

PROYECTO FIN DE GRADO

TÍTULO: CARACTERIZACIÓN DE CRISIS EPILÉPTICAS A PARTIR DE EEG USANDO AUTO-CODIFICADORES VARIACIONALES

AUTOR/A: CLAUDIA BRAVO RUIZ

TITULACIÓN: GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN

TUTOR/A: DAVID LUENGO GARCÍA

DEPARTAMENTO: INGENIERÍA AUDIOVISUAL Y COMUNICACIONES

Miembros del Tribunal Calificador:

PRESIDENTE/A: SANTIAGO HIGUERA

TUTOR/A: DAVID LUENGO

SECRETARIO/A: FRANCISCO CANO

Fecha de lectura: 23 de julio de 2024

Calificación:

VºBº TUTOR/A

El Secretario/La Secretaria,

En primer lugar, me gustaría poder dedicar este proyecto a mi familia, en especial a mis padres, quienes me dieron la oportunidad de estudiar esta carrera y me han animado durante todos estos años a continuar.

También, gracias a Mario, quien ha estado en lo bueno y en lo malo, en los momentos de estrés y de bajón, apoyándome y escuchándome y, en los momentos de alegría y celebración.

Gracias a mis compañeros y amigos, por haber formado un gran equipo en los momentos de estudio, resolviendo dudas entre todos, ayudándonos y avanzando juntos.

Y por último, gracias a David Luengo, por haberme acompañado en la realización de este proyecto con tanta delicadeza y paciencia.

Resumen

La forma de pensar y de vivir de las personas ha ido cambiando a lo largo de la historia debido a los avances tecnológicos. Estas innovaciones han permitido un gran desarrollo en el campo de la medicina, entre otros. El desarrollo de nuevos softwares y herramientas matemáticas, contribuyen a mejorar el diagnóstico que realizan los médicos a la hora de detectar una enfermedad.

En este proyecto se aborda la epilepsia, una enfermedad que afecta a millones de pacientes en el mundo, con nuevos casos cada día, y afectando a la calidad de vida de las personas que la padecen. La presentación de esta enfermedad se centra en los conceptos más básicos, pero suficientes para poder continuar con el estudio de señales del encefalograma a través de algoritmos de *Deep Learning* para su caracterización. Este estudio se ha realizado teniendo en cuenta consideraciones éticas, con datos procedentes de una base de datos pública y anónima, manteniendo su confidencialidad.

Destacando el *Deep Learning* como uno de los grandes avances tecnológicos de los últimos años, éste ha generado gran interés debido a su capacidad para detectar enfermedades causadas por ritmos cerebrales anómalos o patologías a través de redes neuronales. El proyecto está centrado en el desarrollo de un modelo de red neuronal, un autocodificador variacional para la detección de crisis epilépticas. Por ello, el proyecto incluye tecnologías en continua evolución como la inteligencia artificial y el *Deep learning* para mejorar la precisión y la eficiencia en la identificación de crisis epilépticas, contribuyendo a mejorar la calidad de vida de las personas que la sufren.

Dentro de los factores económicos y ambientales, cabe destacar que el coste se puede reducir a largo plazo gracias a estos métodos, ya que el tratamiento de las personas sea menos elevado y la atención de la epilepsia más sencilla. Pero en relación con el impacto ambiental, el entramado de un modelo de *Deep Learning* puede suponer un gran consumo de energía, por lo que el factor de sostenibilidad se ha tenido en cuenta con la implementación de un modelo sencillo pero eficaz.

La metodología que ha seguido el desarrollo de este proyecto se basa en un enfoque de optimización iterativa. Se recopilaban los datos y fueron preprocesados. Posteriormente, ha sido necesario iterar con ellos de manera continua hasta llegar a los resultados deseados con diferentes tipos de normalización de datos, hiperparámetros del modelo, capas del modelo, etc.

Los resultados obtenidos demuestran que el espacio latente del modelo de autocodificador variacional proporciona una mejor representación del espacio latente en comparación con el autocodificador estándar. Se demuestra además, a través de las métricas de rendimiento del CVAE, destacando la precisión, que existe una mejora en la detección de las crisis en el espacio latente del CVAE. Adicionalmente, la capacidad del CVAE para generar nuevas señales proporciona nuevas posibilidades para la simulación y análisis de datos en escenarios controlados.

En conclusión, el modelo CVAE se presenta como una herramienta prometedora para la detección de crisis epilépticas, ofreciendo ventajas significativas en términos de precisión y capacidad de generación de datos. Este trabajo no solo contribuye al avance del conocimiento en el campo del *Deep learning* en la medicina, sino también a futuras investigaciones y aplicaciones prácticas en el diagnóstico y manejo de la epilepsia.

Abstract

The way people think and live has been changing throughout history due to technological advances. These innovations have allowed significant developments in the field of medicine, among others. The development of new software and mathematical tools contributes to the diagnosis made by doctors when detecting a disease.

In this project, epilepsy is presented, a disease that affects millions of patients worldwide, with new cases every day, impacting the quality of life of those who suffer from it. The presentation of this disease focuses on the most basic concepts, but enough to continue with the study of electroencephalogram signals through Deep Learning algorithms for its characterizations. This study has been conducted ethically, using data from a public and Anonymous database, maintaining confidentiality.

Highlighting Deep Learning as one of the technological advances, it has generated great interest due to its ability to detect diseases caused by abnormal brain rhythms or pathologies through neural networks. The project focuses on the development of a neural network model, a variational autoencoder, for the detection of epileptic seizures. Therefore, the Project includes continuously evolving technologies such as artificial intelligence and Deep Learning to improve the accuracy and efficiency in the identification of epileptic seizures, contributing to improving the quality of life of those who suffer from it.

Considering economic and environmental factors, it is worth noting that costs can be reduced in long term thanks to these methods, as the treatment of people becomes less expensive and epilepsy care simple. However, regarding environmental impact, training a Deep Learning model can result in high energy consumption, so the sustainability has been considered with the implementation of a simple yet effective model.

The methodology followed in the development of this project is based on an iterative optimization approach. Data were collected and preprocessed. Subsequently, it was necessary to continuously iterate with them until reaching the desired results with different types of data normalization, model hyperparameters, model layers, etc.

The results obtained demonstrate that the latent space of the variational autoencoder model provides a better representation of the latent space compared to the standard autoencoder. It is demonstrated, through CVAE performance metrics, highlighting precision, an improvement in the detection of seizures in the CVAE's latent space. Additionally, the CVAE's ability to generate new signals provides new possibilities for the simulation and analysis of data in controlled scenarios.

In conclusion, the CVAE model is presented as a promising tool for the detection of epileptic seizures, offering significant advantages in terms of precision and data generation capability. This work not only contributes to advancing knowledge in the field of Deep Learning in medicine but also to future research and practical applications in the diagnosis and management of epilepsy.

INDICE DE CONTENIDOS

Resumen	4
Abstract.....	6
1. Introducción.....	16
1.1 Descripción del PFG	16
1.2 Objetivos	16
1.3 Especificaciones y restricciones de diseño	17
1.4 Metodología de trabajo propuesta.....	17
1.5 Desglose de tareas y cronograma.....	18
1.6 Estructura del proyecto	19
2. Señales en la bioingeniería, encefalograma y epilepsia	20
2.1 Introducción.....	20
2.2 Señales biomédicas.....	21
2.3 Epilepsia	27
2.3.1. Descripción de la epilepsia.....	27
2.3.2. Síntomas.....	27
2.3.3. Causas	28
2.3.4. Diagnóstico.....	29
2.3.5. Tratamiento	29
2.3.6. Clasificación de crisis.....	30
2.4 Discusión	31
3. Detección de Crisis Epilépticas a través de Aprendizaje Máquina	32
3.1. Introducción.....	32
3.2. Aprendizaje Máquina.....	32
3.2.1. Preparación de los datos.....	34
3.3. Aprendizaje profundo y las redes neuronales	35
3.3.1. Extracción de características.....	37
3.4. Sobreajuste	38
3.5. Redes densas.....	39
3.6. Redes neuronales convolucionales.....	39
3.7. Modelos de clasificación	41
3.7.1. Autocodificadores	41
3.7.2. Autocodificador Variacional.....	43
3.7.3. Autocodificador Variacional Condicional.....	44

3.8.	Medidas de rendimiento	45
3.9.	Discusión	46
4.	Metodología y resultados	48
4.1	Introducción	48
4.2	Base de datos	48
4.3	Autocodificador con red densa	52
4.3.1.	Análisis del espacio latente	52
4.3.2.	Análisis del sobreajuste	58
4.4	Autocodificador Variacional Condicional	59
4.4.1.	Definición del CVAE	61
4.4.2.	Sobreajuste	62
4.4.1.	Clasificación de los datos.	63
4.4.2.	Obtención del espacio latente.	64
4.4.3.	Reconstrucción de la señal	65
4.4.4.	Generación de nuevas señales	66
4.5	Discusión	67
5.	Conclusiones y líneas futuras	68
5.1	Conclusiones generales del proyecto	68
5.2	Líneas futuras	68
6.	Referencias	70
Anexo A: Impacto del proyecto		74
A.1.	ODS 3: Salud y Bienestar	74
A.2.	ODS 4: Educación de calidad	74
A.3.	ODS 9: Industria, innovación e infraestructura	74
A.4.	ODS 10: Reducción de las desigualdades	74
Anexo B: Presupuesto del proyecto		76
B.1.	Coste de los medios materiales	76
B.2.	Coste personal	76
B.3.	Coste total del proyecto	77

INDICE DE FIGURAS

<i>Figura 1. Diagrama de Gantt incluyendo las tareas del proyecto.</i>	<i>18</i>
<i>Figura 2. Categorías de la bioingeniería.</i>	<i>20</i>
<i>Figura 3. Potencial transmembrana en una señal biomédica [12].</i>	<i>21</i>
<i>Figura 4. Relación entre puntos de medición y simulación de señales EEG y EMG con ruido y ECG cargadas en Matlab.</i>	<i>22</i>
<i>Figura 5. Anatomía de una neurona [16].</i>	<i>23</i>
<i>Figura 6. Representación de la diferencia de potencial correspondiente a un ritmo Alpha [17].</i>	<i>24</i>
<i>Figura 7. Representación de la diferencia de potencial correspondiente a un ritmo Beta [17].</i>	<i>24</i>
<i>Figura 8. Representación de la diferencia de potencial correspondiente a un ritmo Theta [17].</i>	<i>24</i>
<i>Figura 9. Representación de la diferencia de potencial correspondiente a un ritmo Delta [17].</i>	<i>25</i>
<i>Figura 10. Representación de la diferencia de potencial correspondiente a un ritmo Gamma [17]. ...</i>	<i>25</i>
<i>Figura 11. Sistema internacional 10-20 [20].</i>	<i>26</i>
<i>Figura 12. Diferentes segmentos de señales EEG simuladas [21]. a) En reposo con ojos cerrados, b) Crisis epiléptica, c) En reposo con ojos abiertos con artefacto electrodo "pop", d) artefacto de ruido de interferencia de la red eléctrica de 60 Hz, e) En reposo ojos con abiertos con artefacto de parpadeo.</i>	<i>26</i>
<i>Figura 13. Síntomas más comunes de la epilepsia [25].</i>	<i>28</i>
<i>Figura 14. Causas de la epilepsia.</i>	<i>28</i>
<i>Figura 15. Procedimiento de prueba EEG [30].</i>	<i>29</i>
<i>Figura 16. Actividad del cerebro en los diferentes tipos de crisis epilépticas [30].</i>	<i>31</i>
<i>Figura 17. Aplicaciones de aprendizaje automático y sus aplicaciones [32].</i>	<i>33</i>
<i>Figura 18. Pipeline de Machine Learning [33].</i>	<i>34</i>
<i>Figura 19. De neurona biológica (izquierda) a una neurona artificial o perceptrón (derecha) [37]. ..</i>	<i>35</i>
<i>Figura 20. Componentes en una neurona artificial [38].</i>	<i>36</i>
<i>Figura 21. Representación de algunas funciones de activación [38].</i>	<i>36</i>
<i>Figura 22. Ejemplo de una red neuronal totalmente conectada con múltiples capas ocultas [40]. ...</i>	<i>37</i>
<i>Figura 23. Representación del overfitting [42].</i>	<i>38</i>
<i>Figura 24. Fully connected Neural Network Layer [44].</i>	<i>39</i>
<i>Figura 25. Arquitectura de las CNN aplicada al reconocimiento de dígitos [46].</i>	<i>40</i>
<i>Figura 26. Representación de la función max pooling [45].</i>	<i>41</i>
<i>Figura 27. Autoencoder [47].</i>	<i>42</i>
<i>Figura 28. Autoencoder Variacional [7].</i>	<i>43</i>
<i>Figura 29. Arquitectura de CVAE [51].</i>	<i>44</i>
<i>Figura 30. Organización de los datos.</i>	<i>49</i>
<i>Figura 31. Obtención de las señales para el entrenamiento.</i>	<i>50</i>
<i>Figura 32. Obtención de las señales para la validación.</i>	<i>51</i>
<i>Figura 33. Obtención de las señales con epilepsia.</i>	<i>52</i>
<i>Figura 34. Obtención de los booleanos de las señales con epilepsia.</i>	<i>52</i>

<i>Figura 35. Espacio latente del autoencoder [53].</i>	53
<i>Figura 36. Espacio latente del variational autoencoder [53].</i>	53
<i>Figura 37. Implementación de la técnica PCA en un autoencoder.</i>	54
<i>Figura 38. Implementación de la técnica PCA en el autoencoder variacional.</i>	54
<i>Figura 39. Espacio latente con PCA del autoencoder (izquierda) y del autoencoder variacional (derecha).</i>	55
<i>Figura 40. Implementación de la técnica t-SNE en un autoencoder.</i>	55
<i>Figura 41. Implementación de la técnica t-SNE en el autoencoder variacional.</i>	56
<i>Figura 42. Espacio latente con t-SNE del autoencoder (izquierda) y del autoencoder variacional (derecha).</i>	56
<i>Figura 43. Implementación de la técnica Isomap en el autoencoder.</i>	57
<i>Figura 44. Implementación de la técnica Isomap en el autoencoder variacional.</i>	57
<i>Figura 45. Espacio latente con Isomap del autoencoder (izquierda) y del autoencoder variacional (izquierda).</i>	57
<i>Figura 46. Análisis visual del overfitting en autoencoder.</i>	58
<i>Figura 47. Análisis numérico del overfitting en autoencoder.</i>	59
<i>Figura 48. Transformación StandardScaler [58].</i>	60
<i>Figura 49. Normalización de los datos flatten a través de StandardScaler.</i>	60
<i>Figura 50. Transformación MinMaxScaler[59].</i>	60
<i>Figura 51. Normalización de los datos flatten a través de MinMaxScaler.</i>	60
<i>Figura 52. Esquema del CVAE implementado.</i>	61
<i>Figura 53. Representación gráfica del overfitting sobre el conditional variational autoencoder.</i>	62
<i>Figura 54. Representación numérica del overfitting sobre el conditional variational autoencoder.</i>	62
<i>Figura 55. Métricas de evaluación del CVAE mediante normalización StandardScaler.</i>	63
<i>Figura 56. Métricas del CVAE con la normalización de los datos MinMax.</i>	63
<i>Figura 57. Conditional Variational Autoencoder [60].</i>	64
<i>Figura 58. Análisis del espacio latente en 2D(izquierda) y en 3D (derecha) con la normalización de los datos StandardScaler.</i>	64
<i>Figura 59. Análisis del espacio latente en 2D (izquierda) y en 3D (derecha) con la normalización de los datos MinMax.</i>	65
<i>Figura 60. Señal reconstruida utilizando StandardScaler.</i>	65
<i>Figura 61. Señal reconstruida utilizando normalización MinMaxScaler.</i>	66
<i>Figura 62. Ejemplos de señales generadas sin crisis.</i>	66
<i>Figura 63. Ejemplos de señales generadas con crisis.</i>	67

INDICE DE TABLAS

<i>Tabla 1. Desglose de tareas planificadas.....</i>	<i>18</i>
<i>Tabla 2. Matriz de confusión.....</i>	<i>45</i>
<i>Tabla 3. Valores de la matriz de confusión 2x2.</i>	<i>45</i>
<i>Tabla 4. Información básica de cada paciente de la base de datos CHB-MIT [4].....</i>	<i>48</i>
<i>Tabla 5. Resumen de las técnicas de visualización de los datos.</i>	<i>58</i>

Lista de acrónimos

ACRÓNIMO	DEFINICIÓN
ADN	Ácido Desoxirribonucleico
AE	Autocodificador (<i>Autoencoder</i>)
ANN	Red Neuronal Artificial (<i>Artificial Neural Networks</i>)
BCI	Interfaz cerebro-computador (<i>Brain Computer Interface</i>)
BERT	<i>Bidirectional Encoder Representations from Transformers</i>
CNN	Redes neuronales convolucionales (<i>Convolutional Neural Network</i>)
COIT	Colegio Oficial de Ingenieros de Telecomunicación
ECG	Electrocardiograma
EEG	Encefalograma
EMG	Electromiograma
ENV	Estimulación No Invasiva
FAE	Fármacos anti-elépticos
FN	Falso Negativo
FP	Falso Positivo
GAN	Redes generativas antagónicas (<i>Generative Adversarial Network</i>)
GEMB	<i>The Group of Engineering in Medicine and Biology of The Institute of Electrical and Electronics Engineers</i>
GPT	<i>Generative Pre-Trained Transformer</i>
GPU	Unidades de procedimiento gráfico
GRU	<i>Gated Recurrent Unit</i>
IA	Inteligencia Artificial
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
Isomap	<i>Isometric Mapping</i>
KL	Kullback-Leibler
KNN	<i>K-Nearest Neighbors</i>
LSTM	Redes de memoria a corto plazo (<i>Long short-term memory networks</i>)
MLP	Perceptrón multicapa (<i>Multi-layer perceptron</i>)
MSE	Error cuadrático medio (<i>Mean Squared Error</i>)
ODS	Objetivos de Desarrollo Sostenible
OMS	Organización Mundial de la Salud

ONU	Organización de las Naciones Unidas
PCA	Análisis de Componentes Principales (<i>Principal Component Analysis</i>)
PET	Tomografía de emisión de positrones (<i>Positron Emission Tomography</i>)
RM	Resonancia Magnética
RNN	Redes neuronales recurrentes (<i>Recurrent Neural Network</i>)
SEN	Sociedad Española de Neurología
SPECT	Tomografía computarizada por emisión de fotón único (<i>Single Photon Emission Computed Tomography</i>)
SVM	Máquina de Vectores de Soporte (<i>Support Vector Machine</i>)
TN	Verdadero Negativo
TP	Verdadero Positivo
t-SNE	<i>T-distributed Stochastic</i>
VAE	Autocodificador Variacional (<i>Variational Autoencoder</i>)

1. Introducción

1.1 Descripción del PFG

La tecnología supone avances continuos en los progresos para entender los mecanismos del comportamiento de nuestro organismo. Gracias a ella, somos capaces de prevenir, diagnosticar y curar patologías que ponen en riesgo la vida de las personas [1].

Este proyecto une Ciencia, Salud y Tecnología, centrándose en la detección de ataques epilépticos. La epilepsia está presente en todo el mundo, afectando la vida de más de 50 millones de personas. En concreto, en España, se considera que 400.000 personas sufren esta patología según datos de la Sociedad Española de Neurología (SEN) [2], y aparecen 20.000 casos nuevos cada año [3].

La epilepsia se caracteriza por alterar el sistema nervioso central (neurológico) y trastornar la actividad cerebral. Esto conlleva que el paciente sufra convulsiones repetidas. El proyecto se llevará a cabo mediante el estudio de señales electroencefalográficas (EEGs) correspondientes a la base de datos de CHB-MIT Scalp [4].

A lo largo de este proyecto se desarrolla un modelo basado en el aprendizaje profundo a través de *Variational Autoencoders* partiendo del TFG de Marta Fraile [5], que utilizó *Autoencoders* no variacionales. Un *Variational Autoencoder* (VAE) es un modelo de aprendizaje generativo que utiliza redes neuronales probabilísticas con el fin de aprender a generar cualquier tipo de dato. Se construye a partir de un *autoencoder* (codificador) y un *decoder* (decodificador), a los que se les añade a la salida una función de pérdida para medir la similitud entre el objeto decodificado y el que se codifica. El objetivo es que su espacio latente, a la salida del codificador, represente buenas cualidades para la generación de nuevos datos [6], [7].

1.2 Objetivos

El objetivo de este proyecto se centra en el desarrollo de un auto-codificador variacional que permita discriminar entre comportamiento normal del cerebro y estado ictal, así como la generación de nuevos datos (señales sintéticas).

Es necesario tener en cuenta que este objetivo se divide en dos partes fundamentales del proyecto:

- Desarrollo de un *autoencoder* condicionado durante el entrenamiento para tratar que su espacio latente tenga buenas propiedades.
- Generación de nuevos datos a partir del espacio latente del *autoencoder*.

El desarrollo del proyecto se realizará a partir de las señales electroencefalográficas (EEG) de la base de datos CHB-MIT Scalp disponible en PhysioNet [4], codificando el VAE en Python.

Desde el punto de vista académico, este proyecto incluye los siguientes objetivos:

- Aprendizaje del lenguaje Python.
- Aprendizaje acerca de los modelos generativos y desarrollo de un VAE dentro del ámbito del aprendizaje de máquina.

1.3 Especificaciones y restricciones de diseño

En la realización de este proyecto, contamos desde el primer momento con las siguientes especificaciones y restricciones que enumeramos a continuación:

1. Uno de los objetivos principales es que el espacio latente esté lo suficientemente estructurado para poder utilizar el decodificador con fines generativos, para ello nos fijaremos en la continuidad y la integridad del mismo.
2. Reducción máxima de los falsos positivos y falsas alarmas.
3. El modelo de autocodificador ha de ser flexible en cuanto al número de señales con las que se trabaja, es decir, debemos poder cambiar el número de canales y la frecuencia de muestreo manteniendo una buena fidelidad en la aproximación.
4. El modelo desarrollado deberá poder adaptarse al formato de los datos de entrada.
5. El de modelo de auto-codificador debe ser validado con señales reales obtenidas de PhysioNet [4].

1.4 Metodología de trabajo propuesta

A lo largo del desarrollo de este proyecto, la metodología de trabajo será la siguiente:

Se utilizará un proceso de optimización iterativa para el desarrollo del codificador y decodificador. De esta manera, se ajustará el autocodificador para poder depurar la información utilizada y continuar con la siguiente iteración.

Una vez conseguido el autocodificador, se continuará con la parte opcional del proyecto, es decir con el diseño de un espacio latente regular para poder conseguir señales generadas por el mismo dentro de los márgenes establecidos.

1.5 Desglose de tareas y cronograma

El proyecto se desarrollará a lo largo de 5 meses a tiempo completo, como se indica en la Figura 1. Se dedicarán alrededor de 320 horas para la realización del proyecto. El desglose de las tareas se muestra a continuación, en la Tabla 1:

DESGLOSE DE TAREAS

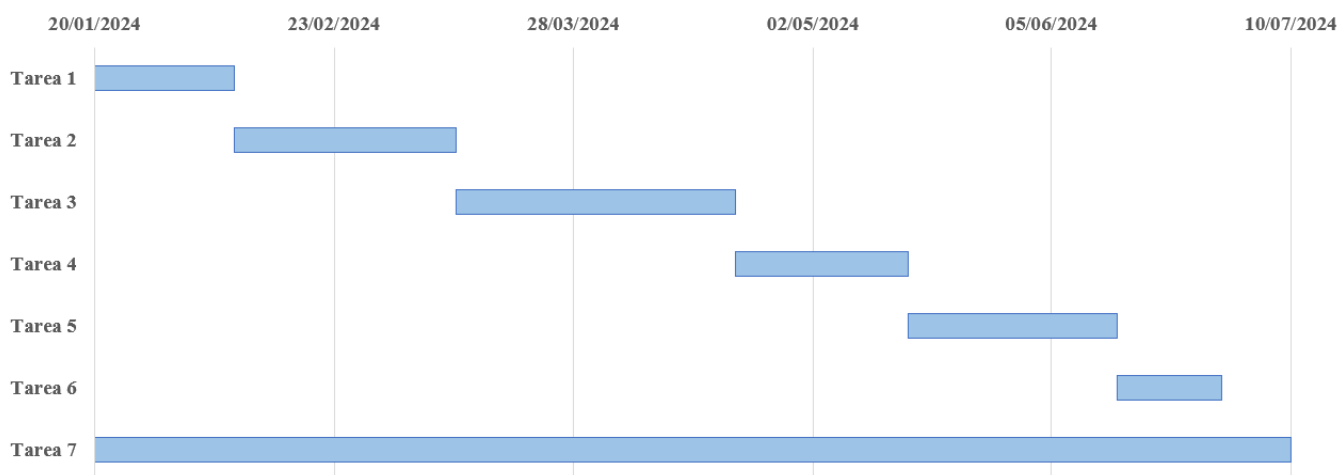


Figura 1. Diagrama de Gantt incluyendo las tareas del proyecto.

- 1) **Tarea 1:** Aprendizaje y estudio del lenguaje de programación Python.
- 2) **Tarea 2:** Estudio del modelo *Variational Autoencoders* y de las técnicas de *Deep Learning*.
- 3) **Tarea 3:** Diseño de la arquitectura del *Variational Autoencoder* (codificador y decodificador).
- 4) **Tarea 4:** Entrenamiento del modelo, para minimizar la función de pérdida, y validación del mismo.
- 5) **Tarea 5:** Muestreo del espacio latente y generación de nuevas muestras.
- 6) **Tarea 6:** Evaluación y ajuste del VAE para mejorar su rendimiento.
- 7) **Tarea 7:** Redacción de la memoria del proyecto y preparación de la presentación.

TAREA	FECHA DE INICIO	DURACION EN DIAS	FECHA DE FIN	DURACIÓN EN HORAS
Tarea 1	20/01/2024	20	09/02/2024	35
Tarea 2	09/02/2024	32	12/03/2024	45
Tarea 3	12/03/2024	40	21/04/2024	45
Tarea 4	21/04/2024	25	16/05/2024	50
Tarea 5	16/05/2024	30	15/06/2024	55
Tarea 6	15/06/2024	15	30/06/2024	30
Tarea 7	20/01/2024	172	10/07/2024	60
Total	20/01/2024	172	10/07/2024	320

Tabla 1. Desglose de tareas planificadas.

1.6 Estructura del proyecto

Este documento contiene 5 capítulos. A continuación, se describen brevemente cada uno de ellos:

- 1) En el primer capítulo se refleja la orientación del desarrollo de este proyecto.
- 2) En el segundo capítulo se expone una breve descripción sobre las señales biomédicas, centrándose en el electroencefalograma como señal bioeléctrica. A partir de esta señal, se presenta la anatomía y fisiología de las neuronas del cerebro a nivel básico. Por último, se presenta el sistema 10/20 como estándar para la medición de esta señal. Además, incluye la explicación de la enfermedad de la epilepsia, describiendo qué es, los síntomas que provocan y por qué ha sido provocada, el diagnóstico que debe ser realizado para detectarlo, el tratamiento para la misma y qué tipos de crisis existen.
- 3) En el tercer capítulo se desarrollan las bases del Aprendizaje Automático, sus características y se añade una explicación sobre los sistemas basados en el mismo. Además, se presenta el aprendizaje profundo, traducido en inglés como *Deep Learning*, los sistemas que lo forman y, por último, se explican las redes neuronales que se han implementado en este proyecto.
- 4) El capítulo cuatro incluye los resultados obtenidos en la realización de este proyecto. Además, se incluye la manera en la que se han implementado las redes en Python.
- 5) En el último capítulo, se presentan las conclusiones generales del proyecto y, se añaden las líneas futuras posibles para realizar en un futuro.
- 6) Por último, los Anexos incluyen el impacto del proyecto en relación con las implicaciones sociales, de salud y bienestar, de educación, de innovación e industria y su posible aportación a los Objetivos de Desarrollo Sostenible(ODS) [8] y el presupuesto del proyecto.

2. Señales en la bioingeniería, encefalograma y epilepsia

2.1 Introducción

La palabra “bioingeniería” nace en 1954 con el profesor Heinz Wolff: “La Bioingeniería consiste en la aplicación de las técnicas y las ideas de la ingeniería a la biología, y concretamente a la biología humana. El gran sector de la Bioingeniería que se refiere especialmente a la medicina puede llamarse más adecuadamente Ingeniería Biomédica” [9].

Con esta definición, partimos de la base de la aplicación de los conocimientos de la ingeniería para poder mejorar y proporcionar resultados en el campo de la medicina. El Comité de Educación de GEMB/IEEE (*The Group of Engineering in Medicin and Biology of The Electrocal and Electronics Engineers*, 1997) distingue tres categorías diferentes dentro del mundo de la bioingeniería como se muestra en la Figura 2, las cuales satisfacen todas las necesidades del campo de la medicina:

- **Bioingeniería o ingeniería biológica:** parte de la biología, siendo la más básica de todas, centrada en el estudio profundo de la ciencia.
- **Ingeniería biomédica o ingeniería médica:** aunque también parte de la biología, está centrada en la parte humana, destinada a definir, controlar y mejorar enfermedades y problemas básicos.
- **Ingeniería clínica:** también incluida en la ingeniería biomédica, pero enfocada en los problemas asistenciales de salud [10].

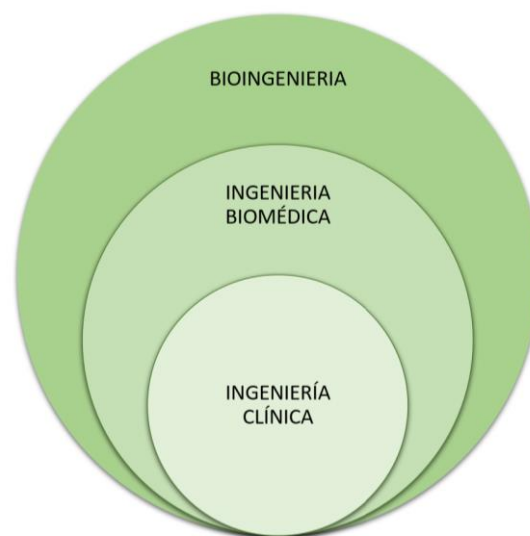


Figura 2. Categorías de la bioingeniería.

Conociendo las tres diferentes ramas de la bioingeniería, en este proyecto nos centraremos en la ingeniería biomédica, conoceremos las señales que podemos conseguir a través de ella, y

mostraremos mayor interés en el electroencefalograma (EEG). Por último, hablaremos de la epilepsia, de sus causas, su tratamiento, etc.

2.2 Señales biomédicas

Según el origen de las señales biomédicas, estas se pueden clasificar dentro de diferentes ramas de la bioingeniería, entre las que se encuentran las señales de bioimpedancia, bioacústicas, biomagnéticas, biomecánicas, bioquímicas, bioópticas y bioeléctricas. En este capítulo, nos centraremos en las últimas mencionadas, las señales bioeléctricas.

2.2.1. Señales bioeléctricas

Las señales bioeléctricas provienen de los sistemas biológicos, son corrientes eléctricas generadas por procesos biológicos dentro del cuerpo humano u otros organismos vivos. Estas señales reflejan la diferencia de potencial que proviene del potencial transmembrana que, medido con microelectrodos y a nivel de células aisladas, resulta ser una señal biomédica como se muestra en la Figura 3 [11].

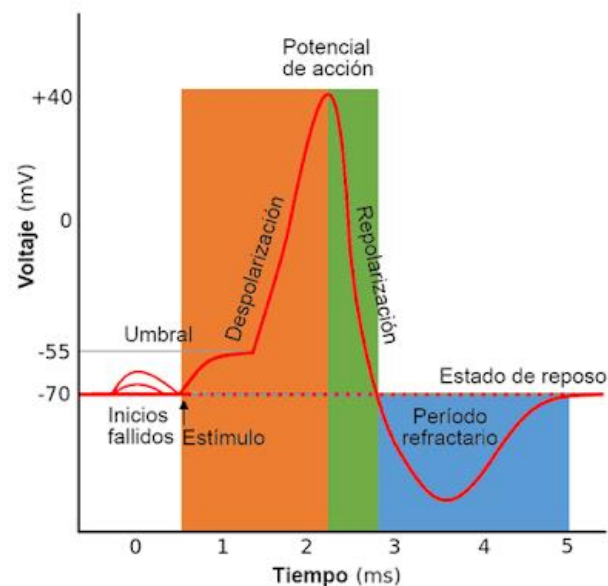


Figura 3. Potencial transmembrana en una señal biomédica [12].

Estas señales suelen estar contaminadas por ruido e interferencias, además de caracterizarse por amplitudes muy pequeñas, por lo que las técnicas que se utilizan deben de ser de calidad y precisión. Estas técnicas típicamente son del tipo no invasivo, es decir, no provocan la necesidad de la rotura del tejido de la piel, sino que se utilizan electrodos colocados en la parte del cuerpo que se vaya a medir.

Ya hemos explicado qué son las señales bioeléctricas y cómo se obtienen. A continuación, se describen muy brevemente, las principales de señales que se caracterizan como señales bioeléctricas, cuya su representación puede verse en la Figura 4.

- **Electroencefalograma**, también conocido como **EEG**. Esta señal registra la actividad eléctrica dentro del cerebro, la cual se mide mediante electrodos colocados en la cabeza del paciente. La actividad que se registra en la práctica es la sincronización de la actividad neuronal en diferentes regiones del cerebro. Esta señal se utiliza, principalmente, para diagnosticar trastornos neurológicos y, para evaluar la actividad del sueño [13].
- **Electromiograma** o **EMG**. Esta señal se caracteriza por registrar la actividad eléctrica de las contracciones musculares. Su uso se centra en el diagnóstico de trastornos musculares, evaluación de lesiones musculares y nerviosas, y para el control de la actividad muscular durante la rehabilitación [14].
- **Electrocardiograma** o **ECG**, el cual registra la actividad eléctrica del corazón. Se mide a través de electrodos, como el EEG, pero situados sobre la piel. Consiste en la medición de la despolarización y repolarización del tejido cardiaco. Se utiliza para el diagnóstico de trastornos cardiacos, como pueden ser las arritmias, enfermedades del corazón y daño en su tejido [15].

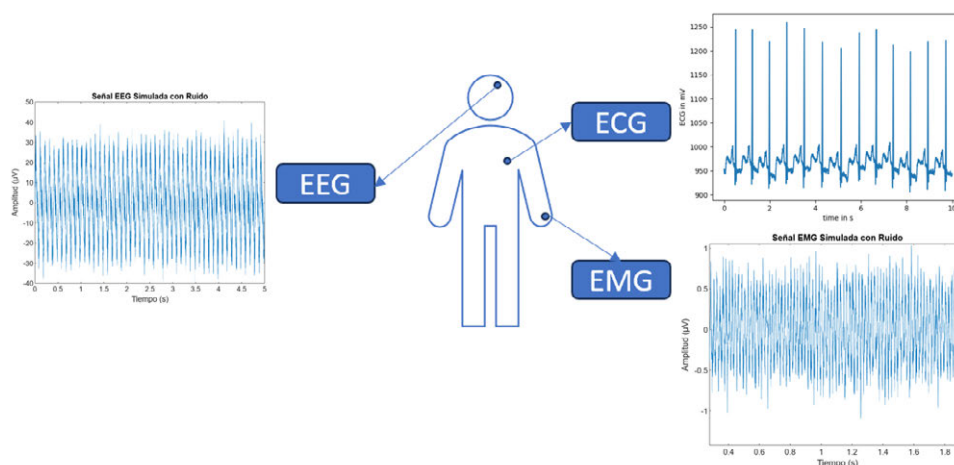


Figura 4. Relación entre puntos de medición y simulación de señales EEG y EMG con ruido y ECG cargadas en Matlab.

2.2.2. Electroencefalograma

El electroencefalograma, mencionado anteriormente como una señal bioeléctrica, está definido como el registro de la actividad eléctrica del cerebro, y es la señal que se utiliza para la medición de la epilepsia.

Para poder entender la señal registrada, se explicará el sistema nervioso y sus componentes, las células.

Las células del cerebro, llamadas **neuronas**, son un tipo de células con la función de recibir y enviar mensajes entre el cuerpo y el encéfalo. Estos mensajes, son los que se envían a través de la corriente

eléctrica generada por ellas mismas, los llamados **potenciales de acción** como el mostrado en la Figura 5, que permite que la comunicación sea rápida y a “largas” distancias.

La anatomía de la neurona comienza con el cuerpo celular, el **soma**, donde se encuentra el núcleo de la neurona, que contiene el material genético de la célula (ADN). En él se sintetizan la mayoría de las proteínas que necesitan producir las neuronas. Del soma, como se puede ver en la Figura 5, salen varias extensiones denominadas **dendritas**, y una más larga, el **axón**.

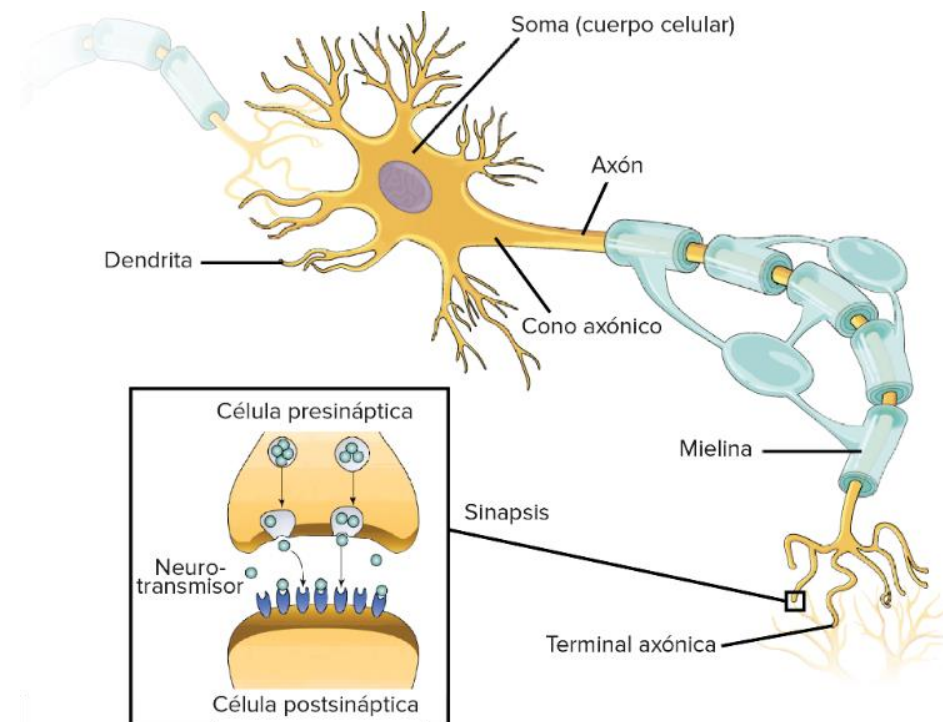


Figura 5. Anatomía de una neurona [16].

Las neuronas tienen las funciones de recibir y procesar la información que reciben, las cuales se realizan en las dendritas y en el cuerpo celular o soma mayoritariamente. Estas señales recibidas se pueden caracterizar por provocar que la neurona genere un impulso eléctrico, **señal excitatoria**, o pueden impedir que la neurona dispare, **señal inhibitoria**.

Son muchas las señales, varios miles, que recibe una neurona en sus dendritas, ya que una neurona puede tener varios conjuntos de dendritas. El conjunto de señales inhibitorias y excitatorias recibidas determinan que una señal se conduzca por el axón, en cuyo caso es necesario que se active la neurona, el impulso nervioso o el potencial de acción.

El axón está recubierto por una sustancia llamada **mielina**, la cual es la responsable de que las señales se transmitan de manera rápida. Sin embargo, esta sustancia nunca se encuentra en las dendritas. Por último, se muestra el extremo del axón, el cual se divide en varias ramas, terminales axónicas, que forman conexiones con las **células blanco**.

La unión entre varias neuronas se forma a partir de las dendritas y el cuerpo celular. Una **sinapsis**, por lo tanto, es el lugar donde se transmite la información desde la primera neurona, llamada neurona presináptica, a la neurona denominada blanco o neurona postsináptica, donde la información se transmite a través de neurotransmisores. Estos neurotransmisores se liberan cuando un potencial de acción viaja por el axón y llega a la terminal axónica. El neurotransmisor está compuesto por unas moléculas responsables de transmitir señales excitatorias o inhibitorias cuando pasan la sinapsis y se unen a receptores en la célula postsináptica [16].

Este proceso, genera el denominado potencial de acción, y la suma de muchos potenciales de acción dan lugar a la señal EEG. A lo largo del tiempo, se puede registrar una diferencia de potencial de la señal EEG entre los que distinguimos los siguientes ritmos:

- **Alpha**

- Banda asociada: de 8 a 13 Hz
- Estado: vigila con ojos cerrados o persona despierta en estado de relajación.

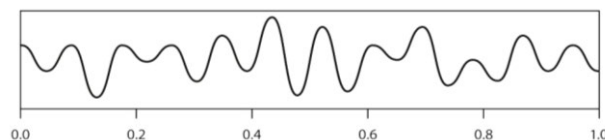


Figura 6. Representación de la diferencia de potencial correspondiente a un ritmo Alpha [17].

- **Beta**

- Banda asociada: de 13 a 30 Hz
- Estado: Concentración o pensamiento activo.

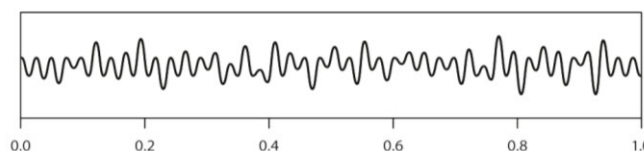


Figura 7. Representación de la diferencia de potencial correspondiente a un ritmo Beta [17].

- **Theta**

- Banda asociada: de 4 a 8 Hz
- Estado: Somnolencia.

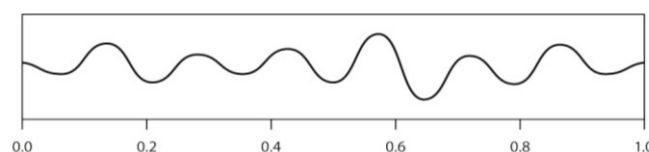


Figura 8. Representación de la diferencia de potencial correspondiente a un ritmo Theta [17].

- **Delta**
 - Banda asociada: entre 0,5 y 4 Hz
 - Estado: Sueño profundo o estado de vigila.

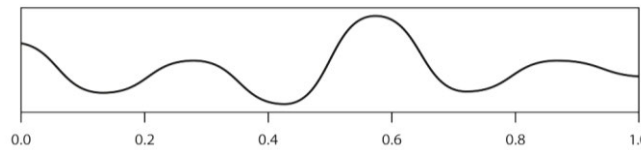


Figura 9. Representación de la diferencia de potencial correspondiente a un ritmo Delta [17].

- **Gamma**
 - Banda asociada: de 30 a 50 Hz
 - Estado: Procesamiento de información que tiene lugar en el córtex cerebral.

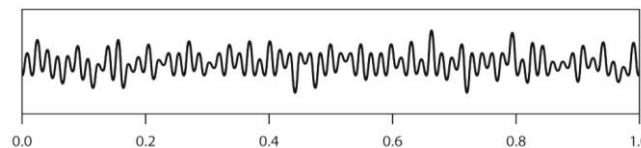


Figura 10. Representación de la diferencia de potencial correspondiente a un ritmo Gamma [17].

Por último, para poder medir una señal EEG, se suele utilizar el método denominado 10/20, el cual es **el sistema internacional** para describir la localización de cada uno de los electrodos. Estos números, el 10 y 20, se refieren al porcentaje de distancia en la que los electrodos deben estar colocados para que la medición sea precisa respecto al eje longitudinal en las diferentes partes del cráneo, la parte frontal, occipital, del lado derecho y del lado izquierdo. Cada una de estas zonas se diferencian por su inicial: **F** (frontal), **T** (temporal), **C** (central), **P** (parietal) y **O** (occipital) y, con dos puntos de referencia denominados inión y nasión.

Las cuatro zonas descritas anteriormente se encargan de diferentes funciones [18]:

- La zona **frontal** es la encargada de los movimientos voluntarios, el habla y de los movimientos faciales.
- La zona **temporal** es la encargada de la coordinación de la comprensión auditiva.
- La zona **parietal** es la encargada de controlar la sensibilidad y la percepción.
- La zona **occipital** es la encargada de coordinar y regular la visión.

Como podemos ver la Figura 11, los electrodos también se caracterizan de unos números, los cuales representan el lado donde están situados, siendo los números 2, 4, 6 y 8 (números pares) los electrodos colocados en el hemisferio derecho, los electrodos 1, 3, 5, 7 y 9 (números impares) los del hemisferio izquierdo y los electrodos colocados en el eje del hemisferio se designan con la letra z [19], [20].

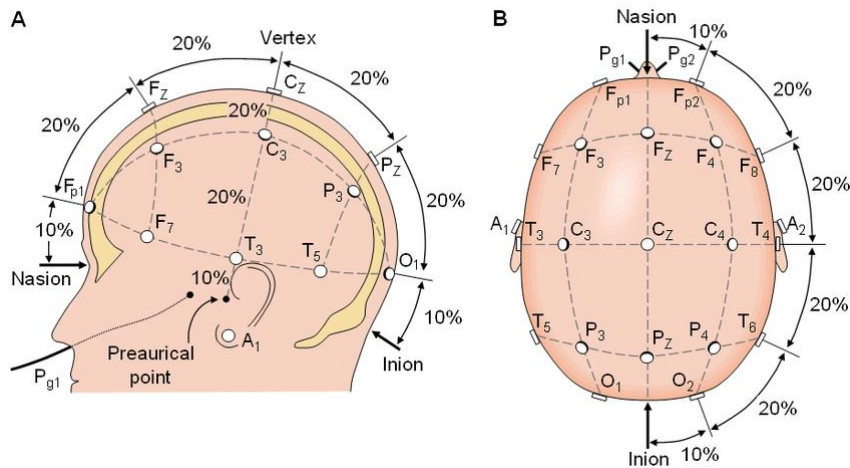


Figura 11. Sistema internacional 10-20 [20].

Por último, como hemos mencionado anteriormente, las señales eléctricas se caracterizan por contener ruido e interferencias que enmascaran la señal. Entre ellas podemos destacar el parpadeo, el movimiento, etc [20].

En la Figura 12 se presentan diferentes ejemplos de registros de EEG. La figura a) muestra la señal registrada en un paciente en situación de reposo con los ojos cerrados, en la imagen b) se registra la señal de un paciente sufriendo una crisis epiléptica, en la figura c) se muestra el registro de un paciente en situación de reposo con los ojos abiertos con artefacto de electrodo pop, en la figura d) se muestra el impacto de una interferencia eléctrica del propio equipo de registro de 60 Hz, en la figura e) se muestra la generación de amplitud detonante debido al registro de una señal EEG de un paciente en reposo, con los ojos abiertos, con artefacto de parpadeo.

“El artefacto del electrodo “pop” puede ocurrir debido a una acumulación de carga estática en un electrodo individual seguido de una liberación rápida de esa carga, como por ejemplo, un movimiento repentino del lugar de contacto del electrodo. Normalmente la aparición de este artefacto es confinada a un solo electrodo. La forma más características del artefacto es una rápida descarga de gran amplitud.” [21].

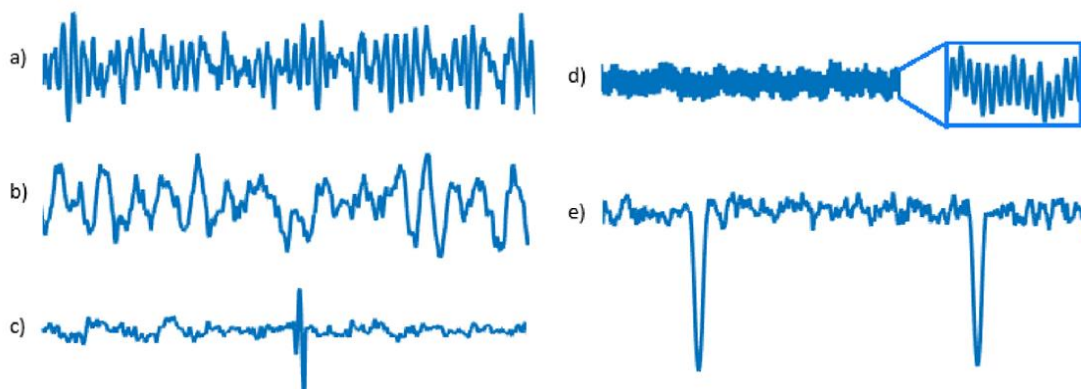


Figura 12. Diferentes segmentos de señales EEG simuladas [21]. a) En reposo con ojos cerrados, b) Crisis epiléptica, c) En reposo con ojos abiertos con artefacto electrodo “pop”, d) artefacto de ruido de interferencia de la red eléctrica de 60 Hz, e) En reposo ojos con abiertos con artefacto de parpadeo.

2.3 Epilepsia

La epilepsia está definida por la OMS como la segunda enfermedad neurológica que más años de vida se ven afectados por la discapacidad [22]. Está reconocida como un gran problema dentro de la salud pública, debido a su prevalencia y repercusión en la vida de los pacientes.

En Europa está presente en el 0,7% de la población, de manera que afecta a unos seis millones de personas, destacando que en España son entre 300.000-400.000 pacientes, con 400.000 casos nuevos por año. Es tal su importancia, que la tasa de incidencia mundial está definida por 61,44 x 100.000 personas/año.

Esta enfermedad afecta a la vida de los pacientes de diferentes maneras, ya que, dependiendo de la gravedad y de la frecuencia de las crisis, así como de los posibles efectos secundarios que aparecen en cada uno de ellos, se cree que la expectativa de vida de los pacientes se reduce entre 2 y 10 años, teniendo estos una tasa de mortalidad entre 2 y 3 veces mayor que del resto de la población [23].

2.3.1. Descripción de la epilepsia

La epilepsia es una enfermedad ancestral. En los pueblos primitivos, fue interpretada como un castigo divino e incluso como una posesión demoníaca. Fueron los griegos quienes la reconocieron como una enfermedad física y, más tarde Hipócrates (460 a.C), la describió como una enfermedad de origen natural y localizada en el cerebro. Finalmente, se llegó a una definición descrita por John Hüngling Jackson en 1873 como *“una descarga súbita, rápida y excesiva de células cerebrales”*. En 1912 se comercializaron los primeros fármacos anti-elépticos, seguidos por el primer encefalograma registrado por Hans Berger en 1929 [24].

Una persona se caracteriza por tener epilepsia cuando ha sufrido dos o más crisis epilépticas, entre las que distinguiremos a lo largo del capítulo entre generalizadas (afectación simultánea de todas las neuronas) y parciales o focales (un grupo delimitado de neuronas), que explicaremos más adelante.

2.3.2. Síntomas

La epilepsia está provocada por crisis epilépticas como, mencionamos anteriormente, en las que se ve afectada la actividad del cerebro, concretamente en las neuronas.

Los síntomas pueden variar debido al tipo de crisis que se sufra. La más conocida y, a su vez la más peligrosa, es la pérdida de conciencia con convulsiones, pero pueden aparecer hasta síntomas motores, sensitivos, visuales o psíquicos sin necesidad de afectar a la conciencia. En la Figura 13 podemos ver los síntomas más comunes y la zona del cerebro afectada. La duración de una crisis epiléptica puede llegar hasta los 5 minutos y afecta tanto a los hombres como a las mujeres [23].



Figura 13. Síntomas más comunes de la epilepsia [25].

2.3.3. Causas

Las causas de la epilepsia pueden llegar a ser determinantes para el pronóstico y el control de esta para cada uno de los pacientes. Como vemos en la Figura 14, podemos diferenciarlas en [27]:

- Causa **genética** o “idiopática”, producida por alteraciones genéticas. Estas causas pueden aparecer en diferentes miembros de una misma familia.
- Causa **estructural**, producida a consecuencia de una lesión en el cerebro, como puede ser traumatismo, ictus y otras lesiones vasculares, tumor, infección, etc.
- Causa **desconocida**, cuando no se puede determinar la causa de la enfermedad, aunque suelen ser lesiones estructurales microscópicas.

Hay que tener en cuenta que hay edades en las que las causas pueden ser determinantes en la aparición de la epilepsia, como una causa genética en las dos primeras etapas de la vida, un traumatismo en adultos jóvenes, un tumor entre los 40 y 50 años de edad, o enfermedades vasculares cerebrales y neurodegenerativas en la tercera edad [26].

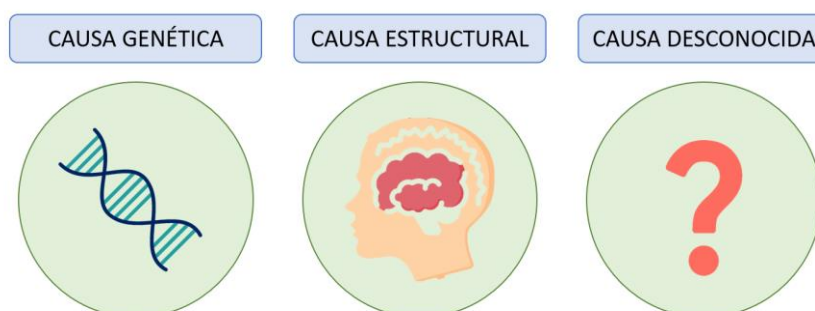


Figura 14. Causas de la epilepsia.

2.3.4. Diagnóstico

El diagnóstico del paciente no se realiza a menos que hayan ocurrido dos o más crisis epilépticas, las cuales no hayan sido provocadas. Es importante destacar que el diagnóstico se realiza a través de una historia clínica, así como de los testigos que hayan vivido las crisis con el paciente. Además, existen exploraciones complementarias que ayudan a mejorar el estudio del paciente, como puede ser el electroencefalograma (EEG), ya mencionado anteriormente, a través de una prueba específica como la mostrada en la Figura 15, la cual nos aporta la información suficiente de la actividad del cerebro en tiempo real, una resonancia magnética (RM) cerebral, la cual permite a los médicos identificar anomalías estructurales, y por último, pruebas de Medicina Nuclear, también denominadas PET/SPECT, las cuales ayudan a identificar las zonas cerebrales que han provocado la crisis [27].

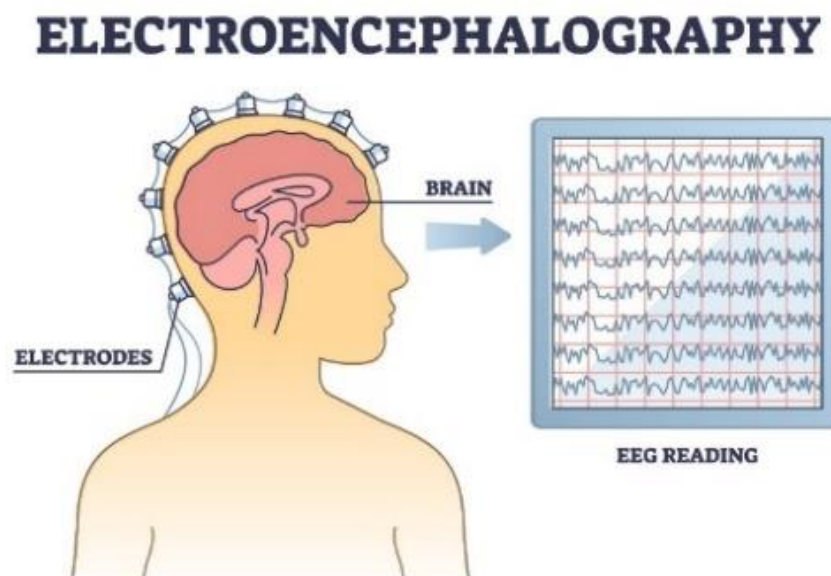


Figura 15. Procedimiento de prueba EEG [30].

2.3.5. Tratamiento

Los tratamientos que existen hoy en día son tan eficaces que en algunos casos pueden llegar a desaparecer las causas y en la mayoría de los casos son capaces de controlar las crisis. A continuación, nombraremos algunos de los principales tratamientos que se utilizan para el tratamiento de esta enfermedad [28]:

- **Medicamentos:** es el tratamiento básico para todos pacientes, se caracteriza por englobar los llamados fármacos anti epilépticos (FAE). El objetivo de estos fármacos es restaurar el equilibrio químico de las neuronas para que, con una toma regular de los mismos, se puedan reducir, controlar e incluso eliminar las crisis epilépticas.

- **Cirugía:** Si para un paciente no se logra disminuir las crisis epilépticas con el tratamiento farmacológico, se suele recurrir a la operación por parte del neurólogo, para poder encontrar y eliminar la zona del cerebro que causa las crisis. Las intervenciones quirúrgicas en la epilepsia son:
 - **Lobectomía o lesionectomía:** extirpación de la parte del cerebro que provoca la crisis.
 - **Callosostomía:** Corte de la unión de los dos hemisferios cerebrales. Se realiza habitualmente en niños con crisis de tipo epiléptica tónica, atónica, o mioclónica (véase en la sección 2.3.6).
- **Dieta cetógena:** El objetivo de esta dieta está pensado para provocar un cambio químico en el cuerpo. Se suele utilizar en casos complicados de niños, ya que con ella se pueden llegar a controlar las crisis. Está basada en un alto nivel de grasas, escasas proteínas y casi nada de hidratos de carbono. Esta dieta controla o reduce las crisis en 2 de cada 3 niños, quienes deben de seguir la dieta rigurosamente.
- **Estimulación del nervio vago (ENV):** es el tratamiento para crisis focales, las cuales no se pueden tratar con medicación, y donde no existe una lesión cerebral. Está destinada para personas mayores de 12 años y consiste en provocar leves pulsaciones eléctricas regulares que envían información al cerebro.

2.3.6. Clasificación de crisis

Como hemos mencionado anteriormente, podemos distinguir entre dos grandes grupos de crisis epilépticas en función de la parte del cerebro afectada. A continuación, las explicaremos más detalladamente [29] y se representan en la Figura 16.

- **Crisis focales:** Este tipo de crisis epilépticas afectan a una zona concreta del cerebro y, el 50% de estas ocurren en niños.
 - **Crisis epiléptica focal elemental:** Durante esta crisis existe un cambio de postura, vista, audición, memoria o sensación. No afecta a la pérdida del conocimiento.
 - **Crisis epiléptica focal compleja:** Conlleva la pérdida del conocimiento y convulsiones rítmicas.
- **Crisis generalizadas:** La mayoría de este tipo de crisis conllevan la pérdida del conocimiento y convulsiones. Se producen en los dos hemisferios del cerebro.
 - **Crisis de ausencias:** Durante esta crisis se pierde el conocimiento durante unos segundos, se detiene la actividad y la mirada se queda fija en un punto.
 - **Crisis mioclónica:** Se sacuden uno o varios grupos de músculos.
 - **Crisis tónica:** Aparece la hipertonia o rigidez de todo el cuerpo.
 - **Crisis atónica:** Hipotonía muscular generaliza y caída al suelo.
 - **Crisis tónico-clónica:** Pérdida del conocimiento seguida de rigidez, caída al suelo y sacudidas rítmicas de manos y piernas.

- **Espasmos:** Flexión o extensión brusca del tronco y de las extremidades repetidamente durante un segundo.

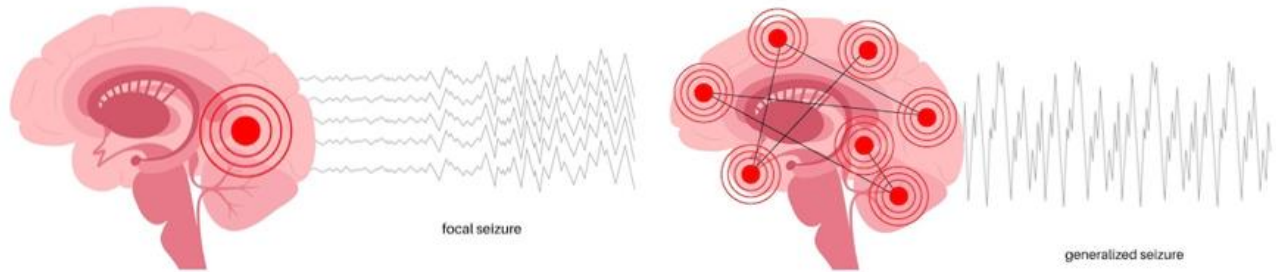


Figura 16. Actividad del cerebro en los diferentes tipos de crisis epilépticas [30].

2.4 Discusión

Este capítulo se ha centrado en la presentación de las señales bioeléctricas, con un enfoque particular en la señal EEG. A través del análisis de la anatomía de las neuronas y la clasificación de las señales bioingenieriles, se han presentado los mecanismos subyacentes de la generación y propagación de los potenciales eléctricos en el cerebro.

Las señales bioeléctricas son fundamentales en la bioingeniería, ya que permiten la monitorización y diagnóstico de diversas condiciones, tanto fisiológicas como patológicas.

El EEG se ha destacado como una herramienta esencial en la neurociencia y neurología. El análisis del EEG, que capta la actividad eléctrica cerebral mediante el sistema 10/20, proporciona un acceso directo a la actividad neuronal. Este sistema de colocación estandarizado asegura la reproducibilidad y comparabilidad de los estudios para poder llegar a mejorar la vida de los pacientes, por ejemplo, mediante la detección de trastornos neurológicos como la epilepsia o el control de interfaces cerebro-computador (BCI).

La epilepsia es uno de los trastornos neurológicos más comunes, caracterizado por la ocurrencia de convulsiones recurrentes y por una actividad eléctrica anormal del cerebro. Estas convulsiones pueden afectar a la vida de las personas de diferentes maneras, debido a la diversidad de sus manifestaciones, por lo que el estudio de esta enfermedad necesita un diagnóstico muy preciso.

Este diagnóstico se basa en la combinación de la observación de dichas convulsiones y pruebas complementarias, como pueden ser el EEG y la RM. Gracias al EEG se pueden identificar patrones específicos que determinan la actividad anormal del cerebro causante de la epilepsia.

Por último, se puede definir la epilepsia como una condición compleja que requiere un enfoque multidisciplinario para su manejo. Para que el paciente afectado pueda ser capaz de llevar una vida plena y completa se combina el diagnóstico preciso, tratamiento adecuado y apoyo integral. Pero también es necesaria una investigación continua y el desarrollo de nuevas terapias para avanzar tanto en el control como en el tratamiento de esta enfermedad.

3. Detección de Crisis Epilépticas a través de Aprendizaje Máquina

3.1. Introducción

A continuación, en este siguiente capítulo nos centraremos en los conceptos básicos del conocido como *Machine Learning* o Aprendizaje de Máquina, indicando cómo está estructurada una red neuronal y, por último, explicaremos el auto-codificador estándar, variacional y variacional condicional.

3.2. Aprendizaje Máquina

El aprendizaje automático, también llamado *Machine Learning* en inglés, forma parte de la rama de la inteligencia artificial (IA), que centrada en el desarrollo de algoritmos y modelos, permiten a las computadoras aprender y mejorar su rendimiento en tareas específicas a partir de datos, lo que sustituye a la tarea de programar una computadora para llevar a cabo cada tarea. Por lo tanto, el objetivo del aprendizaje máquina es crear sistemas que puedan aprender a partir de los datos que se les proporcionan, y realizar predicciones e incluso tomar decisiones a partir de los mismos.

Dependiendo del enfoque y de la técnica utilizada en el campo del aprendizaje máquina, podemos distinguir entre [31]:

- **Aprendizaje supervisado:** Consiste en un entrenamiento utilizando un conjunto de datos que incluye ejemplos de la entrada y la salida deseada correspondiente. El objetivo de esta técnica es el aprendizaje de una función que pueda “mapear” nuevas entradas a salidas esperadas. Entre este tipo de aprendizaje de máquina se encuentran la regresión lineal, la regresión logística, y las máquinas de vectores de soporte (SVM).
- **Aprendizaje no supervisado:** Este modelo se entrena con un conjunto de datos que no incluye las salidas etiquetadas como en el anterior. El objetivo de esta técnica es descubrir patrones intrínsecos o estructuras subyacentes en los datos. Entre las prácticas de esta técnica podemos encontrar el *clustering* o agrupamiento, el análisis de componentes principales (PCA), y los algoritmos de asociación.
- **Aprendizaje por refuerzo:** El modelo aprende a través de la interacción con un entorno, tomando acciones y recibiendo retroalimentación en forma de recompensas o penalizaciones. El objetivo, en este caso, es que el modelo aprenda a tomar decisiones que maximicen la recompensa a largo plazo. Entre los usos de esta técnica se encuentran el Q-learning y los algoritmos basados en políticas.
- **Aprendizaje semi-supervisado:** Este aprendizaje es una combinación del aprendizaje supervisado y no supervisado. En este caso, el conjunto de datos de entrenamiento contiene

datos etiquetados y no etiquetados. El objetivo es aprovechar la información contenida en los datos sin etiquetar para mejorar el aprendizaje del modelo sobre las tareas específicas que se desean abordar. Se suele utilizar en tareas como el reconocimiento de imágenes, el procesamiento de lenguaje natural, y la bioinformática [31].

La Figura 17 muestra un resumen de los 4 tipos de aprendizaje automático y los usos.



Figura 17. Aplicaciones de aprendizaje automático y sus aplicaciones [32].

Para poder poner en práctica las técnicas de *Machine Learning*, es necesario implementar algoritmos de regresión, de estimación Bayesiana, de agrupación o *clustering*, de decisión o clasificación, o de aprendizaje profundo [37].

Para la realización de este proyecto, se han estudiado diferentes técnicas de *Machine Learning*, aunque finalmente, se realizará a través de un aprendizaje supervisado, ya que la base de datos que se va a utilizar contiene etiquetas de los datos. Dado que el objetivo es la detección de crisis epilépticas, los datos se etiquetarán como crisis o no crisis.

El desarrollo de este aprendizaje se realizará mediante una secuencia de pasos de forma ordenada, para poder llegar al resultado deseado. Esta secuencia de pasos se representa a través del pipeline de la Figura 18, la cual transforma un conjunto de datos en un modelo de aprendizaje y, a continuación, se incluye su explicación.

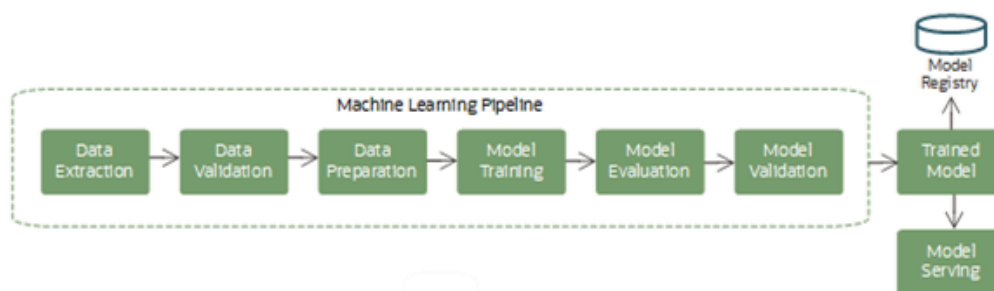


Figura 18. Pipeline de Machine Learning [33].

3.2.1. Preparación de los datos

La fase de preparación de los datos en EEG es muy importante para obtener buenos resultados de los modelos de aprendizaje profundo, ya que la calidad de los datos de entrada afecta directamente al rendimiento del modelo. A continuación, se citan los pasos más generales durante la preparación de los datos [34]:

- **Adquisición de los datos.** Consiste en obtener los registros de EEG de los electrodos colocados en el paciente durante la actividad cerebral, mediante un dispositivo EEG. En este caso, los registros se recogen en la base de datos CHB-MIT Scalp [4].
- **Preprocesamiento de las señales.** Como se ha mencionado en uno de los capítulos anteriores, la señal EEG posee mucho ruido, que puede ser procedente del ruido de la línea eléctrica, de movimientos oculares, etc. Este paso consiste en la eliminación de todo aquel ruido mediante filtrado: filtro paso banda para el rango de trabajo de 0,5 a 100 Hz, Filtro *Notch* centrado en 50 Hz para eliminar el ruido de la red eléctrica, filtro adaptativo para eliminar otros artefactos, etc. Y a continuación, la segmentación de los datos en ventanas de tiempo específicas para el análisis.

En este proyecto, debido a que las señales se han adquirido en un entorno clínico, no se han implementado los algoritmos de filtrado, ya que las señales son bastante limpias. Sin embargo, sí que se utiliza un eventanado de dos segundos a causa de los diferentes estados de la actividad mental, ya que, partiendo de Proyecto de Fin de Grado de Marta Fraile [5], se comprueba que es el que tiene mayor utilidad.

- **Extracción de las características.** Es importante identificar las características más relevantes, o de interés, en el estudio de las señales EEG para su clasificación o regresión. Típicamente, se recogen las características temporales, espectrales y/o espaciales.
- **Normalización y estandarización.** Para poder ayudar a la convergencia del modelo, es necesario ajustar los datos para conseguir una media y una desviación estándar específica (típicamente 0 y 1 respectivamente), aunque esto no es necesario en todos los casos. En este proyecto sí que es necesario que los datos sean normalizados y estandarizados.
- **Selección de las características.** Para poder reducir la complejidad y el tiempo de entrenamiento del modelo, resulta de gran utilidad seleccionar previamente las características más relevantes, y descartar las que no resulten de interés.
- **División de los datos.** Separación de los datos en conjuntos de entrenamiento, validación y prueba (test) para poder evaluar el rendimiento del modelo.
- **Codificación de etiquetas.** Para una tarea de clasificación, las etiquetas de salida deben ser codificadas adecuadamente.

- **Aumentación de los datos.** Generar datos adicionales sintéticamente mediante diferentes técnicas (según la aplicación) como puede ser la rotación, el cambio de escala, el ruido auditivo, etc. Esto permite evitar sobreajustes y mejorar la generalización del modelo.
- **Formato de entrada.** Es de crucial importancia conocer el formato de los datos de entrada para ser procesados por el modelo de aprendizaje profundo. En este proyecto, los datos se han reducido a un formato de dos dimensiones.

3.3. Aprendizaje profundo y las redes neuronales

Dentro de los diferentes métodos que existen en la inteligencia artificial (IA), se encuentra el aprendizaje profundo. Este método consiste en la imitación del funcionamiento del cerebro humano dentro de una computadora u ordenador.

Entre sus múltiples aplicaciones, como puede ser el sector automotriz, aeroespacial, manufacturero, electrónico, etc., se encuentra la investigación médica. Es por esta razón, por la que en este proyecto se utiliza para la detección de crisis epilépticas [35].

El funcionamiento del aprendizaje profundo se basa en algoritmos de redes neurales artificiales (ANN), *Artificial Neural Networks*, las cuales están modeladas siguiendo la estructura del cerebro humano. Es decir, un cerebro humano está compuesto por neuronas, de la misma manera, las redes neuronales por neuronas artificiales, por lo que también son denominadas redes neuronales artificiales.

Las principales características que presentan las ANN para poder parecerse al cerebro humano son las siguientes [36]:

- Capacidad de aprendizaje.
- Capacidad de generalizar automáticamente, es decir, capacidad de distinguir las pequeñas variaciones de los datos de entrada.
- Capacidad de abstraer las características principales de los datos de entrada.

Las ANN básicas están compuestas por unidades llamadas perceptrones. El perceptrón utiliza una arquitectura basada en aprendizaje supervisado. En la Figura 19 podemos ver cómo se modela una neurona con un perceptrón: las dendritas se convierten en las entradas de la red neuronal, la sinapsis en pesos, y el denominado axón, en la salida de la red.

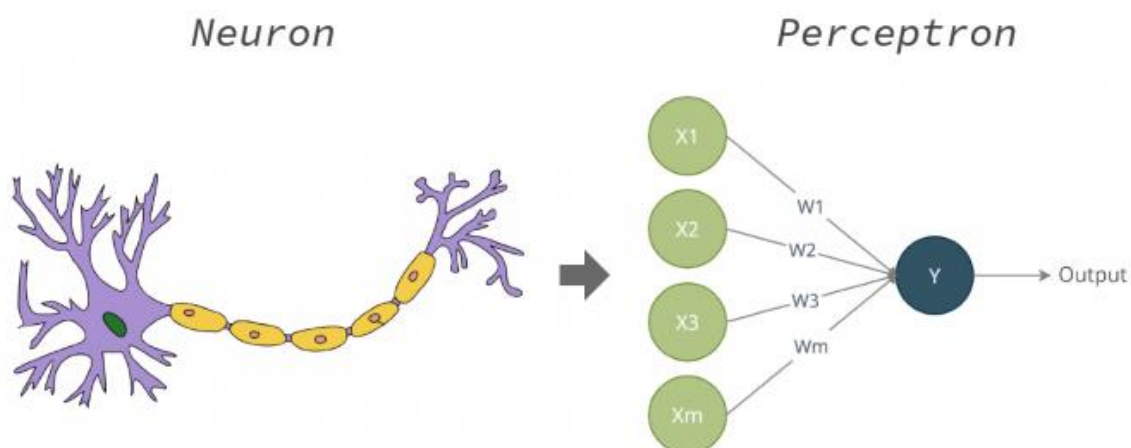


Figura 19. De neurona biológica (izquierda) a una neurona artificial o perceptrón (derecha) [37].

El perceptrón se compone de diferentes entradas desde x_1 hasta x_m , las conexiones o pesos se modelan con los parámetros w_i , y se añade un sesgo (*bias*) w_0 . La Figura 20 muestra esta operación desde un punto de vista matemático incluyendo a función no lineal $f(\cdot)$ al final.

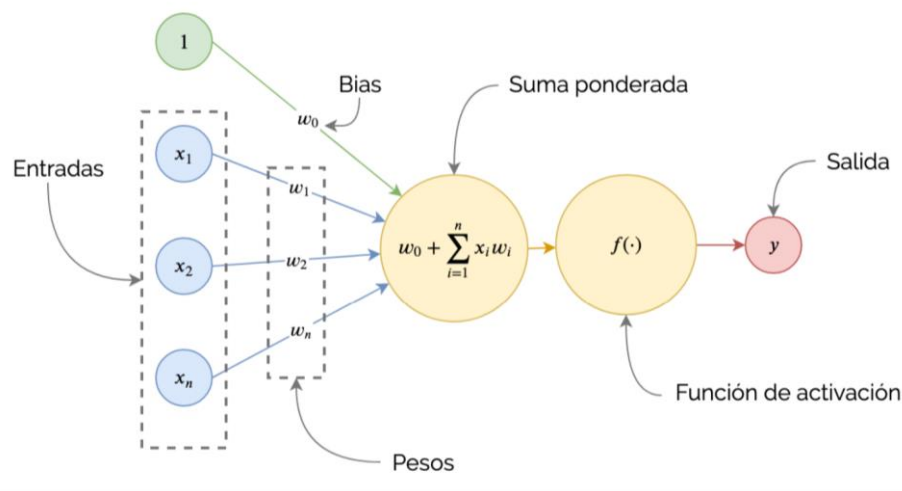


Figura 20. Componentes en una neurona artificial [38].

Como puede verse, el perceptrón realiza las siguientes operaciones para el cálculo de la salida y :

1. Realiza la suma ponderada de las entradas más el sesgo.
 2. Comprueba si la suma de las entradas ponderadas es mayor o menor que el valor umbral y genera la salida denominada y a través de una función de activación no lineal. Existen múltiples opciones de funciones de activación entre las que se encuentran las mostradas en la Figura 21. Dentro de las funciones de activación más comunes encontramos las siguientes [39]:
- **Función Sigmoidal.** Como podemos ver en la Figura 21, esta función “mapea” los datos de entrada en un rango acotado por los valores $[0,1]$.
 - **Función hiperbólica.** Como se muestra en el centro de la Figura 21, esta función “mapea” los valores de entrada, como la función sigmoidal, pero, en este caso, entre los valores $[-1, 1]$.
 - **Función Relu,** también llamada *Rectified Linear Unit*. En esta función, cuando la entrada (z) sea negativa generará una salida igual a cero, y una salida igual a la entrada cuando la entrada sea positiva. Esta es la función usada en este proyecto.

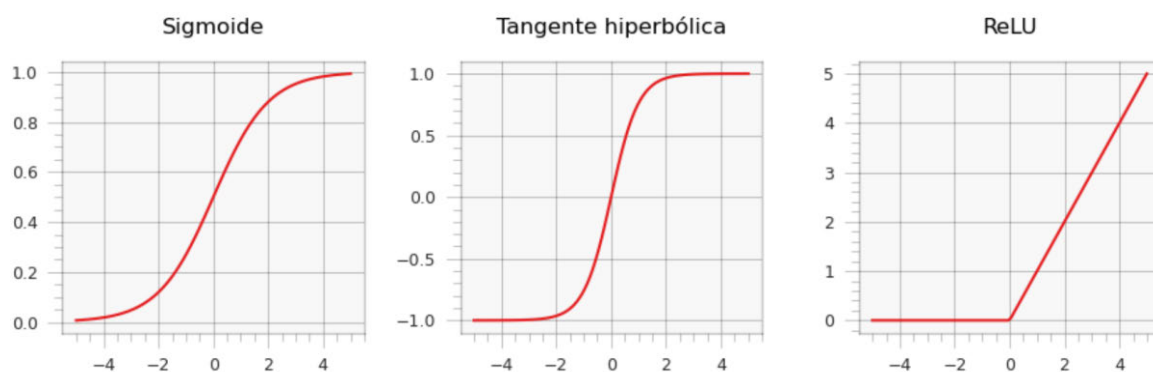


Figura 21. Representación de algunas funciones de activación [38].

3. Transmite la salida y a través de la línea de salida.

Este proceso se repite múltiples de veces, ya que no existe una única neurona, lo que da lugar a al perceptrón multicapa (MLP). Una red neuronal está formada típicamente por millones de neuronas interconectadas que forman, normalmente, tres o más capas diferentes. A continuación, en la Figura 22 se muestra la arquitectura de estas capas y su definición [40]:

- **Capa de entrada:** En esta capa se recibe la información directamente de las fuentes externas.
- **Capas ocultas:** Pueden ser más de una, dependiendo de la red neuronal que se quiera diseñar. Estas capas están "ocultas" dentro de la red, es decir, son capas internas sin contacto con el exterior. El número de capas ocultas oscila entre una y un número elevado. Determinan la topología de la red.
- **Capa de salida:** Es la capa encargada de transferir la información de la red hacia el exterior.

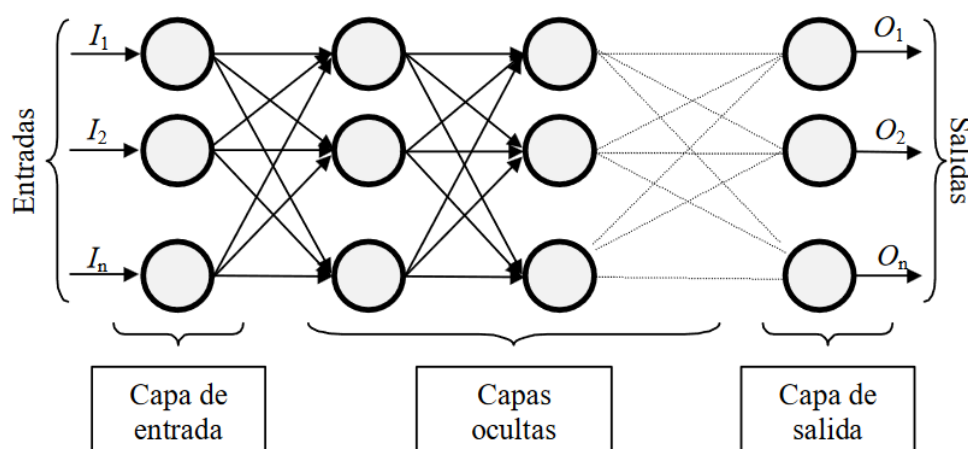


Figura 22. Ejemplo de una red neuronal totalmente conectada con múltiples capas ocultas [40].

3.3.1. Extracción de características

En *Deep Learning*, la extracción de las características es uno de los pasos que se realiza de manera automática a través de técnicas definidas, las cuales lo tienen integrado.

A continuación, se presentan las más comunes [41]:

- **Redes neuronales convolucionales (CNN).** Se caracterizan por su eficacia para la extracción de características en datos visuales.
- **Redes neuronales recurrentes (RNN).** Estas redes son utilizadas para datos secuenciales y temporales, ya que extraen las características a lo largo del tiempo. Entre ellas encontramos:
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- **Autoencoders.** Son redes que se utilizan para la reducción de dimensionalidad y la eliminación de ruido. Entre ellas distinguimos:
 - *Autoencoders* estándar (AE).
 - *Variational Autoencoders* (VAE).
 - *Conditional Variational Autoencoders* (CVAE).

- **Redes generativas Antagónicas (GANs).** Formadas por dos redes (generadora y discriminadora) compitiendo entre sí. La red generadora es la encargada de la creación de datos falsos que la red discriminadora debe de distinguir de los verdaderos. Fueron creadas con el propósito de la generación de nuevos datos.
- **Transformers.** Utilizan unos mecanismos de atención que les permiten centrarse en las partes más relevantes de la entrada, extrayendo sus características más significativas. Son muy útiles para el procesamiento del lenguaje natural. Entre sus modelos encontramos:
 - *Bidirectional Encoder Representations from Transformers (BERT).*
 - *Generative Pre-Trained Transformer (GPT),* la base de ChatGPT.

Todas y cada una de estas técnicas se centran en el aprendizaje de las representaciones de los datos más útiles para la tarea que se va a realizar, de manera que evitan que se realice de manera manual. Una de las principales ventajas del aprendizaje profundo, por lo tanto, es la definición de este paso de manera automática, es decir, la capacidad de aprender representaciones de características complejas de los datos de forma automática.

En este proyecto se ha partido de la técnica de *autoencoder* estándar de Marta Fraile [5], pero se ha modificado para usar un *autoencoder variacional*, tal como se detallará a lo largo de este capítulo y el siguiente.

3.4. Sobreajuste

Una de las desventajas de las redes neuronales es el denominado sobreajuste u *overfitting*. El *overfitting* ocurre cuando el modelo se ajusta demasiado bien a los detalles y el ruido del conjunto de datos de entrenamiento. Esto da lugar a que el modelo consiga un rendimiento excepcionalmente bueno en estos datos, los de entrenamiento, pero no capte patrones generalizables que se puedan aplicar a nuevos datos [42].

Existen múltiples de métodos para evitar que esto ocurra, entre los que se encuentran la validación cruzada, agregar datos de entrenamiento, eliminar funciones, etc [42]. Pero, como el objetivo de este proyecto no se centra en este paso, utilizaremos simplemente la parada anticipada. La parada anticipada consiste en lo mostrado en la Figura 24. Trata de encontrar la duración óptima del entrenamiento para evitar que el modelo se sobre-adapte. Esto es, se trata de parar cuando el error en los datos de validación empiece a aumentar, aun cuando el error en los datos de test siga disminuyendo.

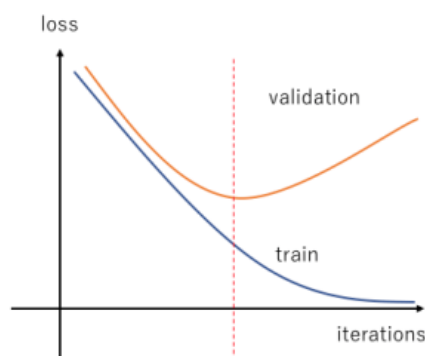


Figura 23. Representación del overfitting [42].

3.5. Redes densas

Las redes densas, también conocidas como *fully connected networks (FCNs)*, son uno de los tipos de redes neuronales, las cuales consisten en que cada neurona está conectada a todas las neuronas de la capa anterior y de la siguiente, como mostramos en la Figura 23. Esta arquitectura de redes neuronales permite que la información fluya directamente de una capa a otra sin necesidad de saltos [43].

Las redes densas se utilizan mayoritariamente en tareas de clasificación y reconocimiento de patrones, en los cuales los datos se caracterizan por tener una relación compleja y no-lineal. Por ello, este proyecto se desarrollará con redes densas.

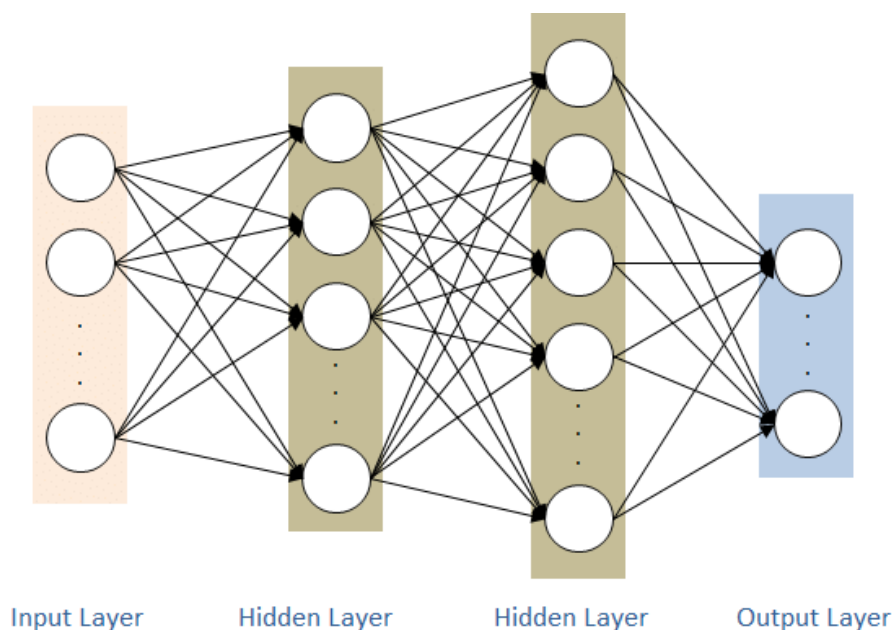


Figura 24. Fully connected Neural Network Layer [44].

3.6. Redes neuronales convolucionales

Las CNN están inspiradas por la organización de la corteza visual de los animales. Se diseñaron con el propósito de procesar datos estructurados en forma de cuadrícula, como imágenes, y han demostrado ser muy eficaces en tareas de reconocimiento y clasificación de imágenes con datos tridimensionales.

Las redes neuronales convolucionales se utilizan actualmente para tareas de clasificación y *computer vision*. Esto es debido a que las CNN son capaces de aprovechar la multiplicación de matrices de álgebra lineal para la identificación de patrones en una imagen. Una de las desventajas de estas redes es el alto coste computacional, lo que requiere el uso de unidades de procesamiento gráfico (GPU) para el entrenamiento de los modelos [45].

A continuación, en la Figura 25 se muestra la arquitectura de una CNN aplicada al reconocimiento de dígitos, la cual está formada por:

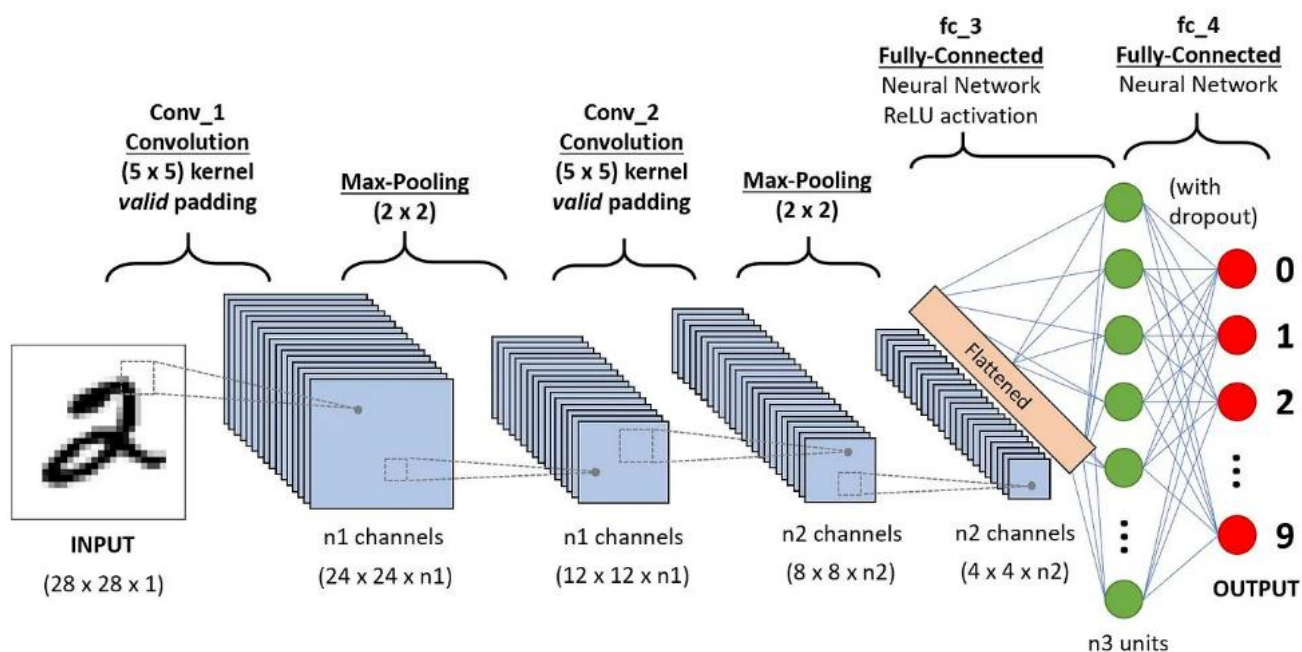


Figura 25. Arquitectura de las CNN aplicada al reconocimiento de dígitos [46].

- **Capas convoluciones** (Filtros o *kernels*, *stride* y *padding*)

Lo primero que se realiza en una CNN es la convolución, es decir, la operación matemática que aplica la función de ventana deslizante a una matriz de píxeles que representa una imagen. El deslizamiento de esta función está definido por el filtro o *kernel*. Pero una red convolucional no está formada únicamente por un filtro o *Kernel*, sino que en ella se aplican tantos filtros como fueran necesarios para reconocer el patrón deseado de la imagen.

- **Capas de pooling** (*max pooling*, *average pooling*)

Estas capas se caracterizan por la extracción de las características más importantes de la matriz convolucionada. Para ello, se aplican operaciones de agregación, las cuales reducen la dimensión de dicha matriz, de manera que se reduce la memoria utilizada en el entrenamiento de la red. Estas capas también tienen importancia para la reducción del sobreajuste.

Entre las funciones más comunes que se aplican se encuentran:

- *Max pooling*: representa el valor máximo del mapa de las características. Se muestra un ejemplo en la Figura 26.
- *Average-pooling*: representa la media de los valores.

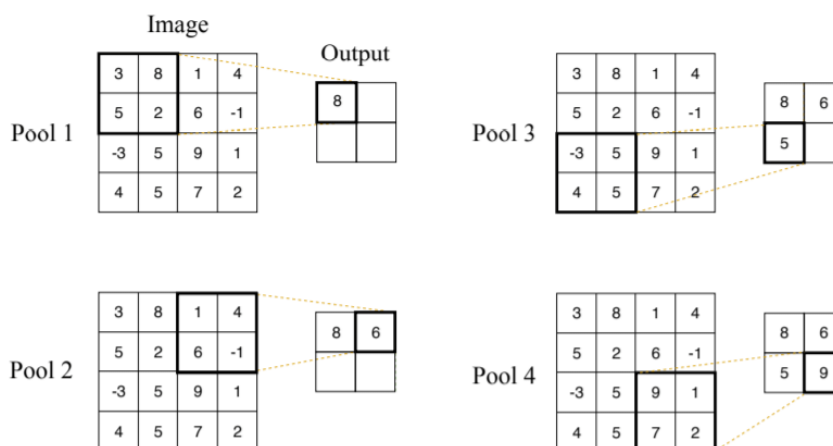


Figura 26. Representación de la función max pooling [45].

- **Capas de activación (ReLU)**

En esta capa se aplica la función ReLU explicada anteriormente, después de cada operación de convolución. Su función es el aprendizaje de las relaciones no lineales de las características de la imagen para poder identificar los patrones diferentes.

- **Capas completamente conectadas (fully connected)**

Estas capas representan las últimas capas de la red neuronal convolucional. Su entrada es un vector, que se obtiene aplanando (“Flattening”) la salida de la última capa convolucional.

3.7. Modelos de clasificación

3.7.1. Autocodificadores

Los auto-codificadores, traducidos del inglés *autoencoders*, son un tipo de redes neuronales profundas. Este tipo de redes consisten en comprimir la entrada en un código, generalmente de dimensión inferior y, a partir de la representación de los datos comprimidos, reconstruir la salida. Es decir, la salida está basada en reconstruir la entrada e intentar que sea lo más parecido posible. Es importante destacar que los datos de salida, aunque sean una reconstrucción, y por lo tanto, no sean iguales que la entrada, deben ser del mismo tamaño, como se muestra en la Figura 27. Este código en el que se comprime, es llamado espacio latente [47].

Aunque normalmente consta de dos bloques (codificador y decodificador), incluiremos también un tercero (espacio latente) debido a que tendrá gran protagonismo en este proyecto.

- **Codificador** o *encoder*, encargado de comprimir los datos de la entrada en el código o espacio latente.
- Código o **espacio latente**, caracterizado por ser una única capa de la ANN. La dimensión de esta capa depende del uso.
- **Decodificador** o *decoder*, encargado de reconstruir los datos de salida a partir del código o espacio latente.

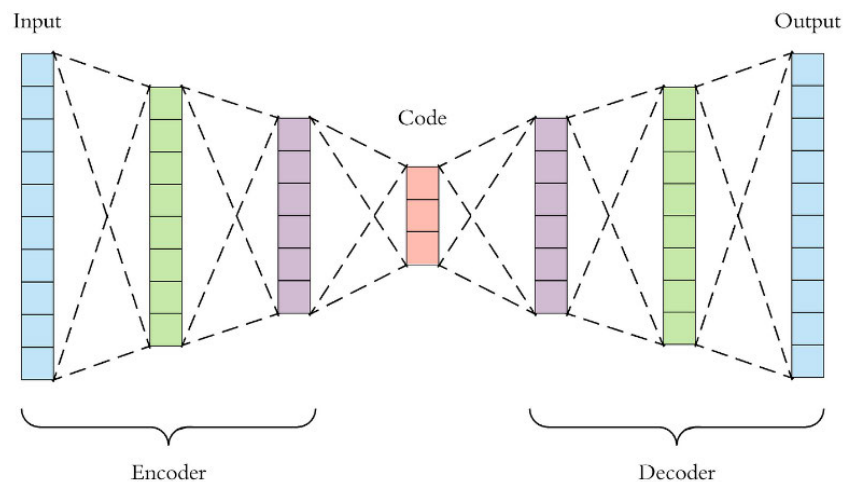


Figura 27. Autoencoder [47].

Este proyecto comienza a partir de un codificador simétrico, es decir, la entrada tiene el mismo tamaño a la salida. En este caso se define de tamaño 128 a 64.

Antes de implementar un *autoencoder*, es necesario definir 4 parámetros:

- **Tamaño del código.** Este parámetro define el número de nodos que formarán la parte intermedia, es decir, el espacio latente. Se destaca que cuanto más pequeño sea este parámetro, mejores resultados de compresión se obtienen.
- **Número de capas.** En la Figura 27, mostrada anteriormente, se muestran dos capas ocultas tanto en el codificador como en el decodificador, pero, el número de capas del *autoencoder* depende de su aplicación.
- **Número de nodos por capa.** Como se ha mencionado anteriormente, un *autoencoder* normalmente está configurado para que tenga mayor dimensión a la entrada y que la dimensión de las sucesivas capas del codificador vaya disminuyendo, pero también puede funcionar de manera inversa.
- **Función de pérdida.** Se utiliza el error cuadrático medio (mse) o entropía cruzada binaria. La utilización de uno u otro depende de los valores de entrada. Si los datos de entrada están acotados en el rango $[0, 1]$, se suele utilizar la entropía cruzada. En este proyecto se utilizará el error cuadrático medio.

Antes de crear el *autoencoder variacional* final, en este proyecto se ha comenzado utilizando un *autoencoder*. Se ha creado una red densa para generar el *autoencoder* basado en el error cuadrático medio (mse). Lo primero que se ha realizado ha sido un estudio del *overfitting*, con datos únicamente sin epilepsia. En el Trabajo de Fin de Grado de Marta Fraile [5] se utiliza un *autoencoder* con la intención de detectar crisis epilépticas, pero ese no es el objetivo de este proyecto, por lo que únicamente estudiamos el número de épocas (*epoch*) que son necesarias para entrenarlo aplicando la parada anticipada y visualizaremos el espacio latente.

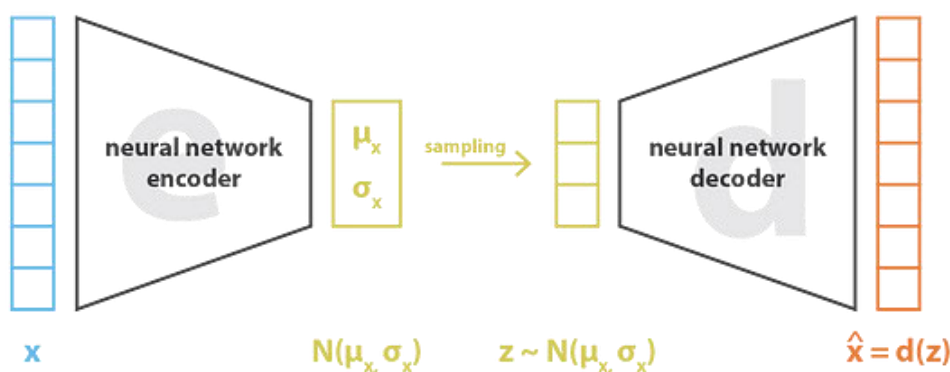
Este proyecto se basa en mejorar el aprendizaje profundo de un *autoencoder* estándar con un autocodificador variacional condicional, para lo que se va a definir a continuación el autocodificador variacional condicional.

3.7.2. Autocodificador Variacional

Un *autoencoder* variacional (VAE) es un modelo generativo que sigue la estructura. Los VAEs son utilizados, mayoritariamente, para la realización de tareas como la generación de datos, reducción de dimensionalidad y detección de anomalías.

A continuación, se explica la **arquitectura** de un *autoencoder* variacional [48], representado en la Figura 28.

- **Codificador** (*encoder*). Se encarga de codificar la entrada como una distribución sobre el espacio latente.
- **Espacio latente**. Es una codificación probabilística (típicamente Gaussiana) de los datos de entrada transformados por el codificador.
- **Decodificador** (*decoder*). Se define como el proceso inverso al del codificador. Debido a la naturaleza probabilística del espacio latente, los VAEs son capaces de generar nuevas salidas tomando muestras de la distribución Gaussiana del espacio latente y pasándolas al decodificador.



$$loss = ||x - \hat{x}||^2 + KL[N(\mu_x, \sigma_x), N(0, I)] = ||x - d(z)||^2 + KL[N(\mu_x, \sigma_x), N(0, I)]$$

Figura 28. Autoencoder Variacional [7].

En esta Figura 28, se muestra la función de pérdida del VAE, definida como la variable *loss*. A continuación, se explica esta función de la **pérdida de reconstrucción** más detalladamente:

Antes de continuar con la explicación, es importante destacar que los *autoencoders* no están diseñados para generar nuevos datos, sino para reconstruir datos a partir de las características del espacio latente.

Sin embargo, en un VAE se busca la generación de nuevos datos. Por ello, no solo se realiza el proceso de reducción, sino que además se determina la distribución de cada característica [49]. Esto implica modificar la función de pérdida [7]:

“Un codificador automático variacional se puede definir como un codificador automático cuyo entrenamiento se regulariza para evitar el sobreajuste y garantizar que el espacio latente tenga buenas propiedades que permitan el proceso generativo.”

El primer objetivo de los VAEs es que el espacio latente sea continuo, ya que, gracias a este espacio, el modelo será capaz de generar nuevos datos. Para ello, el VAE aprende dos vectores del mismo tamaño, los cuales representan la media μ de una Gaussiana para cada dimensión del espacio latente, que almacena el centro aproximado en el que una entrada debe ser codificada, y su desviación estándar σ , que controla cuanto se puede desviar ese centro en cualquiera de sus muestreos, y que dan lugar a un vector de variables aleatorias. Este es el vector que se utiliza para generar la salida pasándola por el decoder.

Por ello, al error cuadrático medio del *autoencoder* se le añade otro término en la función de pérdida, la llamada divergencia KL, la cual mide diferencia existente entre dos distribuciones de probabilidad. Es decir, la divergencia KL se encarga de medir cuánto se parece la distribución normal que queremos tener en el espacio latente a la distribución real obtenida al pasar los datos por el codificador.

3.7.3. Autocodificador Variacional Condicional

Una vez que se ha presentado el *Autoencoder Variacional*, se continúa con el *Conditional Variational Autoencoder*. Este tipo de autoencoder variacional incluye la "condición". Es decir, tanto el codificador como el decodificador reciben una condición a parte de los datos de entrada como se muestra en la Figura 29, llamada "c" [50] que indica la clase a la que pertenecen los datos de entrada.

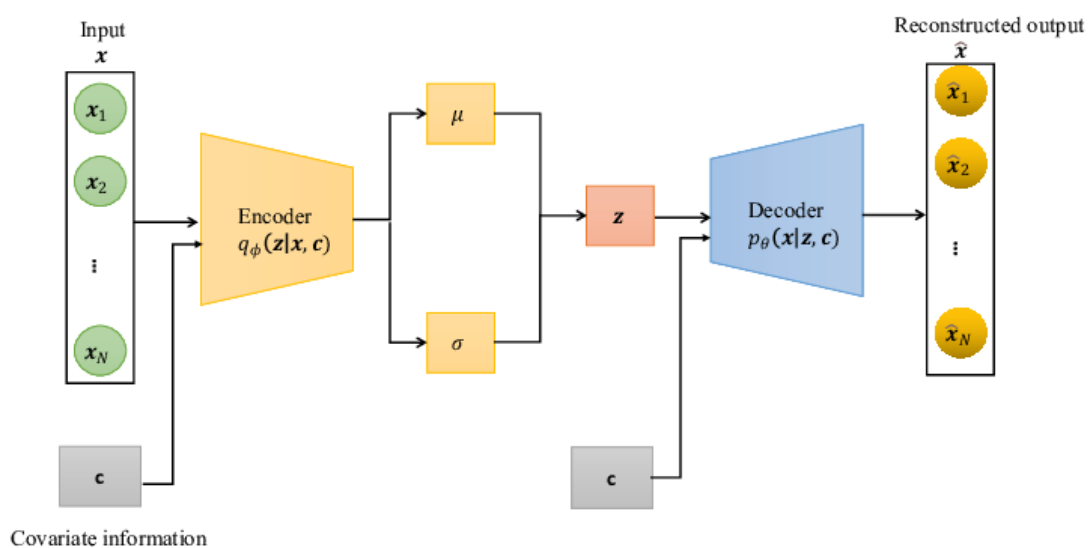


Figura 29. Arquitectura de CVAE [51].

Las características más significativas de los CVAEs son [50]:

- Entradas condicionales: la generación de nuevos datos se produce en función de las condiciones, lo que permite que sea una función más controlada y diversificada.
- Compresión de los datos en un espacio latente condicional que tiende a separar las muestras de las distintas clases.
- Un control más preciso en la generación de datos en comparación con los VAE que no incluyen condición.

En el Capítulo 5 se desarrollará más en detalle la implementación del CVAE de este proyecto.

Por último, cabe destacar que estos modelos se utilizan en diferentes campos como la generación de las imágenes, sistemas de recomendación de contenido, descubrimientos de medicamentos, procesamiento del lenguaje natural, etc [50].

3.8. Medidas de rendimiento

Para comprobar el correcto funcionamiento del modelo implementado existen diferentes formas de definir el rendimiento. Todas estas medidas de rendimiento utilizan la matriz de confusión, la comparación entre señales reales y señales predichas por el modelo.

En este proyecto se ha llevado a cabo una clasificación binaria. A continuación, en la Tabla 2 se muestran los diferentes valores de una matriz de confusión 2x2 en la que podemos ver cómo la parte de **observación** representa los momentos en los que sí existe crisis (positivos) o no (negativos) y la parte de **la predicción**, la cual representa los valores que se esperan obtener.

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (TP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (TN)

Tabla 2. Matriz de confusión.

Los valores de la matriz de confusión 2x2 que se utilizan en este proyecto se detallan en la Tabla 3.

Valor de la posición	Representación
(0,0)	Verdaderos Negativos (TN)
(0,1)	Falsos positivos (FP)
(1,0)	Falsos Negativos (FN)
(1,1)	Verdaderos Positivos (TP)

Tabla 3. Valores de la matriz de confusión 2x2.

Para poder analizar el comportamiento de nuestro modelo, a partir de la matriz de confusión se pueden obtener las siguientes medidas de rendimiento [52]:

- **Precisión:** Este parámetro mide la proporción de crisis que realmente se corresponden con crisis epilépticas. La fórmula para calcular este parámetro es la siguiente:

$$precision = \frac{TP}{TP + FP} \quad (ec.1)$$

- **Recall (Sensitivity):** Este parámetro mide la proporción de casos positivos (crisis epilépticas) detectados correctamente. La fórmula para calcular este parámetro es la siguiente:

$$recall = \frac{TP}{TP + FN} \quad (ec.2)$$

- **Accuracy:** Este parámetro mide la proporción de casos (tanto positivos como negativos) que el modelo ha acertado. La fórmula para calcular este parámetro es la siguiente:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (ec.3)$$

- **Specificity:** Este parámetro representa la proporción de casos negativos (no crisis) que se detectan de forma correcta. La fórmula para calcular este parámetro es la siguiente:

$$specificity = \frac{TN}{TN + FP} \quad (ec.4)$$

- **F_score:** Este parámetro representa la combinación de la precisión y el *recall*, es decir, tiene en cuenta los valores positivos (verdaderos y falsos), así como los falsos negativos, sin tener en cuenta número de casos negativos correctos, dado que este pueda estar desbalanceado. El valor del F-Score varía entre 0 (peor valor posible) y 1 (mejor valor posible).

$$F_{score} = \frac{TP}{TP + \frac{FP + FN}{2}} = 2 * \frac{recall * precision}{recall + precision} \quad (ec.5)$$

3.9. Discusión

El aprendizaje máquina incluye técnicas para que los ordenadores o computadoras puedan aprender a partir de datos. El aprendizaje profundo, una subrama, utiliza redes neuronales inspiradas en la

estructura del cerebro humano. Estas redes neuronales pueden formar redes densas, caracterizadas por tener cada neurona conectada a todas las neuronas de la capa siguiente, siendo fundamentales, pero menos eficientes para ciertos datos o las redes convolucionales, que son esenciales en el procesamiento de imágenes extrayendo las características locales. En cuanto a los modelos de clasificación, los *autoencoders* (AE) y los *autoencoders* variacionales VAE son útiles para la reducción de dimensionalidad y generación de datos, en los que se utilizan medidas de rendimiento como precisión y *recall*, que evalúan la efectividad de estos modelos.

El aprendizaje de máquina, por tanto, aprende de los datos de la entrada. Este método es muy importante dentro del análisis de señales biomédicas para mejorar el análisis de estas señales a partir de aprendizaje profundo, utilizando redes neuronales. En este proyecto, para detectar crisis epilépticas, los VAE se caracterizan por su utilidad ya que, estos tienen la capacidad de aprender representaciones latentes de las señales cerebrales, identificando los patrones anómalos asociados con las crisis epilépticas. Para evaluar la efectividad del VAE, y poder determinar la detección de crisis de manera precisa, se utilizarán las medidas de precisión y *recall*.

4. Metodología y resultados

4.1 Introducción

Este capítulo está dedicado a la explicación de la metodología aplicada para validar los métodos desarrollados y los resultados obtenidos en este proyecto, incluyendo la base de datos que ha sido utilizada. Se mostrarán las partes del código en Python más importantes para la obtención de los resultados que se comentan a continuación de la implementación del *Conditional Variational Autoencoder*.

4.2 Base de datos

La base de datos utilizada en este proyecto es CHB-MIT Scalp EEG [4], una base de datos pública. Esta base de datos incluye registros de EEG pertenecientes a los niños de un Hospital de Boston, quienes padecen de convulsiones intratables. El objetivo de estos registros fue evaluar si era posible someterles a una intervención quirúrgica. Para este estudio, los pacientes fueron monitorizados durante un periodo de tiempo equivalente a varios días, en los que se les había retirado la medicación anticonvulsiva, para así poder caracterizar las crisis que sufrían [4]. De esta base de datos se utilizará el paciente 1, aunque la base de datos incluye 24 pacientes en total. En la Tabla 4 se muestran las características básicas de las señales que proporciona la base de datos de los 24 pacientes.

Patient	Sex	Age	Total recordings duration [h:m:s]	Seizures	Avg. seizure duration [s]	Std	Min/Max duration [s]	Total ictal time [min]
chb01	F	11	40:33:08	7	63.14	28.21	27/101	7.37
chb02	M	11	35:15:59	3	57.33	34.18	9/82	2.87
chb03	F	14	38:00:06	7	57.43	7.76	47/69	6.7
chb04	M	22	156:03:54	4	94.5	26.74	49/116	6.3
chb05	F	7	39:00:10	5	111.6	8.45	96/120	9.3
chb06	F	1.5	66:44:06	10	15.3	2.72	12/20	2.55
chb07	F	14,5	67:03:08	3	108.33	24.85	86/143	5.42
chb08	M	3.5	20:00:23	5	183.8	44.0	134/264	15.32
chb09	F	10	67:52:18	4	69.0	6.67	62/79	4.6
chb10	M	3	50:01:24	7	63.86	15.96	35/89	7.45
chb11	F	12	34:47:37	3	268.67	341.79	22/752	13.43
chb12	F	2	23:41:40	40	36.88	15.98	13/97	24.58
chb13	F	3	33:00:00	12	44.58	20.58	17/70	8.92
chb14	F	9	26:00:00	8	21.12	8.12	14/41	2.82
chb15	M	16	40:00:36	20	99.6	52.23	31/205	33.2
chb16	F	7	19:00:00	10	8.4	2.15	6/14	1.4
chb17	F	12	21:00:24	3	97.67	12.28	88/115	4.88
chb18	F	18	35:38:05	6	52.83	13.17	30/68	5.28
chb19	F	19	29:55:46	3	78.67	1.7	77/81	3.93
chb20	F	6	27:36:06	8	36.75	5.85	29/49	4.9
chb21	F	9	31:00:11	4	49.75	24.7	12/81	3.32
chb22	F	6	26:33:30	3	68.0	7.12	58/74	3.4
chb23	-	-	21:17:47	7	60.57	30.16	20/113	7.07
chb24	F	13	32:49:49	16	31.94	17.83	16/70	8.52

Tabla 4. Información básica de cada paciente de la base de datos CHB-MIT [4].

En esta Tabla 4 se diferencia la duración de las crisis, la duración de la grabación (la mayoría de las grabaciones presentan una duración de 1 hora, excepto en el paciente 10 de dos horas, y las de los pacientes 4, 6, 7, 9 y 23, que tienen una duración de cuatro horas), los datos ictales, así como las diferencias de edades y género. Las grabaciones agrupan los datos de 22 pacientes en total, entre los que se encuentran 5 hombres de 3 a 22 años y 17 mujeres de 1,5 a 19 años. Las grabaciones se caracterizan por una duración de 1 hora aproximadamente, utilizando una frecuencia de muestreo de 256 Hz con precisión de 16 bits. Por último, cabe destacar que todas las grabaciones fueron realizadas siguiendo el estándar Internacional 10/20 en una configuración diferencial.

4.2.1. Preparación de los datos

Antes de comenzar con la explicación de los métodos implementados en este proyecto y sus resultados, es importante conocer cómo están estructurados los datos. La información almacenada en cada una de las señales se caracteriza por tener dos dimensiones, el número de muestras y los canales registrados. Como se ha explicado, la señal EEG se registra a través de un conjunto de electrodos, y en estos registros cada canal es diferencial, es decir, el registro es el resultado de la diferencia de dos señales obtenidas con dos electrodos. Esta manera de registrar las señales da lugar a que la información se forme a partir de un número determinado de muestras y el número de canales diferenciales utilizados. A partir de esto, la señal es dividida en varios segmentos, denominados ventanas. Estas ventanas son necesarias ya que, como se ha comentado en capítulos anteriores, la señal EEG no es estacionaria y sufre muchos cambios debido a la variación de la actividad mental, y por tanto es necesario analizarla en periodos de tiempo más cortos. Para este proceso, la señal es muestreada a 256 Hz. Si una señal, por ejemplo, se caracteriza por una dimensión (1; 512; 21), quiere decir que la señal posee 21 canales de duración 2 segundos (512 muestras con muestreo a 256 Hz). El número de ventanas de cada señal se representa en la dimensión adicional añadida.

En este proyecto se va a entrenar un modelo de aprendizaje supervisado, es decir, las señales tendrán etiquetas que las califiquen como muestras con crisis epiléptica o con ausencia de esta. A continuación, se presenta cómo se han obtenido los ficheros necesarios para la implementación, así como su estructura.

Es importante destacar que, a diferencia de la organización de los datos en el *autoencoder* del TFG de Marta Fraile [5], en este modelo de *conditional variational autoencoder* el modelo se entrenará tanto con datos con crisis como sin ella. En la Figura 30 se presenta la estructura de los datos de manera visual.

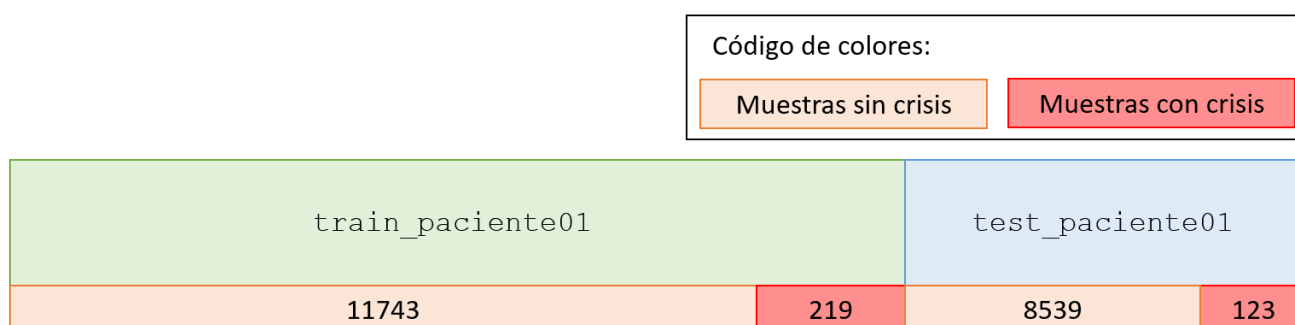


Figura 30. Organización de los datos.

- **Obtención de los datos de entrenamiento.** En la Figura 31 se muestra cómo se ha procesado la señal del paciente 1, obteniendo los ficheros necesarios para el entrenamiento del modelo almacenados en `train_paciente01`. Estas señales son almacenadas en tiempo. Como se ha mencionado, en este modelo se incluyen en el entrenamiento señales con epilepsia (`chb01_03`, `chb01_04`, `chb01_15`, `chb01_16`, `chb01_18`) y sin epilepsia (los restantes).

```
# Get the dataframe with all the info of a patient
patient_data_df_train = generate_patient_data(patient_data_path, registers_train, channel_index)

# Store dataframe as pickle
patient_data_df_train.to_pickle(f'../data/processed/{patient}_data_concatenated.pickle')
```

```
Register chb01_01.edf read and concatenated
Register chb01_02.edf read and concatenated
Register chb01_03.edf read and concatenated
Register chb01_04.edf read and concatenated
Register chb01_05.edf read and concatenated
Register chb01_06.edf read and concatenated
Register chb01_09.edf read and concatenated
Register chb01_14.edf read and concatenated
Register chb01_15.edf read and concatenated
Register chb01_16.edf read and concatenated
Register chb01_17.edf read and concatenated
Register chb01_18.edf read and concatenated
Register chb01_20.edf read and concatenated
```

```
registro_train = concatenar_seizures_filtrado(patient_data_df_train)
```

```
(11740928, 21)
(11740928, 21)
```

```
file = open('train_paciente01', 'wb')

pickle.dump(registro_train, file)

file.close()
```

Figura 31. Obtención de las señales para el entrenamiento.

- **Obtención de los datos de test.** En la Figura 32 se muestra cómo se ha procesado la señal del paciente 1, obteniendo los ficheros necesarios para la validación del modelo almacenados en `test_paciente01`. Estas señales son almacenadas en tiempo. Como se ha mencionado, en este modelo se incluyen en la validación señales con epilepsia (`chb01_21`, `chb01_26`) y sin epilepsia (los restantes).

```
# Get the dataframe with all the info of a patient
patient_data_df_test = generate_patient_data(patient_data_path, registers_test, channel_index)

# Store dataframe as pickle
patient_data_df_test.to_pickle(f'../data/processed/{patient}_data_concatenated.pickle')
```

```
Register chb01_21.edf read and concatenated
Register chb01_24.edf read and concatenated
Register chb01_25.edf read and concatenated
Register chb01_26.edf read and concatenated
Register chb01_27.edf read and concatenated
Register chb01_29.edf read and concatenated
```

```
registro_test = concatenar_seizures_filtrado(patient_data_df_test)
```

```
(4435200, 21)
(4435200, 21)
```

```
file = open('test_paciente01', 'wb')

pickle.dump(registro_test, file)

file.close()
```

Figura 32. Obtención de las señales para la validación.

- **Obtención de los booleanos de los momentos con epilepsia del paciente.** Para poder obtener los booleanos con epilepsia, es necesario separar todas las señales con epilepsia, como se muestra en la Figura 33, de la misma manera que se han cargado las señales de entrenamiento y validación.

```
# Get the dataframe with all the info of a patient
patient_data_df_seizures = generate_patient_data(patient_data_path, registers_epilept, channel_index)

# Store dataframe as pickle
patient_data_df_seizures.to_pickle(f'../data/processed/{patient}_data_concatenated.pickle')
```

```
Register chb01_03.edf read and concatenated
Register chb01_04.edf read and concatenated
Register chb01_15.edf read and concatenated
Register chb01_16.edf read and concatenated
Register chb01_18.edf read and concatenated
Register chb01_21.edf read and concatenated
Register chb01_26.edf read and concatenated
```

```
registro_seizures = concatenar_seizures_filtrado(patient_data_df_seizures)
```

```
(6124800, 21)  
(6124800, 21)
```

```
file = open('epilepsia_paciente01', 'wb')  
  
pickle.dump(registro_seizures, file)  
  
file.close()
```

Figura 33. Obtención de las señales con epilepsia.

Una vez que se obtiene el archivo con las señales con epilepsia, se obtienen los booleanos de los momentos con epilepsia del paciente, implementado tal como se muestran en la Figura 34, y almacenados en `epilepsia_paciente01`. De esta manera, se podrán etiquetar los datos que se van a utilizar. Se almacenan en `targets_paciente01`.

```
targets = obtener_bools_seizures(patient_data_df_seizures)
```

```
file = open('targets_paciente01', 'wb')  
  
pickle.dump(targets, file)  
  
file.close()
```

Figura 34. Obtención de los booleanos de las señales con epilepsia.

4.3 Autocodificador con red densa

En el estudio realizado en el TFG de Marta Fraile [5] se comprobó que la red no presentaba ninguna variación significativa en su rendimiento entre los datos de tiempo y los datos de frecuencia, por lo que, en este desarrollo, se ha decidido trabajar en frecuencia. Sin embargo, sobre dicha red se han realizado diferentes estudios adicionales, como la visualización del espacio latente y el análisis del *overfitting* que sufre la red con diferentes épocas de entrenamiento, los cuales se presentarán a continuación.

4.3.1. Análisis del espacio latente

Una de las diferencias más características entre un *autoencoder* estándar y un *variational autoencoder* es el espacio latente.

Como vemos en la Figura 35, el espacio latente del *autoencoder* estándar (AE) se representa por z y, dado que su objetivo principal es la reconstrucción de las señales de entrada, como se ha detallado en capítulos anteriores, no impone ninguna estructura en el espacio latente.

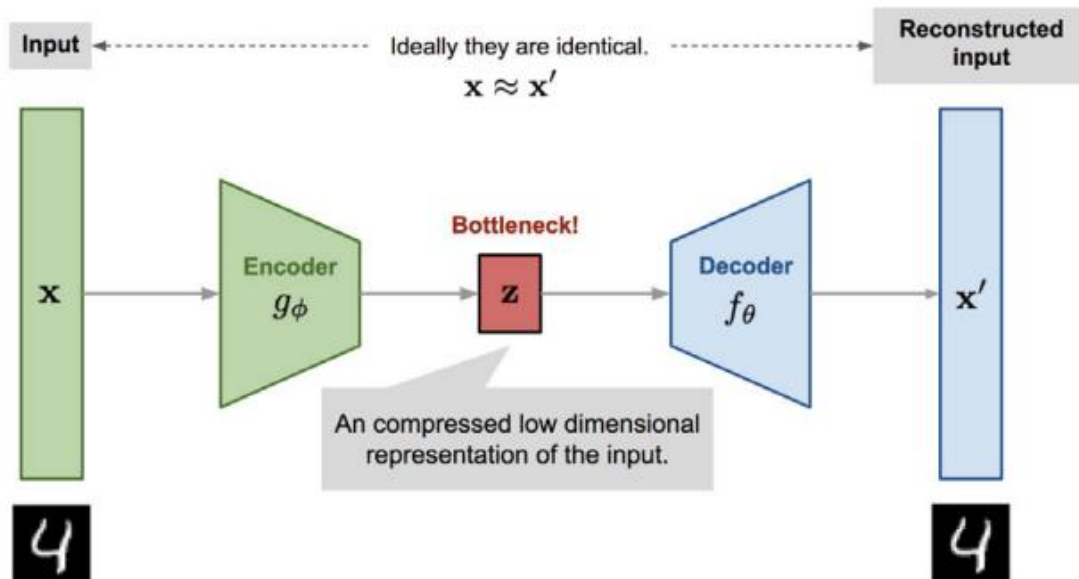


Figura 35. Espacio latente del autoencoder [53].

Sin embargo, si analizamos el espacio latente mostrado en la Figura 36, éste es utilizado por el *decoder* del *autoencoder* variacional no solo para la reconstrucción de señales, sino también para la generación de nuevas muestras similares a las que se le ha introducido a la entrada.

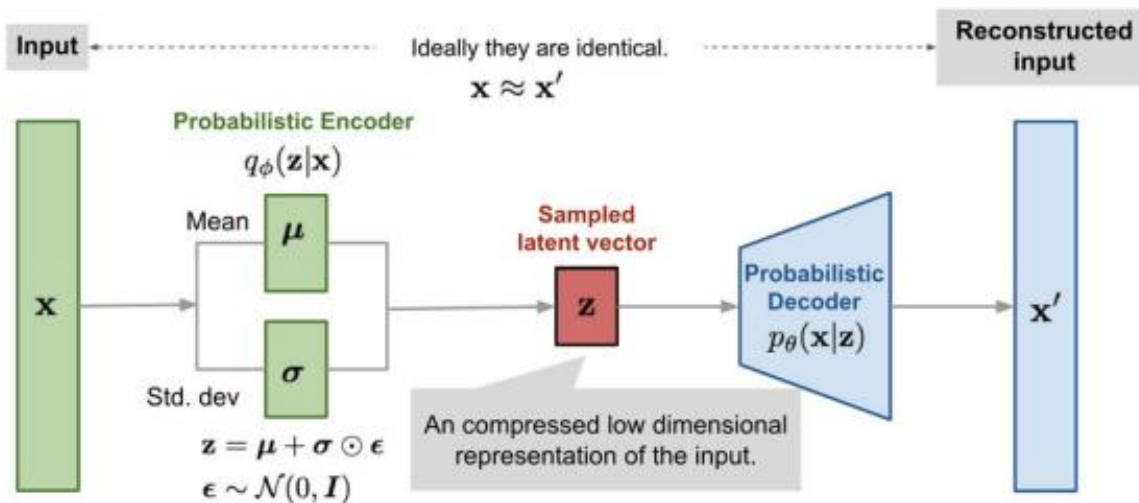


Figura 36. Espacio latente del variational autoencoder [53].

A partir de la Figura 36 se muestra como el *encoder* produce un espacio latente asociado con una distribución de probabilidad. Para poder conseguir esto es necesario disponer de una estructura apropiada en el espacio latente.

Viendo esta diferencia en detalle, se muestra el análisis que se ha desarrollado a través de una red densa de un *autoencoder* y un *variational autoencoder*.

En general, el espacio latente del VAE tiene una dimensión muy elevada. Por lo tanto, para poder visualizarlo es necesario utilizar diferentes técnicas de reducción de dimensionalidad que nos permiten representarlo en dos o tres dimensiones. A continuación, se describen brevemente las tres técnicas usadas junto con los resultados obtenidos:

- **Principal Component Analysis (PCA)**

La técnica denominada PCA, se caracteriza por ser un método lineal, el cual se implementa con el objetivo de reducir la dimensionalidad [54]. Esto se realiza a través de una transformación lineal de los datos de entrada en un conjunto de variables linealmente incorreladas, llamadas componentes principales, que maximizan la varianza de los datos. Los datos, por lo tanto, se proyectan en estas nuevas direcciones.

Su implementación es muy sencilla, tiene una gran eficiencia computacional y es útil para casos en que las relaciones lineales predominan. Como desventaja no pueden capturar relaciones no lineales entre las variables. Se utiliza en análisis exploratorio de datos, compresión de datos, eliminación de ruido, etc [54].

A continuación, se muestra cómo se implementa esta técnica, de manera muy sencilla, y sus resultados. El primer espacio latente obtenido es el del *autoencoder* estándar que, como se muestra en la Figura 37, es muy sencillo de implementar.

```
# Obtener la representación del espacio latente para X_train y X_test_NoCrisis
espacio_latente_train = encoder.predict(X_train)
espacio_latente_test = encoder.predict(X_test_NoCrisis)
```

```
# Reducir la dimensionalidad del espacio latente usando PCA
pca = PCA(n_components=2)
espacio_latente_PCA = pca.fit_transform(espacio_latente_train)
```

Figura 37. Implementación de la técnica PCA en un autoencoder.

En el *autoencoder* variacional se realiza de manera similar aunque, como se detalla en la Figura 38, previamente se deben almacenar las variables `z_mean_train` y `z_log_var_train`, que representan la media y la desviación de la log-varianza del modelo.

```
# Obtener representaciones en el espacio latente para X_train_flat
z_mean_train, z_log_var_train = encoder.predict(X_train_flat)
```

```
# Reducir dimensiones utilizando PCA
pca = PCA(n_components=2)
z_mean_pca = pca.fit_transform(z_mean_train)
```

Figura 38. Implementación de la técnica PCA en el autoencoder variacional.

Una vez implementada la visualización del espacio latente en ambos modelos usando PCA con solo 2 componentes principales, se analizan los resultados en la Figura 39:

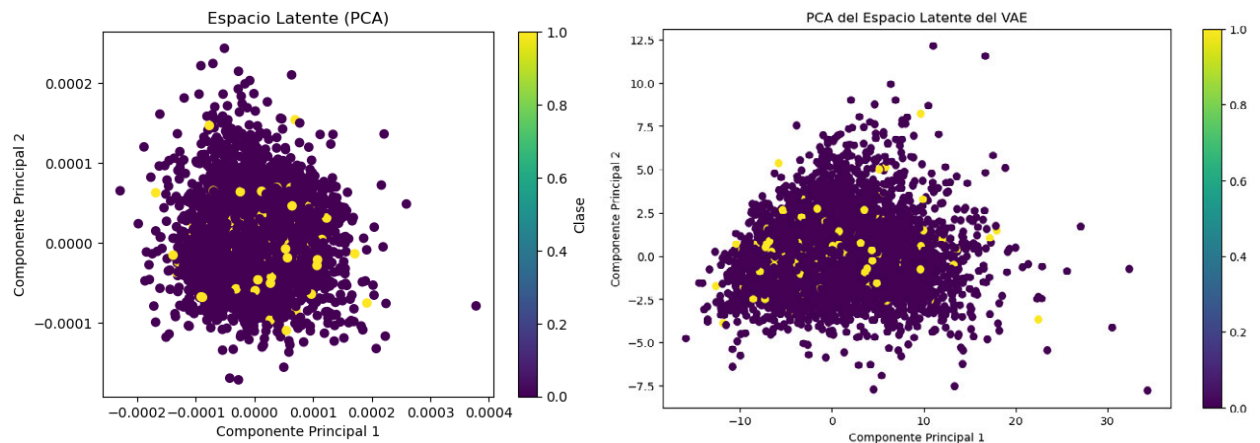


Figura 39. Espacio latente con PCA del autoencoder (izquierda) y del autoencoder variacional (derecha).

Como se puede apreciar, esta técnica no resulta útil en estos modelos, ya que las muestras con crisis (amarillo) están dispersas por todo el espacio. Más adelante se analizará con el *autoencoder* condicional variacional.

- ***T-Distributed Stochastic Neighbor Embedding*** (t-SNE)

Una de las técnicas más utilizadas para la reducción de dimensionalidad es t-SNE [55], la cual se caracteriza por ser no lineal y está basada en las proximidades, como su propio nombre indica. Para ello, se realiza un “mapeo” de los puntos de datos de un espacio de alta dimensionalidad (la del espacio latente) a un espacio de menor dimensionalidad (dos o tres típicamente) de manera que las similitudes entre pares de puntos se preserven tanto como sea posible.

Se utiliza para la visualización, ya que es muy efectivo y preserva bien las relaciones locales, aunque la captura de relaciones no lineales entre las variables es bastante compleja.

Esta técnica presenta varias desventajas frente a las demás, como puede ser el coste computacional, que es el más alto de todos, además de su complejidad para ser implementada. Cabe destacar que no es determinista, lo que implica que puede producir diferentes resultados en ejecuciones distintas. Otra complicación de este método es que cuenta con dos parámetros que se deben ajustar para conseguir resultados óptimos [56].

Se utiliza en visualización de datos complejos como en aprendizaje no supervisado, o *clusters* de datos de alta dimensionalidad [56].

A continuación, se muestra cómo se implementa esta técnica y sus resultados. De la misma manera que con PCA, se ha implementado primero en el *autoencoder* estándar, como se muestra en la Figura 40.

```
# Obtener la representación del espacio latente para X_train y X_test_NoCrisis
espacio_latente_train = encoder.predict(X_train)
espacio_latente_test = encoder.predict(X_test_NoCrisis)

# Aplicar t-SNE directamente a los datos
tsne = TSNE(n_components=2, random_state=42)
espacio_latente_tsne = tsne.fit_transform(espacio_latente_train)
```

Figura 40. Implementación de la técnica t-SNE en un autoencoder.

De manera similar, se implementa en el *autoencoder* variacional (Figura 41):

```
# Aplicar t-SNE
tsne = TSNE(n_components=2, random_state=42)
z_mean_tsne = tsne.fit_transform(z_mean_train)
```

Figura 41. Implementación de la técnica t-SNE en el *autoencoder* variacional.

Como se ha indicado antes, los resultados obtenidos con esta técnica dependen de los valores *perplexity* y *learning-rate*, por lo que se han visualizado los espacios latentes, Figura 42, con distintas combinaciones de estos parámetros.

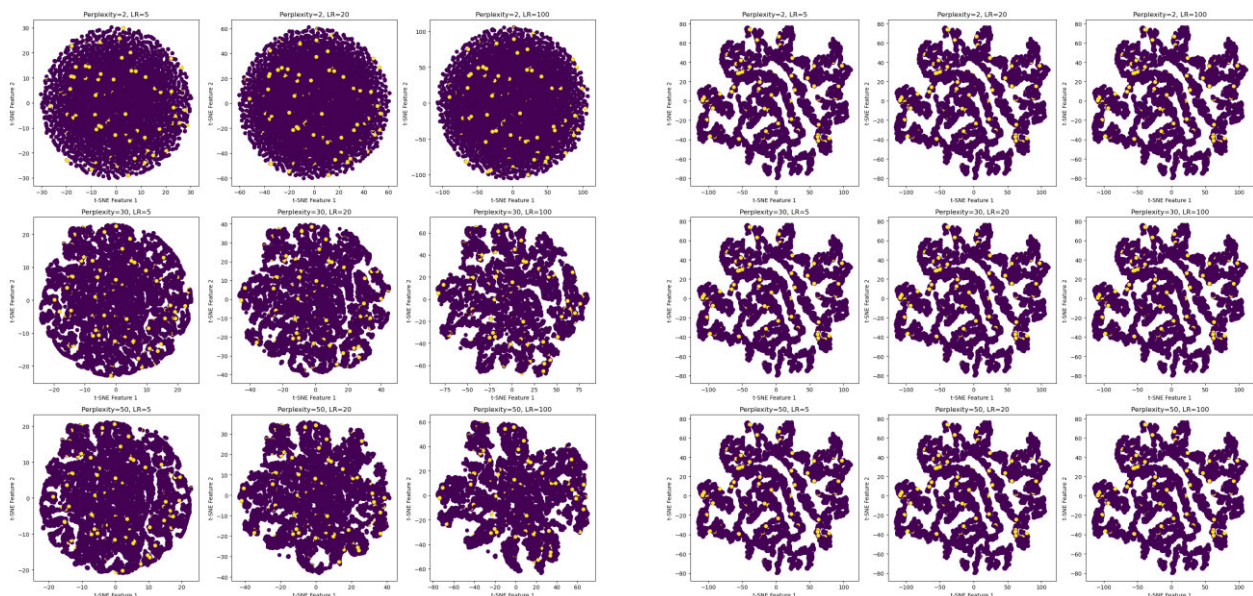


Figura 42. Espacio latente con t-SNE del autoencoder (izquierda) y del autoencoder variacional (derecha).

Al igual que antes, con esta técnica se comprueba que, a pesar de que el coste computacional ha sido mucho más elevado, los resultados obtenidos siguen sin ser satisfactorios.

- **Isometric Mapping** (isomap)

Esta técnica, una vez más, tiene como objetivo la reducción de dimensionalidad. Se caracteriza por ser no lineal, y por estar basada en grafos [57]. Es una extensión del *Multi-Dimensional Scaling* (MDS), [57] que preserva las distancias geodésicas (distancias en el gráfico de vecinos más cercanos) en un espacio de menor dimensionalidad. Se construye a través de un grafo de vecinos más cercanos de los datos y calcula las distancias más cortas entre todos los pares de puntos en este grafo. Luego, aplica MDS a estas distancias.

Este método se caracteriza por capturar las relaciones no lineales y, además, preservar tanto la estructura local como global de los datos. Como principales desventajas, el costo computacional es elevado para grandes conjuntos de datos y es sensible al número de vecinos más cercanos elegidos. Su uso se centra en la visualización y análisis de datos de alta dimensionalidad donde la estructura no lineal no pasa desapercibida.

Como en PCA y t-SNE se ha implementado para los dos modelos, como se muestra en la Figura 43 para el *autoencoder* y en la Figura 44 para el *autoencoder* variacional.

```
#Obtener la reconstrucción de los datos (salida del autoencoder)
espacio_latente_isomap_train = encoder.predict(X_train)
```

374/374 ————— 1s 1ms/step

```
from sklearn.manifold import Isomap
# Aplicar Isomap para reducir la dimensionalidad de las representaciones del espacio latente a 2 dimensiones
isomap = Isomap(n_components=2)
espacio_latente_isomap_train = isomap.fit_transform(espacio_latente_train)
```

Figura 43. Implementación de la técnica Isomap en el autoencoder.

```
# Reducir dimensiones utilizando Isomap
isomap = Isomap(n_components=2)
z_mean_isomap = isomap.fit_transform(z_mean_train)
```

Figura 44. Implementación de la técnica Isomap en el autoencoder variacional.

A continuación, se muestran los resultados obtenidos en la Figura 45.

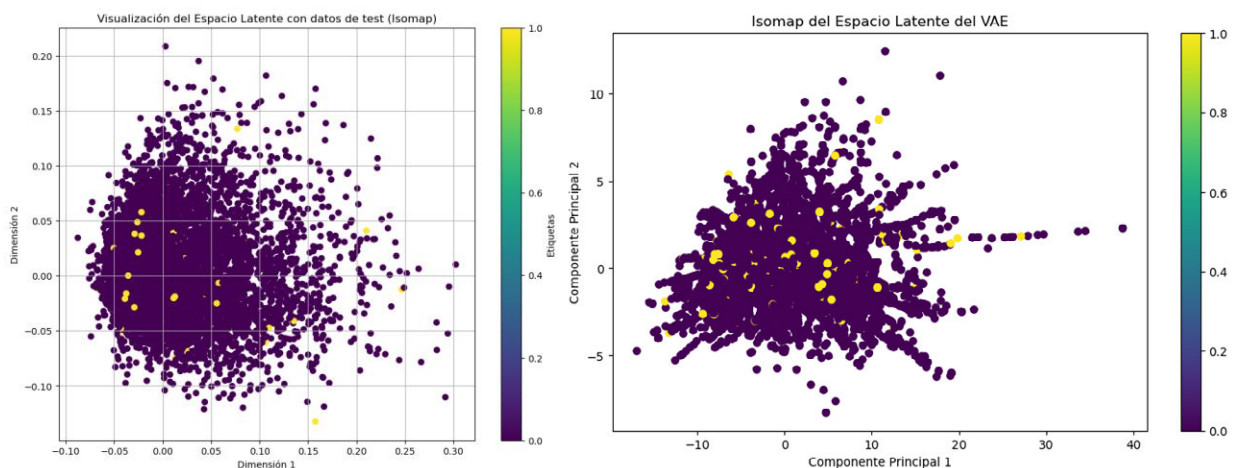


Figura 45. Espacio latente con Isomap del autoencoder (izquierda) y del autoencoder variacional (izquierda).

De la misma manera que en t-SNE, se ha comprobado que la técnica de Isomap tiene un gasto computacional demasiado elevado. Además, en los resultados podemos observar que no se diferencian con claridad las nubes de puntos correspondientes a crisis y no crisis, por lo que no está separando las características de manera adecuada. En la Figura 45 se muestra cómo las señales con crisis de valor 1 (amarillas) están mezcladas con las señales que no presentan crisis (moradas).

Por último, se resumen las tres técnicas, una vez ya implementadas y comprobadas, en la Tabla 5.

	PCA	t-SNE	Isomap
Método	Lineal	No lineal	No lineal
Eficiencia computacional	Eficiente	Muy poco eficiente	Poco eficiente
Preservación de la estructura	Preserva las relaciones lineales y la varianza global	Preserva las relaciones locales	Preserva tanto las relaciones locales como las globales.
Utilidad	Compresión y análisis general	Visualización de alta dimensionalidad	Análisis de estructura no lineal y global

Tabla 5. Resumen de las técnicas de visualización de los datos.

4.3.2. Análisis del sobreajuste

Con el autoencoder del TFG de Marta Fraile [5], también se ha aprovechado la oportunidad para estudiar el *overfitting* que presenta la red del codificador con los datos del `paciente_01` en una red compuesta por dos capas de 64 y 32 neuronas respectivamente.

Para ello, se han guardado los datos en dos variables diferentes, `loss` y `val_loss` y se han representado con entrenamientos desde 15 épocas a 200 como se muestra en la Figura 46, de manera visual, y en la Figura 47, de manera numérica.

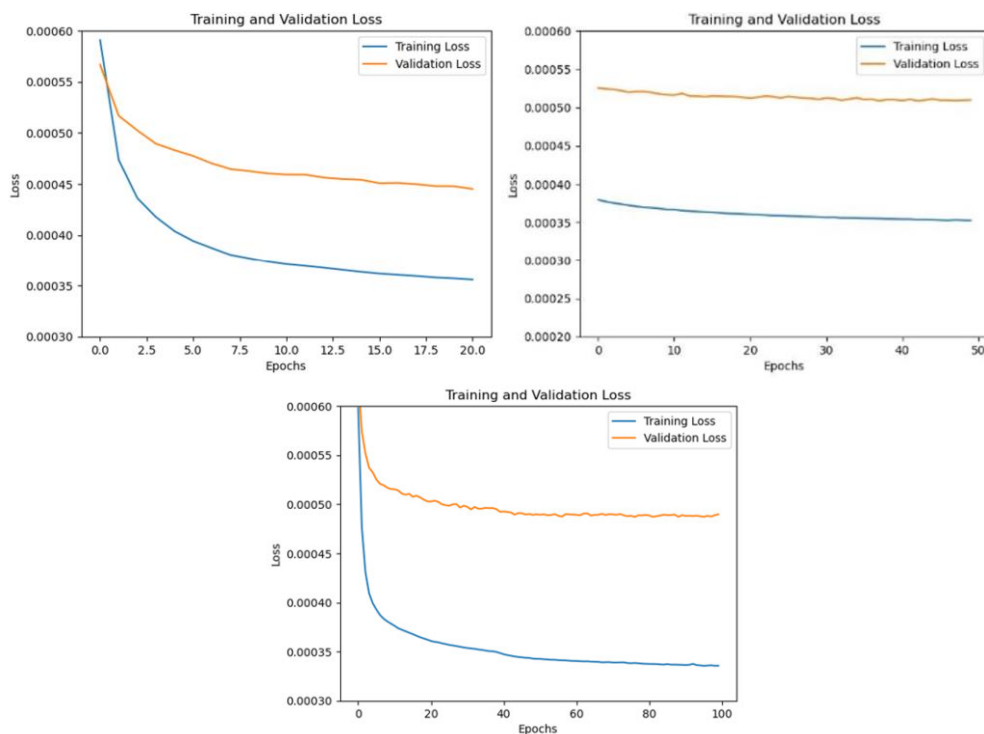


Figura 46. Análisis visual del overfitting en autoencoder.

```

492/492 ██████████ 11s 22ms/step - loss: 3.3588e-04 - val_loss: 4.8902e-04
Epoch 10/25
492/492 ██████████ 11s 22ms/step - loss: 3.3129e-04 - val_loss: 4.8940e-04
Epoch 11/25
492/492 ██████████ 11s 22ms/step - loss: 3.3410e-04 - val_loss: 4.8746e-04
Epoch 12/25
492/492 ██████████ 11s 22ms/step - loss: 3.3406e-04 - val_loss: 4.8839e-04
Epoch 13/25
492/492 ██████████ 11s 23ms/step - loss: 3.2988e-04 - val_loss: 4.8993e-04
Epoch 14/25
492/492 ██████████ 11s 23ms/step - loss: 3.3353e-04 - val_loss: 4.8957e-04
Epoch 15/25
492/492 ██████████ 11s 22ms/step - loss: 3.3564e-04 - val_loss: 4.8911e-04
Epoch 16/25
492/492 ██████████ 11s 23ms/step - loss: 3.3004e-04 - val_loss: 4.8839e-04
Epoch 17/25
492/492 ██████████ 11s 23ms/step - loss: 3.3700e-04 - val_loss: 4.8933e-04
Epoch 18/25
492/492 ██████████ 11s 23ms/step - loss: 3.3453e-04 - val_loss: 4.8802e-04
Epoch 19/25
492/492 ██████████ 11s 23ms/step - loss: 3.2873e-04 - val_loss: 4.8886e-04
Epoch 20/25
492/492 ██████████ 11s 23ms/step - loss: 3.2810e-04 - val_loss: 4.8832e-04
Epoch 21/25
492/492 ██████████ 11s 23ms/step - loss: 3.3308e-04 - val_loss: 4.8870e-04
Epoch 22/25
492/492 ██████████ 12s 23ms/step - loss: 3.2994e-04 - val_loss: 4.8765e-04
Epoch 23/25
492/492 ██████████ 11s 23ms/step - loss: 3.3308e-04 - val_loss: 4.8784e-04

```

Figura 47. Análisis numérico del overfitting en autoencoder.

Como se puede observar en las figuras anteriores existe una gran diferencia entre la pérdida (*loss*) de entrenamiento y validación, lo que indica claramente la presencia de sobreajuste (*overfitting*). Además, a partir de 20 iteraciones (épocas), la diferencia del error es mínima, por lo que se continúa trabajando con este dato para el posterior codificador variacional.

4.4 Autocodificador Variacional Condicional

Con el análisis del *autoencoder*, del *autoencoder variacional* y, con las características seleccionadas, se continúa con la implementación del modelo final: *Autoencoder Variational Condional* (CVAE)

El CVAE se ha construido para separar las dos clases de interés (crisis y no crisis) en el espacio latente y así poder incluir el algoritmo KNN en nuestro análisis y clasificación de las señales EEG. A continuación, se explica la implementación del CVAE y el algoritmo KNN en este proyecto. En primer lugar, como se ha comentado en la preparación de los datos, cabe destacar que el modelo trabajará en aprendizaje supervisado, con datos de entrenamiento y de test/validación que incluyen tanto señales con epilepsia como con ausencia de esta. Estas señales serán etiquetadas, con la variable *targets* con valores binarios: 0 = no crisis, 1 = crisis. Una vez que los datos están cargados, ha sido necesario incluir varias modificaciones sobre los mismos, las cuales se explicarán a continuación. Lo primero de todo, ha sido necesario aplicar la técnica *flattening* sobre los datos, lo que implica transformar los datos multidimensionales en una única dimensión. Después de haber aplanado los datos, ha sido necesario normalizarlos. Este procedimiento se ha realizado de dos maneras diferentes, para poder compararlas y definir cuál es la que más se ajusta al modelo.

Una de ellas es la denominada *StandardScaler* [58]. Esta técnica estandariza los datos. Para ello elimina la media y escala los datos para que la varianza sea igual a 1. Su aplicación se muestra en la Figura 48 y, en la Figura 49 se muestra este tipo de normalización aplicada sobre los datos del entrenamiento y de la validación. A modo visual, se ha impreso la media y la desviación estándar de *X_train_flat* y *X_train_normalized*, donde se puede apreciar que la media es muy cercana a 0 después de la normalización y la varianza es prácticamente igual a 1.

$$z = (x - u) / s$$

Figura 48. Transformación *StandardScaler* [58].

```
# Normalizar los datos: Implica escalar los datos a un rango entre [0,1].
# En este paso no cambian las dimensiones de los datos sino que se normalizan las características
# de los datos, pero no la cantidad de muestras ni las características.
print("Antes de la normalización:")
print("Media de X_train_flat:", np.mean(X_train_flat))
print("Desviación estándar de X_train_flat:", np.std(X_train_flat))

scaler = StandardScaler()

X_train_normalized = scaler.fit_transform(X_train_flat)
X_test_normalized = scaler.transform(X_test_flat)

#Tenemos que asegurar que después de normalizar:
# media cercana a 0
# desviación estándar cercana a 1
print("\nDespués de la normalización:")
print("Media de X_train_normalized:", np.mean(X_train_normalized))
print("Desviación estándar de X_train_normalized:", np.std(X_train_normalized))
```

```
Antes de la normalización:
Media de X_train_flat: 0.20331581875954655
Desviación estándar de X_train_flat: 39.00117844627702

Después de la normalización:
Media de X_train_normalized: 4.617531948602113e-18
Desviación estándar de X_train_normalized: 1.0000000000000007
```

Figura 49. Normalización de los datos *flatten* a través de *StandardScaler*.

La otra técnica utilizada es la denominada *MinMaxScaler* [59]. Esta técnica transforma las características de los datos tal y como se muestra en la Figura 50, es decir, por rango, que por defecto es (0,1). En la Figura 51, se muestra aplicada a los datos del modelo.

```
X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
X_scaled = X_std * (max - min) + min
```

Figura 50. Transformación *MinMaxScaler*[59].

```
from sklearn.preprocessing import MinMaxScaler

# Inicializar el MinMaxScaler
scaler = MinMaxScaler()

# Ajustar el scaler a los datos de entrenamiento y transformarlos
X_train_normalized = scaler.fit_transform(X_train_flat)

# Utilizar el mismo scaler para transformar los datos de prueba
X_test_normalized = scaler.transform(X_test_flat)
```

Python

Figura 51. Normalización de los datos *flatten* a través de *MinMaxScaler*.

Por último, una vez que los datos han sido normalizados, el CVAE trabaja con tensores, por lo que es necesario convertir tanto los datos como sus etiquetas en tensores. Éstos representarán los datos que se utilizarán tanto en entrenamiento, `X_train_tensor` e `y_train_tensor`, como en validación `X_test_tensor` e `y_test_tensor`.

Una vez que están cargados los datos, se continua con la definición del CVAE.

4.4.1. Definición del CVAE

El desarrollo del modelo autocodificador condicional variacional se ha realizado con la definición de una clase denominada CVAE, la cual incluye el codificador y el decodificador necesarios.

Como se muestra en la Figura 52, los datos introducidos en el codificador incluyen tanto las señales como la condición (esto es la clase asociada). Para ello, se define la dimensión de entrada como el resultado de las dimensiones de ambas. La dimensión de las señales está definida por el número de canales y el número de ventanas, como se ha explicado anteriormente. La dimensión de la condición tiene el valor 1, ya que las etiquetas utilizadas solo introducen valores [0, 1] dependiendo de la caracterización de la señal.

Para la definición del codificador, se ha utilizado en cada una de las capas la función de activación ReLU y las capas están totalmente conectadas. La primera capa procesa los datos de entrada usando 128 neuronas y pasándole la salida a la siguiente, que la comprime en 64 neuronas. Por último, a la salida de estas 64 neuronas se calcula la media y la varianza logarítmica que van a caracterizar cada clase del espacio latente.

El decoder está definido mediante el mismo conjunto de capas lineales con funciones de activación ReLU a la salida, al cual le llega una muestra del espacio latente y la convierte usando una capa completamente conectada de 64 neuronas primero y luego de 128. Por último, realiza la transformación de la salida de esta red neuronal al tamaño de la entrada. Cabe destacar que a la salida del decoder ha sido necesario utilizar la función tanh, ya que las amplitudes de los datos normalizados varían entre [-1, 1].

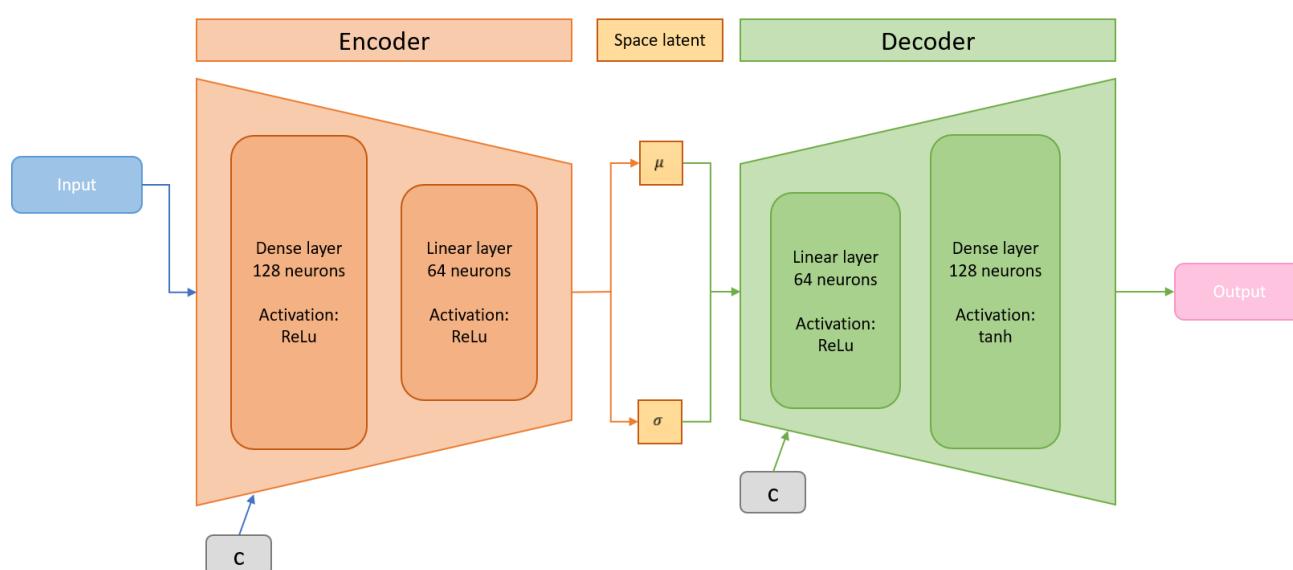


Figura 52. Esquema del CVAE implementado.

A continuación, se estudian cada uno de los resultados que se han obtenido con esta red neuronal.

4.4.2. Sobreajuste

El primer estudio que se ha realizado sobre *el conditional variational autoencoder* es el overfitting. Para ello, como en la red densa, se han almacenado los datos en las dos variables diferentes, `loss` y `val_loss`, y se han representado con entrenamientos desde 20 épocas a 50, como se muestra en la Figura 53, de manera visual, y en la Figura 54, de manera numérica.

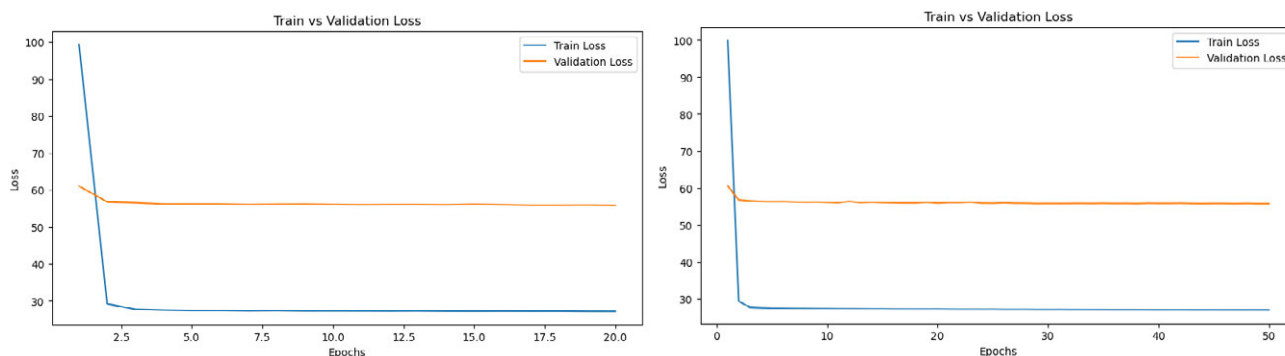


Figura 53. Representación gráfica del overfitting sobre el conditional variational autoencoder.

```
Epoch 1, Train Loss: 99.8334, Val Loss: 60.5277
Epoch 2, Train Loss: 29.3194, Val Loss: 56.7503
Epoch 3, Train Loss: 27.6486, Val Loss: 56.4862
Epoch 4, Train Loss: 27.5011, Val Loss: 56.3312
Epoch 5, Train Loss: 27.4126, Val Loss: 56.3001
Epoch 6, Train Loss: 27.3977, Val Loss: 56.3265
Epoch 7, Train Loss: 27.3569, Val Loss: 56.2121
Epoch 8, Train Loss: 27.3486, Val Loss: 56.1544
Epoch 9, Train Loss: 27.3204, Val Loss: 56.1959
Epoch 10, Train Loss: 27.3123, Val Loss: 56.1032
Epoch 11, Train Loss: 27.2802, Val Loss: 55.9915
Epoch 12, Train Loss: 27.2748, Val Loss: 56.3646
Epoch 13, Train Loss: 27.2619, Val Loss: 56.0313
Epoch 14, Train Loss: 27.2353, Val Loss: 56.1420
Epoch 15, Train Loss: 27.2379, Val Loss: 56.0436
Epoch 16, Train Loss: 27.2033, Val Loss: 55.9809
Epoch 17, Train Loss: 27.1942, Val Loss: 55.9403
Epoch 18, Train Loss: 27.1899, Val Loss: 55.9408
Epoch 19, Train Loss: 27.1930, Val Loss: 56.1233
Epoch 20, Train Loss: 27.2048, Val Loss: 55.8992
Epoch 21, Train Loss: 27.1617, Val Loss: 56.0424
Epoch 22, Train Loss: 27.1452, Val Loss: 56.0455
Epoch 23, Train Loss: 27.1452, Val Loss: 56.1752
Epoch 24, Train Loss: 27.1320, Val Loss: 55.8938
Epoch 25, Train Loss: 27.1430, Val Loss: 55.8681
...
Epoch 47, Train Loss: 26.9754, Val Loss: 55.7452
Epoch 48, Train Loss: 26.9726, Val Loss: 55.8010
Epoch 49, Train Loss: 26.9759, Val Loss: 55.7378
Epoch 50, Train Loss: 26.9672, Val Loss: 55.7457
```

Figura 54. Representación numérica del overfitting sobre el conditional variational autoencoder.

De nuevo puede apreciarse la existencia de overfitting con estos resultados, se ha decidido continuar el desarrollo del modelo con un entrenamiento de 20 épocas, aunque la pérdida en el conjunto de validación apenas disminuye tras las 10 primeras iteraciones.

4.4.1. Clasificación de los datos.

Tras el estudio del *overfitting*, se ha realizado el de las métricas del funcionamiento del clasificador basado en el uso de un KNN en el espacio latente. Estas métricas, explicadas en el capítulo anterior, se han desarrollado como se muestra en las Figura 55 y Figura 56.

La precisión define la proporción de los aciertos sobre el total de las predicciones de una clase, es decir, cuales de todas las señales que el clasificador predijo como pertenecientes a una clase, son realmente de esa clase. El valor *recall*, traducido sensibilidad, es el que mide la proporción de aciertos sobre el número real de muestras positivas.

Macro average (macro avg) es el promedio por cada clase de manera individual. No se tiene en cuenta el número de muestras en cada clase. Es decir, no se tiene en cuenta el desbalance de datos.

Weighted average (weighted avg) es el promedio ponderado, donde sí que se tiene en cuenta el número de muestras por cada clase que se está analizando. Por esta razón es más fiable, ya que tiene en cuenta el desbalanceo de datos y da más valor a las señales con más muestras (no crisis). Sin embargo, si el desbalanceo es muy grande puede dar lugar a resultados engañosos cuando se clasifica muy bien la clase mayoritaria y muy mal la clase minoritaria (que puede ser la clase de interés, como en este caso). Por ese motivo conviene prestar atención también a la macro avg cuando se trabaja con datos muy desbalanceados.

```

Accuracy en datos de prueba: 0.9674
      precision    recall  f1-score   support

     0.0         0.99     0.98     0.98     8539
     1.0         0.01     0.01     0.01      123

 accuracy                   0.97     8662
 macro avg                 0.50     0.49     0.50     8662
 weighted avg              0.97     0.97     0.97     8662

```

Figura 55. Métricas de evaluación del CVAE mediante normalización *StandardScaler*.

```

Accuracy en datos de prueba: 0.9840
      precision    recall  f1-score   support

     0.0         1.00     0.99     0.99     8539
     1.0         0.46     0.82     0.59      123

 accuracy                   0.98     8662
 macro avg                 0.73     0.90     0.79     8662
 weighted avg              0.99     0.98     0.99     8662

```

Figura 56. Métricas del CVAE con la normalización de los datos *MinMax*.

Como conclusión principal, se aprecia claramente que el modelo que incluye los datos normalizados con *StandardScaler* no está funcionando con las señales de crisis. Sin embargo, en el modelo con datos normalizados con *MinMaxScaler*, se puede apreciar que, si se analiza con *macro avg*, se obtienen buenos resultados, aunque la precisión y el *recall* son más bajos debido a la clase minoritaria (crisis).

4.4.2. Obtención del espacio latente.

El espacio latente, en el CVAE se define de la siguiente manera en la representación del esquema de un CVAE completo en la Figura 57:

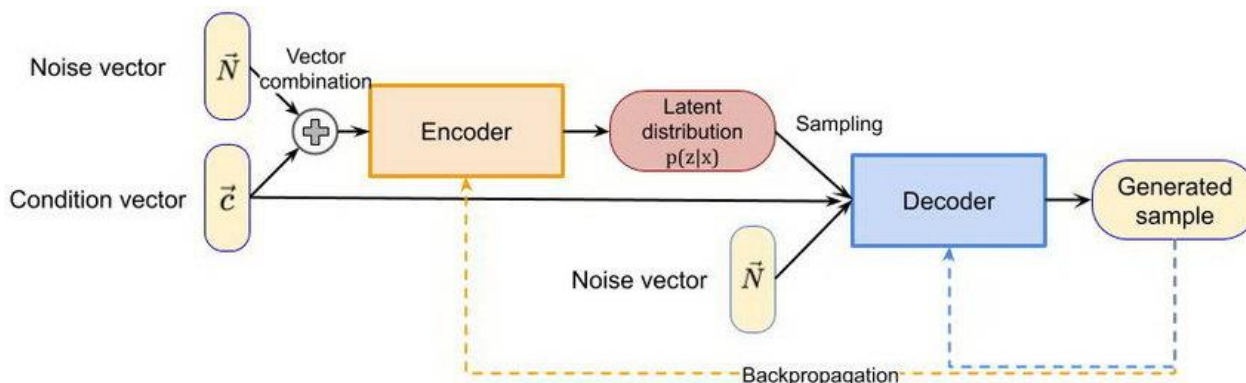


Figura 57. Conditional Variational Autoencoder [60].

Como se muestra en la Figura 57, a diferencia de un *variational autoencoder*, el CVAE incluye una condición a la entrada, es decir, la etiqueta con el valor booleano 0 = no crisis y 1 = crisis. Se ha analizado, como en los casos anteriores, el espacio latente.

A diferencia del *autoencoder* y el *variational autoencoder*, en este caso se ha realizado únicamente el estudio con la técnica PCA debido al coste computacional que supone el análisis de las otras técnicas, pero incluyendo la representación con PCA en tres dimensiones, para asegurarnos la separación de las muestras.

El primer análisis del espacio latente se ha realizado sobre el CVAE con normalización *StandardScaler*, como se muestra en la Figura 58. En la representación en 2D, las señales con crisis se caracterizan por estar en el eje (0,0) superpuestas con las señales sin crisis, pero en la representación en 3D (derecha) se muestra cómo la nube de muestras sin crisis se sitúa por delante de las muestras con crisis, por lo que no podemos definir con claridad suficiente si están totalmente separadas

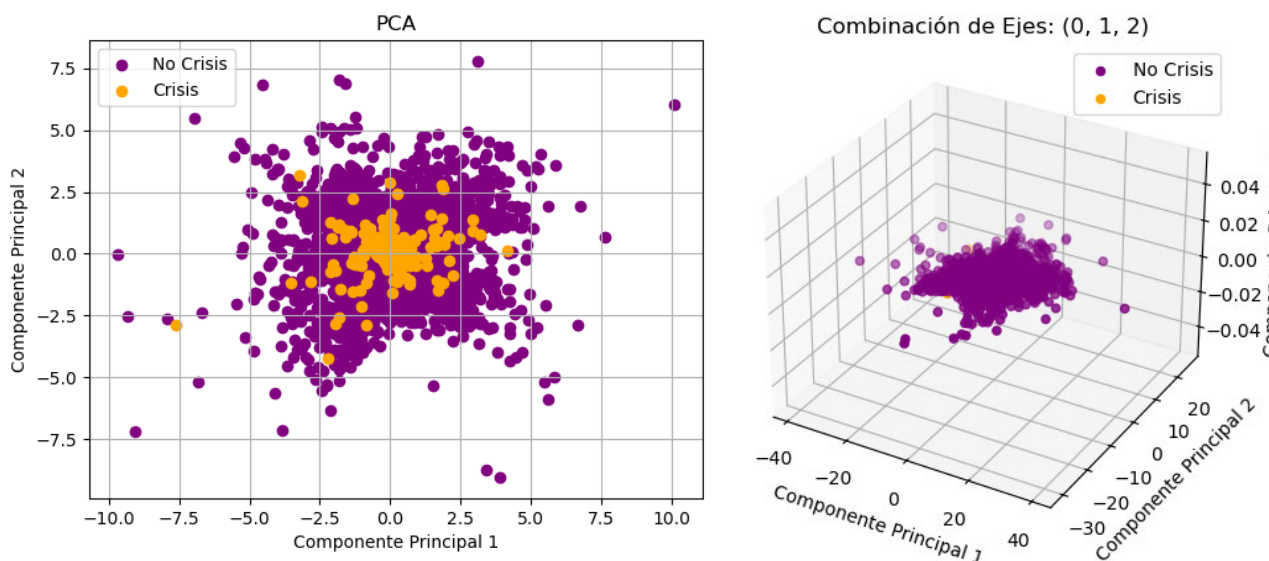


Figura 58. Análisis del espacio latente en 2D(izquierda) y en 3D (derecha) con la normalización de los datos *StandardScaler*.

Sin embargo, también se ha realizado el estudio sobre el CVAE con normalización *MinMaxScaler*, y realmente sí que se separan las señales en diferentes subconjuntos, como se aprecia en la figura en 3D (derecha) de la Figura 59. En la representación en 2D (izquierda), ocurre lo mismo que en la Figura 58, las muestras de no crisis se centran en 0,0.

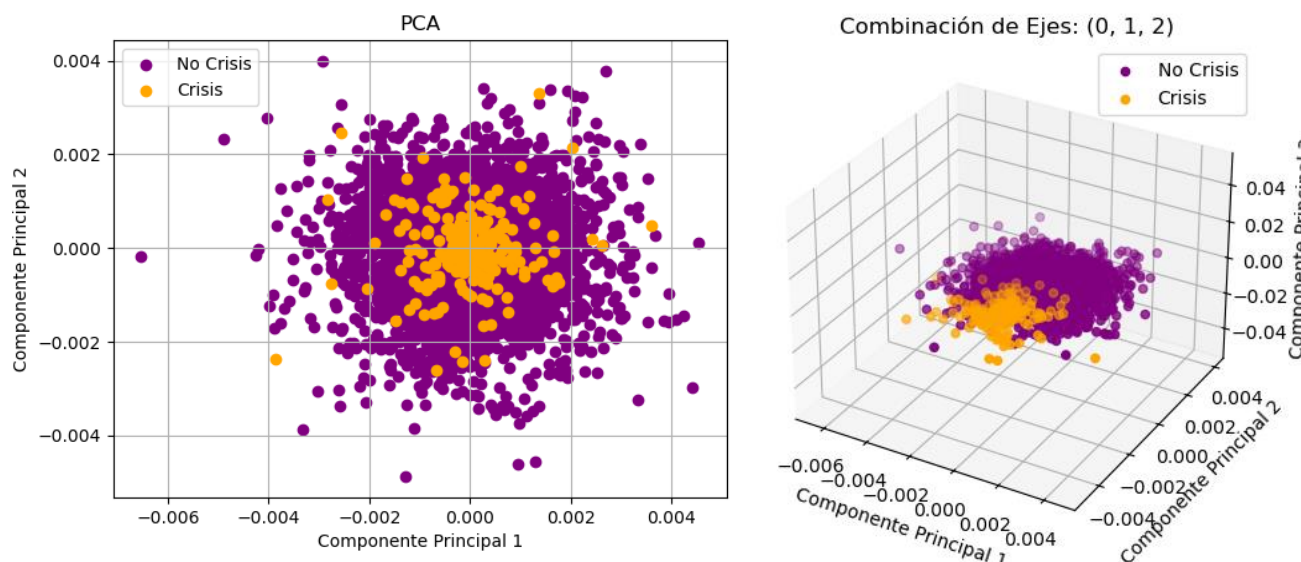


Figura 59. Análisis del espacio latente en 2D (izquierda) y en 3D (derecha) con la normalización de los datos *MinMax*.

4.4.3. Reconstrucción de la señal

Una vez analizado el clasificador y el espacio latente, se estudia cómo se realiza la reconstrucción de las señales, la cual se muestra en las Figura 60 y Figura 61.

Esta reconstrucción, cuando la normalización de las señales de entrada se ha realizado con *StandardScaler*, no funciona correctamente, como se muestra en la Figura 60.

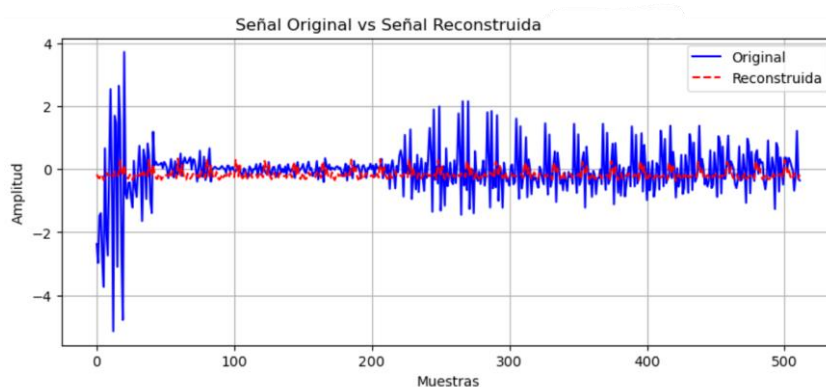


Figura 60. Señal reconstruida utilizando *StandardScaler*.

Sin embargo, la reconstrucción de las señales normalizadas con *MinMaxScaler*, como se puede observar en la Figura 61, es muy buena.

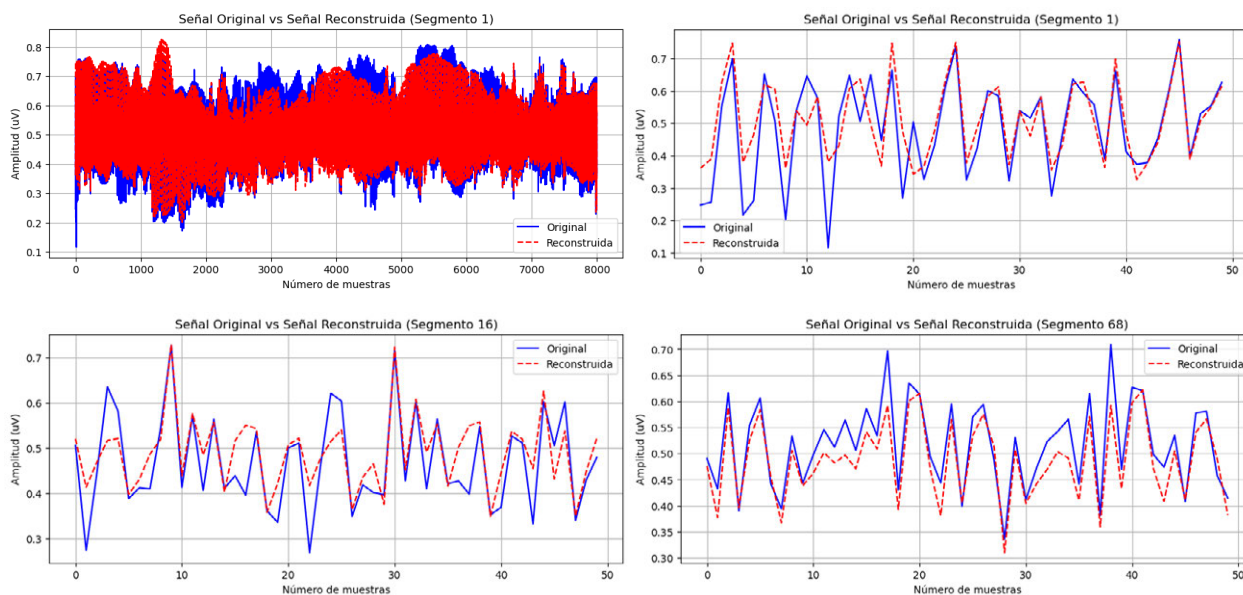


Figura 61. Señal reconstruida utilizando normalización MinMaxScaler.

Con estos resultados, a partir de aquí, se continuará sólo con el CVAE con normalización *MinMaxScaler* ya que, dado que ninguno de los resultados obtenidos ha sido favorable. Se descarta la utilización de *StandardScaler*.

4.4.4. Generación de nuevas señales

Para la generación de nuevas muestras se ha implementado una función en la que, en primer lugar, utilizando el espacio latente con distribución normal y la clase deseada, se generan muestras aleatorias en el mismo. A continuación se decodifican las muestras latentes generando las señales de salida del modelo. Las Figuras 62 y 63 muestran dos ejemplos de señales generadas para segmentos sin crisis y con crisis, respectivamente.

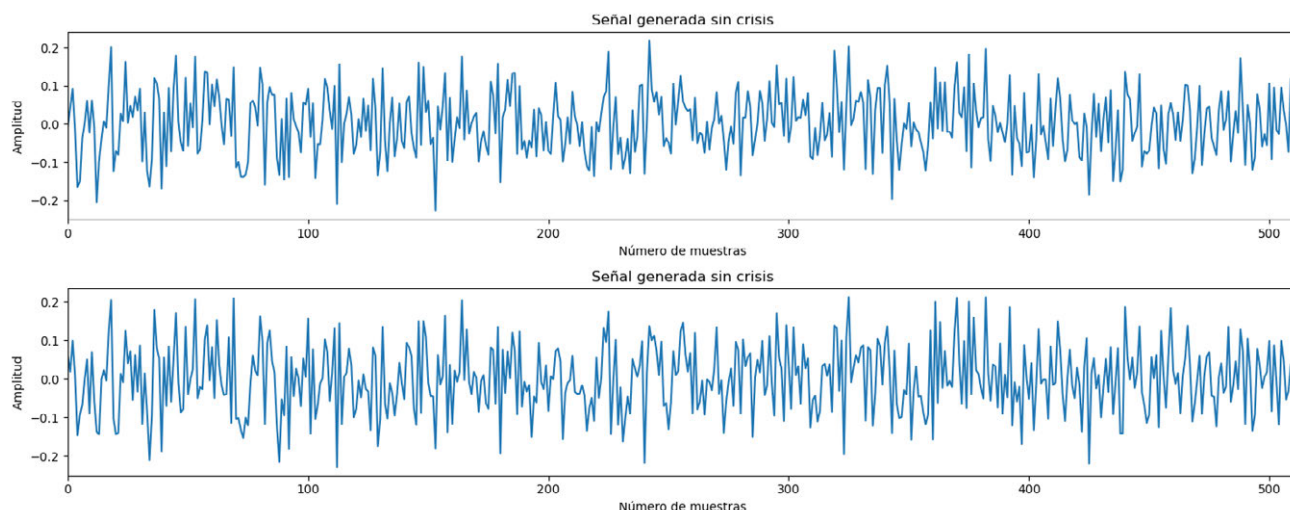


Figura 62. Ejemplos de señales generadas sin crisis.

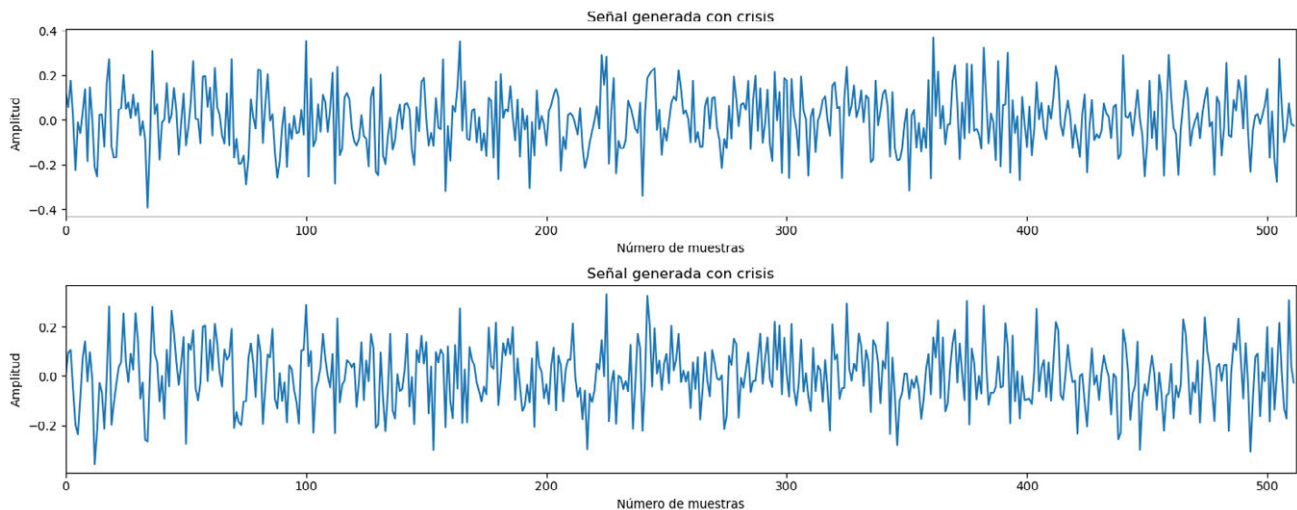


Figura 63. Ejemplos de señales generadas con crisis.

4.5 Discusión

El desarrollo de este proyecto centrado, en el desarrollo de un *conditional variational autoencoder* a partir de un *autoencoder* estándar, ha permitido poder realizar una comparación entre dichos modelos.

El *autoencoder* estándar, formado por una red densa, ha demostrado ciertas limitaciones, como la representación del espacio latente para la evaluación de los datos de test, además de su bajo rendimiento debido a los falsos positivos.

Sin embargo, el CVAE ha sido más efectivo. El CVAE ha sido entrenado con datos tanto de crisis como de no crisis etiquetados con valores booleanos, lo que ha permitido su mejor adaptación a las características más significativas del EEG, con una mejora de su precisión a la hora de distinguir segmentos con crisis y sin crisis.

En relación con el análisis de los datos, cabe destacar la diferencia que existe entre la normalización de los datos, siendo la más eficiente la normalización con la función *MinMaxScaler*.

También cabe destacar la diferencia más importante entre el *autoencoder* y el CVAE mencionada durante el desarrollo de la memoria, la generación de las señales. El CVAE ha sido capaz de almacenar las gaussianas calculadas por el espacio latente de manera muy efectiva para la generación de nuevas señales de tipo 0 (no crisis) y de clasificación 1 (crisis).

Por último, destacamos las ventajas que presenta un *conditional variational autoencoder* para la detección de crisis epilépticas:

La reducción de dimensionalidad de las señales EEG facilita la tarea de clasificación, de una manera más manejable y menos computacionalmente intensa en el espacio latente mediante un simple KNN.

La representación de las señales de crisis y no crisis más complejas de los EEGs ha sido más relevantes gracias a la captura del CVAE de las características no lineales.

La implementación del KNN, una vez obtenido el espacio latente, ha proporcionado un método simple y efectivo para la clasificación sin necesidad de un entrenamiento demasiado complejo.

5. Conclusiones y líneas futuras

5.1 Conclusiones generales del proyecto

La epilepsia es una enfermedad neurológica crónica que afecta a millones de personas en todo el mundo. Esta enfermedad provoca graves consecuencias en las personas que la padecen, por lo que la atención temprana y precisa puede ayudar a su calidad de vida.

El estudio de la epilepsia a través de señales EEG no es sencillo, debido a la afectación de ruidos e interferencias que sufre la señal. En este proyecto se han realizado varios estudios.

El primero ha sido la ampliación del autocodificador del que parte el proyecto, donde se ha evaluado el *overfitting* con diferentes números de épocas de entrenamiento, determinando que 20 épocas proporcionan un buen balance de rendimiento y complejidad, teniendo en cuenta el coste computacional que suponía el entrenamiento con más épocas y sin mejora de resultados significativos. Además, se ha comparado el espacio latente en dos dimensiones de este autocodificador estándar con un modelo autocodificador variacional, donde no se observan grandes diferencias significativas en la visualización en 2D.

Finalmente, se desarrolla un *conditional variational autoencoder* (CVAE) para la detección de crisis epilépticas, el objetivo principal del proyecto. En este modelo se comparan dos técnicas de normalización de datos, *StandardScaler* y *MinMaxScaler*. *MinMaxScaler* ha resultado ser mejor técnica en términos de rendimiento. Se estudia de nuevo el *overfitting* como en el *autoencoder*, con varias épocas de entrenamiento, tomando como decisión final usar 20 épocas como la configuración final para el desarrollo del proyecto. Uno de los estudios, del rendimiento del modelo, fue bastante satisfactorio con la normalización *MinMaxScaler*, especialmente en la reconstrucción de las señales, aunque los resultados son mejorables para las señales con crisis epilépticas. Sin embargo, con la normalización *StandardScaler* se confirma la importancia de la normalización de los datos, ya que la reconstrucción no ha sido buena. Por último, se utilizó el CVAE para la generación de nuevas señales con la normalización *MinMaxScaler* y, aunque sí se ha visto que es capaz de generarlas, ofrece un margen de mejoras adicionales y debe realizarse un estudio más profundo acerca de la calidad y precisión de las señales generadas.

Finalmente, como conclusión principal cabe destacar que el desarrollo del *conditional variational autoencoder* ha demostrado que este modelo tiene capacidad para discriminar entre crisis eilépticas y segmentos sin crisis. Además, el análisis del espacio latente permitirá, en el futuro, comprender mejor la diferencia entre las características de los periodos ictales y no ictales, y la generación de nuevas secuencias servirá para entrenar otros algoritmos de aprendizaje automático.

5.2 Líneas futuras

Las líneas futuras que ofrece este proyecto para el estudio de la detección de ataques epilépticos a través del CVAE implementado son:

Como se ha observado en las métricas de rendimiento, el CVAE no está funcionando de manera satisfactoria con las señales de crisis en comparación con las muestras de no crisis. Por lo que, podrían implementarse otros métodos de clasificación en el espacio latente, como el *RandomForest*. Este algoritmo de aprendizaje automático supervisado construye múltiples árboles de decisión aleatorios

a partir de los datos disponibles (las representaciones en el espacio latente en nuestro caso) y realiza la predicción promediando o haciendo coincidir las predicciones de los árboles individuales.

Otra de las líneas futuras que ofrece este proyecto es la prueba del modelo con los 23 pacientes restantes de la base de datos utilizada, ya que, por falta de tiempo, no se ha incluido en este proyecto. Con este análisis, podría verificarse el buen funcionamiento del CVAE para diferentes pacientes y su robustez para poder detectar crisis epilépticas en diferentes pacientes. Además, aparte de la base de datos CHB-MIT Scalp, se podrían incluir en el estudio las señales del Hospital de la Princesa. Estas señales requieren el desarrollo del preprocesamiento de los datos de manera diferente a la ya implementada, debido a que se disponen en un formato diferente, por lo que trabajar con las mismas presenta retos que exceden los objetivos de este proyecto.

Por último, el CVAE de este proyecto ha sido desarrollado siguiendo una estructura similar a la del autoencoder del TFG de Marta Fraile. Otra línea futura para el modelo podría ser su modificación con cambios de la estructura del modelo, una función de coste que incluyera una pérdida de contraste, probar distintas combinaciones del tipo y número de capas, etc.

6. Referencias

- [1] «Avances tecnológicos en la salud: mejoras aplicadas a la medicina - Hospital Virgen del Mar». [En línea]. Disponible en: <https://www.hospitalvirgendelmar.es/noticia/la-tecnologia-aplicada-a-la-salud-los-ultimos-y-mejores-avances/20>
- [2] Redacción, «Cada año se diagnostican más de 2,4 millones de nuevos casos de epilepsia en el mundo», Médicos y Pacientes - Portal informativo de la Organización Médica Colegial. [En línea]. Disponible en: <https://www.medicosypacientes.com/articulo/cada-ano-se-diagnostican-mas-de-24-millones-de-nuevos-casos-de-epilepsia-en-el-mundo/>
- [3] I. M. López-Tercero, «Detección de crisis epilépticas a partir de señales EEG mediante índices basados en el algoritmo de Lempel-Ziv», Trabajo Fin de Grado en Ingeniería Electrónica Industrial I Automática, UPC, 2013
- [4] Gutttag, J., «CHB-MIT Scalp EEG Database v1.0.0». PhysioNet [En línea]. Disponible en: <https://physionet.org/content/chbmit/1.0.0/>
- [5] M.Fraile, «Detección de crisis epilépticas mediante aprendizaje profundo», Trabajo Fin de Grado en Ingeniería de Sistemas de Telecomunicación, UPM, Junio 2023
- [6] S. Kullback y R. A. Leibler, «On Information and Sufficiency», The annals of mathematical statistics, 22(1), 79-86, 1951.
- [7] J. Rocca, «Understanding Variational Autoencoders (VAEs)», Medium. [En línea]. Disponible en: <https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>
- [8] «Objetivos y metas de desarrollo sostenible - Desarrollo Sostenible». [En línea]. Disponible en: <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible>
- [9] G.A.M. Chávez «Desarrollo e implementación de un prototipo biomédico para la determinación de urea aplicado al tratamiento de hemodiálisis», Tesos de maestría, Instituto tecnológico y de estudios de Monterrey, Diciembre 2000.
- [10] D.M.Terán Pérez, «Bioingeniería», ALFAOMEGA GRUPO EDITOR, S.A. de C.V, Enero 2017. [En línea]. Disponible en: <https://content.e-bookshelf.de/media/reading/L-18426488-c64f6bc570.pdf>
- [11] P.S. Saldana, «Procesamiento de señales biomédicas en electrocardiogramas y electroencefalogramas», Monografias.com, Ingeniería Electrónica, Universidad Politécnica Salesiana. [En línea]. Disponible en: <https://www.monografias.com/trabajos88/procesamiento-senales-biomedicas-electrocardiogramas-y-electroencefalogramas/procesamiento-senales-biomedicas-electrocardiogramas-y-electroencefalogramas>
- [12] F. Alda, «Sistema Nervioso II: señales eléctricas y químicas en las neuronas». Blog.ia, 2.0, Junio de 2016. [En línea]. Disponible en: <http://b-log-ia20.blogspot.com/2016/06/>
- [13] Enciclopedia Médica A.D.A.M., «Electroencefalograma». [En línea]. Disponible en: <https://medlineplus.gov/spanish/ency/article/003931.htm>
- [14] Enciclopedia Médica A.D.A.M., «Electromiografía». [En línea]. Disponible en: <https://medlineplus.gov/spanish/ency/article/003929.htm>
- [15] Enciclopedia Médica A.D.A.M., «Electrocardiograma». [En línea]. Disponible en: <https://medlineplus.gov/spanish/ency/article/003868.htm>

- [16] Artículo, Khan Academy «Función y estructura de la neurona». [En línea]. Disponible en: <https://es.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/overview-of-neuron-structure-and-function>
- [17] Ilustración vectorial, «conjunto de oscilación de ondas cerebrales. ondas cerebrales beta, alfa, theta, delta, gamma. ritmo humano, tipos, amplitud de ondas mentales.», Vector Pro [En línea]. Disponible en: <https://es.vecteezy.com/arte-vectorial/5232812-conjunto-de-ondas-cerebrales-oscilacion-beta-alfa-theta-delta-gamma-ondas-cerebrales-humano-ritmo-tipos-amplitud-de-ondas-mentales-vector-ilustracion>
- [18] Dr.M.Guzmán, «Lóbulos del cerebro», Kenhub, Octubre 2023 [En línea]. Disponible en: <https://www.kenhub.com/es/library/anatomia-es/lobulos-del-cerebro>
- [19] Ó. R.Varo, «Diseño e implementación de redes neuronales para la detección de crisis epilépticas», Trabajo Fin del Grado en Ingeniería de Sistemas de Comunicación, UPM, Julio de 2022.
- [20] Unidad temática, «Fisiología de la actividad eléctrica del cerebro». Departamento de Fisiología, Facultad de Medecina, UNAM.
- [21] F.Sánchez y A.M.Hernández «Modelado autorregresivo de señales encefalográficas para simuladores médicos», Ingeniería y Desarrollo, Diciembre 2017. [En línea]. Disponible en: https://www.researchgate.net/figure/Figura-7-Diferentes-segmentos-de-senales-EEG-simuladas-a-Reposo-ojos-cerrados-b_fig1_324441020
- [22] World Health Organization, «Epilepsy». [En línea]. Disponible en: <https://www.who.int/health-topics/epilepsy>
- [23] F.J. López González, V. Villanueva, M. Falip, M. Toledo, D.Campos y J.Serratos, «Recomendaciones diagnóstico terapéuticas de la SEN2019». 2019. [En línea]. Disponible en: https://www.sen.es/pdf/guias/GuiaEpilepsia_2023.pdf
- [24] «El término Epilepsia deriva del griego ‘epilambaneim que significa “coger por sorpresa”», Asociación Gipuzkoana de Epilepsia. [En línea]. Disponible en: <https://epilepsiagipuzkoa.eus/epilepsia/historia-y-origen>
- [25] Clínica Universidad de Navarra «Epilepsia: causas, síntomas y pronóstico» [En línea]. Disponible en: <https://www.cun.es/enfermedades-tratamientos/enfermedades/epilepsia>
- [26] «Médica Sur: excelencia médica, calidez humana». [En línea]. Disponible en: <https://www.medicasur.com.mx/>
- [27] UCBCares, «Pruebas para diagnosticar la epilepsia: EEG, RM y TAC». [En línea]. Disponible en: <https://ucbcares.es/pacientes/epilepsia/es/content/945786100/pruebas-diagnosticas-epilepsia>
- [28] Vivir con Epilepsia, «Diagnóstico y tratamiento de la epilepsia». [En línea]. Disponible en: <https://vivirconepilepsia.es/diagnostico-tratamiento-epilepsia>
- [29] Vivir con Epilepsia, «Guía Epilepsia y Adultos», [En línea]. Disponible en: <https://vivirconepilepsia.es/guia-epilepsia-y-adultos>
- [30] Freepik, «Convulsión generalizada y parcial. Epilepsia y actividad cerebral anormal. Vector de investigación médica». [En línea]. Disponible en: https://www.freepik.es/vector-premium/convulsion-generalizada-parcial-epilepsia-actividad-cerebral-anormal-vector-investigacion-medica_18477319.htm

- [31] decide4AI, «Tipos de aprendizaje que usan los algoritmos de Machine Learning», *Analítica predictiva y machine learning*, Noviembre 2022. [En línea]. Disponible en: <https://decidesoluciones.es/tipos-de-aprendizaje-algoritmos-machine-learning/>
- [32] Ai Lab School, «¿Cuáles son los tipos de machine learning?». [En línea]. Disponible en: <https://ailabschool.com/cuales-son-los-tipos-de-machine-learning/>
- [33] Artículo, «Pipelines». Oracle, Octubre 2023 [En línea]. Disponible en: <https://docs.oracle.com/es-ww/iaas/data-science/using/pipelines-about.htm>
- [34] Admin, «Cómo Preparar Datos para Modelos de Machine Learning Eficientes». [En línea]. Disponible en: <https://data-universe.org/como-preparar-datos-para-modelos-de-machine-learning-eficientes/>
- [35] Amazon, «¿Qué es el aprendizaje profundo? - Explicación del aprendizaje profundo - AWS». [En línea]. Disponible en: <https://aws.amazon.com/es/what-is/deep-learning/#:~:text=El%20aprendizaje%20profundo%20es%20un,inspira%20en%20el%20cerebro%20humano.>
- [36] X. B. Olabe, «Redes neuronales artificiales y sus aplicaciones». Escuela Superior Técnica de Bilbao, UPV-EHU. [En línea]. Disponible en: https://ocw.ehu.es/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf
- [37] «Neural Networks: Creating a Perceptron Model in Python», Circuitrocks. [En línea]. Disponible en: <https://learn.circuit.rocks/neural-networks-creating-a-perceptron-model-in-python>
- [38] L.Castillo y C.Maldonado, «Redes Neuronales», Abril 2020. [En línea]. Disponible en: https://rpubs.com/luis_castillo/redes_neuronales
- [39] M.Sotaquirá, «La Función de Activación», blog, Septiembre 2018. [En línea]. Disponible en: <https://www.codificandobits.com/blog/funcion-de-activacion/>
- [40] C. A. Ruiz, M. S. Basualdo, y D. J. Matich, «Redes Neuronales: Conceptos Básicos y Aplicaciones.», [En línea]. Disponible en: <https://www.codificandobits.com/blog/funcion-de-activacion/PA>
- [41] «Deep Learning - Técnicas y Aplicación en Empresas», PredictLand AI. [En línea]. Disponible en: <https://www.predictland.com/tecnicas-ia/deep-learning/>
- [42] I. Talbi, « 7 méthodes pour éviter l'overfitting », *La revue IA*, Septiembre 2020. [En línea]. Disponible en: <https://larevueia.fr/7-methodes-pour-eviter-loverfitting/>
- [43] J.Segura, «¿Qué son las Redes Neuronales Densas y Cómo Funcionan?» [En línea]. Disponible en: <https://www.learningheroes.com/aprende-inteligencia-artificial/que-son-las-redes-neuronales-densas-y-como-funcionan>
- [44] M.Scanlon, «Anotación semántica de imágenes aéreas mediante aprendizaje profundo, aprendizaje por transferencia y datos de entrenamiento sintéticos », Tesis para Maestría en Ciencias de la Computación (Análisis de Datos). [En línea]. Disponible en: https://www.researchgate.net/figure/Fully-connected-feed-forward-neural-network-architecture_fig2_329016767
- [45] Artículo, «¿Qué son las redes neuronales convolucionales?», IBM. [En línea]. Disponible en: <https://www.ibm.com/es-es/topics/convolutional-neural-networks>

- [46] S. Saha, «A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way», Medium. [En línea]. Disponible en: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [47] A. Dertat, «Applied Deep Learning - Part 3: Autoencoders», Medium. [En línea]. Disponible en: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>
- [48] Artículo, «Variational AutoEncoders», GeeksforGeeks. Diciembre 2023 [En línea]. Disponible en: <https://www.geeksforgeeks.org/variational-autoencoders/>
- [49] D.S.Hernández, «ODA14 U5 | Autoencoders varacionales (VAE)», Genially, Marzo 2024. [En línea]. Disponible en: <https://view.genially.com/65eb31306e9b790015425456/presentation-oda14-u5-or-autoencoders-varacionales-vae>
- [50] «Conditional Variational Autoencoders», Deepgram, Junio 2024. [En línea]. Disponible en: <https://deepgram.com/ai-glossary/conditional-variational-autoencoders>
- [51] K.Abdelli, H.Griesser, C.Neumeyr, R.Hohenleitner y S.Pachnicke «Degradation prediction of Semiconductor Lasers Using conditional variational autoencoder.», Noviembre 2022. [En línea]. Disponible en: https://www.researchgate.net/figure/Structure-of-the-conditional-variational-autoencoder-CVAE_fig2_365190062
- [52] Departamento de Matemática Aplicada, «02.3 Ratios e indicadores que califican una clasificación — Introducción al Aprendizaje Automático». Libro [En línea]. Disponible en: https://dcain.etsin.upm.es/~carlos/bookAA/02.3_MetodosdeClasificacion-Ratioseindicadores.html
- [53] L. Weng, «From Autoencoder to Beta-VAE». GitHub, Agosto 218. [En línea]. Disponible en: <https://lilianweng.github.io/posts/2018-08-12-vae>
- [54] J.A.Rodrigo, «Análisis de Componentes Principales (Principal Component Analysis, PCA) y t-SNE». Documento, Junio 217 [En línea]. Disponible en: https://cienciadedatos.net/documentos/35_principal_component_analysis
- [55] M. Wattenberg, F. Viégas, y I. Johnson, «How to Use t-SNE Effectively», *Distill*, vol. 1, n.º 10, p. e2, oct. 2016, doi: 10.23915/distill.00002.
- [56] «TSNE», scikit-learn. [En línea]. Disponible en: <https://scikit-learn/stable/modules/generated/sklearn.manifold.TSNE.html>
- [57] J.V.-Aguirre, G.D.-Santacoloma, C.D.Acosta y G.C.-Dominguez. «Comparación de métodos de Reducción de Dimensión Basados en Análisis por Localidades». [En línea]. Disponible en: <https://www.redalyc.org/pdf/3442/344234320008.pdf>
- [58] «StandardScaler», scikit-learn. [En línea]. Disponible en: <https://scikit-learn/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>
- [59] «MinMaxScaler», scikit-learn. En línea]. Disponible en: <https://scikit-learn/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
- [60] F.Sergeev, N.Jain, I.Knunyants, G.Kostenkov y E.Trofimova, «Fast simulation of the LHCb electromagnetic calorimeter resonance using VAEs and GANs», Journal of physics Conference, Enero 2021. [En línea]. Disponible en: https://www.researchgate.net/figure/Conditional-variational-autoencoder-scheme_fig2_348713080

Anexo A: Impacto del proyecto

En este capítulo se ponen de manifiesto las implicaciones sociales, de salud y seguridad, ambientales, económicas, tecnológicas e industriales relacionadas con este proyecto, así como la posible aportación a los ODS (Objetivos de Desarrollo Sostenible) [8].

Al vincular este proyecto con dichos objetivos, no solo se resalta la importancia de la salud individual, sino también su impacto en el desarrollo a nivel global.

A continuación, se enumeran y justifican los ODS relacionados con este proyecto.

A.1. ODS 3: Salud y Bienestar

Este proyecto está alineado con el ODS, 3 ya que uno de sus principales objetivos se centra en asegurar una vida sana y promover el bienestar para todas las personas de todas las edades. Con la detección de crisis epiléptica se puede mejorar significativamente la calidad de vida de las personas afectadas, así como la morbilidad asociada, ya que conlleva intervenciones más rápidas y efectivas.

A.2. ODS 4: Educación de calidad

El objetivo ODS 4 relaciona con este proyecto mediante dos propósitos diferentes.

El primero de estos propósitos está relacionado con asegurar que las personas con epilepsia puedan detectar y manejar sus crisis, lo que puede permitirles continuar con su educación sin interrupciones significativas, promoviendo una educación inclusiva y equitativa.

Por otra parte, el desarrollo y la implementación de este proyecto requiere educación y capacitación en áreas de salud, ingeniería y tecnología, promoviendo una educación de calidad en estas disciplinas.

A.3. ODS 9: Industria, innovación e infraestructura

En relación con el ODS 9, utilizar algoritmos de Inteligencia Artificial avanzada como los *variational autoencoders* representa una innovación tecnológica significativa en el ámbito de la salud, fomentando el desarrollo de infraestructuras sanitarias más efectivas, resilientes y sostenibles. Esto también fomenta la investigación y la innovación en el campo de la salud.

A.4. ODS 10: Reducción de las desigualdades

Por último, la tecnología puede hacer que la atención médica de calidad sea más accesible para personas con epilepsia en distintas partes del mundo, especialmente en áreas con acceso limitado a servicios médicos avanzados, reduciendo desigualdades en la atención sanitaria.

Anexo B: Presupuesto del proyecto

En este anexo, se han estimado los costes necesarios para la realización del estudio propuesto. Para ello, se tienen en cuenta tanto los medios materiales como los recursos humanos empleados.

B.1. Coste de los medios materiales

En la Tabla B.1 se muestran los costes derivados del material empleado, incluyendo el gasto en licencias, hardware y software.

Coste de los Medios Materiales			
Material	Tangible	Intangible	Coste (€)
Ordenador personal S540-13IML Laptop (ideapad) – Type 81XA	X		1.156,00
Monitor ARZOPA A1 Travel Monitor 15.6" FHD 1080 Display	X		119,99
Ratón Mi Dual Mode Wireless	X		9,99
Paquete Microsoft Office 365 Personal		X	69,95
Anaconda (Jupyter Notebook y Spider)		X	0,00
Base de datos PTB (Physionet)		X	0,00
Encuadernación del proyecto	X		150,00
Espacio Google One Drive		X	0,49
			TOTAL: 1506,42€

Tabla A. 1. Desglose de los Medios Materiales.

B.2. Coste personal

El coste del personal de este proyecto consta del estudiante que ha realizado el proyecto, así como del personal de dirección, quien ha dirigido el proyecto.

El sueldo medio de un Ingeniero de Sistemas de Telecomunicación se encuentra en torno a los 30.000€ anuales, según el Colegio Oficial de Ingenieros de Telecomunicación (COIT). Este proyecto ha sido realizado por un Ingeniero con experiencia laboral de dos años. En estas condiciones se toma como valor de referencia un sueldo entorno a los 23.000€ brutos anuales de media.

Aunque el trabajo no se ha realizado a jornada completa, los cálculos se van a realizar sobre 40h/semana para obtener el resultado total de una manera más sencilla. Por lo tanto, con un trabajo realizado 40 horas por semana, se realizarán 160 horas mensuales, lo que equivale a 1916,6€ (brutos al mes).

$$1.916,6\text{€}/160\text{h} \approx 11,98 \text{€}/\text{h}$$

Con este cálculo, y con 320h dedicadas en total, se calcula que el coste es de:

$$11,98\text{€}/\text{h} * 320\text{h} = 3833,33\text{€}$$

Como se ha mencionado anteriormente, el proyecto ha necesitado de dirección. Esta dirección se estima que tiene un valor del 7% del total del coste del material junto con el coste del personal limitado a 30.000€/anuales. Se calcula a continuación:

$$(\text{coste del material} + \text{coste del personal}) * 0,07 = 373,79\text{€}$$

Coste del Personal	
Ingeniero recién licenciado	3833,33
Ingeniero Docente	373,79
TOTAL: 4207,11€	

Tabla A. 2. Desglose del coste del personal.

B.3. Coste total del proyecto

Por último, se ha calculado el coste total del proyecto en la Tabla A.3. Este coste se basa en la suma de la Tabla A.1. y de la Tabla A.2.

Coste Total del Proyecto (€)	
Medios materiales	1506,42
Personal	4207,11
TOTAL: 5713,53€	

Tabla A. 3. Desglose del coste total del proyecto.