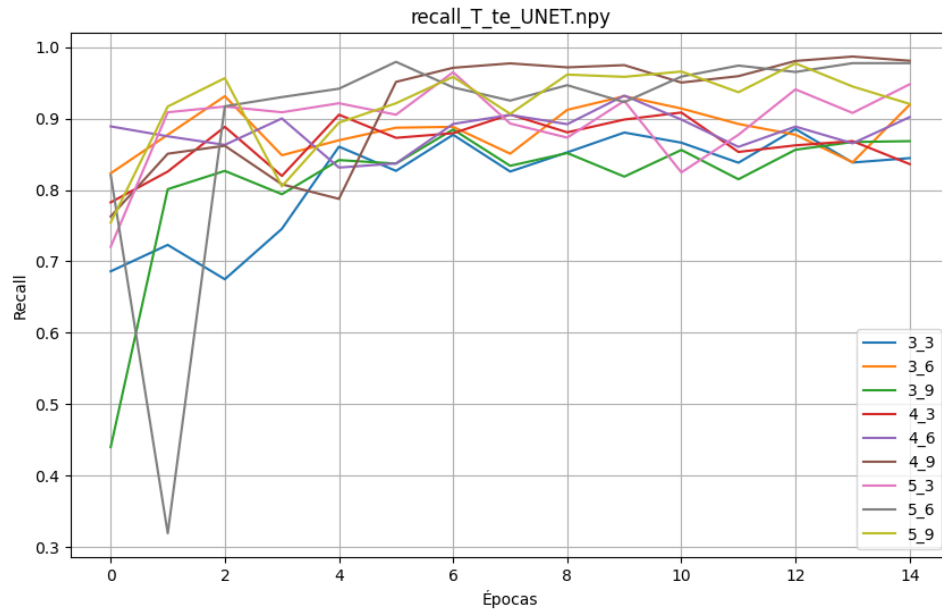


Figura 46: Recall de la onda T durante el entrenamiento con las bases de datos QT y Lobachevsky obtenida con los datos de entrenamiento para cada configuración.

4.4 Arquitectura U-NET con LSTM

La arquitectura U-NET permite detectar características y morfologías de la red con lo que es capaz de generar una respuesta adecuada para la delineación de ECGs. Sin embargo, esta red no tiene en consideración la dimensión temporal, muy importante en la delineación de la ECG por ser una señal muy redundante y periódica. Por ello se ha introducido una red de memoria de corto-largo

plazo (LSTM) que es capaz de capturar características temporales de la señal introducida, la cual ha sido añadida en el cuello de botella (entre el encoder y decoder de la red U-NET).

Para poder capturar información temporal, se modifican las células de una red RNN añadiendo un bloque de memoria temporal que es controlado por la entrada y salida.

Una unidad LSTM (Long Short-Term Memory) es una extensión sofisticada de una unidad estándar de red neuronal, diseñada para superar las limitaciones inherentes en el procesamiento de secuencias temporales. Mientras que una unidad estándar de red neuronal consiste únicamente en una función de activación aplicada a la entrada, una unidad LSTM introduce varios pasos adicionales que le permiten manejar dependencias temporales de manera más efectiva.

En la figura 47, el círculo amarillo representa la función de activación, que generalmente es una tangente hiperbólica. Esta función de activación se aplica a la entrada a_i , y el resultado se multiplica por un factor b_i . A continuación, se añade el valor de activación interno del paso de tiempo anterior, multiplicado por un factor b_ϕ , a través de una conexión recurrente. Estos factores $((b_i, b_\phi, b_\omega \in (0,1)))$ son controlados por unidades adicionales, representadas en azul en la figura.

Estas unidades adicionales son las compuertas de entrada, salida y olvido. Las compuertas suman las activaciones de la capa anterior con las activaciones de la capa actual del instante de tiempo anterior, así como la propia activación interna de la célula LSTM. Posteriormente, el valor resultante se comprime mediante una función sigmoide, regulando así la información que se almacena y se transmite a través del tiempo.

Esta estructura de compuertas y conexiones recurrentes permite a la unidad LSTM almacenar y actualizar información de instantes de tiempo anteriores, capturando de manera efectiva las relaciones temporales presentes en la entrada. En esencia, la LSTM actúa como una memoria diferenciable, similar a la memoria de un computador, lo que le confiere la capacidad de mantener y utilizar información relevante a lo largo de secuencias extensas. Esta característica es fundamental para tareas que requieren el procesamiento de datos secuenciales, como la delineación de ondas en electrocardiogramas, procesamiento de lenguaje natural y reconocimiento de voz.

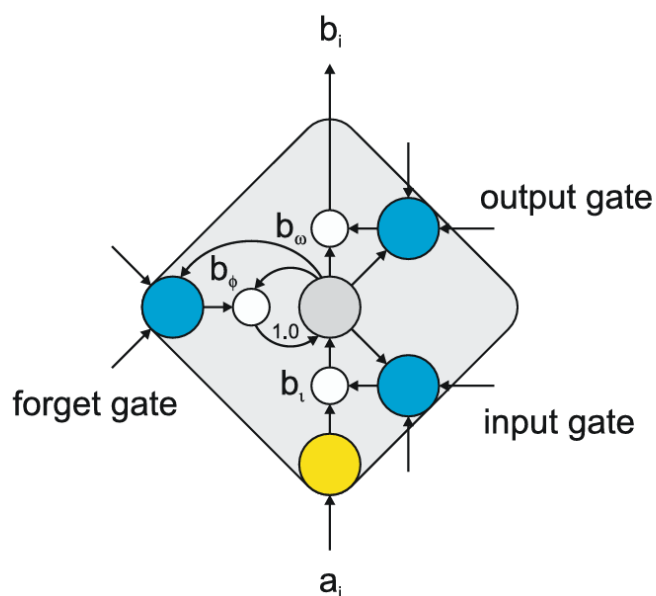


Figura 47 Esquema de una célula LSTM. Disponible en: [lstm-neural-networks-for-language-modeling-1aw3c4nqzw.pdf \(typeset.io\)](https://arxiv.org/pdf/1610.10166v1.pdf)

De forma simplificada, la compuerta de entrada es la encargada de controlar la información nueva que se almacena en la célula, la compuerta de olvido determina la información que se descarta de la memoria de la célula (el bucle recurrente es el encargado de guardar el estado (información) en la propia célula) y la compuerta de salida decide la información que se propaga [15].

4.5 Arquitectura U-NET con conexiones residuales

Las conexiones residuales, también conocidas como "skip connections" o "shortcuts," son una técnica clave en la arquitectura de redes neuronales profundas, introducida con el propósito de mitigar problemas asociados con el entrenamiento de redes muy profundas, tales como la desaparición del gradiente y la degradación del rendimiento.

En términos conceptuales, una conexión residual es un atajo que permite que la información salte uno o más niveles de procesamiento en la red. En lugar de que los datos pasen secuencialmente a través de cada capa, una conexión residual agrega la entrada original de una capa a su salida, lo que se puede formalizar como $y = F(x, \{W_i\})$ donde x es la entrada, $F(x, \{W_i\})$ representa la función de la capa (o capas) con parámetros $\{W_i\}$, y y es la salida final. Este esquema, representado en la figura 48, introduce un camino directo para el flujo de la información y los gradientes, facilitando el aprendizaje de funciones de identidad y simplificando la optimización [16].

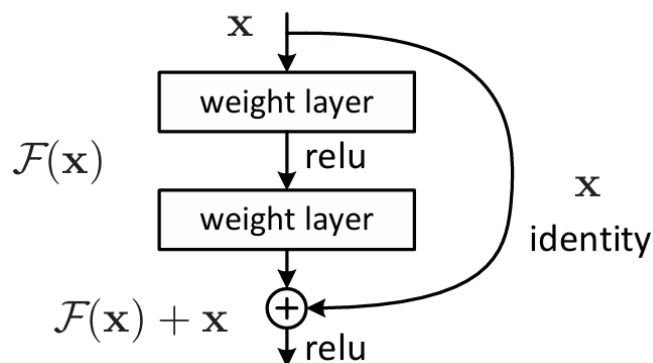


Figura 48: Esquema de una conexión residual. Disponible en:

<https://doi.org/10.48550/arXiv.1512.03385>

Las ventajas de las conexiones residuales son multifacéticas. En primer lugar, facilitan el entrenamiento de redes muy profundas al prevenir la desaparición del gradiente, un fenómeno donde los gradientes se vuelven extremadamente pequeños en las capas inferiores durante la retropropagación (back propagation), dificultando el ajuste de los parámetros. Las conexiones residuales, al proporcionar caminos adicionales para el flujo de los gradientes, aseguran que incluso las capas más profundas reciban señales significativas durante el entrenamiento [16].

Además, las conexiones residuales ayudan a resolver el problema de la degradación del rendimiento. Al aumentar la profundidad de las redes, es posible que el rendimiento se estanque o incluso disminuya después de cierto punto. Las conexiones residuales permiten que las capas aprendan funciones de identidad de manera más efectiva, asegurando que la adición de nuevas capas no empeore el rendimiento del modelo [16].

4.6 Arquitectura U-NET con mecanismos de atención

Los mecanismos de atención en redes neuronales son técnicas que permiten a los modelos de aprendizaje automático centrarse en las partes más relevantes de los datos de entrada cuando realizan una tarea específica. Estos mecanismos han demostrado ser especialmente efectivos en tareas como la traducción automática, la generación de texto, el reconocimiento de voz y el procesamiento de imágenes, entre otras.

En su esencia, los mecanismos de atención permiten que una red neuronal asigne diferentes niveles de importancia a diferentes partes de la entrada. En el contexto del análisis de ECG, los mecanismos de atención permiten a una red neuronal identificar y enfocarse en las partes más importantes de la señal de ECG en diferentes momentos, mejorando la precisión de la delineación de las ondas y segmentos. Esto es especialmente útil dado que la señal de ECG puede contener ruido y variaciones que no son relevantes para la tarea de delineación.

Estos mecanismos se utilizan habitualmente en modelos de encoder-decoder, como es el caso de la red U-NET de este trabajo. Concretamente, estos mecanismos de atención toman los estados ocultos del encoder h_i y la salida anterior del decoder s_{t-1} para calcular una puntuación que indica qué tan bien se alinean los elementos de la secuencia de entrada con la salida actual en la posición t [17].

$$e_{t,i} = a(s_{t-1}, h_i) \quad (7)$$

Los pesos se calculan utilizando una función de activación softmax.

$$\alpha_{t,i} = \text{softmax}(e_{t,i}) \quad (8)$$

Estos pesos son los que identifican las secciones de los datos que tienen mayor relevancia. Por ello, el decoder se alimenta mediante un vector único llamado vector de contexto, calculado mediante la suma ponderada de todos los estados ocultos del codificador h :

$$c_t = \sum_{i=1}^T \alpha_{t,i} h_i \quad (9)$$

ANEXO V: Evaluación

5.1 Postprocesado

La red propuesta en este estudio es una red U-NET de una única derivación (single-lead). Aunque se dispone de una base de datos con 12 derivaciones, como es el caso de la base de datos Lobachebsky, la red procesa cada derivación de manera individual, por lo que no es capaz de aprender las relaciones entre las distintas derivaciones. Esta limitación representa una desventaja en la evaluación de la red, dado que los cardiólogos delinearon esta base de datos considerando las 12 derivaciones de manera simultánea. Como consecuencia, puede ocurrir que en determinadas secciones de una grabación, una derivación no detecte la onda P a pesar de su presencia, algo que el cardiólogo identifica y marca como onda P. No obstante, la red empleada en este estudio no tiene la capacidad de capturar estas relaciones interderivacionales.

Para abordar esta problemática, se ha desarrollado una función en Matlab que analiza cada derivación de la grabación de forma individual, onda por onda. Si se identifica una zona de la señal sin presencia de ondas en una derivación específica, pero con ondas presentes en las demás derivaciones, la función promedia los bordes de la onda detectada en las otras derivaciones y genera la onda correspondiente en la derivación faltante. De manera similar, si una onda se detecta

únicamente en una derivación específica, la función elimina dicha onda. El umbral establecido es de 6 derivaciones: si más de 6 derivaciones presentan una onda en una zona concreta de la señal, esta onda se establecerá en todas las derivaciones; de lo contrario, se procederá a eliminarla.

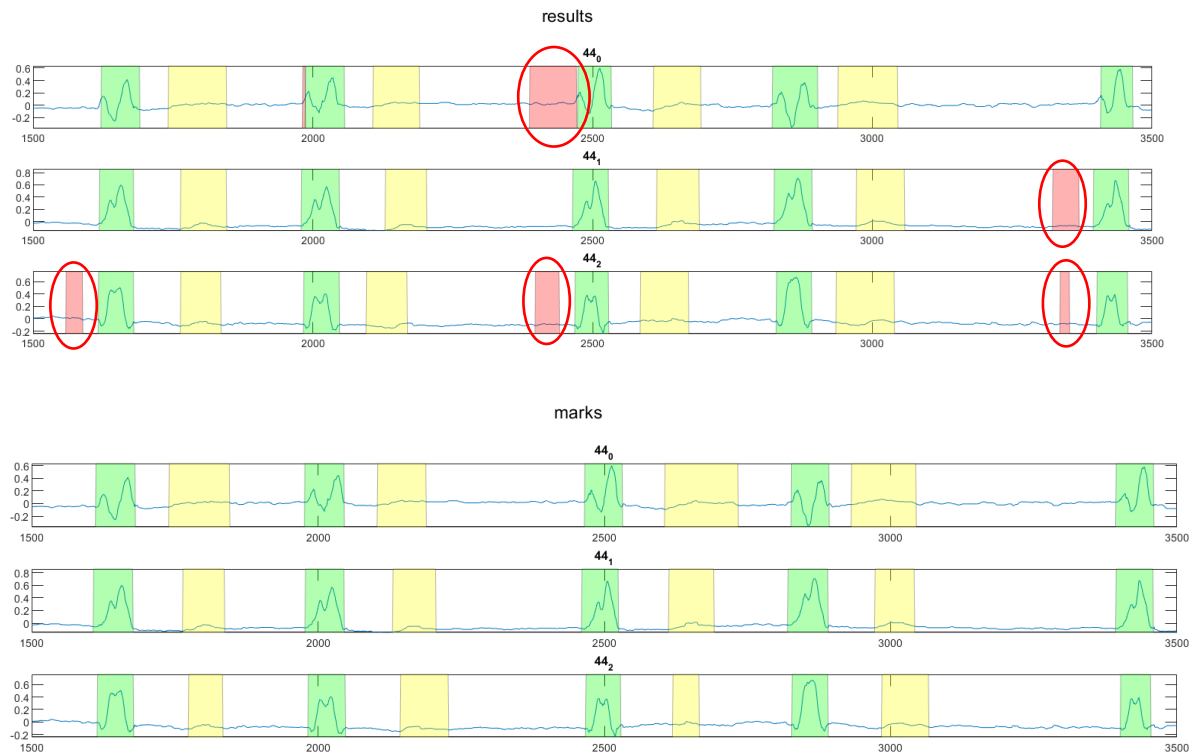


Figura 49, 50: Ejemplo de falso positivo en la delineación de un ECG, marcas y resultados de la red neuronal.

En las Figuras 49 y 50 se observa un ejemplo en el que la red neuronal ha detectado ondas P que no están presentes en las anotaciones realizadas por los cardiólogos. Aunque solo se han representado tres derivaciones, existen doce derivaciones en total, la mayoría de las cuales no presentan ondas P. Al aplicar el postprocesamiento previamente descrito, se obtienen los resultados mostrados en la Figura 51.

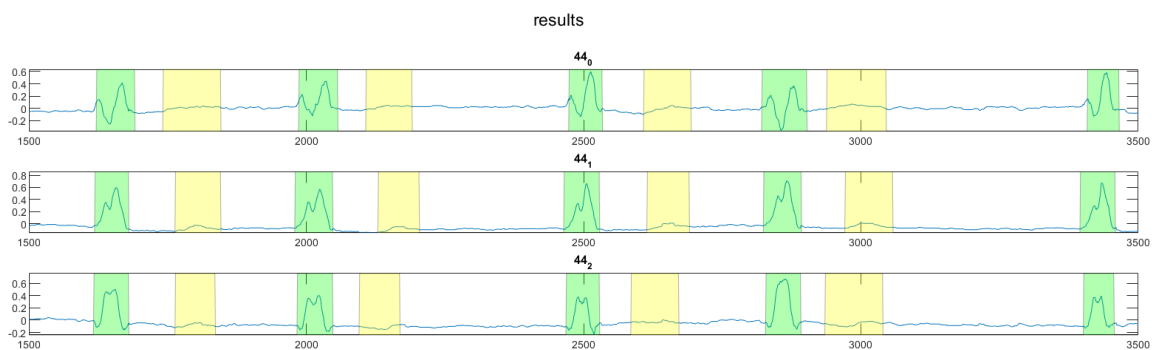


Figura 51: Resultados obtenidos tras aplicar postprocesado

Se han eliminado las ondas P teniendo en cuenta el resto de derivaciones, provocando una mejora de los resultados del delineador.

En las figuras 52 y 53 se encuentra la situación contraria.

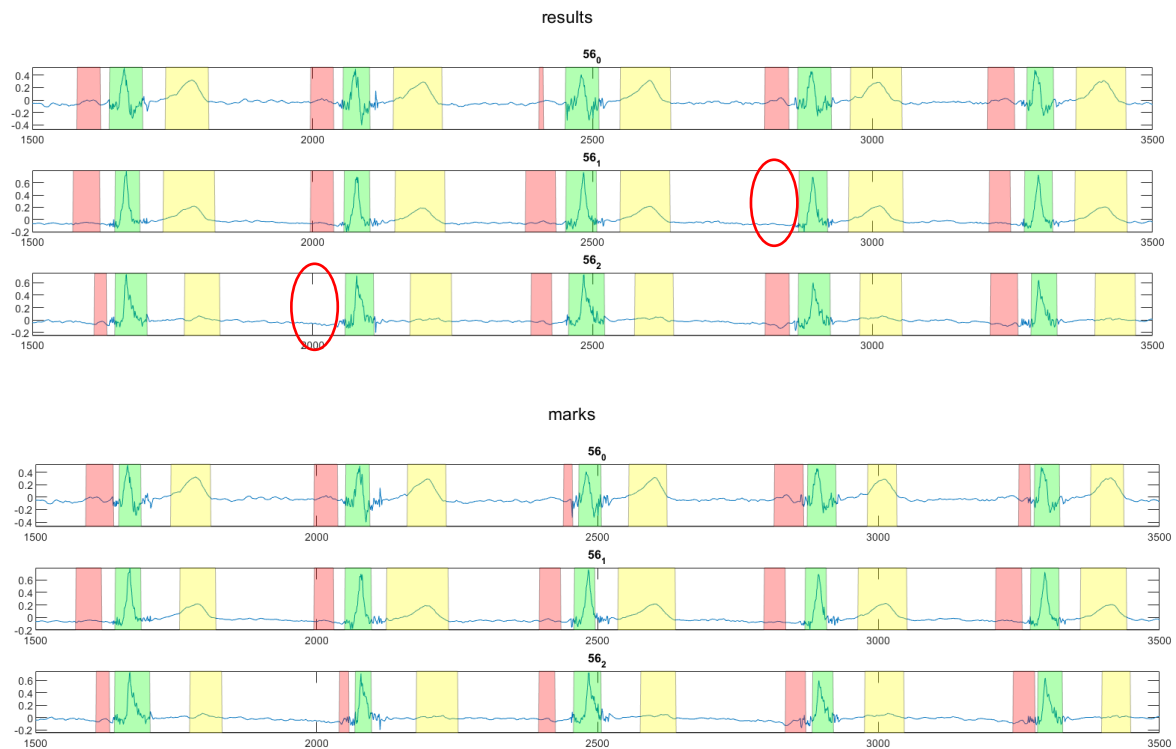


Figura 52, 53: Ejemplo de falso negativo en la delineación de un ECG, marcas y resultados de la red neuronal.

Aplicando el postprocesado se obtiene, como se observa en la figura 54, una delineación más acertada, a pesar de no ser ideal por no coincidir exactamente con los bordes de las marcas de esa delineación.

La única solución posible sería crear la misma red con 12 entradas, una para cada derivación del ECG, y cuyas salidas fueran las ondas P, T, el complejo QRS y null que abarcan todas las derivaciones (multi-lead).

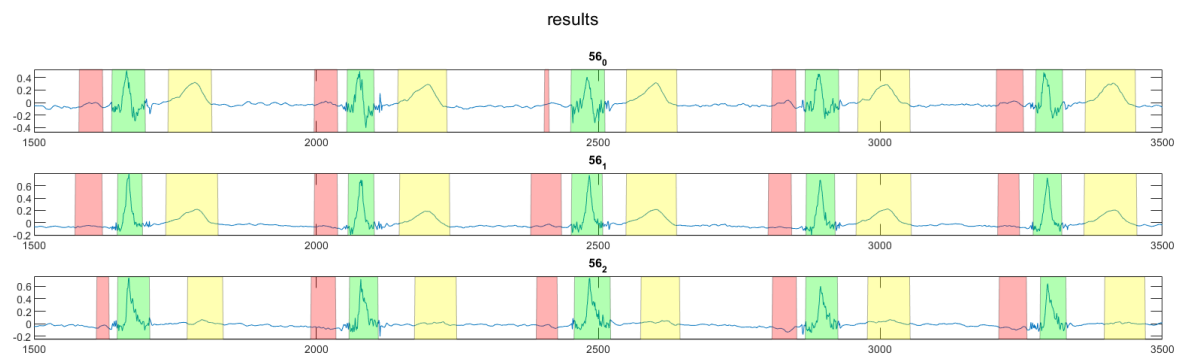


Figura 54: Resultados obtenidos tras aplicar postprocesado

5.2 Evaluación single-lead y multilead

En este estudio se han planteado dos estrategias para calcular las métricas en la etapa de evaluación. La primera estrategia, denominada evaluación "single-lead", consiste en calcular las métricas de cada derivación individualmente para luego obtener un promedio de estas. Este método es particularmente adecuado dado que la red utilizada en este trabajo está diseñada para operar sobre una única derivación.

La segunda estrategia, denominada evaluación multilead, implica el cálculo de métricas a partir de una única derivación compuesta que considera las 12 derivaciones de la señal ECG, proceso que se denominará delineación multilead con reglas. Este método se basa en transformar las anotaciones de los cardiólogos en máscaras binarias, tras lo cual se procede a la integración de las 12 derivaciones (unión). Esta integración proporciona una representación precisa del inicio y el final de una onda en toda su extensión, considerando las variaciones en el tiempo de propagación del impulso eléctrico entre las distintas derivaciones.

Para las ondas detectadas por la red neuronal, el procesamiento se realiza de manera diferente. Dado que no se puede asegurar la detección correcta de las ondas en las 12 derivaciones, se sigue el siguiente procedimiento: para cada onda, se verifica si al menos otras cinco derivaciones presentan inicios y finales dentro de un rango predefinido a partir de los inicios y finales de la onda analizada. Si esta condición se cumple, la onda se considera válida para la integración; de lo contrario, se descarta. Este proceso se repite para todas las derivaciones, eliminando aquellas ondas con inicios y finales atípicos.

Finalmente, se realiza la unión de las ondas válidas, obteniendo así una derivación única para la evaluación. Este enfoque asegura que solo se consideren las ondas que cumplen con criterios estrictos de coherencia y validez, garantizando la fiabilidad y precisión de los resultados obtenidos en la delineación del ECG.

Realmente, al realizar un electrocardiograma, la señal resultante es una única derivación que surge como combinación de todas ellas, por lo que el método gold-standard es un método de evaluación acorde con la realidad. Sin embargo, como ya se ha mencionado anteriormente, se está estudiando una red single-lead con lo cual este método de evaluación no es el más adecuado.

Bibliografía

- [1] Maggio, Ana & Bonomini, María & Laciari, Eric & Arini, Pedro. (2012). Quantification of Ventricular Repolarization Dispersion Using Digital Processing of the Surface ECG. 10.5772/23050.
- [2] Chatterjee, S., Thakur, R.S., Yadav, R.N., Gupta, L. and Raghuvanshi, D.K. (2020), Review of noise removal techniques in ECG signals. IET Signal Process., 14: 569-590. <https://doi.org/10.1049/iet-spr.2020.0104>
- [3] Webster J. (1992). Medical Instrumentation. Application and Design, J. Webster (Editor), 2nd Edition, Houghton Mifflin Co., ISBN 0-395-59492-8, Boston, USA
- [4] Sörnmo L. & Laguna P. (2005). Bioelectrical Signal Processing in Cardiac and Neurological Applications, Elsevier Academic Press, ISBN 0-12-437552-9, Amsterdam, The Netherlands
- [5] Montesinos López OA, Montesinos López A, Crossa J. Multivariate Statistical Machine Learning Methods for Genomic Prediction [Internet]. Cham (CH): Springer; 2022. Chapter 10, Fundamentals of Artificial Neural Networks and Deep Learning. 2022 Jan 14. Available from: <https://www.ncbi.nlm.nih.gov/books/NBK583971/> doi: 10.1007/978-3-030-89010-0_10
- [6] "What is the ReLU Activation Function?," *Builtin*, <https://builtin.com/machine-learning/relu-activation-function>. [Accessed: Jun. 8, 2024]
- [7] Danyang, Zhu & Lu, Siyuan & Wang, Meiqi & Lin, Jun & Wang, Zhongfeng. (2020). Efficient Precision-Adjustable Architecture for Softmax Function in Deep Learning. IEEE Transactions on Circuits and Systems II: Express Briefs. PP. 1-1. 10.1109/TCSII.2020.3002564.
- [8] T. Gonsalves and J. Upadhyay, "Integrated deep learning for self-driving robotic cars," in *Artificial Intelligence for Future Generation Robotics*, R. N. Shaw, A. Ghosh, V. E. Balas, and M. Bianchini, Eds. Elsevier, 2021, pp. 93-118. doi: 10.1016/B978-0-323-85498-6.00010-1.
- [9] Srivastava Nitish, Hinton Geoffrey E., Krizhevsky Alex, Sutskever Ilya, and Salakhutdinov Ruslan. 2014. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 15, 1 (2014), 1929–1958.
- [10] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: theoretical analysis and applications," *Proc. 40th Int. Conf. Machine Learning (ICML'23)*, vol. 202, JMLR.org, Art. no. 992, pp. 23803–23828, 2023.
- [11] MedlinePlus, "Niveles de potasio en la sangre," [Online]. Available: <https://medlineplus.gov/spanish/ency/article/001101.htm>. [Accessed: Jun. 10, 2024].
- [12] << Bigeminy: Causes, Symptoms & Treatment (clevelandclinic.org)>> Available <https://my.clevelandclinic.org/health/diseases/24576-bigeminy>. [Accessed: Jul. 9, 2024]
- [13] << Ectopic Heartbeat: Symptoms, Causes & What It Means for Your Health (clevelandclinic.org)>> Available: <https://my.clevelandclinic.org/health/diseases/23000-ectopic-heartbeat> [Accessed: Jun. 9, 2024]
- [14] Jimenez-Perez G, Alcaine A, Camara O. Delineation of the electrocardiogram with a mixed-quality-annotations dataset using convolutional neural networks. Sci Rep. 2021

Jan 13;11(1):863. doi: 10.1038/s41598-020-79512-7. PMID: 33441632; PMCID: PMC7806759.

- [15] Sundermeyer, M., Schlüter, R., Ney, H. (2012) LSTM neural networks for language modeling. Proc. Interspeech 2012, 194-197, doi: 10.21437/Interspeech.2012-65
- [16] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.48550/arXiv.1512.03385>
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 6000-6010.
- [18] S. Kumar, "The Cardiac Cycle," CVIL at UCSD. [Online]. Disponible en: <https://cvil.ucsd.edu/wp-content/uploads/2017/02/cardiac-cycle.pdf>. [Accessed: Jun. 8, 2024]