



Benemérita Universidad Autónoma de Puebla

---

Facultad de Ciencias Físico Matemáticas

---

Optimización del rendimiento de una RNC para clasificar  
señales EMG mediante un diseño factorial de  
hiperparámetros

Tesis presentada al

**Posgrado en Física Aplicada**

como requisito parcial para la obtención del grado de

**MAESTRO EN CIENCIAS**

por

Arturo Ariel Márquez Carranza

Asesorado por

Dr. Eduardo Moreno Barbosa

Dr. Juan Moisés Arredondo Velázquez

Puebla Pue.  
Septiembre de 2025





Benemérita Universidad Autónoma de Puebla

---

Facultad de Ciencias Físico Matemáticas

---

Optimización del rendimiento de una RNC para clasificar  
señales EMG mediante un diseño factorial de  
hiperparámetros

Tesis presentada al

**Posgrado en Física Aplicada**

como requisito parcial para la obtención del grado de

**MAESTRO EN CIENCIAS**

por

Arturo Ariel Márquez Carranza

Asesorado por

Dr. Eduardo Moreno Barbosa

Dr. Juan Moisés Arredondo Velázquez

Puebla Pue.  
Septiembre de 2025



**Título:** Optimización del rendimiento de una RNC para clasificar señales EMG mediante un diseño factorial de hiperparámetros

**Estudiante:** ARTURO ARIEL MÁRQUEZ CARRANZA

COMITÉ

---

Dr. Javier Miguel Hernández López  
Presidente

---

Dr. Lucio Fidel Rebolledo Herrera  
Secretario

---

Dr. Jorge Velázquez Castro  
Vocal

---

Dr. Eduardo Moreno Barbosa  
Asesor

---

Dr. Juan Moisés Arredondo Velázquez  
Coasesor



# Agradecimientos

Agradezco al SECIHTI por el apoyo económico, el cual hizo posible la realización de este trabajo de investigación.

Quiero expresar mi más profundo agradecimiento a mi familia, en especial a mis padres y a mi hermano, por su constante apoyo, paciencia y motivación a lo largo de este proceso. Su confianza en mí ha sido fundamental para culminar esta etapa.

Finalmente, agradezco a mis asesores por su orientación y colaboración durante el desarrollo de esta tesis, así como a los miembros del jurado por sus observaciones y sugerencias que contribuyeron a enriquecer este trabajo.



# Índice general

<b>1. Introducción</b>	<b>3</b>
1.1. Descripción	3
1.2. Motivación	3
1.3. Descripción del problema	4
1.4. Hipótesis	5
1.5. Objetivo general	5
1.6. Objetivos específicos	5
1.7. Alcance	5
<b>2. Estado del arte</b>	<b>7</b>
2.1. RNCs para señales EMG	7
2.2. Optimización de hiperparámetros en clasificación de señales EMG	10
<b>3. Marco teórico</b>	<b>17</b>
3.1. Señales electromiográficas (EMG)	17
3.1.1. Definición de la electromiografía	17
3.1.2. Origen de las señales EMG	18
3.1.3. Propiedades de las señales EMG	24
3.1.4. Adquisición y electrodos	24
3.2. Diseño factorial	26
3.2.1. Diseño factorial $2^k$	26
3.2.2. Diseño factorial $3^k$	29
3.2.3. Diseño D-óptimo	32
3.3. Redes Neuronales de Convolución	34
3.3.1. Componentes	35
3.3.2. Entrenamiento de una RNC	37
3.3.3. Funciones de activación	39
3.3.4. Tipos de Convolución	42
3.3.5. Tipos de conexiones	44
<b>4. Materiales y métodos</b>	<b>49</b>
4.1. Bases de datos	49
4.2. Preprocesamiento de los datos	51
4.3. Selección de hiperparámetros	52
4.4. Diseño D-Óptimo	53
4.5. Diseño e implementación de las redes	54
4.5.1. Estructura general de las redes	54
4.5.2. Detalles del entrenamiento	55
4.6. Medición del tiempo de inferencia	56

<b>5. Resultados y discusión</b>	<b>59</b>
5.1. Resultados por base de datos . . . . .	59
5.1.1. Base de datos 1 . . . . .	59
5.1.2. Base de datos 2 . . . . .	63
5.1.3. Base de datos 3 . . . . .	67
5.2. Comparación entre bases de datos . . . . .	71
5.3. Tiempo de inferencia . . . . .	74
<b>6. Conclusión</b>	<b>79</b>
<b>Bibliografía</b>	<b>81</b>

# Resumen

De acuerdo con la Academia Nacional de Medicina de México, en el país se realizan aproximadamente 2,190 amputaciones transradiales por año. Esto implica una afectación significativa en la calidad de vida de las personas, ya que su capacidad de movilidad se ve reducida. Para abordar este problema, diversos estudios han propuesto el desarrollo de modelos de clasificación de gestos manuales basados en electromiografía, específicamente electromiografía de superficie, que permite registrar y analizar las señales mioeléctricas generadas por la contracción muscular. El objetivo de dichos modelos es su implementación en sistemas protésicos que permitan reemplazar de forma eficiente el miembro perdido.

Para clasificar los movimientos manuales se utiliza el enfoque moderno de clasificación, el cual omite la etapa de ingeniería de características y emplea modelos de aprendizaje profundo capaces de extraer automáticamente las características relevantes. Sin embargo, para obtener un alto rendimiento en términos de precisión, es necesario ajustar adecuadamente los hiperparámetros del modelo.

En este estudio se implementa un diseño D-óptimo para determinar combinaciones eficientes de cinco hiperparámetros en una red neuronal convolucional orientada a la clasificación de 18 gestos manuales a partir de señales electromiográficas de superficie. Para ello, se utilizaron tres subconjuntos de la base de datos pública Ninapro (DB1, DB2 y DB3), que contienen señales tanto de sujetos sanos como de sujetos con amputación transradial a distintos niveles.

Se evaluaron 30 modelos sobre los tres subconjuntos de datos, obteniéndose precisiones de  $97.09\% \pm 1.42\%$  para DB1,  $99.02\% \pm 0.25\%$  para DB2, y una destacada precisión de  $98.62\% \pm 0.84\%$  para DB3 (conjunto de datos correspondiente a sujetos amputados). Además, se midió el tiempo de inferencia y se determinó que los tres modelos más eficientes de cada base de datos presentan tiempos inferiores a los 200 ms al ser ejecutados en un procesador AMD Ryzen 7 5700G y, en algunos casos, en una Raspberry Pi 4 modelo B.



# Capítulo 1

## Introducción

### 1.1. Descripción

En este estudio se busca mejorar el rendimiento de una Red Neuronal de Convolución (RNC) para la clasificación de señales electromiográficas (señales EMG) correspondientes a movimientos manuales utilizando una base de datos pública. Para lograr este objetivo, se empleará un enfoque de diseño factorial para explorar la influencia de diferentes combinaciones de hiperparámetros de la red neuronal en su capacidad de clasificación de las señales EMG. El propósito último es mejorar la precisión de la red neuronal en este contexto específico, con posibles aplicaciones en campos como la rehabilitación, la interfaz cerebro-computadora y la robótica asistencial.

La evaluación del rendimiento de la red neuronal se realizará utilizando métricas estándar como la precisión, la sensibilidad, la exactitud y la puntuación F1. Se llevarán a cabo pruebas estadísticas utilizando software especializado para identificar la combinación óptima de hiperparámetros. A partir de estos resultados, se establecerán criterios para garantizar un rendimiento óptimo de la RNC en la clasificación de señales mioeléctricas.

### 1.2. Motivación

De acuerdo con la Academia Nacional de Medicina de México, en el país se realizan aproximadamente 75 amputaciones diarias. Esto implica que la calidad de vida de un gran número de personas se ve afectada por la pérdida de uno de sus miembros. Es por esta razón que surge la necesidad de crear prótesis funcionales que reemplacen de forma eficiente a la extremidad amputada.

Es aquí cuando interviene el uso de señales biológicas, específicamente la electromiografía. La electromiografía (EMG) es una técnica para analizar las señales mioeléctricas que se producen por las variaciones fisiológicas de las fibras musculares, centrándose en la activación neuromuscular voluntaria. En el caso de las extremidades superiores, los músculos controlan la posición y el movimiento de la mano mediante contracciones que generan potenciales de acción. Estas señales pueden ser adquiridas de manera no invasiva a través de electrodos superficiales, lo que las convierte en señales electromiográficas superficiales (sEMG) [31].

Actualmente, las señales EMG son objeto de estudio por parte de numerosos investigadores debido a su potencial para crear sistemas de control basados en el reconocimiento de patrones mioeléctricos. Estas señales se analizan en busca de características que puedan asociarse con movimientos específicos de las extremidades. Se han realizado diversos trabajos de investigación centrados en la clasificación de señales sEMG utilizando clasificadores como máquinas de vectores de soporte (SVM) [56], árboles de decisión, vecinos más cercanos (KNN) y análisis discriminante lineal (LDA) [68], entre otros.

En estos estudios, las señales biológicas se adquieren primero, ya sea mediante sensores superficiales o bases de datos públicas. Luego, se caracterizan utilizando métricas del dominio del tiempo, como el valor absoluto medio (VAM), la media cuadrática (MC), los cruces por cero (CC), la longitud de forma de onda (LFO) y el cambio de signo en la pendiente (CSP) [26]. Estas características se utilizan para entrenar el clasificador seleccionado y evaluar su rendimiento. Es importante mencionar que existen otras características (no necesariamente del dominio del tiempo) que pueden ser extraídas de las señales y usarse en distintas combinaciones en conjunto con un determinado clasificador. Sin embargo, no existe una regla para determinar la combinación con la que se obtendrá la mayor precisión de clasificación.

Con el advenimiento del aprendizaje profundo, este problema de clasificación se ha abordado utilizando redes neuronales convolucionales de una y dos dimensiones [62], [34], [35]. La ventaja de este enfoque radica en que no se requiere una etapa de caracterización de las señales; la red aprende las propiedades de las señales sEMG directamente de la señal instantánea. Sin embargo, la determinación de los hiperparámetros de la red mediante prueba y error limita el alcance del rendimiento óptimo en la clasificación, además de no explorar el impacto de las interacciones entre factores en el rendimiento de la red.

La importancia de este estudio radica en la optimización del rendimiento de una RNC para clasificar señales EMG mediante el ajuste sistemático de hiperparámetros usando un enfoque de diseño factorial.

### 1.3. Descripción del problema

La caracterización de las señales mioeléctricas representa un desafío debido a su comportamiento estocástico y no lineal, influenciado por una variedad de factores físicos y fisiológicos como la fatiga muscular, el desplazamiento de los electrodos, la postura del brazo y las diferencias entre sujetos [38].

Como se ha mencionado anteriormente, el enfoque tradicional para clasificar señales sEMG implica la caracterización y entrenamiento de un clasificador utilizando métricas extraídas. Sin embargo, determinar las propiedades óptimas a extraer de cada señal para lograr la máxima eficiencia en la clasificación resulta desafiante. Se tiene evidencia de que existe un límite en la cantidad de características que pueden usarse; se ha concluido que, a pesar de emplear un gran número de características (20 características), no se obtiene un aumento significativo en el rendimiento del clasificador en comparación con estudios que emplean una menor cantidad de métricas (6 características) [61].

La caracterización típicamente se centra en el dominio del tiempo, aunque algunos estudios exploran propiedades del dominio espacial y herramientas matemáticas como el análisis de Fourier y la transformada wavelet. Sin embargo, esta etapa consume tiempo y no garantiza un rendimiento óptimo del clasificador, ya que cada característica debe evaluarse en conjunto con el algoritmo de clasificación. Además, Triwiyanto *et al.* demostraron que una combinación adecuada de características produce mejores resultados en comparación con la clasificación basada en propiedades individuales [62]. Su trabajo expone que se debe poner atención no solo en las características que se extraen, sino también en la sinergia entre ellas.

Para evitar esta etapa de caracterización, se ha recurrido al aprendizaje profundo utilizando redes neuronales de convolución (RNC) de una y dos dimensiones. Este enfoque ofrece ventajas como el uso de la señal instantánea sin procesar y la capacidad de emplear el aprendizaje de transferencia para adaptar una red a un usuario específico [60]. Sin embargo, en las investigaciones sobre clasificación de señales sEMG, los parámetros de la RNC suelen ser determinados por prueba y error, lo que limita el rendimiento máximo del modelo. Además, no se comprende completamente el efecto de las interacciones entre los parámetros y cómo cada parámetro individualmente afecta al modelo.

## 1.4. Hipótesis

Un diseño factorial para optimizar los hiperparámetros de una RNC mejora la precisión en la clasificación de señales EMG.

## 1.5. Objetivo general

Optimizar el rendimiento de una RNC para la clasificación de señales electromiográficas de superficie (EMG), enfocándose en la precisión y el tiempo de procesamiento mediante el ajuste de hiperparámetros.

## 1.6. Objetivos específicos

1. Recopilar y preparar un conjunto de datos adecuado de señales EMG de movimientos manuales para el entrenamiento y evaluación de la red neuronal.
2. Definir los hiperparámetros a estudiar y diseñar el experimento con un enfoque factorial.
3. Diseñar y desarrollar una arquitectura de RNC que sea adecuada para la clasificación de señales EMG.
4. Analizar los resultados experimentales para identificar combinaciones eficientes de hiperparámetros que mejoren el rendimiento de la RNC en la clasificación de señales EMG.

## 1.7. Alcance

El alcance de este trabajo se centra en la optimización del rendimiento de una RNC en términos de precisión para clasificar señales sEMG de movimientos manuales.

Para lograr este objetivo, se aplicará un diseño factorial sobre los hiperparámetros de la red neuronal. La elección de este método proporcionará una estructura sistemática para explorar cómo diferentes combinaciones de hiperparámetros afectan el rendimiento de la RNC en la clasificación de las señales sEMG. Esto permite una evaluación exhaustiva de las diferentes configuraciones de la red y sus efectos en el rendimiento del modelo.

Además, con base en los valores óptimos de los parámetros, se pueden proponer criterios o principios que puedan ser utilizados en el aprendizaje profundo para garantizar el máximo rendimiento del algoritmo en la clasificación de señales sEMG. Esta contribución es valiosa para proporcionar pautas prácticas para futuras investigaciones en este campo.

Este estudio se realizó teniendo en cuenta la posibilidad de implementar la RNC en una prótesis en una futura investigación, sin embargo, en este trabajo no se aborda ningún aspecto relacionado con el desarrollo de un modelo de prótesis. También se resalta que para el entrenamiento, validación y prueba de las redes diseñadas se usarán únicamente bases de datos públicas, por lo que tampoco se abarcan aspectos relacionados a la adquisición de las señales mioeléctricas.



## Capítulo 2

# Estado del arte

En este capítulo se explorarán dos aspectos clave de la clasificación de señales EMG utilizando redes neuronales convolucionales (RNCs): las aplicaciones de las RNCs en la clasificación de señales EMG y las estrategias de optimización de hiperparámetros para mejorar su rendimiento. La primera sección se enfocará en el uso de las RNCs en este campo, revisando estudios previos que destacan su efectividad y los retos que presentan las señales EMG. La segunda sección se centrará en las metodologías utilizadas para la optimización de hiperparámetros, un componente esencial en el diseño de modelos precisos y eficientes, y cómo su ajuste influye en el rendimiento de los modelos para la clasificación de estas señales. La organización del capítulo sigue esta estructura, primero abordando los enfoques en el uso de RNCs y luego las estrategias de optimización de los modelos.

### 2.1. RNCs para señales EMG

En esta primera sección se presentan artículos que muestran la eficiencia de las RNCs aplicadas a diversas tareas usando señales electromiográficas. En la figura 2.1 se presenta un organigrama de los artículos, donde se muestra de forma general lo que se estudia en cada uno.

Triwiyanto et al. [62] propusieron una metodología para clasificar movimientos de la mano con redes neuronales convolucionales (RNC) utilizando señales EMG crudas, eliminando la necesidad de extracción manual de características. Se empleó una base de datos con señales de dos canales de 10 sujetos, implementando una RNC con ocho capas y 100 filtros por convolución, optimizada con Adam, que demostró ser el algoritmo más eficiente en comparación a otros.

Los resultados mostraron que la RNC superó a K vecino más cercanos (KNN), máquina de soporte vectorial (SVM) y análisis discriminante lineal, alcanzando precisiones entre 77 % y 93 %. Además, se redujo el tiempo de procesamiento sin afectar la precisión, demostrando que el uso de dos canales no ofrecía mejoras significativas. Este enfoque simplifica el procesamiento y mejora la clasificación de gestos mioeléctricos, facilitando su aplicación en prótesis controladas por EMG.

Rahimian et al. [49] propusieron una RNC con convoluciones dilatadas para clasificar gestos manuales a partir de señales sEMG, eliminando la necesidad de redes recurrentes y reduciendo el costo computacional. Utilizando la base de datos NinaPro DB2, segmentaron las señales en ventanas de 100 ms con un paso de 10 ms y diseñaron una arquitectura con tres bloques de convolución y dos bloques de identidad, ampliando el campo receptivo mediante convoluciones dilatadas.

El modelo alcanzó una precisión del 92.5 % con datos refinados y 90.1 % con datos no refinados, superando en 8.71 % a otros métodos del estado del arte. Su eficiencia computacional y precisión lo hacen ideal para aplicaciones en prótesis mioeléctricas.

Ozdemir et al. [48] propusieron un modelo de aprendizaje profundo para reconocer gestos manuales mediante señales EMG, utilizando una red residual (ResNet-50) y la transformada de Fourier de tiempo reducido (STFT) para generar imágenes espectrograma. Este enfoque mejora la

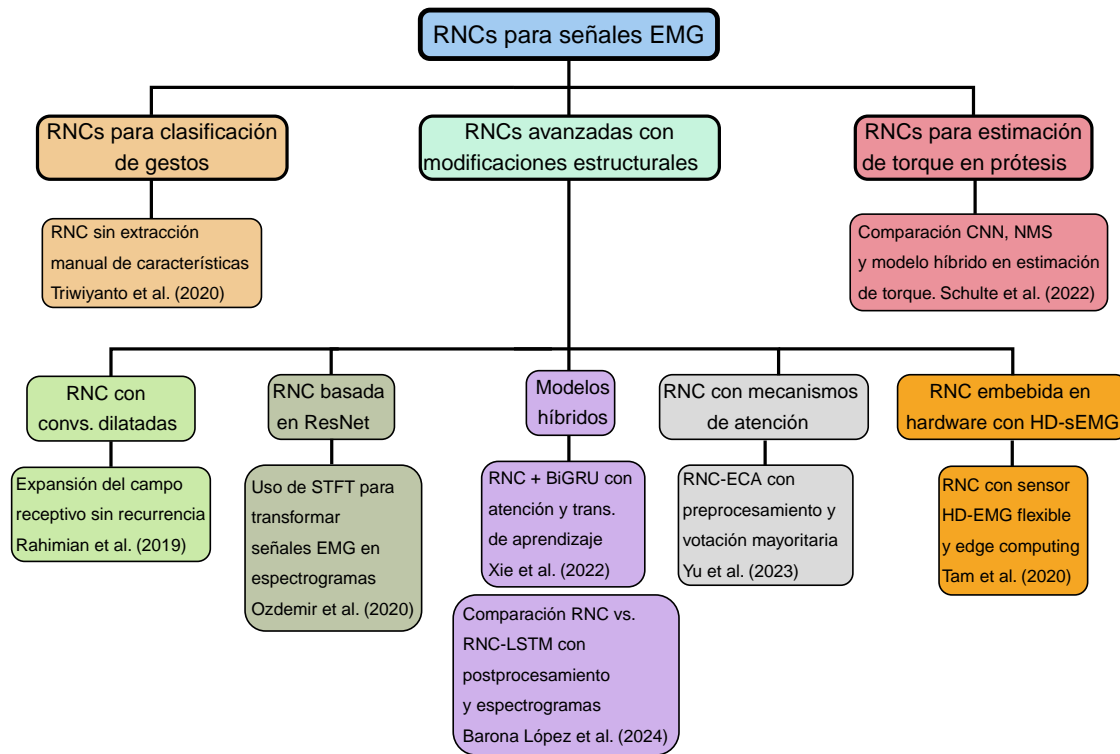


Figura 2.1: Organigramma de los artículos correspondientes al uso de RNCs para procesar señales EMG.

precisión y la velocidad de predicción, reduciendo el error en sistemas biomédicos controlados por sEMG.

La adquisición de señales se hizo con de 30 voluntarios (15 hombres y 15 mujeres) mediante el dispositivo BIOPAC MP36, con electrodos en cuatro músculos. Los participantes realizaron siete gestos manuales, generando 4200 imágenes espectrograma. Las señales fueron segmentadas y filtradas para eliminar ruido.

El modelo, entrenado con imágenes de tamaño  $64 \times 64 \times 3$  y optimizado con Adam durante 40 épocas, logró una precisión de 99.59 %. Este enfoque no solo mejora la precisión, sino que también optimiza la sincronización, siendo adecuado para aplicaciones biomédicas de control no invasivo mediante EMG.

Xie et al. [66] propusieron la red Bi-ConvGRU con atención para el reconocimiento de gestos manuales con señales EMG, orientada al control en tiempo real de prótesis. El modelo combina redes convolucionales unidimensionales para la extracción de características y redes GRU bidireccionales para capturar correlaciones temporales entre los canales. La principal novedad es el mecanismo de atención, que mejora la robustez del sistema frente al ruido y la variabilidad de las señales. Además, se aplica el aprendizaje de transferencia, entrenando inicialmente con datos de sujetos no amputados y ajustando el modelo con datos de amputados, lo que reduce el tiempo de entrenamiento.

Los resultados en la base de datos Ninapro mostraron una precisión de 88.7 % en el reconocimiento de 18 gestos, con una latencia de 160 ms, lo que hace al modelo adecuado para aplicaciones de control en tiempo real de prótesis.

Barona López et al. [13] propusieron un modelo para el reconocimiento de gestos de mano utilizando señales EMG, combinando redes RNC y LSTM con un algoritmo de post-procesamiento

para mejorar la precisión en presencia de ruido y variabilidad entre sujetos. En la figura 2.2, tomada de [13], se ilustra su metodología.

Entrenado con el dataset EMG-EPN-612, el modelo alcanzó una precisión de clasificación del  $92.00\% \pm 8.82\%$  y de reconocimiento del  $90.55\% \pm 9.45\%$ , superando al modelo RNC puro. El post-procesamiento aumentó la precisión, aunque el modelo RNC-LSTM requería significativamente más parámetros, lo que aumentó la complejidad computacional.

Este enfoque destaca el uso de técnicas de aprendizaje profundo y post-procesamiento para mejorar el reconocimiento de gestos EMG, con potencial para aplicaciones en prótesis y dispositivos biomédicos.

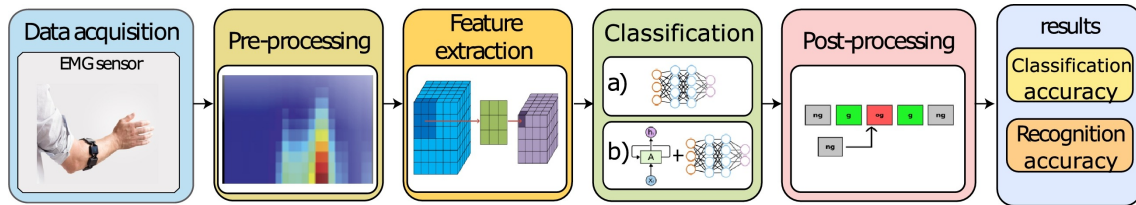


Figura 2.2: Etapas del modelo de reconocimiento de gestos de mano propuesto por Barona López et al. [13].

Yu et al. [69] desarrollaron un modelo de RNC mejorado con un mecanismo de atención de canales (RNC-ECA) para el reconocimiento de gestos con señales EMG, orientado al control en tiempo real de prótesis. El modelo incluye un módulo ECA que mejora la capacidad de discriminación al extraer características clave de los datos EMG, y utiliza la Transformada Hilbert Rápida (FHT) para reducir el ruido.

La arquitectura combina convoluciones 2D para capturar patrones temporales y espaciales, y se optimiza con capas de normalización y abandono. Evaluado en un microprocesador NVIDIA Jetson Nano, el modelo alcanzó una frecuencia de predicción de 100 fps y un tiempo de respuesta de 0.30 segundos. En el conjunto de prueba, obtuvo una precisión promedio del 95.45 %, superando otros modelos como RNC-LSTM, LSTM, SVM y KNN.

Tam et al. [59] presentaron un sistema de reconocimiento de gestos en tiempo real para el control de prótesis de mano multi-articuladas, basado en una RNC embebida y señales electromiográficas de alta densidad o HD-EMG (uso de  $\geq 32$  electrodos). Utiliza un sensor de 32 electrodos en un circuito flexible y un preprocesamiento de dominio cruzado (frecuencia-tiempo-espacial) que mejora la homogeneidad de los datos, alcanzando una precisión del 98.15 % mediante votación mayoritaria.

El modelo, optimizado para hardware embebido con Nvidia Jetson Nano, reduce la latencia a 100-200 ms. A diferencia de sistemas comerciales, permite un control fluido mediante contracciones musculares naturales. Los experimentos demostraron que la combinación de técnicas avanzadas y RNC optimizada proporciona un reconocimiento preciso y confiable, mejorando la precisión, latencia y facilidad de uso de las prótesis mioeléctricas.

Además, el sistema se puede recalibrar y afinar con facilidad, lo que permite su uso por pacientes sin conocimientos técnicos.

Schulte et al. [52] compararon tres enfoques para estimar el torque de la articulación de la rodilla en control de prótesis transfemorales: (1) RNC, (2) un modelo neuromusculoesquelético (NMS), y (3) un modelo híbrido que combina ambos. Diez sujetos realizaron actividades sin carga mientras se obtenían señales EMG de cuatro músculos, preprocesadas y normalizadas.

El rendimiento se evaluó mediante el error cuadrático medio normalizado (NRMSE). Los resultados mostraron que la RNC tuvo el mejor desempeño global (NRMSE promedio de 9.2 %), aunque mostró variabilidad entre días. El modelo NMS, aunque menos preciso (NRMSE de 14.3 %), mostró mayor robustez a través de los días. El modelo híbrido combinó las fortalezas de ambos, alcanzando un NRMSE intermedio de 12.4 %.

Este estudio resalta que las RNCs son precisas, mientras que los modelos híbridos proporcionan un balance entre precisión y robustez, lo cual es clave para el desarrollo de sistemas de control de prótesis basados en EMG.

## 2.2. Optimización de hiperparámetros en clasificación de señales EMG

A continuación se presentan estudios enfocados en la optimización de hiperparámetros para clasificar gestos manuales usando señales EMG. En la figura 2.3 se presenta el organigrama de los artículos de esta sección, donde se muestra de forma general lo que se hace en cada uno y cómo se relacionan.

Sehat et al. [53] presentaron un modelo de RNC para la clasificación de gestos manuales usando señales EMG del antebrazo, con un enfoque innovador en la optimización de hiperparámetros mediante un algoritmo genético. El objetivo fue mejorar la precisión en la clasificación de ocho gestos, optimizando parámetros como el número de capas de convolución, neuronas en la capa densa y el número de kernels.

El modelo inicial alcanzó una precisión de 91.86 %, mientras que, tras la optimización, la precisión aumentó a 96.4 % en tiempo real y 99.6 % en modo offline. Sin embargo, algunos gestos, como el de reposo y el de doble toque, seguían siendo difíciles de distinguir debido a la similitud en las posiciones de las manos.

El estudio concluye que, aunque la optimización genética mejoró la precisión, persisten desafíos relacionados con el costo computacional y la clasificación de gestos complejos, lo que subraya la importancia de la optimización de hiperparámetros en el desarrollo de modelos de clasificación para señales EMG.

El estudio de Asif et al. [7] se centra en el impacto de los hiperparámetros en RNC para la clasificación de gestos manuales usando señales EMG. Se destaca la importancia de optimizar estos valores, dada la variabilidad estocástica de las señales mioeléctricas entre sujetos, en lugar de confiar en enfoques empíricos. El análisis estadístico se considera crucial para mejorar la precisión y diseñar sistemas de control robustos para prótesis mioeléctricas.

El modelo RNC consistió en 15 capas, incluidas convolución, normalización y agrupación con activación ReLU. Se ajustaron hiperparámetros como la tasa de aprendizaje y el número de épocas mediante búsqueda en cuadrícula. Los resultados mostraron que tasas de aprendizaje de 0.0001 y 0.001 lograron precisiones superiores al 90 % en algunos gestos (ver figura 2.4).

El estudio concluye que la optimización mejora significativamente el rendimiento de la red, aunque no se puede generalizar un conjunto de hiperparámetros para todos los sujetos. Una limitación del estudio es la portabilidad del sistema, ya que los sistemas de bajo consumo no pueden replicar el rendimiento de GPUs dedicadas.

Kulwa et al. [36] investigaron el impacto del tamaño de la ventana, el solapamiento y el tamaño del kernel, en la clasificación de señales EMG mediante RNC. Su objetivo fue encontrar la combinación óptima de estos valores para mejorar el rendimiento. Propusieron un modelo de RNC multi-escala con menor complejidad computacional, utilizando convolución dilatadas para extraer características generales y detalladas de las señales.

El estudio utilizó tres bases de datos: CapgMyo, HD-sEMG y NinaPro. Las señales fueron segmentadas en ventanas de 25 a 200 ms, con solapamientos del 0 %, 25 %, 50 % y 75 %. La red utilizó diferentes tamaños de kernel ( $3 \times 3$ ,  $5 \times 5$ , y  $7 \times 7$ ), y se evaluó con validación cruzada de 5 pliegues. Los resultados mostraron que un mayor solapamiento (75 %) y un kernel de  $7 \times 7$  mejoraron la precisión pero aumentaron la latencia, mientras que la configuración de kernel  $7 \times 7$  y solapamiento del 50 % mantuvo la latencia por debajo de los 200 ms.

Este trabajo ofrece una guía práctica para seleccionar hiperparámetros en modelos RNC para señales sEMG, optimizando precisión y latencia de manera computacionalmente eficiente.

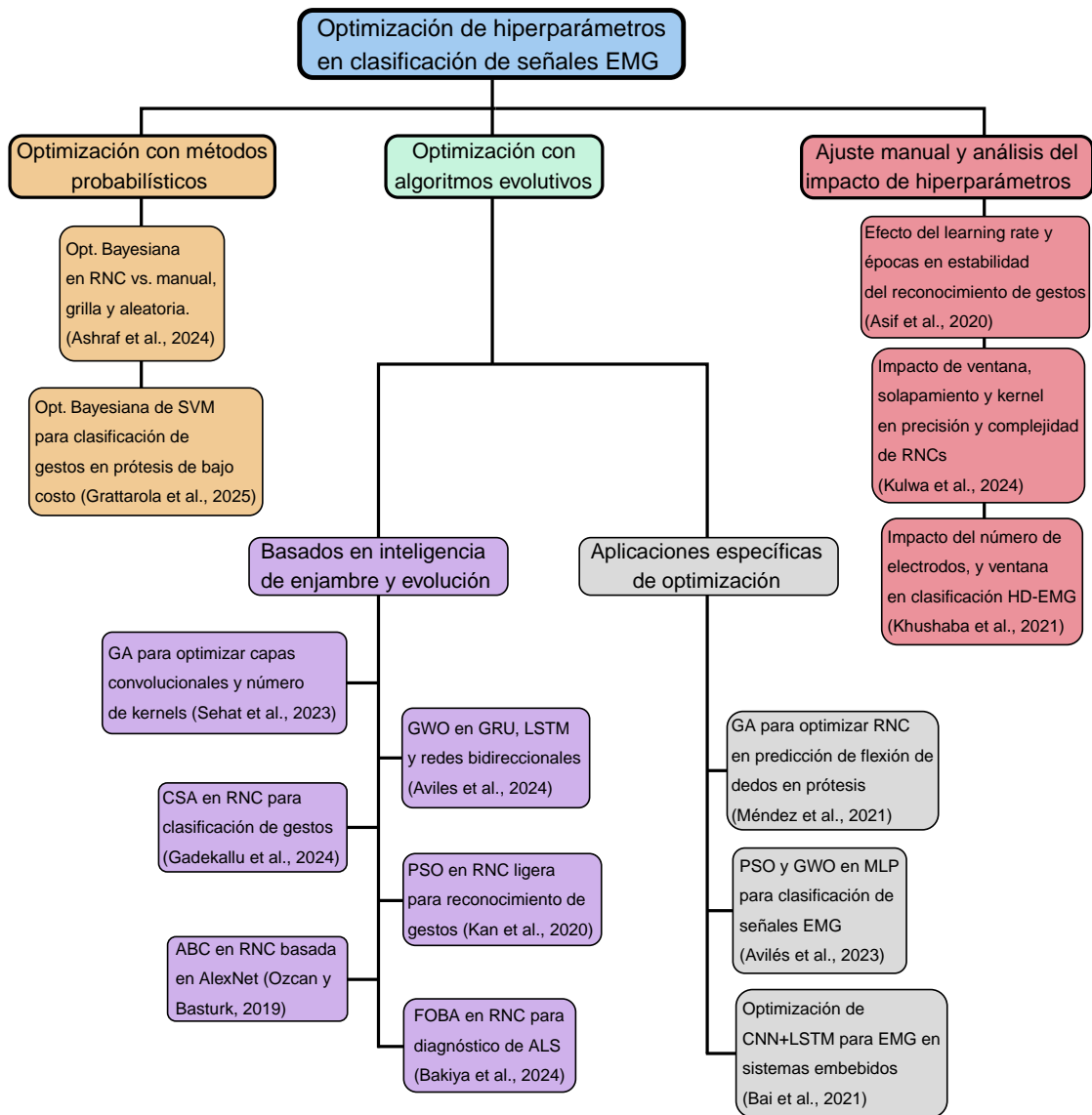


Figura 2.3: Organigrama de los estudios enfocados en la optimización de diversos hiperparámetros, la mayoría de ellos usando RNCs.

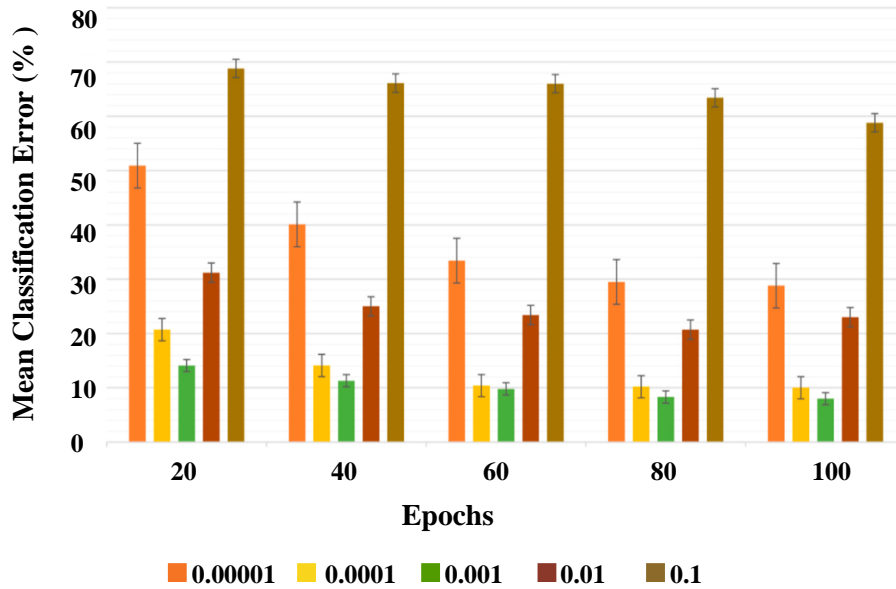


Figura 2.4: Error de clasificación medio promediado para todos los sujetos para cada tasa de aprendizaje a lo largo de distintos números de épocas de entrenamiento. Imagen tomada de [7]

El estudio de Ashraf et al. [6] se enfoca en la optimización de hiperparámetros en RNC para la clasificación de señales HD-EMG, con el objetivo de mejorar el reconocimiento de gestos en sistemas de prótesis. Los autores evaluaron diferentes técnicas de segmentación, comparando segmentación superpuesta y disjunta, y aplicaron la optimización Bayesiana para ajustar hiperparámetros clave como la tasa de aprendizaje, el momentum y la regularización L2. Utilizando datos de 20 voluntarios que emplearon la banda *Myo Armband*, se encontró que la segmentación superpuesta con ventanas de 200 ms y un solapamiento del 80 % optimiza el rendimiento.

Los resultados mostraron que esta combinación de segmentación y optimización de hiperparámetros redujo la tasa de error de clasificación media a  $0,08 \pm 0,03$ . Además, los autores destacaron que la optimización Bayesiana mejora la eficiencia del ajuste de los hiperparámetros y la reproducibilidad de los resultados, lo que podría beneficiar el control mioeléctrico en sistemas protésicos y otras interfaces hombre-máquina. En la figura 2.5 se muestra su arquitectura optimizada.

El estudio de Aviles et al. [9] se enfoca en la optimización de redes neuronales recurrentes (RNR), como LSTM, GRU y redes bidireccionales, para la clasificación de señales EMG, aplicando un algoritmo de optimización de lobos grises (GWO) para ajustar hiperparámetros. Este enfoque busca mejorar la precisión y reducir el costo computacional en comparación con métodos tradicionales como la búsqueda en cuadrícula. Se optimizaron hiperparámetros clave, como la cantidad de neuronas, el tamaño del lote, el número de épocas y la tasa de aprendizaje.

Utilizando datos de nueve participantes que realizaron cinco movimientos de la extremidad superior, los resultados mostraron que la arquitectura LSTM fue la más eficiente, alcanzando una precisión de 100 % en la primera fase experimental, la figura 2.6 muestra el error de clasificación. En la segunda fase, LSTM mantuvo una alta precisión de 98.46 %, superando los otros modelos. Los resultados sugieren que la combinación de RNR y el algoritmo de lobos grises mejora la precisión y el tiempo de entrenamiento, presentando una solución prometedora para la clasificación de señales EMG en aplicaciones biomédicas.

Gadekallu et al. [19] proponen un modelo de clasificación de gestos de mano utilizando RNC,

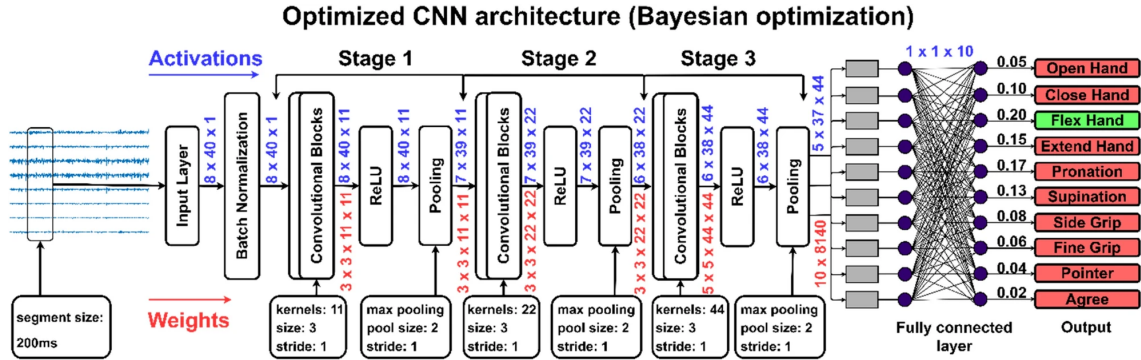


Figura 2.5: La arquitectura de la RNC se diseñó usando optimización Bayesiana para ajustar los hiperparámetros. Consiste en dos bloques convolucionales, cada uno con una capa convolucional, una capa de normalización por lotes y una capa ReLU. Imagen obtenida de [6]

optimizado mediante el algoritmo de búsqueda de cuervos. Este algoritmo, inspirado en el comportamiento de los cuervos, selecciona automáticamente los hiperparámetros más adecuados, como el número de capas, el tamaño del lote y las funciones de activación y pérdida. Su objetivo es mejorar la precisión y reducir los tiempos de entrenamiento en comparación con métodos tradicionales como la búsqueda aleatoria.

Utilizando una base de datos de gestos de mano de Kaggle, los resultados mostraron una precisión de 100 % en entrenamiento y prueba en solo 16 minutos, superando a otros modelos optimizados con algoritmos como GWO y el de enjambre de partículas. El estudio demuestra que el algoritmo de búsqueda de cuervos, combinado con RNC, mejora la clasificación de gestos al optimizar automáticamente los hiperparámetros y reducir los tiempos de cómputo.

Kan et al. [29] proponen una RNC ligera optimizada mediante el algoritmo de optimización por enjambre de partículas (PSO) para mejorar la clasificación de señales EMG. Su enfoque busca aumentar la precisión y reducir la complejidad del modelo, utilizando una arquitectura con tres capas de convolución y dos capas densas. La PSO ajusta hiperparámetros como el número de filtros, el tamaño de los kernels y la tasa de aprendizaje, optimizando automáticamente el modelo con la entropía cruzada como función de aptitud. Su metodología se muestra en la figura 2.7.

El estudio, basado en la base de datos Ninapro, demuestra que esta metodología supera a otras técnicas en precisión y velocidad, alcanzando un 90.06 % en movimientos de muñeca y 86.67 % en movimientos de mano. Además, la RNC ligera reduce significativamente el tiempo de clasificación en comparación con métodos tradicionales, lo que la hace ideal para aplicaciones en interacción humano-máquina.

Méndez et al. [42] emplearon un algoritmo genético para optimizar hiperparámetros en una RNC aplicada al control proporcional de movimientos de dedos con señales EMG. La optimización ajustó el número de capas convolucionales, filtros, tasa de aprendizaje y estrategias de regularización, reduciendo el tiempo de ajuste y mejorando la precisión del modelo.

Utilizando datos de 4 sujetos, el modelo optimizado superó el método tradicional de ingeniería de características en la regresión de ángulos de flexión de los dedos, logrando una pérdida de validación de 5.34. Este enfoque demostró ser eficiente al reducir la complejidad computacional y mejorar la precisión en la predicción de movimientos mioeléctricos.

Avilés et al. [10] utilizaron optimización de enjambre de partículas (PSO) y el optimizador de lobos grises (GWO) para ajustar hiperparámetros en un perceptrón multicapa (MLP) aplicado a la clasificación de señales EMG. Se optimizaron parámetros como el número de neuronas, capas ocultas, tasa de aprendizaje, épocas y tamaño del lote.

Trabajando con datos de nueve sujetos y cinco movimientos capturados con cuatro canales

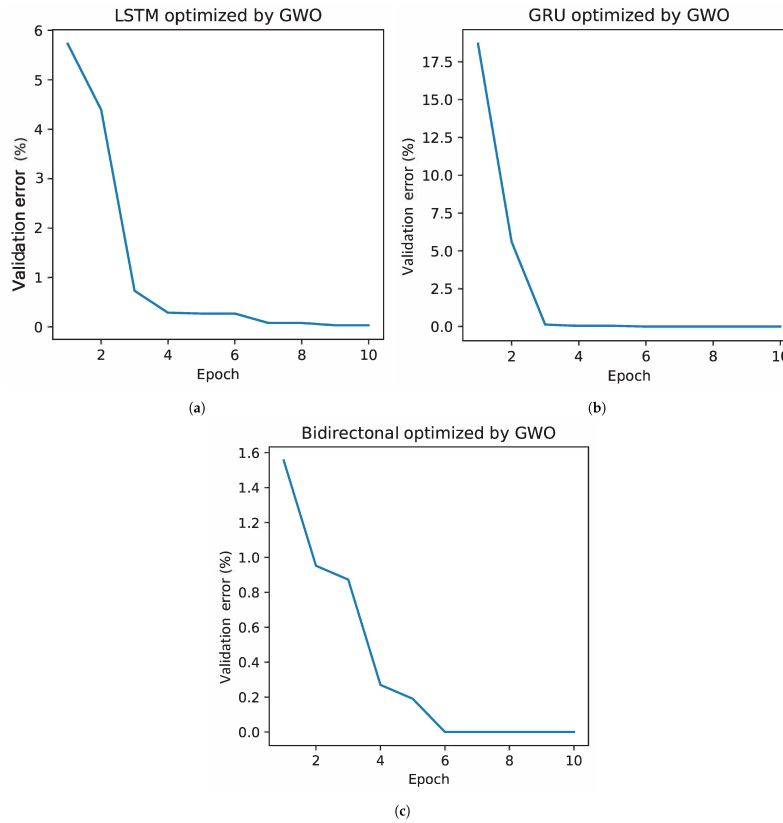


Figura 2.6: Disminución en el error de clasificación gracias a la selección de hiperparámetros con GWO. Imagen tomada de [9].

bipolares, el estudio comparó ambos optimizadores, encontrando que GWO superó a PSO, logrando una precisión del 97 % frente al 93 %. Además, GWO permitió reducir la complejidad del modelo sin comprometer la generalización.

Este trabajo demuestra la efectividad de los algoritmos metaheurísticos para mejorar modelos de aprendizaje profundo en clasificación de señales EMG, proporcionando un enfoque adaptable a distintas aplicaciones biomédicas.

Ozcan y Basturk [47] emplearon transferencia de aprendizaje y optimización metaheurística en AlexNet para el reconocimiento de gestos de la mano, ajustando hiperparámetros como el número de épocas, tamaño del lote y tasa de aprendizaje con colonia de abejas artificial (ABC), PSO y GA.

Utilizando bases de datos de imágenes de lenguaje de señas, el modelo optimizado con ABC obtuvo 98.40 % y 98.09 % de precisión, superando a otros métodos. Aunque no trabaja con señales EMG, el estudio demuestra la eficacia de combinar transferencia de aprendizaje con optimización de hiperparámetros, ofreciendo un enfoque adaptable al reconocimiento de señales mioeléctricas.

Bakiya et al. [12] optimizaron una RNC para clasificación de señales EMG mediante el algoritmo de murciélago con cálculo fraccional (FOBA), ajustando hiperparámetros como los pesos iniciales, frecuencia del pulso y tasa de emisión. Utilizando la base EMGLAB con 150 registros de bíceps braquiales en condiciones normales, con miopatía y ALS, transformaron las señales en imágenes tiempo-frecuencia mediante la transformada pseudo-Wigner-Ville suavizada (SP-WVT).

El modelo optimizado con FOBA alcanzó una precisión del 99.11 %, superando la versión estándar del algoritmo de murciélago en eficiencia y convergencia. Este enfoque demuestra el potencial

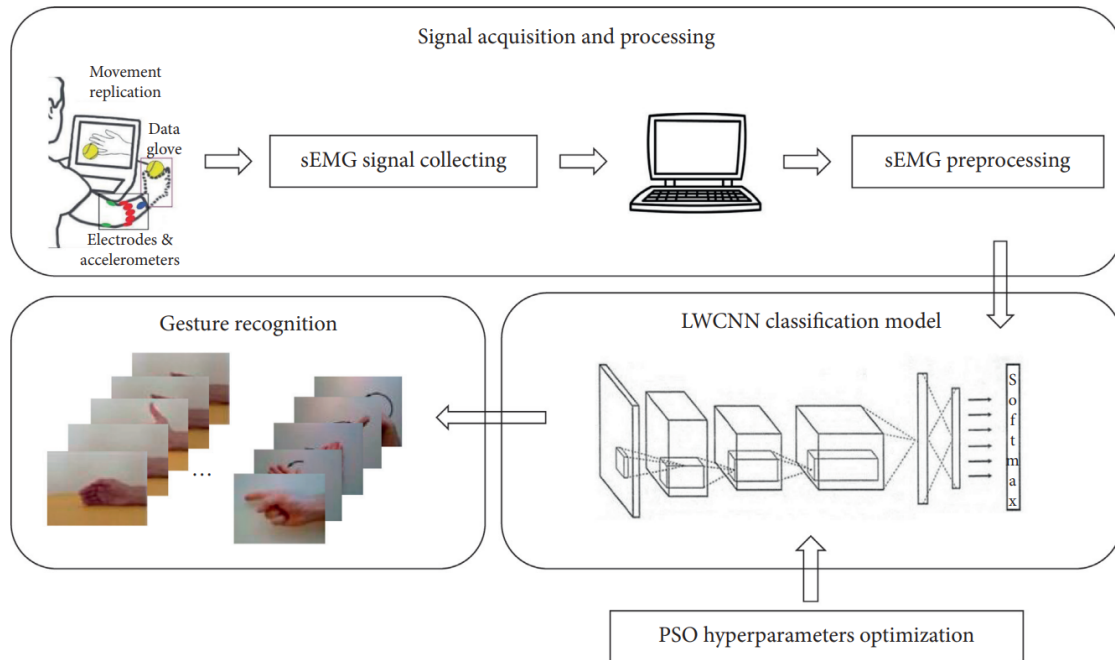


Figura 2.7: Los autores propusieron una RNC ligera para clasificar gestos manuales usando señales EMG. Optimizaron los hiperparámetros mediante el algoritmo de PSO. Imagen obtenida de [29].

de la optimización fraccional en el diagnóstico asistido por computadora de desórdenes neuromusculares.

Bai et al. [11] desarrollaron un modelo híbrido RNC-LSTM optimizado para el reconocimiento de gestos con señales EMG, enfocado en implementación en hardware embebido. Utilizando un módulo propio de 18 electrodos, las señales fueron transformadas con FFT y RMS para mejorar su representación.

Con datos de 10 sujetos y 16 gestos predefinidos, el modelo alcanzó una precisión del 91.40 % con un tamaño de solo 1 MB, permitiendo su implementación en microprocesadores STM32 para el control en tiempo real de prótesis de mano.

Grattarola et al. [21] desarrollaron un sistema de reconocimiento de patrones de agarre para prótesis de bajo costo basado en señales EMG adquiridas con el Myo armband. Utilizando la base NinaPro DB5 (10 sujetos, 8 electrodos, 200 Hz), extrajeron 8 características en el dominio tiempo-frecuencia y emplearon un SVM optimizado con técnicas bayesianas.

El modelo, estructurado con 36 clasificadores binarios para diferenciar 9 patrones de agarre, alcanzó una precisión del 85 %, demostrando su viabilidad para prótesis accesibles y de bajo costo.

Khushaba et al. [30] analizaron el impacto del número de electrodos y el tamaño de la ventana en la precisión de la clasificación de señales HD-EMG, utilizando tres bases de datos con hasta 256 electrodos y un clasificador LDA. Se evaluaron ventanas de 32 ms a 256 ms, encontrando que con 64 electrodos el error cae por debajo del 15 %, y con más de 128 se reduce a menos del 10 %.

El estudio demuestra que aumentar los electrodos mejora la precisión, mientras que ventanas más grandes afectan negativamente el rendimiento. Esto proporciona un enfoque sistemático para optimizar estos parámetros, equilibrando precisión y latencia en la clasificación de señales EMG.

El uso de RNCs para clasificar señales EMG ha sido preciso, pero aún no se comprende bien cómo interactúan sus hiperparámetros ni su efecto en el rendimiento. Aunque se han optimizado algunos mediante métodos como PSO, GA y optimización Bayesiana, falta un análisis sistemático

de su interacción. Además, aspectos estructurales como el número de capas o la función de activación han sido poco explorados. Esto dificulta el desarrollo de estrategias eficientes, por lo que es necesario estudiar la interacción entre múltiples hiperparámetros para mejorar la precisión y reducir la complejidad computacional.

# Capítulo 3

## Marco teórico

En esta sección se presentan los fundamentos teóricos necesarios para el desarrollo del trabajo. El contenido está estructurado en tres secciones: primero, se describen las señales electromiográficas (EMG), su origen y propiedades; posteriormente, se aborda el diseño experimental, en particular el diseño factorial y su optimización mediante el diseño D-óptimo, el cual permite reducir la cantidad de experimentos sin perder información relevante; finalmente, se introduce el concepto de redes neuronales convolucionales (RNCs) y sus componentes e hiperparámetros.

### 3.1. Señales electromiográficas (EMG)

#### 3.1.1. Definición de la electromiografía

La electromiografía es una técnica experimental que implica el desarrollo, grabación y análisis de señales mioeléctricas. Estas señales eléctricas son generadas por cambios en el estado de la membrana de las fibras musculares. La electromiografía puede clasificarse en dos tipos: neurológica, donde se analizan las respuestas musculares producidas por estimulación eléctrica externa en condiciones estáticas; y kinesiológica, que se concentra en estudiar la activación neuromuscular durante actividades posturales, movimientos funcionales o regímenes de entrenamiento [32].

Una de las ventajas más notables de la electromiografía superficial es que es una técnica no invasiva, es decir, no requiere penetrar la piel para cuantificar la actividad muscular. Este método permite una adquisición fácil y segura de las señales mioeléctricas, además de que el observador puede apreciar la actividad del músculo en estado de reposo y los continuos cambios que se generan durante un movimiento.

Mediante el uso de arreglos de múltiples sensores, es posible observar la relación entre distintos músculos y movimientos. La electromiografía superficial complementa la información obtenida a través de otras técnicas como la palpación, las pruebas musculares y la observación de las posturas o movimientos. De esta manera, los profesionales de la salud pueden verificar aspectos fundamentales como la congruencia de la palpación con la actividad que se está realizando, si los músculos se activan antes o después de los patrones de reclutamiento esperados, si la tarea realizada activa el músculo que debería o incluso si los músculos se desactivan durante la ejecución de un movimiento [17].

Otra ventaja de la electromiografía superficial es su valor en la terapia neuromuscular, ya que la información extraída puede emplearse como retroalimentación para que los pacientes sean capaces de afinar la respuesta muscular de acuerdo con las indicaciones verbales del terapeuta. Por ejemplo, si la tarea es mantener los músculos relajados, el paciente puede observar si realmente está cumpliendo el objetivo o necesita ajustar su patrón de reclutamiento. Adicionalmente, es posible comparar las señales mioeléctricas grabadas al inicio de un tratamiento con las adquiridas

---

en sesiones más recientes. Esto sirve para retroalimentar y motivar al paciente [32].

Así como se mencionan las ventajas de la electromiografía superficial, también es necesario abordar sus limitaciones. Una de sus debilidades clave está relacionada con la cantidad de músculos que se pueden estudiar, dado que el sistema neuromuscular es complejo, extenso e intrincado. Emplear un sistema de adquisición con uno o dos canales limita considerablemente la cantidad de información obtenida. Para abordar esta cuestión, se pueden emplear arreglos con múltiples sensores, de modo que se capture más información; o bien, se puede optar por analizar distintas regiones musculares en estado de reposo para seleccionar aquellas que sean de interés para un estudio particular.

De igual forma, los patrones de sustitución muscular presentan otro desafío en el estudio de las señales de electromiografía. Se sabe que el sistema neuromuscular puede realizar una misma tarea a través de diferentes grupos musculares, lo que podría llevar a la errónea conclusión de que las señales son inconsistentes o poco confiables. Para enfrentar este problema, se sugiere estudiar el caso normal, que se refiere a los patrones típicos o estándar de actividad muscular esperados durante ciertos movimientos o acciones en personas sin anomalías neuromusculares. Entender el caso normal significa conocer cómo debería verse la actividad muscular en condiciones típicas, lo cual es esencial para interpretar los registros de sEMG con precisión y evitar malentendidos sobre la fiabilidad de los datos.

Otra dificultad que se presenta es la interferencia cruzada, que se refiere a la situación en la que la señal registrada por un electrodo corresponde a la actividad eléctrica de un músculo adyacente, en lugar del músculo que se pretende monitorear. Esto puede generar problemas en la especificidad de las señales EMG, ya que puede volverse imposible aislar la actividad de un músculo específico. Por lo tanto, es fundamental considerar que algunos sitios de colocación de electrodos ofrecen una mayor especificidad que otros.

Una cuestión estrechamente relacionada con el punto anterior es la carencia de un método estándar de colocación de electrodos. La consecuencia de esto es que las señales del mismo músculo adquiridas en dos clínicas o estudios puedan diferir en el patrón de actividad debido a las diferencias en la posición de los electrodos.

Es necesario aclarar que la electromiografía no es una medida de fuerza ni del esfuerzo realizado al ejecutar un movimiento. Más bien, lo que se mide es la actividad eléctrica del músculo estudiado. Las señales EMG de dos grupos musculares no pueden compararse directamente, sino que primero deben ser normalizadas mediante alguna técnica. Por ejemplo, se pueden referenciar los grupos musculares a una contracción isométrica voluntaria máxima (MVIC) y trabajar con el porcentaje de MVIC.

Finalmente, una debilidad técnica de la electromiografía superficial tiene que ver con los electrodos y cables, que pueden interferir con la ejecución de un movimiento o hacer que el paciente se sienta inhibido al realizar una postura o tarea. En la literatura, se recomienda disponer de una variedad de electrodos para elegir el tipo correcto según el patrón de movimiento que se desee estudiar [17].

### 3.1.2. Origen de las señales EMG

Para entender el origen de las señales de electromiografía, es necesario profundizar en el nivel básico del sistema nervioso muscular. La unidad responsable del control neuronal que desencadena la contracción muscular es la unidad motora, que incluye su sistema motor alfa compuesto por la neurona motora inferior, el axón y las fibras musculares que inerva (ver figura 3.1). El término “unidad” subraya el hecho de que todas las fibras musculares de una unidad motora actúan conjuntamente durante la inervación. El axón motor inferior se ramifica, lo que le permite conectarse con las fibras musculares en la placa motora terminal, donde se generan las sinapsis neuromusculares [17].

El fenómeno de la excitabilidad de las fibras musculares mediante el control neuronal se describe utilizando un modelo de membrana semipermeable [32]. Para explicar la biofísica de las señales

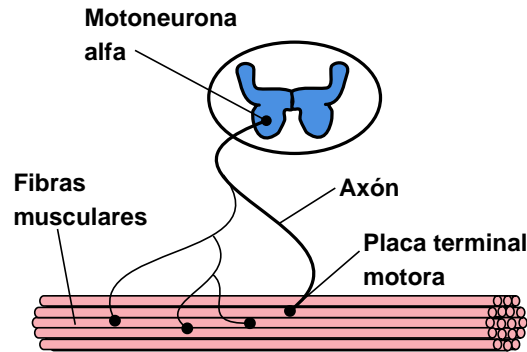


Figura 3.1: Unidad motora. Figura tomada y adaptada de [32].

EMG, es esencial primero entender el estado de una célula en reposo. El equilibrio iónico entre el interior y el exterior de la célula muscular produce un potencial de reposo (aproximadamente  $-90$  mV), lo que significa que el potencial dentro de la fibra es  $90$  mV más negativo que en el medio extracelular [39].

Varios factores determinan este potencial de membrana. Uno de ellos es la existencia de la bomba sodio-potasio (bomba  $\text{Na}^+\text{-K}^+$ , una proteína transportadora encargada de bombear continuamente tres iones de sodio al exterior y dos iones de potasio al interior de la membrana. El bombeo de más iones de sodio hacia el exterior que iones de potasio hacia el interior provoca una pérdida continua de carga positiva, generando alrededor de  $-4$  mV en el medio intracelular [22], [43].

Otra contribución fundamental proviene de la difusión de iones de sodio y potasio a través de los canales de fuga de la membrana. La concentración de potasio es mayor en el interior que en el exterior de la membrana (la proporción es 35:1). Supongamos que, por ahora, la membrana es únicamente permeable al potasio. Debido al alto gradiente de concentración de potasio, existe una marcada tendencia de que los iones adicionales en el interior se difundan hacia el medio extracelular a través de la membrana, creando electronegatividad en el interior debido a la salida de carga positiva y la permanencia de aniones que no se difunden. En cuestión de milisegundos, la diferencia de potencial entre el interior y el exterior, llamada potencial de difusión, se vuelve lo suficientemente grande como para impedir la salida de más iones de potasio [22]. La figura 3.2 ilustra los componentes mencionados en una célula, la dirección de salida y entrada de los iones es indicada para los canales de cada ion.

La ecuación de Nernst describe la relación entre el potencial de difusión y la diferencia de concentración de iones a través de la membrana. El potencial de Nernst es el nivel de potencial de difusión que se opone exactamente a la difusión neta de un ion. Su magnitud depende de la proporción de concentración de un determinado ion en ambos lados de la membrana; cuanto mayor es la proporción, mayor es la tendencia del ion a difundirse hacia un medio, y también deberá aumentar el potencial de Nernst para prevenir la difusión adicional. Para calcular el potencial de Nernst de cualquier ion univalente a la temperatura corporal normal ( $37^\circ$ ), se utiliza la ecuación (3.1), conocida como la ecuación de Nernst:

$$\text{EMF} = \pm \frac{61}{z} \times \log \frac{\text{Concentración interior}}{\text{Concentración exterior}} \quad (3.1)$$

donde EMF es la fuerza electromotriz en milivoltios y  $z$  es la carga eléctrica del ion ( $+1$  en el caso de  $\text{K}^+$ ). Al utilizar esta ecuación, se considera que el potencial en el medio extracelular es cero, mientras que el potencial de Nernst corresponde al interior de la membrana. El potencial se considera positivo (+) si el ion que se difunde del interior al exterior tiene carga negativa, y

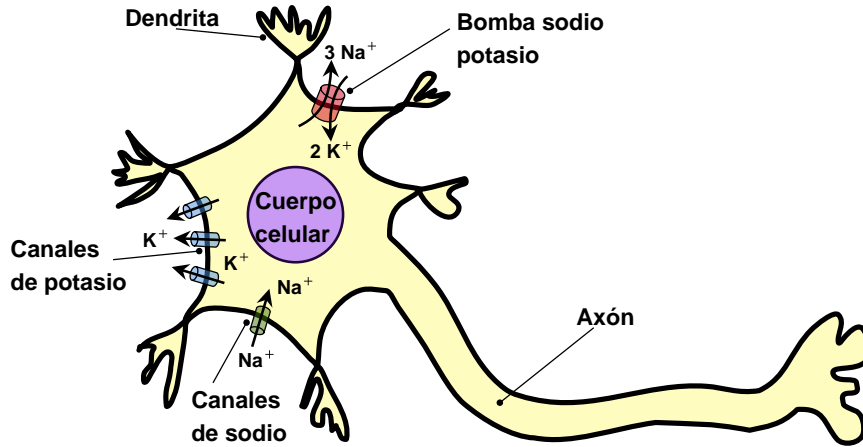


Figura 3.2: Principales componentes encargados de mantener, cambiar y restaurar el potencial de membrana.

negativo (−) si es un ion positivo. En el caso del potasio, con su proporción de concentración de 35:1, el potencial de Nernst es de -94 mV [22], [39].

Para los iones de sodio (que también tienen carga positiva), ocurre lo contrario: la proporción desde el interior al exterior es de 0.1, lo que provoca una difusión de iones hacia el espacio intracelular, y el potencial de membrana queda con la polaridad opuesta al generado por la difusión del potasio, es decir, con negatividad fuera de la membrana. El potencial de Nernst dentro de la membrana es de +61 mV [22].

Cuando una membrana es permeable a más de un tipo de ion, el potencial de difusión depende de la polaridad de la carga eléctrica de cada ion, la permeabilidad de la membrana a cada ion, y la concentración de iones dentro y fuera de la membrana. La ecuación (3.2), conocida como la ecuación de Goldman, se utiliza para calcular el potencial de membrana en el interior cuando intervienen dos iones positivos univalentes ( $K^+$  y  $Na^+$ ) y un ion negativo univalente ( $Cl^-$ ) [22].

$$EMF = -61 \times \log \frac{C_{Na^+i}PNa^+ + C_{K^+i}PK^+ + C_{Cl^-e}P_{Cl^-}}{C_{Na^+e}PNa^+ + C_{K^+e}PK^+ + C_{Cl^-i}P_{Cl^-}} \quad (3.2)$$

Aquí  $C$  denota la concentración de un ion específico, el subíndice  $i$  hace referencia al interior de la membrana, mientras que el subíndice  $e$  se refiere al exterior; la permeabilidad a un ion específico se representa con  $P$ .

En la fibra nerviosa normal, la permeabilidad al potasio es cien veces mayor que al sodio, por lo que la principal contribución proviene de la difusión de iones  $K^+$ , esto se ilustra en la figura 3.2, donde se muestran más canales de potasio que de sodio. Utilizando la ecuación de Goldman, se obtiene que el potencial dentro de la membrana generado por la difusión de sodio y potasio a través de los canales de fuga es de -86 mV. Por tanto, el potencial de membrana neto es de -90 mV cuando todos los factores mencionados operan en conjunto [22].

La importancia de comprender el potencial de membrana radica en que la estimulación de la neurona no es más que un cambio en dicho potencial, conocido como **potencial graduado**. Este potencial se genera por la apertura de canales iónicos regulados durante la sinapsis neuronal y puede ser negativo o positivo, dependiendo del tipo de ion que atraviese el canal [39]. Por ejemplo, la apertura de canales de sodio permite la entrada de iones  $Na^+$ , lo que despolariza la neurona al hacer que el potencial de membrana sea más positivo. En contraste, si se abren canales de potasio, los iones  $K^+$  salen de la célula, causando su hiperpolarización [41], [22], como se muestra en la figura 3.3.

Para que la señal nerviosa se transmita a otras células, el cambio en el potencial de membrana

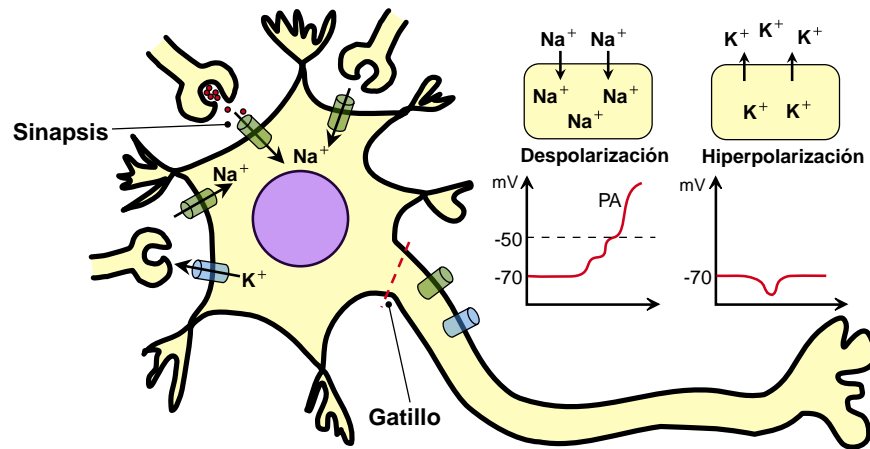


Figura 3.3: Los potenciales graduados se generan por la sinapsis neuronal. La entrada de iones positivos al medio intracelular produce la despolarización (potencial positivo). La suma de varios potenciales graduados puede dar lugar a un potencial de acción (PA).

debe ser lo suficientemente significativo como para activar la zona gatillo, ubicada en el segmento inicial del axón, donde se encuentran más canales regulados por voltaje [43]. Sin embargo, la resistencia del citoplasma provoca que la entrada de iones se disipe, disminuyendo el potencial en la zona gatillo. Por ello, es necesario sumar varios potenciales graduados para alcanzar el potencial umbral que activa los canales del axón. Esta suma puede ser espacial, al abrirse un mayor número de canales en la membrana (como en la figura 3.3 donde se muestran varias terminales axónicas para la sinapsis), o temporal, cuando un canal permanece abierto por más tiempo. Si la suma alcanza el umbral de la zona gatillo, se activan los canales regulados por voltaje y se generan los **potenciales de acción** [39].

El potencial de acción es una señal de conducción que se propaga a lo largo del axón y se inicia cuando el potencial de membrana supera el umbral necesario para abrir los canales de sodio y potasio (pasos 1 y 2 de la figura 3.4). Los canales de sodio tienen una compuerta de activación, que permanece cerrada en reposo, y una compuerta de inactivación. Cuando el potencial de membrana se vuelve más positivo (aproximadamente de  $-90$  mV a  $-50$  mV), la compuerta de activación se abre, permitiendo el influjo de iones de sodio. Esto hace que la membrana se vuelva de 500 a 5000 veces más permeable al sodio [22], provocando un aumento súbito del potencial de membrana, conocido como **despolarización** (paso 2 en la figura 3.4).

Cuando el potencial de membrana alcanza los  $+30$  mV, la compuerta de inactivación se cierra, deteniendo la entrada de iones de sodio. Simultáneamente, se abren los canales de potasio, permitiendo la salida de iones  $K^+$  (paso 3 de la figura 3.4). La disminución de la entrada de sodio y el aumento en la salida de potasio aceleran el proceso de **repolarización**, restaurando el potencial de membrana en reposo (pasos 4-7 de la figura 3.4). Todo este proceso, que involucra la entrada y salida de iones, ocurre en solo unas pocas diezmilésimas de segundo. Además, una característica distintiva del potencial de acción es que su intensidad es invariable, alcanzando siempre los  $+30$  mV [22].

Los potenciales de acción viajan a lo largo del nervio motor hasta sus extremos en la fibra muscular, donde el nervio libera un neurotransmisor llamado acetilcolina. Esta sustancia actúa en la placa terminal motora, la región donde el axón hace sinapsis con la fibra muscular (inciso a de la figura 3.5), y provoca la apertura de canales regulados por acetilcolina en la membrana de la fibra (inciso b de la figura 3.5). A través de estos canales, iones de sodio se difunden hacia el interior de la célula, causando una despolarización local que se propaga a lo largo de los túbulos transversales (túbulos T) que atraviesan la fibra muscular de un extremo al otro [22], [39].

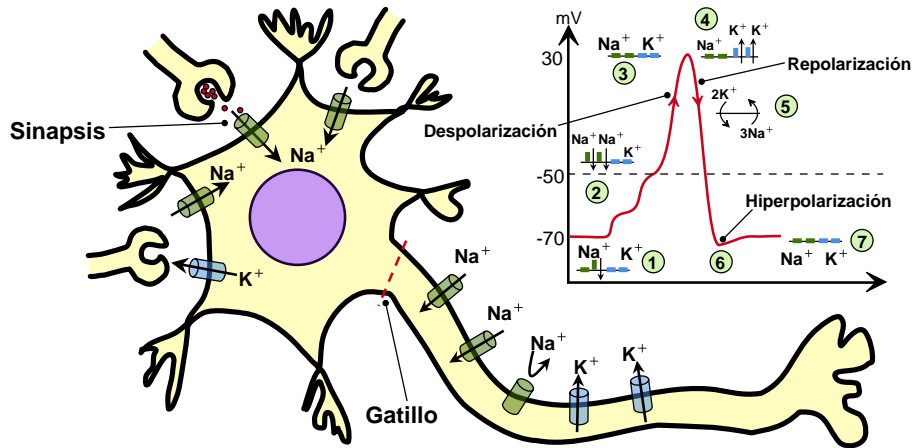


Figura 3.4: Generación del potencial de acción. En el axón se muestran los canales iónicos regulados por voltaje, primero se abren los canales de sodio y, una vez que se cierran, se abren los de potasio.

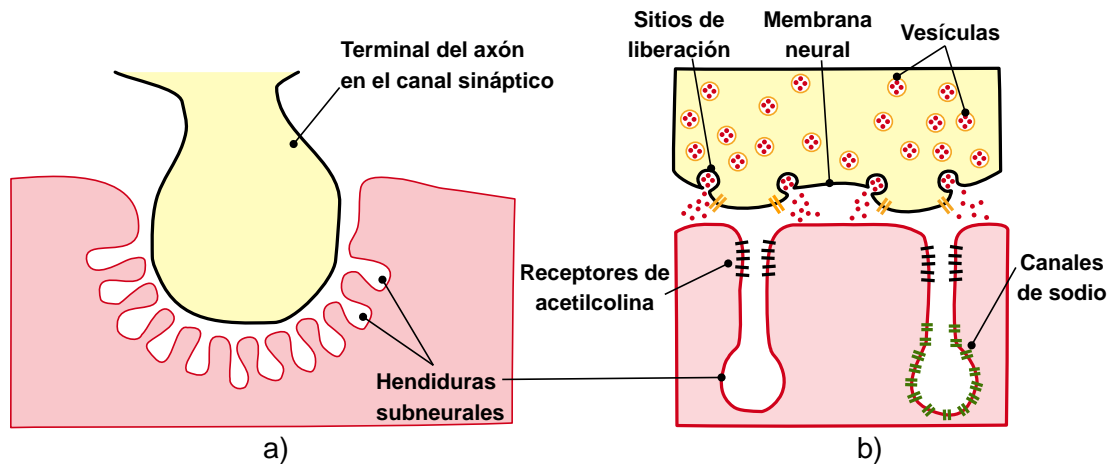


Figura 3.5: Placa terminal motora, a) punto de contacto entre una única terminal axónica y la membrana de la fibra muscular, b) liberación de acetilcolina desde las vesículas sinápticas en la membrana neural de la unión neuromuscular, cerca de los receptores en la membrana muscular, en las hendiduras subneuronales. Figura tomada y adaptada de [22].

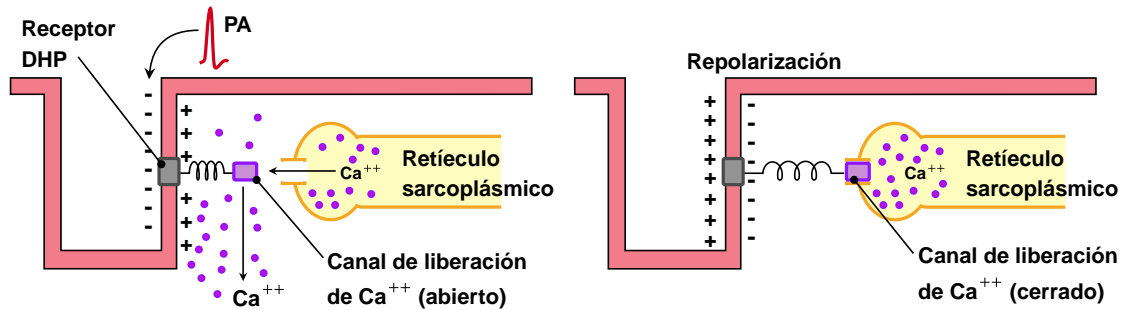


Figura 3.6: El acoplamiento excitación-contracción en el músculo esquelético involucra un potencial de acción (PA) que activa los receptores DHP, lo que abre los canales de  $Ca^{++}$  en el retículo sarcoplásmico, permitiendo la contracción muscular (izquierda). Durante la repolarización, los canales se cierran y el  $Ca^{++}$  es bombeado de vuelta al retículo sarcoplásmico (derecha). Figura tomada y adaptada de [22].

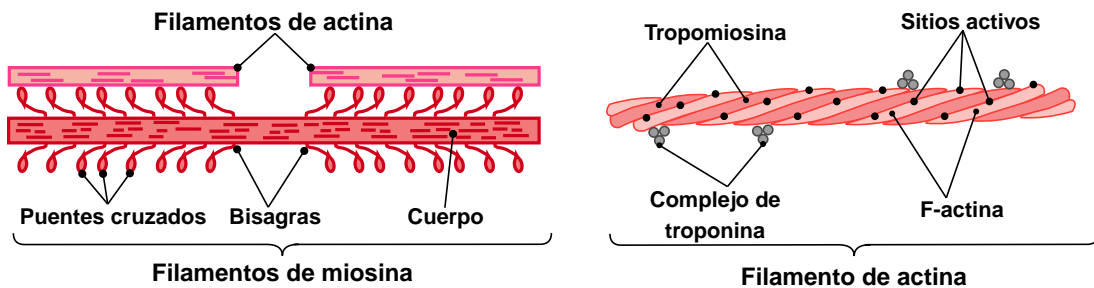


Figura 3.7: Filamento de miosina con sus cabezas que interactúan con los filamentos de actina a través de los puentes cruzados. Los filamentos de actina están formados por moléculas de F-actina y tropomiosina, y el complejo de troponina presente en la tropomiosina es el encargado de iniciar la contracción muscular. Figura tomada y adaptada de [22].

Este cambio en el voltaje es detectado por receptores de dihidropiridina (DHP) localizados en los túbulos T, que están conectados a canales de liberación de calcio en el retículo sarcoplásmico adyacente. La apertura de los canales DHP provoca la apertura de los canales de calcio, permitiendo que el calcio acumulado en el retículo sarcoplásmico salga hacia el citoplasma de la célula muscular, como se muestra en el panel izquierdo de la figura 3.6. El calcio es esencial para el proceso de contracción, como se explicará a continuación [22].

La contracción muscular ocurre gracias a un mecanismo de filamento deslizante, que se activa cuando interactúan los filamentos de actina, los filamentos de miosina y los iones de calcio liberados durante la etapa de acoplamiento excitación-contracción [39]. Los sitios activos de los filamentos de actina deben unirse a las cabezas de las moléculas de miosina, pero estos sitios están inhibidos por un complejo de troponina-tropomiosina (figura 3.7). Este efecto inhibitor se anula en presencia de iones de calcio, aunque el proceso exacto mediante el cual esto ocurre aún se desconoce. Una vez que los filamentos de actina son activados por los iones de calcio, los puentes cruzados del filamento de miosina se atraen hacia los sitios activos del filamento de actina, provocando así la contracción muscular. Aunque el mecanismo preciso de este fenómeno aún no se comprende completamente, existen algunas hipótesis que intentan explicarlo [22].

Una de estas hipótesis es la teoría de la caminata continua, la cual sostiene que, cuando las

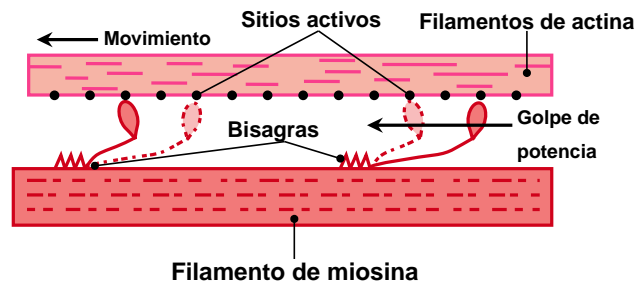


Figura 3.8: Mecanismo de la caminata para la contracción del músculo. Figura tomada y adaptada de [22].

cabezas del filamento de miosina se unen a los sitios activos del filamento de actina, estas cabezas se inclinan hacia el brazo (un movimiento conocido como golpe de potencia) y arrastran el filamento de actina con ellas. Luego, las cabezas se desconectan de los sitios activos a los que se habían unido, regresan a su posición extendida y se conectan a otro sitio activo, produciendo otro golpe de potencia que hace que el filamento de actina avance un paso más, como se muestra en la figura 3.8. En resumen, las cabezas de miosina se doblan hacia adelante y hacia atrás, “caminando” a lo largo del filamento de actina para moverlo hacia el centro del filamento de miosina [22].

### 3.1.3. Propiedades de las señales EMG

Las señales electromiográficas presentan diversas propiedades que dependen de factores fisiológicos y físicos. Una de sus principales características es la variabilidad de su amplitud, que oscila entre microvoltios ( $\mu V$ ) y un bajo rango de milivoltios (menos de 10 mV). De acuerdo con [63], la energía útil de la señal se encuentra en el rango de frecuencias de 0 a 500 Hz, con la mayor concentración de energía entre 50 y 150 Hz. La frecuencia de la señal está directamente relacionada con la velocidad de conducción del potencial de acción en las fibras musculares.

El registro de los cambios generados por la descarga de las fibras musculares de una unidad motora (UM) se conoce como potencial de acción de unidad motora (PAUM). En condiciones normales, la amplitud media de los PAUMs es de aproximadamente 0.5 mV y su duración varía entre 8 y 14 ms, dependiendo del tamaño de las unidades motoras [20].

La amplitud de la señal EMG generalmente aumenta con la fuerza de contracción muscular, aunque esta relación no es estrictamente lineal debido a factores como la sincronización de las unidades motoras y la fatiga muscular. Con el tiempo, la fatiga induce cambios en la señal, como una reducción en la frecuencia media y un aumento en la amplitud debido al reclutamiento de unidades motoras adicionales, justo como se indica en la figura 3.9. Además, factores externos como el desplazamiento de los electrodos, la postura del brazo y la variabilidad en la presión de los electrodos pueden afectar la calidad y estabilidad de la señal, lo que representa un desafío en aplicaciones de clasificación de movimientos [38].

### 3.1.4. Adquisición y electrodos

Las señales EMG se adquieren a través de electrodos que detectan la actividad eléctrica generada por los músculos. La elección y colocación de los electrodos son factores críticos para obtener señales de alta calidad.

Las señales electromiográficas de superficie se captan mediante electrodos bipolares adheridos a la piel. Desde 1948, han sido utilizadas en el control de prótesis de miembros superiores y son

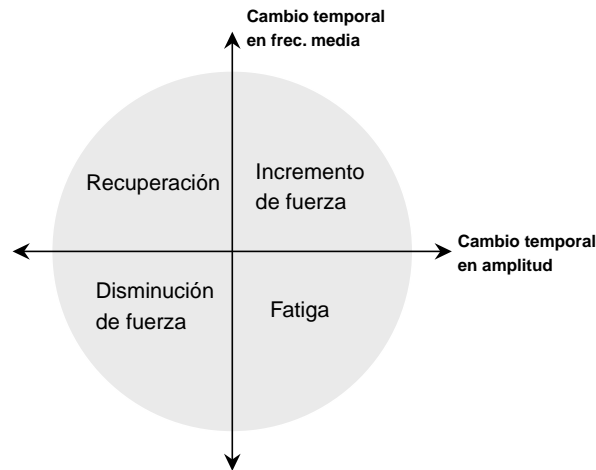


Figura 3.9: El aumento de la amplitud EMG y la frecuencia media indica mayor fuerza, mientras que el aumento de amplitud y frecuencia media baja sugiere fatiga. La disminución de la amplitud de la señal con frecuencia media baja indica menor fuerza, y con frecuencia media alta, recuperación. Diagrama tomado y adaptado de [38].

esenciales en aplicaciones de diagnóstico clínico, rehabilitación y control mioeléctrico de dispositivos activos [51].

Dado que la amplitud típica de las señales EMG varía entre 0 y 10 mV, se requiere un amplificador diferencial de alta ganancia (aproximadamente 5000 veces) para mejorar la señal sin distorsionarla. Posteriormente, se aplica un procesamiento de filtrado para eliminar:

- Ruido de alta frecuencia, generado por dispositivos electrónicos cercanos.
- Interferencia de la red eléctrica (60 Hz), que se atenúa con un filtro notch. Sin embargo, este filtrado puede afectar información útil de la señal, ya que el rango de frecuencias más relevantes para la EMG se encuentra entre 50 y 150 Hz. Tras la amplificación y filtrado, la señal se digitaliza mediante un convertidor analógico-digital (ADC) y se procesa con software especializado para su análisis en aplicaciones como clasificación de movimientos y control de prótesis.

La correcta ubicación de los electrodos es esencial para capturar señales representativas. Se recomienda la asistencia de personal médico especializado para identificar las regiones musculares óptimas. Actualmente, existen iniciativas como SENIAM (Surface ElectroMyoGraphy for the Non-Invasive Assessment of Muscles), que proponen estándares para el posicionamiento de electrodos y el procesamiento de señales EMG [24]. El número de electrodos también influye en la precisión y funcionalidad del sistema:

- Un mayor número de electrodos proporciona más información y mejora la precisión en la clasificación de movimientos.
- Un menor número de electrodos reduce el tiempo de cómputo en sistemas de clasificación y minimiza la interferencia entre canales (crosstalk), fenómeno en el que las señales de músculos adyacentes pueden mezclarse [51].

Los electrodos bipolares de superficie son los más utilizados debido a su facilidad de aplicación y comodidad para el usuario. Se pueden clasificar en dos tipos principales [1]:

1. **Electrodos húmedos (Ag/AgCl):** son los más comunes y proporcionan una mejor calidad de señal debido a su baja impedancia de contacto con la piel. Utilizan un gel conductor que mejora la conductividad y estabilidad de la señal. Su principal desventaja es que requieren reemplazo frecuente y pueden causar irritación en la piel tras un uso prolongado.
2. **Electrodos secos (acero inoxidable, titanio, aluminio, etc.):** no requieren gel conductor, lo que los hace más cómodos y reutilizables. Su impedancia de contacto es más alta al inicio, pero disminuye con el tiempo hasta estabilizarse después de aproximadamente 15 minutos. Son más adecuados para aplicaciones de uso prolongado, aunque pueden presentar mayor ruido en comparación con los electrodos húmedos. Su estabilidad en la impedancia tras cierto tiempo ha demostrado ser suficiente para aplicaciones como el control de prótesis, sin afectar significativamente la precisión de los movimientos.

## 3.2. Diseño factorial

Los diseños factoriales son ampliamente usados en experimentos que involucran múltiples factores donde es necesario estudiar el efecto conjunto que tienen los factores sobre una respuesta. Los más importantes corresponden a los diseños con  $k$  factores, los cuales pueden ser cuantitativos o cualitativos. A continuación se presenta una introducción a los diseños factoriales  $2^k$ . Si bien el enfoque principal será el diseño  $3^k$ , los diseños  $2^k$  son la base más simple y sus conceptos ayudan a comprender mejor el principio básico de los diseños factoriales.

### 3.2.1. Diseño factorial $2^k$

Para un diseño con  $k$  factores y cada factor con dos niveles, se necesitan  $2 \times 2 \times \dots \times 2 = 2^k$  observaciones para completar una réplica. Este tipo de diseño es llamado diseño factorial  $2^k$ . Este diseño permite estudiar  $k$  factores con el menor número de ejecuciones posibles en un diseño factorial completo. Es ampliamente utilizado en experimentos de cribado de factores para identificar factores activos de un grupo amplio.

En la teoría de este tipo de diseño se supone que los factores están fijos, el diseño es completamente aleatorio, y se satisfacen las suposiciones de normalidad. Se asume que los factores tienen solo dos niveles y, por tanto, que la respuesta es aproximadamente lineal en el rango de los niveles seleccionados. En muchas situaciones iniciales, esta suposición es razonable [44].

El diseño  $2^2$  incluye solo dos factores,  $A$  y  $B$ , cada uno con dos factores. Los niveles de los factores pueden ser llamados arbitrariamente “bajo” y “alto”, y se tienen cuatro combinaciones de tratamiento, como se muestra en la figura 3.10. Por convención, el efecto de un factor se denota con una letra mayúscula, de modo que  $A$  se refiere al efecto del factor  $A$ ,  $B$  al efecto del factor  $B$  y  $AB$  al efecto de la interacción. En este tipo de diseño los niveles bajo y alto son denotados por  $-$  y  $+$ , respectivamente, en los ejes de  $A$  y  $B$ . Por lo tanto,  $-$  en el eje  $A$  representa el nivel bajo de  $A$ . Las combinaciones de tratamientos y sus respectivas respuestas se representan por letras minúsculas, de modo que el nivel alto de un factor en el tratamiento se denota por la correspondiente letra minúscula, mientras que el nivel bajo se representa con la ausencia de esa letra. Por ejemplo  $a$  representa la combinación de  $A$  en el nivel alto y  $B$  en el nivel bajo. Por convención, (1) se usa para denotar el nivel bajo de los dos factores [44].

En un diseño factorial de dos niveles, el efecto promedio de un factor se puede definir como el cambio en la respuesta producido por un cambio en el nivel de ese factor, promediado sobre los niveles del otro factor [44]. Esto implica que el efecto de  $A$  en el nivel bajo de  $B$  sea  $[a - (1)]/n$ , y su efecto en el nivel alto de  $B$  es  $[ab - b]/n$ , donde  $n$  es el número de réplicas. El promedio de estas dos cantidades es el efecto principal de  $A$ :

$$A = \frac{1}{2n} \{[ab - b] + [a - (1)]\} = \frac{1}{2n} [ab + a - b - (1)] \quad (3.3)$$

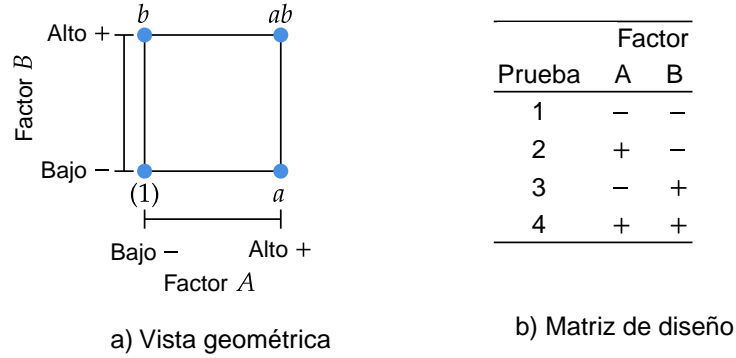


Figura 3.10: Combinaciones de tratamientos en un diseño  $2^2$ .

De la misma forma, el efecto principal de  $B$  se define como el promedio del efecto de  $B$  en el nivel bajo de  $A$  ( $[b - (1)]/n$ ) y el efecto de  $B$  en el nivel alto de  $A$  ( $[ab - a]/n$ ):

$$B = \frac{1}{2n} \{[ab - b] + [b - (1)]\} = \frac{1}{2n} [ab + b - a - (1)] \quad (3.4)$$

Por otra parte, la interacción se define como la diferencia promedio entre el efecto de  $A$  en el nivel alto de  $B$  y el efecto de  $A$  en el nivel bajo de  $B$ :

$$AB = \frac{1}{2n} \{[ab - b] - [a - (1)]\} = \frac{1}{2n} [ab + (1) - a - b] \quad (3.5)$$

Cabe señalar que es posible determinar los efectos mediante la suma del promedio de la respuesta en el nivel alto de un factor más el promedio de la respuesta en el nivel bajo de ese mismo factor [5]. Es importante considerar la magnitud y dirección (signo) de los efectos de factores para determinar cuales variables tienen mayor probabilidad de ser importantes en la respuesta. El análisis de varianza generalmente es usado para confirmar esta interpretación. Otra medida importante son las sumas de cuadrados, que se utilizan para evaluar si las diferencias observadas entre las medias de los niveles de factores son estadísticamente significativas. Estas sumas se calculan en términos de los contrastes, que a su vez son combinaciones lineales de medias de los niveles de factores usadas para probar hipótesis específicas sobre los efectos de los factores en el experimento. Por ejemplo, en la ecuación (3.3) se usa el contraste

$$\text{Contraste}_A = ab + a - b - (1) \quad (3.6)$$

La suma de cuadrados para cada contraste puede ser calculada fácilmente usando el cuadrado del contraste y el número total de observaciones [44]. Para el diseño  $2^2$  con  $n$  réplicas y factores  $A$  y  $B$ , las sumas de cuadrados son

$$SS_A = \frac{[ab + a - b - (1)]^2}{4n} \quad (3.7)$$

$$SS_B = \frac{[ab + b - a - (1)]^2}{4n} \quad (3.8)$$

$$SS_{AB} = \frac{[ab + (1) - a - b]^2}{4n} \quad (3.9)$$

para  $A$ ,  $B$  y  $AB$ , respectivamente. La suma de cuadrados total para este caso es

$$SS_T = \sum_{i=1}^2 \sum_{j=1}^2 \sum_{k=1}^n y_{ijk}^2 - \frac{y_{...}^2}{4n} \quad (3.10)$$

Efectos	(1)	<i>a</i>	<i>b</i>	<i>ab</i>
<i>A</i>	-1	+1	-1	+1
<i>B</i>	-1	-1	+1	+1
<i>AB</i>	+1	-1	-1	+1

Tabla 3.1: Coeficientes de contrastes.

Combinación de tratamiento	Efecto factorial			
	<i>I</i>	<i>A</i>	<i>B</i>	<i>AB</i>
(1)	+	-	-	+
<i>a</i>	+	+	-	-
<i>b</i>	+	-	+	-
<i>ab</i>	+	+	+	+

Tabla 3.2: Signos algebraicos para calcular los efectos en un diseño  $2^2$ .

donde  $y_{ijk}$  denota la observación en el  $i$ -ésimo nivel de  $A$ , el  $j$ -ésimo nivel de  $B$  y la  $k$ -ésima réplica. El término  $y_{...}/4n$  representa la media global [14]. La suma de cuadrados del error se calcula mediante una resta:

$$SS_E = SS_T - SS_A - SS_B - SS_{AB} \quad (3.11)$$

Usualmente se reportan las combinaciones de tratamientos en el orden estándar, esto es, (1),  $a$ ,  $b$  y  $ab$ . De este modo, los coeficientes de contrastes usados para calcular los efectos pueden ser escritos como se muestra en la tabla 3.1 [3] [44].

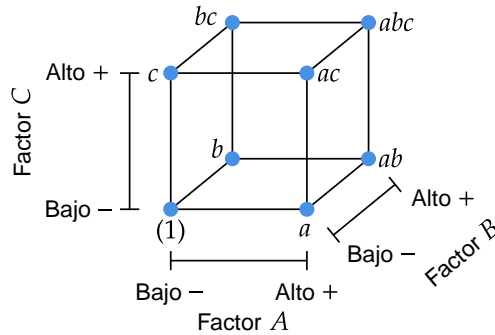
Cabe resaltar que los coeficientes para calcular el efecto de la interacción  $AB$  son el resultado del producto entre los coeficientes para calcular los efectos principales. Los coeficientes de contraste siempre valen  $+1$  o  $-1$ , es por eso que resulta conveniente usar tablas de signos como la tabla 3.2 para determinar el signo apropiado para cada combinación de tratamiento. Las columnas contienen a los efectos principales ( $A$  y  $B$ ), la interacción  $AB$  y el promedio global del experimento entero  $I$ , mientras que las columnas corresponden a las combinaciones de tratamientos. Los contrastes para estimar un efecto determinado se hallan al hacer el producto elemento a elemento entre la columna del efecto deseado y la columna de las combinaciones de tratamientos, por ejemplo, para estimar  $AB$ , el contraste es  $+(1) - a - b + ab$ , justo como en (3.5).

En los diseños factoriales se suele expresar la respuesta del experimento mediante un modelo de regresión, el cual es una aproximación natural e intuitiva. Para el diseño  $2^2$ , el modelo de regresión toma la forma

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \epsilon \quad (3.12)$$

donde  $x_1$  es una variable codificada que representa al factor  $A$ ,  $x_2$  es la variable codificada para  $B$ ,  $x_1 x_2$  corresponde a la interacción, y los coeficientes  $\beta$  son los coeficientes de regresión. Para factores en dos niveles, estos coeficientes se obtienen al dividir los efectos estimados por dos porque los coeficientes de regresión miden el efecto del cambio de una unidad en  $x$  sobre la respuesta  $y$ , y los efectos estimados están basados en cambios de dos unidades (de  $-1$  a  $+1$ ). El término  $\epsilon$  es el error aleatorio, el cual es aproximadamente normal e independientemente distribuido con media 0 y varianza constante  $\sigma^2$  [5] [44] [3].

El modelo de regresión también permite hacer predicciones u obtener el valor ajustado de  $y$  en los cuatro puntos del diseño. Una de las suposiciones claves en el diseño factorial es que los datos vienen de una distribución normal. Las gráficas de probabilidad normal de los residuos se emplean para evaluar la normalidad de un conjunto de datos; los residuos son las diferencias entre los valores observados de la respuesta y el valor ajustado [44]. Los residuos deben seguir una línea recta en la gráfica de probabilidad normal para poder afirmar que están normalmente distribuidos [5]. También es común complementar esta prueba con un gráfica de residuos contra los valores predichos, y se espera que los datos no sigan un patrón.



a) Vista geométrica

Prueba	Factor		
	A	B	C
1	-	-	-
2	+	-	-
3	-	+	-
4	+	+	-
5	-	-	+
6	+	-	+
7	-	+	+
8	+	+	+

b) Matriz de diseño

Figura 3.11: Combinaciones de tratamientos para un diseño  $2^3$ .

Otros gráficos que suelen ser útiles para el análisis del experimento son las superficies de respuesta y los gráficos de contorno para visualizar la dirección de mejora potencial, es decir, observar si la respuesta tiene tendencia a aumentar o disminuir al variar un factor.

Como se mencionó anteriormente, los contrastes y las sumas de cuadrados están relacionados. Las sumas de cuadrados en los diseños factoriales sirven para descomponer la variabilidad total de los datos y evaluar la importancia de los factores y sus interacciones en un experimento. Esto es crucial para comprender qué factores influyen en la variable de respuesta y en qué medida lo hacen, permitiendo tomar decisiones informadas basadas en el análisis estadístico. En general, para un diseño  $2^k$  los contrastes se pueden calcular al expandir el lado derecho de la siguiente expresión

$$\text{Contraste}_{AB\dots K} = (a \pm 1)(b \pm 1) \cdots (k \pm 1) \quad (3.13)$$

Para la expansión simplemente se desarrolla el producto de términos y se reemplaza el 1 por (1) en la expresión final. El signo se considera positivo cuando un factor se no incluye en el efecto, y es negativo cuando se incluye. Por ejemplo, en un diseño factorial  $2^3$  (ver figura 3.11), el contraste para  $AC$  sería

$$\begin{aligned} \text{Contraste}_{AC} &= (a - 1)(b + 1)(c - 1) \\ &= abc + ac + b + (1) - ab - bc - a - c \end{aligned}$$

Con los contrastes hallados, es fácil calcular los efectos y las sumas de cuadrados mediante las ecuaciones (3.14) y (3.15), respectivamente

$$AB \cdots K = \frac{1}{n2^{k-1}} (\text{Contraste}_{AB\dots K}) \quad (3.14)$$

$$SS_{AB\dots K} = \frac{1}{n2^k} (\text{Contraste}_{AB\dots K})^2 \quad (3.15)$$

### 3.2.2. Diseño factorial $3^k$

En este arreglo factorial, cada uno de los  $k$  factores tiene tres niveles. Los efectos y las interacciones se siguen representando con letras mayúsculas, y los niveles pueden denominarse arbitrariamente como “bajo”, “intermedio” y “alto”. Para denotar estos niveles también se emplean los dígitos 0 (bajo), 1 (intermedio) y 2 (alto), de modo que cada combinación de tratamiento es representada

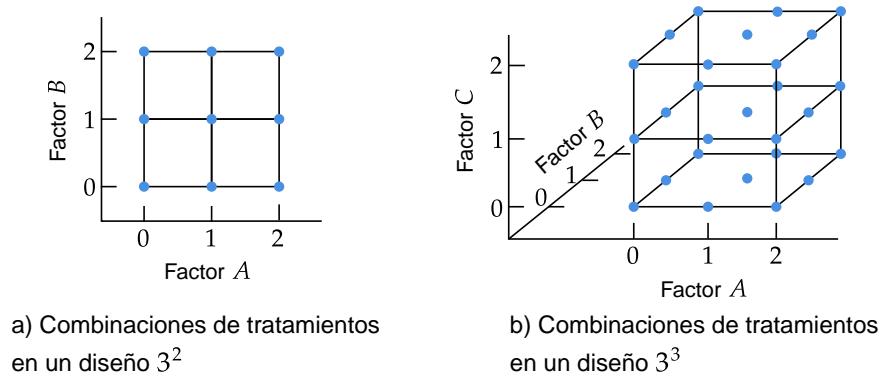


Figura 3.12: Diseños factoriales de 3 niveles.

		Factor B		
		0	1	2
Factor A	0	-2 <sub>Q</sub>	-2 <sub>R</sub>	-6 <sub>S</sub>
	1	3 <sub>R</sub>	5 <sub>S</sub>	10 <sub>Q</sub>
	2	-1 <sub>S</sub>	11 <sub>Q</sub>	-2 <sub>R</sub>

(a)

		Factor B		
		0	1	2
Factor A	0	-2 <sub>Q</sub>	-2 <sub>R</sub>	-6 <sub>S</sub>
	1	3 <sub>S</sub>	5 <sub>Q</sub>	10 <sub>R</sub>
	2	-1 <sub>R</sub>	11 <sub>S</sub>	-2 <sub>Q</sub>

(b)

Figura 3.13: Ejemplo de cuadrados latinos ortogonales donde sus letras son  $Q$ ,  $R$ , y  $S$ . Cada celda contiene al total de una combinación de tratamiento.

por  $k$  dígitos, donde el  $k$ -ésimo número indica el nivel del  $k$ -ésimo factor. Por ejemplo, en un diseño  $3^3$ , 012 denota la combinación de tratamiento donde  $A$  se encuentra en el nivel bajo,  $B$  se halla en el nivel intermedio y  $C$  en el nivel alto. Este tipo de arreglo está pensado para tratar con la curvatura de la función de respuesta [44].

El caso más simple de un diseño con sus factores en tres niveles corresponde al de dos factores, es decir, al diseño  $3^2$ . Este arreglo consiste de  $3^2 = 9$  combinaciones de tratamientos, justo como se ilustra en la figura 3.12a. Cada efecto principal tiene asociada una parte lineal y una parte cuadrática. Por otro lado, la interacción  $AB$  puede ser dividida de dos formas. En el primer método se considera que todos los factores son cuantitativos y se divide a la interacción en sus cuatro componentes:  $AB_{L \times L}$ ,  $AB_{L \times Q}$ ,  $AB_{Q \times L}$  y  $AB_{Q \times Q}$ ; esto se logra al ajustar un modelo que incluya los términos  $\beta_{12}x_1x_2$ ,  $\beta_{122}x_1x_2^2$ ,  $\beta_{112}x_1^2x_2$  y  $\beta_{1122}x_1^2x_2^2$ . La suma de cuadrados de la interacción es dada por la suma total de las sumas de cuadrados de sus componentes [44].

El segundo método está basado en los cuadrados latinos ortogonales, que generalmente se asocia al caso de factores cualitativos. Los totales para cada combinación de de tratamiento se colocan en forma de cuadrícula, donde las filas corresponden al factor  $A$  y las columnas al factor  $B$ , justo como en la figura 3.13. Se dice que dos cuadrados latinos son ortogonales cuando, al superponer uno sobre el otro, cada letra en el primer cuadrado aparecerá exactamente una vez con cada letra del otro cuadrado. Luego, los totales para cada letra en el primer cuadrado se suman y se realiza la suma de cuadrados; este proceso se repite para el segundo cuadrado, y la suma de estas dos componentes da como resultado  $SS_{AB}$  [44] [46].

Cuadrado (a)	Cuadrado (b)
$Q : x_1 + x_2 = 0 \pmod{3}$	$Q : x_1 + 2x_2 = 0 \pmod{3}$
$R : x_1 + x_2 = 1 \pmod{3}$	$R : x_1 + 2x_2 = 1 \pmod{3}$
$S : x_1 + x_2 = 2 \pmod{3}$	$S : x_1 + 2x_2 = 2 \pmod{3}$

Tabla 3.3: Cada letra ocupa una celda específica de acuerdo a una regla.

La suma de cuadrados calculada del cuadrado (a) es llamada componente de la interacción  $AB$ , y la del cuadrado (b) se conoce como componente de la interacción  $AB^2$ . Esta terminología es usada porque, si denotamos a los niveles de  $A$  y  $B$  por  $x_1$  y  $x_2$ , respectivamente, se halla que las letras ocupan las celdas en un patrón específico:

En esta tabla, los niveles (0,1,2) son representados por  $x$ , y  $x_1 + x_2 = 0 \pmod{3}$  significa que los valores que se consideran para la suma de  $x_1$  y  $x_2$  son aquellos que, al dividirse por 3, dejan un residuo de 0. Por ejemplo, si se consideran las celdas donde aparece  $Q$  en el cuadrado (a) de la figura 3.13 se observa que

$$\begin{aligned} x_1 + x_2 = 0 + 0 = 0 &\implies 0/3 = 0 \text{ residuo } 0 \\ x_1 + x_2 = 1 + 2 = 3 &\implies 3/3 = 1 \text{ residuo } 0 \\ x_1 + x_2 = 2 + 1 = 3 &\implies 3/3 = 1 \text{ residuo } 0 \end{aligned}$$

Cabe mencionar que en las expresiones de la forma  $A^p B^q$  el exponente del primer factor siempre se establece como 1 por convención [44], si no fuera así entonces toda la expresión se eleva al cuadrado y los exponentes se reducen mediante la aritmética modular con módulo 3, es decir, los exponentes resultantes se dividen entre 3 y se toma el residuo como nuevo exponente. Por ejemplo,  $A^2 B$  se reduce a  $AB^2$ :

$$(A^2 B) = (A^2 B)^2 = A^4 B^2 = AB^2$$

Ya que la interacción  $AB$  se forma a partir de  $AB$  y  $AB^2$ , generalmente no se muestran estas componentes en el análisis de varianza. Además, no hay relación alguna entre estos dos elementos y las sumas de cuadrados para  $AB_{L \times L}$ ,  $AB_{L \times Q}$ ,  $AB_{Q \times L}$  y  $AB_{Q \times Q}$ .

Ahora, supongamos que hay tres factores ( $A$ ,  $B$ , y  $C$ ) bajo estudio, y cada uno de ellos tiene tres niveles arreglados en un diseño factorial (figura 3.12b). Este arreglo se conoce como diseño  $3^3$ , las 27 combinaciones de tratamientos resultantes tienen 26 grados de libertad. Cada efecto principal tiene dos grados de libertad, las interacciones de dos factores poseen cuatro grados de libertad (producto de los dos grados de cada factor), y la interacción de los tres factores tienen ocho grados de libertad.

Al igual que en el diseño  $3^2$ , si los factores son cuantitativos es posible descomponer los efectos principales en sus partes lineal y cuadrática, mientras que las interacciones de dos factores se dividen en efectos lineal  $\times$  lineal, lineal  $\times$  cuadrático, cuadrático  $\times$  lineal y cuadrático  $\times$  cuadrático. La interacción de tres factores también se divide de ese modo, aunque esta descomposición no es útil en la práctica. Por otro lado, también es posible partir las interacciones de dos factores en sus componentes  $AB$ ,  $AB^2$ ,  $BC$ ,  $BC^2$ ,  $AC$  y  $AC^2$ , y como en el caso  $3^2$ , estas componentes no tienen significado físico. Por último, la interacción de tres factores  $ABC$  también se divide en cuatro componentes con dos grados de libertad cada una:  $AB^2 C^2$ ,  $AB^2 C$ ,  $ABC^2$  y  $ABC$ . Nuevamente, la primera letra solo puede tener exponente igual a uno [44]. Las sumas de cuadrados para los efectos

principales y las interacciones son las siguientes [46]:

$$SS_A = \sum_{i=0}^2 \frac{y_{i..}^2}{3^2 n} - \frac{y_{...}^2}{3^3 n} \quad (3.16)$$

$$SS_B = \sum_{j=0}^2 \frac{y_{.j.}^2}{3^2 n} - \frac{y_{...}^2}{3^3 n} \quad (3.17)$$

$$SS_C = \sum_{k=0}^2 \frac{y_{...k}^2}{3^2 n} - \frac{y_{...}^2}{3^3 n} \quad (3.18)$$

$$SS_{AB} = \sum_{i=0}^2 \sum_{j=0}^2 \frac{y_{ij.}^2}{3n} - \frac{y_{...}^2}{3^3 n} - SS_A - SS_B \quad (3.19)$$

$$SS_{AC} = \sum_{i=0}^2 \sum_{k=0}^2 \frac{y_{i.k.}^2}{3n} - \frac{y_{...}^2}{3^3 n} - SS_A - SS_C \quad (3.20)$$

$$SS_{BC} = \sum_{j=0}^2 \sum_{k=0}^2 \frac{y_{.jk.}^2}{3n} - \frac{y_{...}^2}{3^3 n} - SS_B - SS_C \quad (3.21)$$

$$SS_{ABC} = \sum_{i=0}^2 \sum_{j=0}^2 \sum_{k=0}^2 \frac{y_{ijk.}^2}{n} - \frac{y_{...}^2}{3^3 n} - SS_A - SS_B - SS_C \quad (3.22)$$

$$- SS_{AB} - SS_{AC} - SS_{BC} \quad (3.23)$$

$$SS_T = \sum_{i=0}^2 \sum_{j=0}^2 \sum_{k=0}^2 \sum_{l=0}^n y_{ijkl} - \frac{y_{...}^2}{3^3 n} \quad (3.24)$$

$$SS_E = SS_T - SS_A - SS_B - SS_C - SS_{AB} - SS_{AC} - SS_{BC} - SS_{ABC} \quad (3.25)$$

La notación del subíndice “punto” indica una suma sobre el subíndice que reemplaza. Los conceptos de los diseños  $3^2$  y  $3^3$  se extienden a diseños con  $k$  factores, cada uno con tres niveles, de modo que se genera un diseño  $3^k$ . Las combinaciones de tratamientos también son representadas en notación digital, y se pueden calcular sumas de cuadrados para los  $k$  efectos principales y sus interacciones. El tamaño del diseño crece rápidamente con  $k$ : por ejemplo, un diseño  $3^3$  tiene 27 combinaciones y un diseño  $3^4$  tiene 81. Generalmente, se considera una sola réplica debido al gran tamaño de los diseños, lo que lleva a agrupar interacciones de orden superior para estimar el error. Cuando el número de combinaciones se vuelve inmanejable se puede seleccionar un subconjunto de puntos mediante criterios de optimalidad como se verá a continuación.

### 3.2.3. Diseño D-óptimo

Como se ha visto a lo largo de esta sección, el número de combinaciones aumenta exponencialmente al añadir más factores. Para un diseño con  $k$  factores, cada uno con  $L$  niveles, se tendrán  $L^k$  combinaciones. Por ejemplo, si se establecen 7 factores con 3 niveles cada uno, el número total de combinaciones es  $3^7 = 2187$ , lo que resulta en una cantidad inmanejable de experimentos. Sin embargo, a partir de un diseño factorial completo, se puede generar un diseño óptimo. En este enfoque, se selecciona un subconjunto de combinaciones de tratamientos basándose en criterios de probabilidad y estadística.

Entre los métodos de diseño óptimo, el diseño D-óptimo se destaca por su capacidad para maximizar la información obtenida del experimento, minimizando al mismo tiempo el número total de combinaciones necesarias. Este tipo de diseño permite realizar una estimación más eficiente de los parámetros del modelo, lo que resulta especialmente valioso cuando se trabaja con un número elevado de factores y niveles.

Un diseño D-óptimo  $\xi^*$  busca maximizar el determinante de la matriz de información de Fisher,  $|M(\xi)|$ . Este determinante está relacionado inversamente con el volumen del elipsoide de confianza  $p$ -dimensional sobre los parámetros del modelo asociado al diseño, un diseño D-óptimo minimiza ese volumen para un número dado de ejecuciones  $n$ , por lo tanto, equivalentemente se busca minimizar el determinante de  $|M^{-1}(\xi)|$  [15]. La matriz de información en este contexto se define como

$$M(\xi) = X^T X \quad (3.26)$$

donde  $X$  es la matriz de diseño. Para ilustrar esto vamos a considerar un diseño  $2^2$ , que tiene combinaciones de tratamientos: (1),  $a$ ,  $b$ , y  $ab$ ; el modelo para este diseño es

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \epsilon$$

con  $x_1$  y  $x_2$  como los efectos principales de los factores en la escala  $\pm 1$ , y  $x_1 x_2$  como la interacción. Cada combinación de tratamiento puede escribirse en términos del modelo:

$$\begin{aligned} (1) &= \beta_0 + \beta_1(-1) + \beta_2(-1) + \beta_{12}(-1)(-1) + \epsilon_1 \\ a &= \beta_0 + \beta_1(1) + \beta_2(-1) + \beta_{12}(1)(-1) + \epsilon_2 \\ b &= \beta_0 + \beta_1(-1) + \beta_2(1) + \beta_{12}(-1)(1) + \epsilon_3 \\ ab &= \beta_0 + \beta_1(1) + \beta_2(1) + \beta_{12}(1)(1) + \epsilon_4 \end{aligned}$$

y esto se puede representar en forma matricial:

$$\vec{y} = X\vec{\beta} + \vec{\epsilon}$$

donde

$$\vec{y} = \begin{bmatrix} (1) \\ a \\ b \\ ab \end{bmatrix} \quad X = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad \vec{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_{12} \end{bmatrix} \quad \vec{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{bmatrix}$$

La matriz de diseño para este caso es  $X$ . Las estimaciones de mínimos cuadrados de los parámetros del modelo son los valores de  $\beta$  que minimizan la suma de los cuadrados de los errores del modelo. Esas estimaciones son dadas por [44]

$$\hat{\beta} = (X^T X)^{-1} X^T \vec{y}.$$

Para este ejemplo,

$$X^T X = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

La matriz resultante es diagonal porque el diseño  $2^2$  es ortogonal, y su determinante es

$$|(X^T X)| = 256$$

El diseño D-óptimo maximiza el determinante de  $X^T X$  para minimizar la varianza de los coeficientes de regresión en un modelo. Para un diseño de cuatro corridas en un espacio de diseño limitado a  $\pm 1$ , se elige un diseño  $2^2$  que maximiza este determinante, lo que reduce el volumen de la región de confianza conjunta de los coeficientes del modelo [44].

La eficiencia D de un diseño arbitrario  $\xi$  en comparación con el diseño óptimo  $\xi^*$  se define como

$$D_{\text{ef}} = \left\{ \frac{|M(\xi)|}{|M(\xi^*)|} \right\}^{1/p} \quad (3.27)$$

donde  $p$  es el número de parámetros que se necesitan para ajustar el modelo. La comparación de las matrices de información permite medir la eficiencia de un diseño experimental sin considerar el número de observaciones. Tomar la raíz  $p$ -ésima de la razón de los determinantes da una medida de eficiencia que tiene las dimensiones de una varianza, independientemente de la dimensión del modelo [15] [4]. Por ejemplo, dos réplicas de un diseño para el cual se obtiene  $D_{\text{ef}}$  serían igual de eficientes que una sola réplica del diseño óptimo. Por otra parte, para comparar dos diseños,  $\xi_1$  y  $\xi_2$ , se define la eficiencia D relativa:

$$D_{\text{ef-rel}} = \left\{ \frac{|M(\xi_1)|}{|M(\xi_2)|} \right\}^{1/p} \quad (3.28)$$

A diferencia de la eficiencia D, la eficiencia D relativa puede tomar valores mayores a uno, en cuyo caso se dice que el diseño  $\xi_1$  es mejor que  $\xi_2$  respecto al criterio del determinante.

Otra propiedad de la eficiencia D de un diseño es que es invariante a transformaciones lineales no degeneradas del modelo. Por lo tanto, un diseño D-óptimo para el modelo  $\eta = \beta^T f(x)$  también es un D-óptimo para el modelo  $\eta = \gamma^T g(x)$ , si  $g(x) = Af(x)$  y  $|A| \neq 0$ . Aquí,  $\beta$  y  $\gamma$  son vectores  $p$ -dimensionales que contienen a los  $p$  parámetros del modelo, mientras que  $f(x)$  es el vector de expansión de los factores [15] [4], por ejemplo, para un diseño  $2^3$  el modelo será

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{123} x_1 x_2 x_3$$

de modo que las componentes del vector de expansión  $f(x)$  son 1,  $x_1$ ,  $x_2$ ,  $x_3$ ,  $x_1 x_2$ ,  $x_1 x_3$ ,  $x_2 x_3$  y  $x_1 x_2 x_3$ .

Para construir un diseño D-óptimo se usa un método secuencial. Si se tienen  $N$  puntos en el diseño del experimento entonces el determinante de la matriz de información asociada será

$$|M(N)| = |X^T X|$$

Al añadir un nuevo punto, el determinante será

$$|M(N+1)| = |X^T X + f(x)f^T(x)|$$

y esto puede reescribirse como una actualización multiplicativa del determinante  $|M(N)|$  [4]:

$$|M(N+1)| = |X^T X| \left\{ 1 + f(x)(X^T X)^{-1} f^T(x) \right\} = |M(N)| \left\{ 1 + \frac{d(x, \xi_N)}{N} \right\} \quad (3.29)$$

donde  $d(x, \xi_N)$  es la función de varianza en  $x$  dado el diseño  $\xi_N$  [40] [4], y se define como

$$d(x_i, \xi) = f^T(x_i) M^{-1}(\xi) f(x_i). \quad (3.30)$$

Aquí, la  $N$  tiene el propósito de normalizar la varianza del nuevo punto respecto al tamaño del diseño actual. Por lo tanto, los nuevos puntos que se añaden al diseño deben ser tales que  $d(x_i, \xi_N)$  sea máximo para obtener el mayor incremento posible en el determinante de la matriz de información  $|M(N)|$ .

### 3.3. Redes Neuronales de Convolución

Las redes neuronales convolucionales (RNC) son un tipo especializado de red neuronal profunda diseñado para procesar datos con una estructura de grilla, como imágenes y señales espacio-temporales. Su arquitectura está inspirada en la organización del córtex visual en el cerebro humano, lo que les permite extraer automáticamente características relevantes de los datos mediante operaciones de convolución. A diferencia de las redes densas tradicionales, las RNCs aprovechan propiedades como la localización espacial y la compartición de pesos para reducir la complejidad

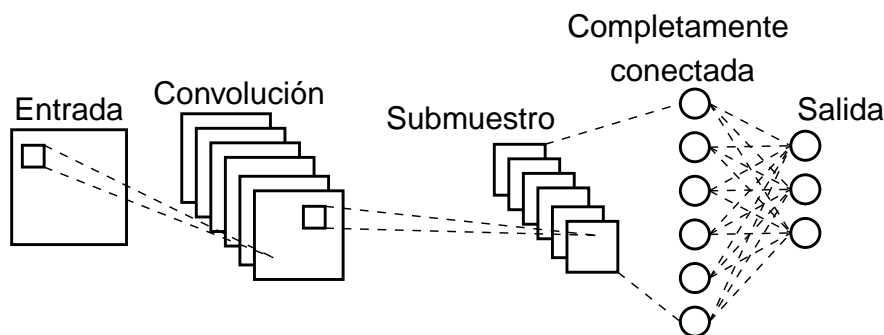


Figura 3.14: Componentes principales de una red neuronal de convolución.

computacional y mejorar su capacidad de generalización. Estas características las han convertido en una herramienta clave para tareas como la clasificación, segmentación y análisis de señales complejas.

Las RNC son construcciones matemáticas que generalmente se componen de tres tipos de capas o bloques: capas de convulsión, de submuestreo, y completamente conectadas. Las primeras dos capas se encargan de la extracción de características, mientras que la capa completamente conectada es la responsable del mapeo de las características extraídas hacia la salida final. La convulsión es una operación que juega un papel clave en la RNC. Los datos que entran a la red se ordenan en un arreglo bidimensional, y una pequeña grilla de parámetros llamada filtro o *kernel* se desplaza por toda el mapa de entrada para la extracción de características. Dado que la salida de una capa sirve como entrada para la siguiente capa, las características extraídas pueden volverse más complejas de forma progresiva y jerárquica, es decir, una primera capa de convulsión aprenderá pequeños patrones locales como bordes, una segunda capa de convulsión aprenderá patrones más complejos que se componen de las características de las primeras capas, y eso continúa de este modo [67].

El proceso de optimizar los parámetros del filtro es llamado entrenamiento, y se realiza con la finalidad de minimizar la diferencia entre la salida final de la RNC y el objetivo verdadero; esto se logra a través de un algoritmo de optimización llamado propagación hacia atrás y gradiente descendente, entre otros.

### 3.3.1. Componentes

Una arquitectura típica de RNC consiste en repeticiones de un grupo de múltiples capas de convulsión y una capa de submuestreo, seguidas por una o más capas completamente conectadas, como se muestra en la figura 3.14. Al proceso mediante el cual los datos de entrada son transformados en una salida a través de las capas se le conoce como propagación hacia adelante.

#### Capa de convulsión

Una capa de convulsión es un componente esencial en las arquitecturas de redes neuronales convolucionales (RNC), encargado de extraer características relevantes de los datos. Estas capas combinan operaciones lineales, como la convulsión, con funciones de activación no lineales para capturar patrones complejos.

La convulsión es una operación lineal utilizada para la extracción de características. Consiste en aplicar un pequeño arreglo de números, llamado *kernel*, sobre la entrada, representada como un tensor. Primero se calcula el producto elemento a elemento entre cada uno de los números del *kernel* y el tensor de entrada en cada posición del tensor, luego se suman para obtener el valor de salida en la posición correspondiente al tensor de salida (ver figura 3.15), llamado mapa de características. Este procedimiento se repite aplicando múltiples filtros para formar un número

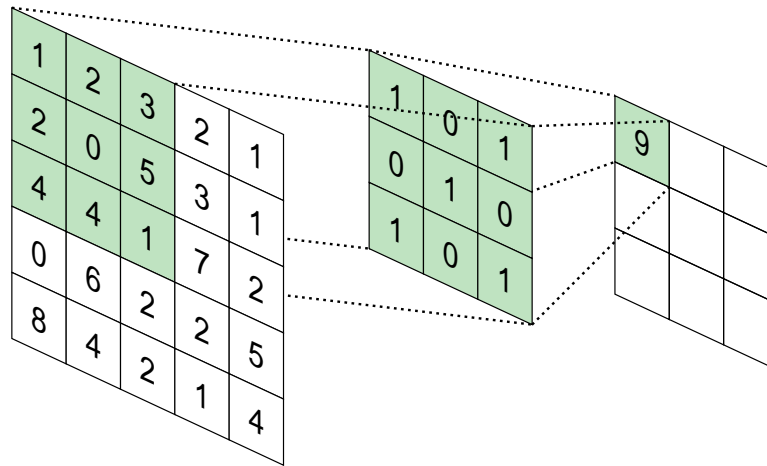


Figura 3.15: Operación de convolución.

arbitrario de mapas de características, por lo que diferentes kernels pueden considerarse diferentes extractores de características [67].

La operación de convolución que se ha descrito no permite que el centro de cada kernel se ubique en los elementos más exteriores del tensor de entrada, por lo las dimensiones (alto y ancho) del mapa de características se ven reducidas en comparación con las del tensor de entrada. Para abordar esta cuestión se usa la técnica de relleno (padding), típicamente el relleno con ceros, que consiste en agregar filas y columnas de ceros a cada lado del tensor de entrada para que el centro de cada filtro pueda ubicarse en los elementos más externos de dicho tensor y mantener su tamaño original. La ventaja de mantener el tamaño radica en la posibilidad de agregar más capas, ya que de lo contrario cada mapa de características se vería reducido sucesivamente [16].

Otro parámetro que define la convolución es el paso (stride), que se refiere a la distancia entre dos posiciones sucesivas de un kernel. El valor por defecto del paso es de uno, sin embargo un valor mayor puede ser usado para reducir las dimensiones de los mapas de características. La convolución con un paso mayor a uno generalmente no se usa en modelos de clasificación [16], en lugar de eso se emplean operaciones de submuestreo como *max-pooling*.

Como ya se mencionó anteriormente, la compartición de pesos es una propiedad importante de las RNC, ya que permiten que los patrones aprendidos sean invariantes bajo traslaciones porque los kernels se desplazan en todas las posiciones de la imagen y detectan los patrones locales. Además, aprenden las jerarquías espaciales de los patrones al reducir las dimensiones de los mapas de características en conjunto con operaciones de submuestreo. Cabe mencionar que la compartición de pesos incrementa la eficiencia del modelo al reducir el número de parámetros a aprender en comparación con las redes neuronales completamente conectadas [67].

Durante el entrenamiento se identifican los valores de los kernels que son más eficientes para una tarea en específico dado un conjunto de datos. La salida de una operación lineal como la convolución pasan entonces a través de una función de activación, de la cual se hablará más adelante.

### Capa de submuestreo

Una capa de submuestreo realiza una operación de reducción de dimensionalidad en los mapas de características, introduciendo invariancia a pequeñas traslaciones y distorsiones, y disminuyendo el número de parámetros que se deben aprender. No contiene parámetros a entrenar, pero el tamaño del filtro, el paso y el relleno son hiperparámetros similares a los de las convoluciones.

Una de las operaciones de submuestreo más populares es *max-pooling*, en la cual se extraen parches de los mapas de características de entrada, y la salida corresponde al valor máximo de

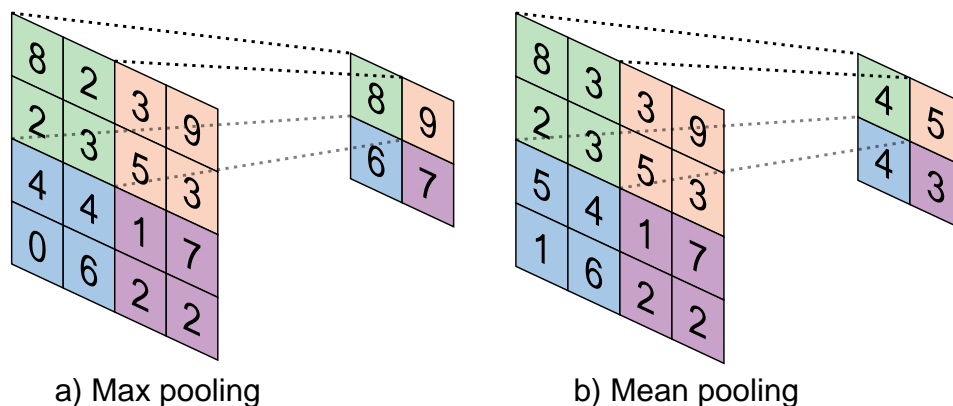


Figura 3.16: Operaciones de submuestreo.

cada parche, descartando el resto de valores (figura 3.16a). En la práctica es común usar max-pooling con un tamaño de  $2 \times 2$  y un paso de 2, de modo que el largo y ancho de la imagen se reducen a la mitad, mientras que el número de mapas de entrada permanece sin cambios [67].

Otra forma de operación es *mean-pooling*, donde también se extraen parches de los mapas de características de entrada pero, a diferencia de max-pooling, la salida corresponde al valor promedio de cada parche (figura 3.16b).

### Capa completamente conectada

Usualmente, los mapas de características de la última capa de convolución o submuestreo son aplanados, es decir, transformados en arreglos de números unidimensionales o vectores que son conectados a una o más capas completamente conectadas (también conocidas como capas densas) donde cada entrada está conectada a cada salida mediante un conjunto de pesos. Las características extraídas son entonces mapeadas por estas capas completamente conectadas hacia la salida final de la red, tales como la probabilidad de pertenencia a una clase en el caso de tareas de clasificación. La última capa densa tiene tantos nodos de salida como clases, y cada capa es seguida por una función de activación no lineal [67] [2].

Cabe destacar que la función de activación de la última capa completamente conectada usualmente es diferente del resto, y se debe elegir de acuerdo a la tarea que va a realizar el modelo. Por ejemplo, una función de activación aplicada a la tarea de clasificación con múltiples clases es la función softmax, la cual normaliza los valores de salida de la última capa densa a probabilidades de la clase objetivo, donde cada valor va de 0 a 1 y la suma total de ellos resulta en la unidad.

### 3.3.2. Entrenamiento de una RNC

El entrenamiento de una red neuronal de convolución es un proceso para encontrar los valores de los parámetros en las capas de convolución y en las capas densas que minimicen la diferencia entre las predicciones de salida y las etiquetas de verdad fundamental dadas para un conjunto de datos. Para llevar a cabo esta tarea se emplea el algoritmo de propagación hacia atrás en conjunto con la función de costo y un método de optimización, que se describen en los siguientes apartados. La eficiencia de un modelo que opera con determinados pesos se mide a través de la función de costo que se calcula con la propagación hacia adelante, y luego esos parámetros son actualizados usando el valor de costo con el método de propagación hacia atrás y un algoritmo de búsqueda como el gradiente descendente, entre otros [67] [33].

El método de propagación hacia atrás consiste en actualizar los pesos o parámetros usando el gradiente, e involucra propagar el error de la capa de salida hacia atrás, a través de todas las capas

de la red para actualizar los correspondientes pesos. Esto se logra con la regla de la cadena de cálculo para hallar la derivada de la función de costo con respecto a cada parámetro del modelo. La función de costo, también conocida como función de pérdida, mide la diferencia entre las predicciones de salida de la red y las etiquetas de verdad fundamental. A través del entrenamiento del modelo se busca optimizar la función de costo. Matemáticamente, la propagación hacia atrás puede ser representada como sigue [33]:

1. **Propagación hacia adelante:** se calcula la salida de la red para un entrada  $x$  a través de las operaciones de las capas.
2. **Cálculo del error:** se evalúa la función de costo  $C(y, \hat{y})$  donde  $y$  es la etiqueta verdadera y  $\hat{y}$  es la salida predicha.
3. **Propagación hacia atrás:** se calcula el gradiente de  $C$  con respecto a los pesos usando la regla de la cadena:

$$\frac{\partial C}{\partial w_{i,j}^{(l)}} = \frac{\partial C}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial w_{i,j}^{(l)}} \quad (3.31)$$

donde  $z_i^{(l)}$  es la suma ponderada de las entradas a la neurona  $i$  en la capa  $l$ , y  $w_{i,j}^{(l)}$  es el peso que conecta a la neurona  $i$  en la capa  $l - 1$  con la neurona  $j$  en la capa  $l$

4. **Actualización de parámetros:** los pesos se ajustan usando un optimizador como gradiente descendente (3.32) o Adam:

$$w_{i,j}^{(l)} = w_{i,j}^{(l)} - \eta \cdot \frac{\partial C}{\partial w_{i,j}^{(l)}} \quad (3.32)$$

donde  $\eta$  es la tasa de aprendizaje, y es un hiperparámetro que controla el tamaño de las actualizaciones aplicadas a los pesos.

5. Los pasos se repiten para todas las muestras del conjunto de datos usado para el entrenamiento.
6. Los pasos 1-5 se repiten para un determinado número de iteraciones o hasta que se alcance la convergencia.

Para calcular los gradientes primero se necesitan definir las siguientes variables [45]: la suma ponderada de las entradas a la neurona  $i$  en la capa  $l$  se define como

$$z_i^{(l)} = \sum_j w_{i,j}^{(l)} a_j^{(l-1)} + b_i^{(l)} = \sum_j w_{i,j}^{(l)} \sigma(z_j^{(l-1)}) + b_i^{(l)} \quad (3.33)$$

Donde  $a$  representa a la función de activación y, considerando esta notación, la activación  $a_i^{(l)}$  de la neurona  $i$  en la capa  $l$  está relacionada a las activaciones en la capa  $l - 1$  por la ecuación

$$a_i^{(l)} = \sigma \left( \sum_j w_{i,j}^{(l)} a_j^{(l-1)} + b_i^{(l)} \right) \quad (3.34)$$

Ahora bien, una ecuación para el error  $\delta^{(l)}$  en términos del error en la siguiente capa,  $\delta^{(l+1)}$ , es

$$\delta^{(l)} = [(w^{(l+1)})^T \delta^{(l+1)}] \cdot \sigma'(z^{(l)}), \quad (3.35)$$

mientras que para el error,  $\delta^L$ , de la capa de salida  $L$  se tiene la ecuación (3.36) que es una expresión componente por componente

$$\delta_i^L = \frac{\partial C}{\partial a_i^L} \sigma'(z_i^L) \quad (3.36)$$

Esta ecuación puede ser reescrita en su forma matricial:

$$\delta^L = \nabla_a C \cdot \sigma'(z^L). \quad (3.37)$$

También es necesario calcular la tasa de cambio del costo con respecto a cada uno de los sesgos (b) de la red:

$$\frac{\partial C}{\partial b_i^l} = \delta_i^l \quad (3.38)$$

Esta ecuación indica que el error  $\delta_i^l$  es igual a la tasa de cambio del costo con respecto al sesgo  $b_i^l$ , y dicho error puede ser calculado usando las ecuaciones (3.35) y (3.36).

Considerando la notación y usando la regla de la cadena, la ecuación (3.31) puede ser reescrita en términos del error:

$$\frac{\partial C}{\partial w_{ij}^l} = a_j^{l-1} \delta_i^l. \quad (3.39)$$

Esta ecuación indica cómo calcular la tasa de cambio  $\partial C / \partial w_{ij}^l$  en términos del error  $\delta^l$  y las activaciones  $a^{l-1}$ .

Las siguientes ecuaciones resumen el algoritmo de propagación hacia atrás y ayudan a entender lo que sucede cuando una red neuronal aprende [45]:

$\delta^L = \nabla_a C \cdot \sigma'(z^L)$	(3.40a)
$\delta^{(l)} = [(w^{(l+1)})^T \delta^{(l+1)}] \cdot \sigma'(z^{(l)})$	(3.40b)
$\frac{\partial C}{\partial b_i^l} = \delta_i^l$	(3.40c)
$\frac{\partial C}{\partial w_{ij}^l} = a_j^{l-1} \delta_i^l$	(3.40d)

Como se mencionó anteriormente, estas expresiones dan información sobre el entrenamiento del modelo. Por ejemplo, consideremos el término  $\sigma'(z^{(l)})$  de la ecuación (3.40a). Si la función de activación  $\sigma$  se vuelve muy plana entonces  $\sigma'(z^{(l)}) \approx 0$  y esto implica que el correspondiente peso en la capa final aprenderá lentamente. En este caso se dice que la neurona de salida se ha saturado [45], es decir, que la derivada de la activación es muy cercana a cero y como resultado el parámetro ha dejado de aprender.

Un comportamiento similar se obtiene para (3.40b), específicamente para el término  $\sigma'(z^{(l)})$ . Si la neurona está cerca de la saturación entonces  $\delta^l$  será un valor pequeño y en consecuencia los respectivos pesos aprenderán lentamente.

Por otra lado, de (3.40d) se observa que cuando la activación de entrada es baja,  $a_j^{l-1} \approx 0$ , el gradiente  $\partial C / \partial w$  también tiende a ser bajo, de modo que nuevamente se tiene un aprendizaje lento para los pesos en cuestión.

En resumen, un peso aprenderá lentamente si la neurona de entrada tiene una activación baja o si la neurona de salida está saturada, es decir, si tiene una activación alta o baja.

### 3.3.3. Funciones de activación

Las funciones de activación son funciones que devuelven una salida específica para una entrada dada considerando sus pesos. Una función de activación permite que las neuronas activadas puedan ser retenidas y mapeadas mediante una función no lineal, lo cual puede ser utilizado para resolver problemas no lineales. La función de activación se usa para mejorar la capacidad de expresión del modelo de red neuronal, lo que le otorga a la red neuronal el significado de inteligencia artificial [64].

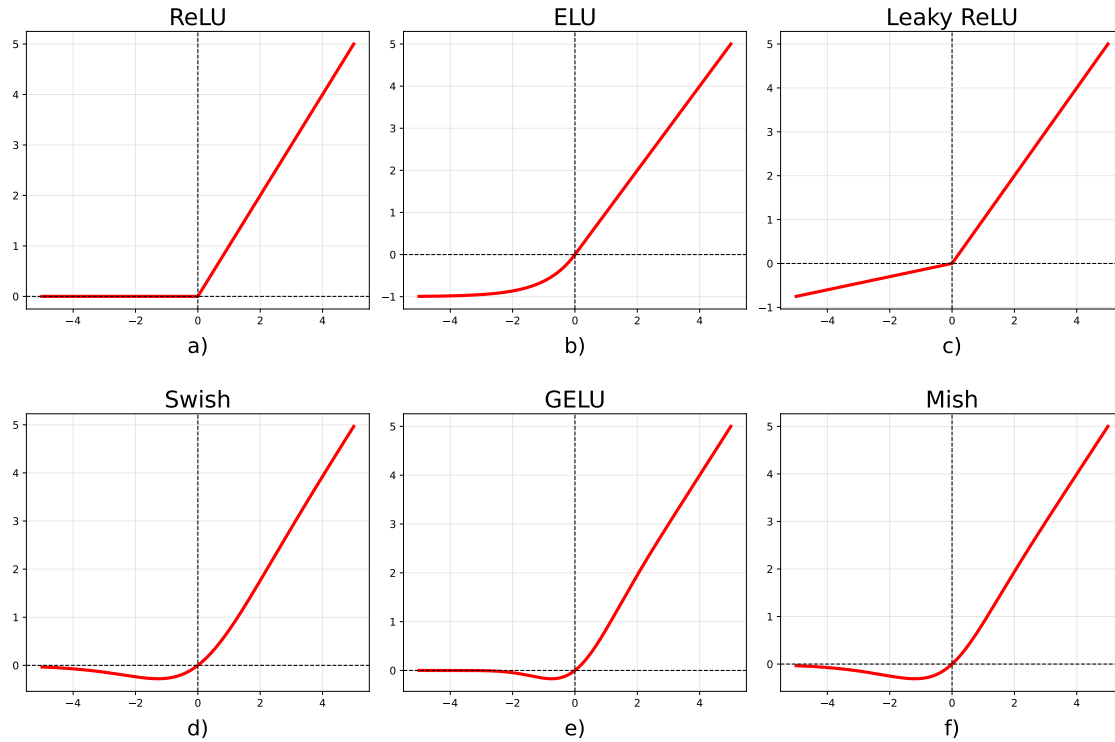


Figura 3.17: Funciones de activación empleadas en este estudio.

En una capa de una red neuronal sin función de activación, la salida de cada neurona se puede describir mediante una suma ponderada de las entradas:

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n + b \quad (3.41)$$

donde  $x_1, x_2, \dots, x_n$  son las entradas a la neurona,  $w_1, w_2, \dots, w_n$  son los pesos asociados a esas entradas,  $b$  es el sesgo de esa neurona, y  $z$  es la salida antes de aplicar una función de activación. Sin la función de activación, esta operación es puramente lineal, lo que significa que cualquier combinación de capas en la red también sería una combinación lineal de las entradas. Por lo tanto, la red solo podría resolver problemas que son lineales en su naturaleza. Esto limita la capacidad del modelo para aprender patrones complejos. En términos matemáticos, si tenemos múltiples capas, cada una con una suma ponderada, la salida de la red neuronal completa seguirá siendo lineal, es decir, incluso si tenemos varias capas, el modelo puede ser reducido a una única capa de suma ponderada. Este comportamiento es similar al de un perceptrón simple, que no puede resolver problemas no lineales como el problema XOR.

Cuando se introduce una función de activación no lineal  $\sigma(z)$ , el modelo ahora tiene la capacidad de capturar relaciones no lineales entre las entradas y las salidas:

$$a = \sigma(z) = \sigma(w_1x_1 + w_2x_2 + \dots + w_nx_n + b) \quad (3.42)$$

Aquí  $a$  es la salida activada de la neurona y  $\sigma(z)$  representa una función de activación no lineal. Esta no linealidad es clave para permitir que la red neuronal aprenda y represente relaciones complejas entre las variables de entrada, lo que le permite clasificar datos que no son separables linealmente. A continuación se presentan algunas funciones de activación.

### Unidad de Rectificación Lineal (ReLU)

Esta función, conocida ampliamente como ReLU por sus siglas en inglés, está definida como

$$\text{ReLU} = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (3.43)$$

El rango de esta función es  $[0, +\infty)$ . Esta función es menos compleja en términos computacionales comparada a otras funciones como la sigmoide o la tangente hiperbólica, pues también puede expresarse como  $\text{ReLU} = \max(0, x)$ . Además, dado que la función ReLU no se satura, no hay problemas con la propagación del gradiente. El inconveniente de ReLU es el problema del gradiente desvaneciente para las entradas negativas. A pesar de tener el problema del gradiente desvaneciente, la función de activación ReLU ha sido utilizada de manera muy extensa en los modelos de aprendizaje profundo [18] [54]. La gráfica de esta función se muestra en el inciso (a) de la figura 3.17.

### Unidad lineal exponencial (ELU)

Esta función, comúnmente conocida como ELU, aborda el problema de la disminución del gradiente que tiene la función ReLU cuando trata con entradas negativas (lo que genera salidas y derivadas nulas, generando un problema llamado “neuronas muertas”). Esta función está definida como

$$\text{ELU} = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \quad (3.44)$$

donde  $\alpha$  es un parámetro positivo que controla la saturación para valores negativos. La función ELU toma valores negativos cuando  $x < 0$  (inciso b de la figura 3.17), acercando la salida promedio a cero y mitigando el gradiente desvaneciente. Su gradiente no nulo para entradas negativas evita neuronas muertas, y su curvatura suave en el origen facilita la optimización [18] [27].

### Unidad de Rectificación Lineal con Fuga (Leaky ReLU)

El problema principal con la función ReLU es el desvanecimiento de gradiente, consecuencia de no usar valores negativos. La unidad de rectificación lineal con fuga (Leaky ReLU) es una extensión de ReLU que usa las entradas negativas al definirse como [18]

$$\text{LReLU} = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \quad (3.45)$$

La función Leaky ReLU tiene un rango de salida de  $(-\infty, \infty)$ . El parámetro  $\alpha$ , que se define en el intervalo  $(0, 1)$ , controla la pendiente en la región negativa. Encontrar la pendiente adecuada es un desafío importante para Leaky ReLU. Diferentes problemas y redes pueden requerir diferentes valores de  $\alpha$ , aunque se recomienda establecerlo en 0.01. Si la tasa de aprendizaje es demasiado alta, Leaky ReLU puede sufrir el problema de explosión del gradiente durante la propagación hacia adelante [54] [27]. La función se presenta en el inciso c de la figura 3.17.

### Función Swish

La activación Swish, propuesta por Ramachandran et al. [50], está definida como

$$\text{Swish} = x \cdot \sigma(\beta x) = \frac{x}{1 + e^{-\beta x}} \quad (3.46)$$

donde  $\sigma(\beta x) = 1/(1 + e^{-\beta x})$  es la función sigmoide, y  $\beta$  es un parámetro que ajusta la forma de la activación entre las funciones lineales y ReLU (ver inciso d de la figura 3.17). El rango de

la salida de Swish es  $(-\infty, \infty)$ . Los valores bajos de  $\beta$  conducen a una función lineal, mientras que los valores altos ajustan esta función a ReLU. Por ejemplo, cuando  $\beta = 0$  la función Swish se transforma simplemente en  $x$ . Por lo tanto,  $\beta$  es un parámetro que puede controlar la no-linealidad de acuerdo a factores como la base de datos o la complejidad de la red. Swish ha mostrado ser superior a ReLU en ciertas tareas [18] [27] [58].

### Unidad lineal de erro Gaussiano (GELU)

La GELU, propuesta por Hendrycks y Gimpel, es una función de activación que introduce un enfoque probabilístico y suavidad en la transición entre valores negativos y positivos de las entradas (inciso e de la figura 3.17). GELU pondera la entrada de una neurona  $x$  multiplicándola por una variable  $m$  que sigue una distribución de Bernoulli, donde la probabilidad de éxito es  $P(X \leq x)$ , con  $X \sim N(0, 1)$ . Esta probabilidad se deriva del hecho de que, después de procesos como la normalización por lotes, las entradas tienden a seguir una distribución normal estándar. La salida esperada de esta transformación probabilística se expresa como [18] [27]:

$$\mathbb{E}(xm) = x\mathbb{E} = xP(X \leq x) \quad (3.47)$$

Donde  $P(X \leq x)$  es la función de distribución acumulada de la distribución Gaussiana. Esto se calcula frecuentemente mediante la función error (erf). La fórmula completa de GELU es

$$\text{GELU}(x) = x \left[ 1 + \text{erf}\left(\frac{x}{2}\right) \right] \quad (3.48)$$

que puede aproximarse como  $0,5x(1 + \tanh[\sqrt{2/\pi}(x + 0,044715x^3)])$  [27]. La función GELU presenta varias ventajas sobre funciones como ReLU, ya que suaviza la transición alrededor del origen, evitando problemas como neuronas muertas y proporcionando una no linealidad más flexible.

### Función Mish

La función de activación Mish, introducida por Misra en 2019, es una función auto-regularizada, no monótona y suave [27]. Se define matemáticamente como:

$$\text{Mish}(x) = x \cdot \tanh(\text{softplus}(x)), \quad (3.49)$$

Mish tiene continuidad  $C^\infty$ , está acotada inferiormente y no tiene límite superior. Aunque en un principio no es evidente, Mish está estrechamente relacionada con Swish, ya que la primera derivada de Mish puede expresarse en términos de Swish [27]. Mish ha demostrado igualar o mejorar el rendimiento de arquitecturas de redes neuronales en comparación con funciones de activación como ReLU, Leaky ReLU y Swish [54]. El principal desafío de Mish reside en su mayor complejidad debido a la composición de múltiples funciones, lo que puede ser una limitación en redes profundas [18], pero su suavidad y capacidad de autorregulación pueden ofrecer beneficios en la optimización y rendimiento de modelos en tareas de visión por computadora. La gráfica de esta función corresponde al inciso f de la figura 3.17.

### 3.3.4. Tipos de Convolución

Además de la convolución estándar, la cual ha sido descrita previamente, en este trabajo se utilizan también las convoluciones dilatada y separable en profundidad, las cuales se describen a continuación.

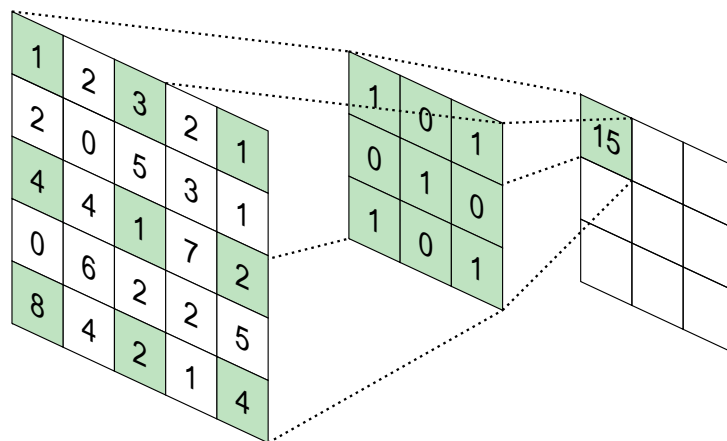


Figura 3.18: Convolución dilatada con un filtro de  $3 \times 3$  y una tasa de dilatación de 2.

### Convolución dilatada

La convolución dilatada es una extensión de la convolución estándar que utiliza filtros convolucionales ampliados. A diferencia de las RNCs convencionales, que típicamente emplean filtros pequeños de tamaño  $2 \times 2$  o  $3 \times 3$  para mantener bajo el número de parámetros y la complejidad computacional, aquí se usan filtros dilatados. Estos filtros introducen espacios (ceros) entre los valores consecutivos del filtro. Un filtro dilatado de tamaño  $k \times k$  y tasa de dilatación  $d$  efectivamente agranda su tamaño a [71]:

$$[k + (k - 1)(d - 1)] \times [k + (k - 1)(d - 1)] \quad (3.50)$$

Esto permite que la convolución cubra un área mayor (campo receptivo más amplio) sin aumentar el número de parámetros (ver figura 3.18), lo que mantiene la complejidad computacional sin cambios. El aumento del campo receptivo permite capturar más información contextual, lo que es útil en aplicaciones donde los patrones se distribuyen más ampliamente, como en el análisis de señales. Una convolución dilatada puede representarse matemáticamente por la siguiente ecuación [65]:

$$y[i] = \sum_k x[i + d \cdot k]w[k] \quad (3.51)$$

donde  $x$  y  $y$  son los mapas de características de entrada y salida, respectivamente;  $w[k]$  es el kernel convolucional de tamaño  $k \times k$ , y  $d$  es el factor de dilatación. Cuando  $d = 1$ , la convolución dilatada se convierte en una convolución estándar.

El principal beneficio de la convolución dilatada es que permite una expansión exponencial del campo receptivo sin perder resolución o cobertura del mapa de características.

### Convolución separable en profundidad

La convolución separable en profundidad es una variante de la convolución estándar que tiene como objetivo reducir el costo computacional y la cantidad de parámetros de una RNC, sin comprometer significativamente su capacidad de aprendizaje.

Como se muestra en la figura 3.19, se compone de dos pasos [28] [55]:

1. Convolución separable en profundidad: Aplica una convolución independiente a cada canal de entrada de la imagen o tensor. En lugar de aplicar un filtro 3D a todos los canales simultáneamente (como se hace en una convolución estándar), se aplica un filtro 2D a cada

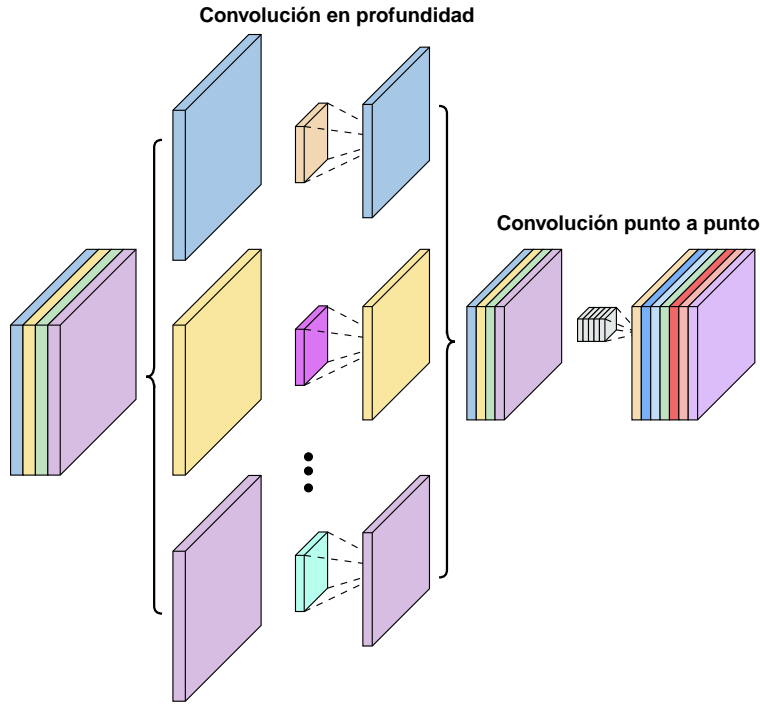


Figura 3.19: Convolución separable en profundidad.

canal por separado. En este paso no se altera el número de canales. La convolución de profundidad puede ser escrita como

$$\mathcal{O}'_{h',w',m} = \sum_{i=1}^D \sum_{j=1}^D \mathcal{D}_{i,j,m} \cdot \mathcal{J}_{h_i,w_j,m} \quad (3.52)$$

donde  $\mathcal{D}_{i,j,m}$  es el kernel de la convolución de profundidad de tamaño  $D \times D$  aplicado al canal  $m$ ;  $\mathcal{O}'_{h',w',m}$  denota la salida en  $(h', w')$ , y  $\mathcal{J}_{h_i,w_j,m}$  representa la entrada en la posición  $\mathcal{J}_{h_i,w_j,m}$ .

2. Convolución punto a punto: Después de la convolución separable en profundidad, se usa una convolución  $1 \times 1$  para combinar la información a lo largo de los canales de salida obtenidos en el paso anterior. Este paso no cambia el tamaño de los mapas de entrada, pero sí el número de canales. La fórmula para esta operación es

$$\mathcal{O}_{h',w',n} = \sum_{m=1}^M \mathbf{P}_{m,n} \cdot \mathcal{O}'_{h',w',m} \quad (3.53)$$

Aquí,  $\mathbf{P}_{m,n}$  es el kernel de convolución punto a punto que opera sobre  $\mathcal{O}'_{h',w',m}$  para generar  $\mathcal{O}_{h',w',n}$ .

### 3.3.5. Tipos de conexiones

#### Conexiones directas

Las conexiones directas en redes neuronales son aquellas donde las salidas de una capa se conectan directamente como entrada de una capa posterior sin ninguna transformación intermedia.

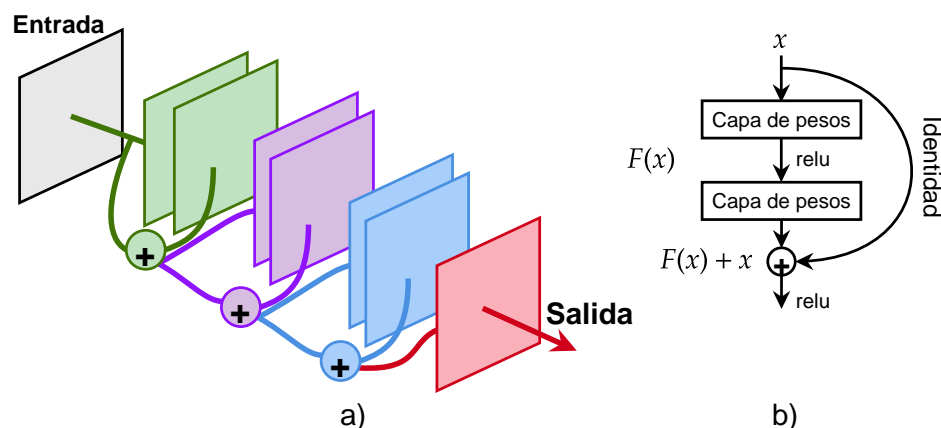


Figura 3.20: Conexión residual. (a) Ejemplo en un modelo RNC. (b) Esquema que ilustra el salto de capas intermedias.

Este tipo de conexión es la forma más sencilla de flujo de datos en una red, y a menudo se utiliza en arquitecturas estándar de redes neuronales profundas donde cada capa toma como entrada la salida de la capa inmediatamente anterior. Las conexiones directas pueden ser descritas como:

$$x_l = \mathcal{H}_l(x_{l-1}, W_l) \quad (3.54)$$

donde  $x_{l-1}$  es la salida de la capa anterior y  $\mathcal{H}_l$  representa la transformación aplicada en la capa  $l$  (por ejemplo, una combinación de convolución, activación y normalización) [57].

Este tipo de conexión tiene una ventaja en su simplicidad, pero presenta ciertos desafíos en redes profundas, como la pérdida de gradientes o la saturación de activaciones cuando se trata de un número elevado de capas. Debido a esto, surgieron variantes como las conexiones residuales o densas, que mejoran el flujo de información y gradientes.

### Conexiones residuales

Las conexiones residuales fueron introducidas para solucionar el problema de degradación del rendimiento en redes neuronales profundas, donde aumentar el número de capas no siempre mejora los resultados [23]. Este tipo de conexión permite que la entrada de una capa se pase directamente a una capa posterior, saltando una o más capas intermedias, lo que facilita el flujo de la información. Esta conexión se ilustra en la figura 3.20.

Matemáticamente, en lugar de aprender la transformación completa  $\mathcal{H}(x)$  de la entrada  $x$ , la red aprende una función residual  $\mathcal{F}(x) = \mathcal{H}(x) - x$ , de modo que la salida del bloque residual es [70] [23]:

$$y = \mathcal{F}(x) + x \quad (3.55)$$

Esto facilita que la red aprenda la diferencia entre la entrada y la salida deseada, en lugar de la transformación completa, lo que reduce el problema de gradientes desvanecientes, mejora la estabilidad del entrenamiento y permite que las redes profundas alcancen un mejor rendimiento.

Los bloques residuales suelen estar compuestos por dos o más capas de convolución, y al final se añade la conexión residual (o atajo) que suma la entrada original a la salida final [23]. Si las dimensiones no coinciden, se aplica una transformación para ajustar el tamaño de los datos. Las conexiones residuales son fundamentales en arquitecturas modernas como ResNet, donde se han logrado mejoras significativas en tareas de clasificación.

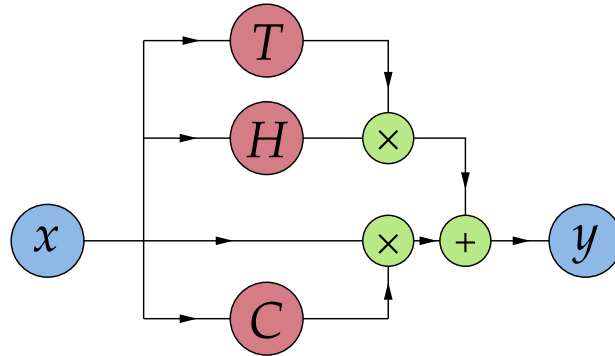


Figura 3.21: Esquema de la conexión highway, donde se implementan las puertas de transformación  $T$ , y la puerta de transporte  $C$ .

### Conexiones highway

Las conexiones highway son una forma más sofisticada de conexiones residuales que permiten el control del flujo de información a través de la red utilizando mecanismos de “puertas”. Estas puertas deciden qué cantidad de la información original pasa a las siguientes capas, y qué cantidad se transforma a través de una función no lineal [37].

En una red highway, la capa típica de una red feedforward realiza una transformación no lineal  $\mathcal{H}(x, W_{\mathcal{H}})$  sobre su entrada  $x$ , generando una salida  $y$ . Esto se puede expresar como:

$$y = \mathcal{H}(x, W_{\mathcal{H}}) \tag{3.56}$$

Aquí,  $\mathcal{H}(x, W_{\mathcal{H}})$  es generalmente una transformación afín seguida de una función de activación no lineal, como ReLU o tangente hiperbólica. En las conexiones highway, además de  $\mathcal{H}(x, W_{\mathcal{H}})$ , se definen dos transformaciones no lineales adicionales [57]:  $\mathcal{T}(x, W_{\mathcal{T}})$  y  $\mathcal{C}(x, W_{\mathcal{C}})$ . Estas representan las “puertas de transformación” y “puertas de transporte” respectivamente, y determinan cuánto de la información de la entrada  $x$  se transforma o se transporta directamente a la salida (ver figura 3.21):

$$y = \mathcal{H}(x, W_{\mathcal{H}}) \cdot \mathcal{T}(x, W_{\mathcal{T}}) + x \cdot \mathcal{C}(x, W_{\mathcal{C}}) \tag{3.57}$$

Donde  $\mathcal{T}(x, W_{\mathcal{T}})$  es la puerta de transformación, que controla cuánto de  $x$  se transforma, y  $\mathcal{C}(x, W_{\mathcal{C}})$  es la puerta de transporte, que controla cuánto de  $x$  se transporta sin cambios. Generalmente  $\mathcal{T}$  es una convolución seguida de la aplicación de la función sigmoide [57] [37]. Para simplificar, a menudo se usa la relación  $\mathcal{C} = 1 - \mathcal{T}$ , lo que lleva a

$$y = \mathcal{H}(x, W_{\mathcal{H}}) \cdot \mathcal{T}(x, W_{\mathcal{T}}) + x \cdot [1 - \mathcal{T}(x, W_{\mathcal{T}})] \tag{3.58}$$

Esto permite que la capa highway sea más flexible que una capa estándar. En particular se tiene que

$$y = \begin{cases} x \text{ (sin cambios)} & \text{si } \mathcal{T}(x, W_{\mathcal{T}}) = 0 \\ \mathcal{H}(x, W_{\mathcal{H}}) \text{ (transformación)} & \text{si } \mathcal{T}(x, W_{\mathcal{T}}) = 1 \end{cases}$$

En resumen, las conexiones highway permiten que una capa ajuste su comportamiento entre transformar la entrada o simplemente dejarla pasar. Mediante puertas de transformación, deciden qué parte de la señal se modifica y cuál se preserva. Esta flexibilidad ayuda a que la red aprenda eficientemente manteniendo información importante a lo largo de las capas.

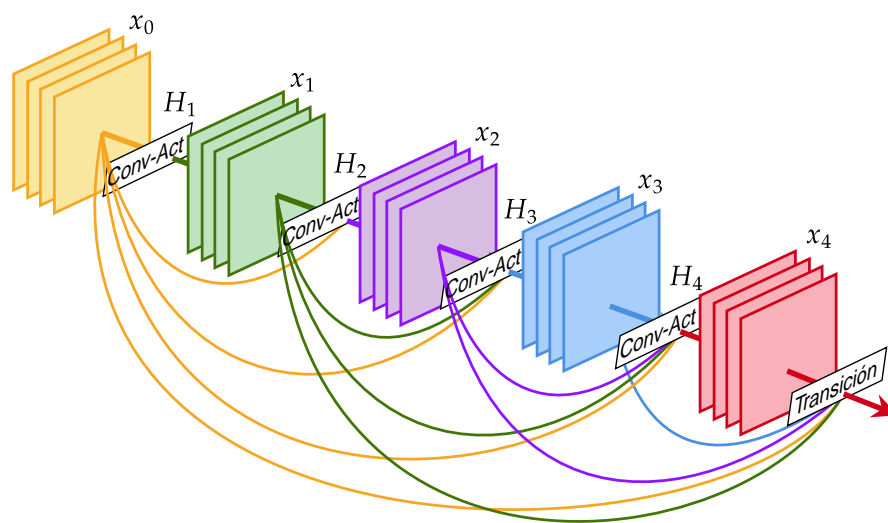


Figura 3.22: Esquema de las conexiones densas, donde todas una determinada capa recibe como entrada los mapas de características de todas las capas anteriores.

### Conexiones densas

Consideremos una imagen  $x_0$  que pasa a través de una red neuronal de convolución, la cual se componen de  $L$  capas que implementan una transformación lineal  $\mathcal{H}_l(\cdot)$  cada una. Esta transformación puede ser una función compuesta de múltiples operaciones como convolución, submuestreo, activación o normalización. La salida de la capa  $l$  se denota como  $x_l$ .

Para mejorar el flujo de información entre las capas, Huang et al. introdujeron las conexiones densas en las DenseNets [25]. Mientras que las conexiones residuales añaden la entrada original a la salida de una capa, las conexiones densas conectan todas las capas entre sí. De esta forma, la capa  $l$  recibe como entrada los mapas de características de todas las capas precedentes (ver figura 3.22):

$$x_l = \mathcal{H}_l([x_0, x_1, \dots, x_{l-1}]) \quad (3.59)$$

donde  $[x_0, x_1, \dots, x_{l-1}]$  denota la concatenación de los mapas de características generados en las  $l$  capas anteriores.

Este tipo de conexión mejora el flujo de gradientes y características a través de la red, lo que mitiga el desvanecimiento del gradiente en redes profundas. Además, permite que cada capa acceda directamente a los gradientes de las capas anteriores, mejorando la eficiencia del aprendizaje y reduciendo la redundancia de características al reutilizar información previamente calculada. Como resultado, las DenseNets tienden a ser más compactas y eficientes en parámetros que otras arquitecturas, al tiempo que muestran una regularización mejorada debido a la presencia de múltiples rutas de información.



## Capítulo 4

# Materiales y métodos

Esta sección describe los recursos y procedimientos empleados en la investigación. En primer lugar, se presenta la base de datos utilizada, detallando sus características y la información contenida. Luego, se explica el preprocesamiento aplicado a las señales EMG para su adecuada utilización en el modelo. Posteriormente, se describe el proceso de selección de hiperparámetros y la construcción del diseño D-óptimo, utilizado para reducir el número de experimentos manteniendo su validez estadística. Finalmente, se aborda el diseño e implementación de las redes neuronales convolucionales, explicando su estructura y los criterios utilizados para su evaluación.

La metodología se diseñó para garantizar un análisis exhaustivo desde la recopilación de datos hasta la evaluación de resultados. En la figura 4.1 se presenta un diagrama que resume los pasos realizados.

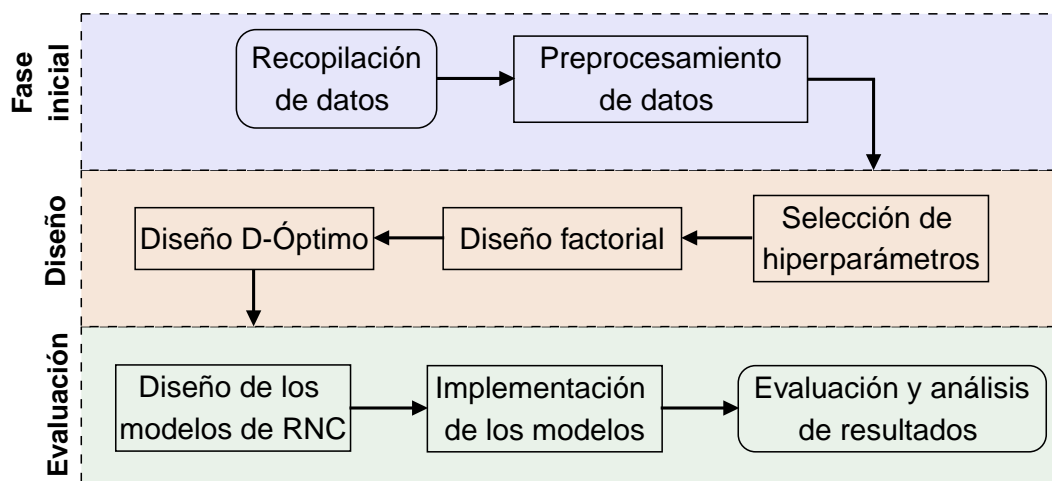


Figura 4.1: Etapas de la metodología.

### 4.1. Bases de datos

En esta investigación se utilizaron tres subconjuntos de la base de datos Ninapro (Non Invasive Adaptive Prosthetics) [8], la cual recopila datos de 67 sujetos no amputados y 11 sujetos amputados. Los ejercicios incluidos en las tres sub-bases son los mismos; en este estudio, se empleó específicamente el Ejercicio B, que comprende 17 movimientos isométricos e isotónicos de la

Base	Sujetos	Tipo de sujetos	No. de electrodos
Ninapro DB1	27	Sin amputaciones	10 Otto Bock
Ninapro DB2	40	Sin amputaciones	12 Delsys
Ninapro DB3	11	Amputación transradial	12 Delsys

Tabla 4.1: Parámetros de las bases de datos que se emplearon en el estudio.

mano y la muñeca. En la Tabla 4.1 se presentan los parámetros principales de las bases de datos utilizadas, seguidos de una descripción detallada de cada una.

*Base de datos 1 (DB1):* la base de datos 1 de Ninapro está diseñada para el estudio de movimientos de la mano en sujetos no amputados. Esta base incluye señales electromiográficas (EMG), cinemática de la mano y, en algunos casos, dinámica de movimientos. Los datos fueron recolectados de 27 sujetos intactos (20 hombres y 7 mujeres) con una edad promedio de  $28 \pm 3,4$  años.

La adquisición de las señales EMG se realizó utilizando 10 electrodos Otto Bock de doble diferencial, con un sistema de amplificación y rectificación RMS incorporado. Estos electrodos fueron dispuestos de manera uniforme alrededor del antebrazo en puntos estratégicos para captar la actividad muscular. La frecuencia de muestreo de las señales fue de 100 Hz.

Como se mencionó anteriormente, se utilizó específicamente el Ejercicio B, que consta de 17 gestos isométricos e isotónicos, realizados bajo las siguientes condiciones:

- Duración de cada movimiento: 5 segundos de ejecución seguidos por 3 segundos de reposo.
- Número de repeticiones: Cada gesto se repitió 10 veces.
- Instrucciones: Los sujetos imitaron los movimientos mostrados en una pantalla, concentrándose en una ejecución precisa y constante.

Las señales recolectadas en esta base de datos fueron procesadas para garantizar su calidad. Se implementó sincronización con marcas de tiempo precisas y un ajuste de etiquetas mediante un algoritmo de razón de verosimilitud para corregir posibles desfases en los movimientos.

*Base de datos 2 (DB2):* la base de datos 2 de Ninapro amplía el alcance de la base anterior al incluir señales EMG, cinemática de la mano y dinámica de movimientos en un grupo más amplio de sujetos intactos. Los datos se recolectaron de 40 sujetos no amputados (28 hombres y 12 mujeres) con una edad promedio de  $29,9 \pm 3,9$  años.

Para la adquisición de las señales EMG, se utilizaron 12 electrodos Delsys de doble diferencial, los cuales operan con una frecuencia de muestreo de 2 kHz y una precisión de ruido inferior a 750 nV RMS. Ocho de estos electrodos se colocaron de manera uniforme alrededor del antebrazo a la altura de la articulación radiohumeral, mientras que los otros cuatro se posicionaron en puntos específicos sobre los siguientes músculos:

- Flexor superficial de los dedos
- Extensor superficial de los dedos
- Bíceps braquial (cuando es aplicable)
- Tríceps braquial (cuando es aplicable)

Los movimientos fueron realizados bajo las siguientes condiciones:

- Duración de cada movimiento: 5 segundos de ejecución seguidos por 3 segundos de reposo.

- Número de repeticiones: cada gesto se repitió 6 veces.
- Instrucciones: los sujetos imitaron los movimientos mostrados en una pantalla, concentrándose en una ejecución fluida y precisa.

Durante el procesamiento de los datos, se sincronizaron todas las señales mediante interpolación lineal y se aplicó un filtrado para eliminar interferencias eléctricas de 50 Hz y sus armónicos. Adicionalmente, las etiquetas de los movimientos fueron ajustadas para corregir discrepancias relacionadas con los tiempos de reacción de los sujetos.

*Base de datos 3 (DB3):* La base de datos 3 de Ninapro está diseñada específicamente para estudiar el control de prótesis de mano en sujetos amputados. Esta base incluye datos obtenidos de 11 sujetos transradiales amputados (todos hombres, con una edad promedio de  $42,36 \pm 11,96$  años).

La adquisición de las señales EMG se realizó con el mismo arreglo de electrodos usado en la Base de datos 2, considerando los mismos músculos.

El protocolo de adquisición fue igual al de DB2, los sujetos realizaron los gestos de la siguiente forma:

- Duración de cada movimiento: 5 segundos de ejecución, seguidos por 3 segundos de reposo.
- Número de repeticiones: cada movimiento se repite 6 veces.
- Instrucciones: a los sujetos amputados se les pidió que imitaran los movimientos proyectados en una pantalla, tratando de realizar los gestos de manera imaginaria (es decir, sin ejercer fuerza real), concentrándose en la activación muscular.

Durante el procesamiento, las señales fueron sincronizadas mediante marcas de tiempo de alta precisión y alineadas a una frecuencia uniforme de 2 kHz. Además, se ajustaron las etiquetas de los movimientos utilizando un algoritmo de corrección basado en la razón de verosimilitud para abordar las discrepancias relacionadas con los tiempos de reacción de los sujetos. Como parte del preprocesamiento, las señales EMG fueron filtradas para eliminar interferencias de 50 Hz y sus armónicos mediante un filtro de Hampel, garantizando datos limpios y fiables para el análisis.

## 4.2. Preprocesamiento de los datos

El preprocesamiento de las señales EMG se realizó en Matlab utilizando un método de segmentación con solapamiento. Las señales fueron divididas en ventanas de 200 ms con un solapamiento del 80 % del tamaño de la ventana (160 ms), lo que permitió mantener una continuidad adecuada para preservar información relevante entre ventanas consecutivas.

La distribución de las base de datos sigue un patrón alternado de actividad y reposo. Para cada movimiento ( $mov_c$ , donde  $c = 1, 2, 3, \dots, 17$ ), la secuencia tiene la forma:

$$\text{reposo, } mov_c, \text{ reposo, } mov_c, \dots \quad (c = 1, 2, \dots, 17)$$

Este ciclo se repite 10 veces para la primera base de datos y 6 veces para DB2 y DB3 antes de pasar al siguiente movimiento. Este diseño asegura la captura tanto de las señales características de cada movimiento como de los intervalos de reposo, necesarios para diferenciar las clases.

Para preparar los datos (de forma individual para cada base de datos), primero se agruparon todas las muestras de una misma clase  $c$  en matrices de tamaño  $n \times m_c$ , donde  $n$  corresponde al número de electrodos y  $m_c$  al número de puntos de la clase  $c$ . Durante este proceso, se descartó una pequeña cantidad de puntos en cada clase, con el objetivo de garantizar que el número de segmentos generados (considerando el solapamiento) fuera exacto.

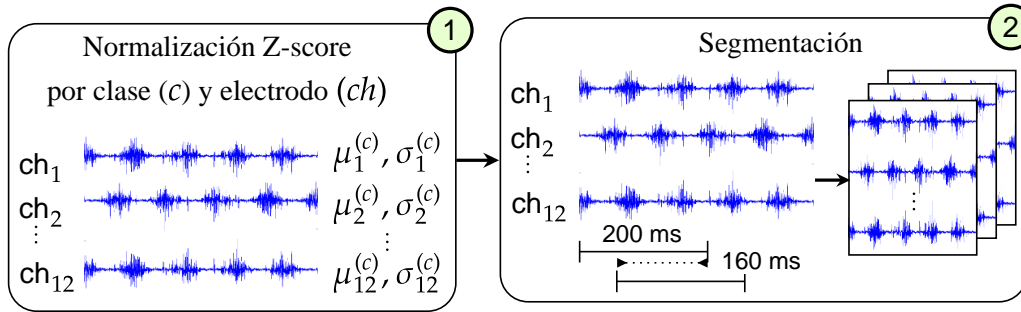


Figura 4.2: Diagrama que muestra la construcción de las señales 2D, una dimensión representa el número de electrodos y la otra dimensión el tamaño de la ventana.

Posteriormente, se aplicó una normalización Z-score a los datos, la cual se realizó individualmente para cada movimiento y de forma independiente para cada electrodo  $ch$ . La normalización se calculó utilizando la ecuación:

$$Z_{ch,i}^{(c)} = \frac{x_{ch,i}^{(c)} - \mu_{ch}^{(c)}}{\sigma_{ch}^{(c)}} \quad (4.1)$$

donde  $x_{ch,i}^{(c)}$  representa la  $i$ -ésima muestra del canal  $ch$  correspondiente a la clase  $c$ , y  $Z_{ch,i}^{(c)}$  es su valor estandarizado. Este proceso se repitió para los datos correspondientes a los electrodos restantes.

Finalmente, se segmentaron las señales empleando el esquema mencionado. Considerando una frecuencia de muestreo de 100 Hz o 2 kHz dependiendo del conjunto de datos, y un tamaño de ventana de 200 ms, los segmentos resultantes tenían dimensiones de  $12 \times 400$ , donde la primera dimensión corresponde al número de electrodos y la segunda al número de puntos en cada segmento. En la figura 4.2 se indica la construcción de las señales de dos dimensiones con base en el método de segmentación con solapamiento.

### 4.3. Selección de hiperparámetros

El siguiente paso fue definir el espacio de búsqueda para la exploración de los hiperparámetros. Basándose en el estado del arte, se identificaron los hiperparámetros menos explorados con potencial para optimizar el rendimiento de la red en términos de precisión. El espacio de búsqueda definido se muestra en la tabla 4.2

Hiperparámetro	Rango considerado
Activación	ReLU, ELU, Leaky ReLU, Swish, GELU, Mish
Conexiones	Directas, Residuales, Densas, Highway
Convolución	Estándar, Dilatada, Separable en profundidad.
Capas paralelas	0, 1, 2, 3
Capas secuenciales	2, 3

Tabla 4.2: Rango de los hiperparámetros considerados para el ajuste.

La elección de estos hiperparámetros y sus valores responde a una combinación de tendencias en el estado del arte y características propias del problema:

1. **Funciones de activación:** Se incluyeron funciones tradicionales como ReLU y ELU debido a su popularidad y eficiencia computacional, así como funciones más recientes como Swish y

Mish, las cuales han mostrado mejoras en problemas con alta no linealidad. GELU, por otro lado, se incluyó por su buen desempeño en arquitecturas modernas como transformers.

2. **Tipos de conexión entre capas:** Las conexiones residuales y densas son ampliamente utilizadas para mitigar el problema del gradiente desaparecido, especialmente en redes profundas. Las conexiones highway fueron seleccionadas por su capacidad para adaptarse dinámicamente al flujo de información, mientras que las conexiones directas sirvieron como base para comparaciones.
3. **Tipos de convolución:** Las convoluciones separables en profundidad (SP) son conocidas por su eficiencia computacional al reducir el número de parámetros, mientras que las dilatadas permiten capturar características en múltiples escalas sin aumentar la dimensionalidad. Las convoluciones estándar fueron incluidas como punto de referencia.
4. **Capas paralelas adicionales y secuenciales:** Se evaluaron configuraciones con hasta tres capas paralelas adicionales, dado que estas pueden ayudar a modelar características complejas en paralelo. Para las capas secuenciales, se limitaron las opciones a dos y tres para evitar arquitecturas demasiado profundas, considerando la estructura de las señales EMG y la necesidad de minimizar el sobreajuste.

Finalmente, el espacio de búsqueda fue diseñado para mantener un equilibrio entre exhaustividad y viabilidad computacional. Las combinaciones posibles permiten explorar una amplia gama de configuraciones sin exceder los recursos disponibles, garantizando una evaluación robusta de los modelos.

## 4.4. Diseño D-Óptimo

Para explorar de manera eficiente el espacio de hiperparámetros definido, se utilizó un diseño experimental D-óptimo, construido a partir del diseño factorial completo. Este enfoque permite seleccionar un subconjunto representativo de combinaciones experimentales que maximicen la información obtenida sobre los factores de interés, reduciendo el número de experimentos necesarios sin comprometer la calidad del análisis.

Se generó inicialmente el diseño factorial completo considerando todas las combinaciones posibles de los hiperparámetros. Este diseño incluyó:

- 6 funciones de activación.
- 4 tipos de conexión.
- 3 tipos de convolución.
- 4 opciones para el número de capas paralelas.
- 2 opciones para el número de capas secuenciales.

En total, el diseño factorial completo abarcó  $6 \times 4 \times 3 \times 4 \times 2 = 576$  combinaciones. A partir del diseño factorial completo, se aplicó el criterio de D-optimalidad utilizando Minitab, seleccionando 30 combinaciones que maximizan la información estadística. Este número fue determinado considerando la complejidad del problema, las restricciones computacionales y la necesidad de evaluar múltiples configuraciones en un tiempo razonable. En la tabla 4.3 se presentan las primeras diez combinaciones del diseño D-óptimo.

Cada una de las combinaciones seleccionadas por el diseño D-óptimo fue implementada y evaluada mediante el entrenamiento de una red neuronal convolucional. Las métricas de rendimiento, como precisión y F1-score, fueron registradas para cada configuración, con el objetivo de identificar patrones y tendencias relevantes.

Activación	Conexión	Convolución	Par.	Sec.
Mish	Directas	Estándar	3	3
Mish	Highway	Separable en profundidad	2	2
GELU	Directas	Separable en profundidad	3	2
LReLU	Directas	Estándar	2	2
Swish	Highway	Estándar	3	2
Swish	Directas	Dilatada	1	3
GELU	Densas	Estándar	0	3
Mish	Residuales	Dilatada	0	2
ELU	Residuales	Estándar	1	2
ReLU	Densas	Dilatada	3	2

Tabla 4.3: Primeros diez puntos del diseño D-Óptimo.

## 4.5. Diseño e implementación de las redes

Cada red fue diseñada de manera independiente según la combinación específica de hiperparámetros seleccionada en el diseño D-óptimo, resultando en un total de 30 arquitecturas únicas. Como punto de partida, se tomó una arquitectura del estado del arte [36], adaptándola para incorporar variaciones según los hiperparámetros explorados.

### 4.5.1. Estructura general de las redes

De manera general, las redes desarrolladas se organizan en bloques consecutivos que combinan operaciones convolucionales y mecanismos de estabilización del entrenamiento. La estructura de cada bloque incluye:

**1. Capas convolucionales en paralelo:**

Dependiendo de la configuración seleccionada, se incorporaron hasta tres capas convolucionales adicionales en paralelo. En caso de no utilizar capas adicionales, el bloque incluye únicamente la capa de convolución principal.

**2. Función de activación:**

Cada salida de las capas convolucionales se somete a una función de activación específica (ReLU, ELU, Leaky ReLU, Swish, GELU o Mish) según el hiperparámetro correspondiente.

**3. Concatenación de salidas paralelas:**

Cuando se utilizan múltiples capas convolucionales en paralelo, sus salidas se concatenan antes de pasar al siguiente paso.

**4. Normalización por lotes:**

Cada bloque finaliza con una capa de normalización por lotes (batch normalization), cuyo objetivo es estabilizar el entrenamiento al regularizar las distribuciones de las activaciones intermedias.

Después de los bloques principales, la arquitectura incluye un bloque adicional que incorpora:

- Dos capas convolucionales de  $1 \times 1$ , cada una seguida por su respectiva función de activación y una capa de normalización por lotes.
- Una capa de submuestreo (global max pooling) para reducir la dimensionalidad de las características aprendidas.

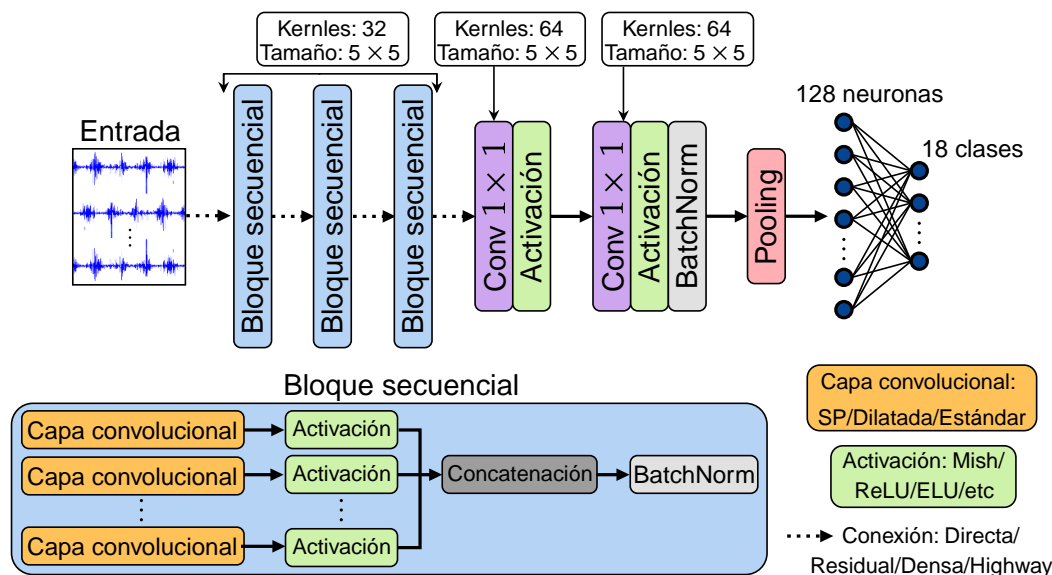


Figura 4.3: Arquitectura de referencia para generar los modelos utilizados en este estudio. Los bloques secuenciales se componen de una convolución, función de activación, concatenación y normalización por lotes. La activación, convolución, número de capas paralelas y bloques secuenciales, así como el tipo de conexión entre ellos, se define según los hiperparámetros seleccionados mediante el diseño D-óptimo.

Finalmente, la red se conecta a una capa totalmente conectada (densa) con 128 neuronas, que actúa como un cuello de botella antes de la salida final. La capa de salida contiene 18 neuronas, correspondientes a las 17 clases de movimiento más el estado de reposo, y emplea la activación softmax para producir probabilidades por clase.

El uso de capas convolucionales en paralelo permite a las redes extraer características de diferentes escalas de la señal, mientras que las capas  $1 \times 1$  actúan como filtros de reducción dimensional para evitar sobreajuste. Además, la incorporación de normalización por lotes asegura estabilidad en el entrenamiento, especialmente en configuraciones con muchas capas. La capa densa intermedia permite una transformación no lineal efectiva antes de la salida, mientras que la capa softmax facilita la clasificación multi-clase con alta precisión.

Entre las ventajas de este diseño se tiene que la estructura modular de los bloques permite adaptarse a cualquier combinación de hiperparámetros seleccionada. Además, la reducción dimensional mediante submuestreo y capas  $1 \times 1$  reduce los costos computacionales en las etapas finales de la red, y la combinación de capas convolucionales, funciones de activación avanzadas y normalización asegura un aprendizaje robusto y generalizable.

La Figura 4.3 ilustra los componentes principales de la arquitectura base, junto con los hiperparámetros que se mantuvieron con valores constantes. Además, muestra los cinco factores experimentales que fueron modificados en este estudio. Cada uno de los 30 modelos generados tiene una combinación única de función de activación, tipo de conexión, tipo de convolución, número de capas paralelas adicionales y número de capas secuenciales.

#### 4.5.2. Detalles del entrenamiento

El desarrollo y entrenamiento de las redes se llevó a cabo utilizando TensorFlow 2.10.0, con soporte para GPU mediante cuDNN y cuDNN Toolkit, lo que permitió entrenar de manera eficiente

incluso las arquitecturas más complejas. Cada modelo se entrenó de forma independiente, asegurando que las configuraciones específicas de hiperparámetros fueran evaluadas sin interferencias.

Para evaluar el desempeño de cada red, se empleó un protocolo de validación cruzada de 5 pliegues, con 30 épocas de entrenamiento por pliegue. El 80 % de los datos disponibles de cada sujeto se utilizó para el entrenamiento, mientras que el 20 % restante se reservó para pruebas.

El entrenamiento se llevó a cabo utilizando el optimizador Adam, con los parámetros  $\beta_1 = 0,9$ ,  $\beta_2 = 0,999$ , y  $\epsilon = 1 \times 10^{-7}$ , una tasa de aprendizaje fija de 0.0001 y un tamaño de lote de 16. Para prevenir el sobreentrenamiento, se implementó la técnica de early stopping, configurando un criterio de paciencia de 10 épocas consecutivas sin mejora en la precisión de validación.

Dentro de los bloques convolucionales, se emplearon filtros de tamaño  $5 \times 5$ , con 32 filtros por capa convolucional. En los bloques adicionales de convolución  $1 \times 1$ , cada capa contenía 64 filtros. Esta configuración permite mantener un equilibrio entre capacidad de aprendizaje y eficiencia computacional.

Cada red fue entrenada de manera independiente usando los datos de cada sujeto, lo cual permite evaluar el rendimiento de cada configuración en función de la precisión por sujeto, y así identificar las combinaciones de hiperparámetros que ofrecen el mejor desempeño para la clasificación de señales EMG.

El desempeño de las redes se evaluó principalmente mediante la métrica de precisión (accuracy), definida como

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}, \quad (4.2)$$

Los términos utilizados en la ecuación (4.2) se definen como sigue:  $TP$  representa los verdaderos positivos, que corresponden a las clasificaciones correctas de instancias positivas;  $TN$  denota los verdaderos negativos, es decir, las clasificaciones correctas de instancias negativas;  $FP$  identifica los falsos positivos, donde una instancia negativa es incorrectamente clasificada como positiva; y  $FN$  se refiere a los falsos negativos, que ocurren cuando una instancia positiva es clasificada erróneamente como negativa.

Además de evaluar las métricas de rendimiento, los resultados fueron sometidos a pruebas estadísticas para identificar diferencias significativas entre las combinaciones de hiperparámetros. Estas pruebas incluyeron un análisis de varianza (ANOVA) y la prueba post hoc de Tukey. Este enfoque permitió no solo medir la efectividad de cada modelo, sino también establecer la influencia de los hiperparámetros seleccionados sobre las métricas de rendimiento.

Cabe remarcar que tanto el preprocesamiento de los datos como el entrenamiento de cada red se realizaron de manera independiente para cada sujeto de cada una de las tres bases de datos. Este enfoque permite capturar las características específicas de las señales EMG de cada individuo, asegurando que los modelos sean entrenados y evaluados bajo condiciones personalizadas y evitando la contaminación entre los distintos conjuntos de datos.

## 4.6. Medición del tiempo de inferencia

Es fundamental medir el tiempo de inferencia de los modelos de RNC para evaluar su viabilidad en aplicaciones reales, ya que el tiempo de respuesta del cuerpo humano es inferior a 200 ms. Por lo tanto, un sistema de prótesis eficiente no debe superar dicho límite.

De acuerdo con [36], el tiempo de inferencia puede calcularse mediante la siguiente ecuación:

$$T = \frac{1}{2}W + \frac{1}{2}V + T_e \quad (4.3)$$

donde  $W$  representa el tamaño de la ventana utilizada en la segmentación,  $V$  el solapamiento y  $T_e$  el tiempo de respuesta de la RNC, que abarca desde el preprocesamiento de la señal hasta la obtención de la predicción del modelo.

El tiempo de inferencia depende tanto de los parámetros de segmentación como del hardware empleado. Para la evaluación del tiempo de inferencia se seleccionaron tres procesadores con el fin de representar escenarios contrastantes de uso. El **AMD Ryzen 7 5700G** (CPU 1) corresponde a un procesador de alto rendimiento y costo medio, empleado como referencia de capacidad computacional en equipos de escritorio modernos. El **Intel Celeron 1017U** (CPU 2) representa un procesador de gama baja y bajo costo, común en computadoras portátiles con recursos limitados, lo que permite analizar el comportamiento de los modelos en condiciones restringidas. Finalmente, la **Raspberry Pi 4 modelo B** (CPU 3) se incluyó como un dispositivo portátil de muy bajo costo y reducido consumo energético, frecuentemente utilizado en aplicaciones embebidas y de cómputo en el borde. De esta forma, la comparación entre los tres procesadores permite evaluar la viabilidad de las arquitecturas propuestas en contextos que van desde entornos de alto rendimiento hasta dispositivos de bajo consumo.

No se consideró el uso de una GPU, ya que, en la práctica, se busca que los componentes de una prótesis sean de bajo costo, compactos y energéticamente eficientes. Las GPUs suelen requerir mayor consumo energético y disipación térmica, lo que podría aumentar el tamaño y la complejidad del sistema.

Se consideraron los 30 modelos evaluados con las bases de datos DB1 y DB3, ya que las entradas de DB2 son del mismo tamaño que las de la base de datos de sujetos amputados (DB3), y esta última es más relevante.



## Capítulo 5

# Resultados y discusión

Este capítulo presenta los resultados de entrenar y evaluar 30 arquitecturas de redes neuronales convolucionales (RNCs) diseñadas mediante un esquema D-óptimo, probadas en tres bases de datos de señales EMG de diferentes condiciones y características. El objetivo es identificar las combinaciones de hiperparámetros que maximizan la precisión, analizar la robustez del modelo frente a distintos conjuntos de datos y evaluar su tiempo de respuesta.

Los resultados se dividen en dos partes: (1) un análisis por base de datos, donde se evalúan las 30 redes independientemente para cada base, considerando la precisión de las 8 mejores y 8 peores combinaciones de hiperparámetros, análisis estadísticos y una evaluación detallada del mejor modelo; y (2) una comparación entre bases de datos, que permite identificar patrones y diferencias en el desempeño, evaluando la robustez del modelo.

Finalmente, se mide el tiempo de predicción de las mejores arquitecturas, que debe ser menor a 200 ms para asegurar su viabilidad en aplicaciones en tiempo real. El enfoque busca proporcionar una visión completa de los resultados, evaluar la robustez del modelo y validar su eficiencia temporal.

### 5.1. Resultados por base de datos

#### 5.1.1. Base de datos 1

Los resultados que se presentan a continuación se obtuvieron al evaluar los 30 modelos en un subconjunto de 15 personas de la base de datos Ninapro DB1 (sujetos sin amputaciones). La precisión del conjunto de prueba se registró para cada uno de los experimentos. Para evaluar el impacto de cada hiperparámetro en la precisión de las RNCs se realizó un análisis de varianza (ANOVA). Los resultados del ANOVA se muestran en la tabla 5.1

Usando un valor de significancia  $\alpha = 0,05$  se determinó que los factores con mayor impacto son el tipo de convolución en primer lugar, después el número de capas paralelas adicionales y finalmente el número de bloques secuenciales. El tipo de conexión no es estadísticamente significativo ( $p > 0,05$ ), mientras que el tipo de activación sí lo es, sin embargo, se realizó una prueba de Tukey para comparar las diferencias entre niveles y se determinó que no existe una tendencia clara para la función de activación. en la tabla 5.2 se muestran los resultados de la prueba de Tukey.

Factor	SS	GdL	Estadístico $F$	$p$ -valor
Activacion	0.0139	5	3.97	$1,58 \times 10^{-3}$
Conexión	0.0045	3	2.14	$9,41 \times 10^{-2}$
Convolución	0.1470	2	104.78	$6,74 \times 10^{-38}$
Paralelas	0.1169	1	166.70	$1,56 \times 10^{-32}$
Secuenciales	0.0365	1	52.01	$2,45 \times 10^{-12}$
Residual	0.3065	437		

Tabla 5.1: Análisis de varianza realizado sobre los resultados obtenidos con la base Ninapro DB1. El nivel de significancia usado fue  $\alpha = 0,05$ . La suma de cuadrados es denotada por SS.

Factor	Grupo 1	Grupo 2	Diferencia media	IC 95% inferior	IC 95% superior	$p$ -valor
<b>Convolución</b>	Dilatada	Estándar	0.0351	0.0261	0.044	0.0
	Dilatada	Sep. Prof.	-0.0128	-0.0215	-0.004	0.0018
	Estándar	Sep. Prof.	-0.0478	-0.057	-0.0387	0.0
<b>Capas paralelas</b>	0	1	0.0167	0.005	0.0283	0.0014
	0	2	0.0375	0.0259	0.0492	0.0
	0	3	0.045	0.0338	0.0563	0.0
	1	2	0.0209	0.0089	0.0329	0.0001
	1	3	0.0172	0.0103	0.0241	0.0
<b>Capas sec.</b>	2	3	0.0172	0.0103	0.0241	0.0

Tabla 5.2: Resultados de la prueba de Tukey para los factores con mayor impacto.

Este análisis indica que las convoluciones dilatada y estándar son superiores a la convolución separable en profundidad en este contexto. Además se encontró una diferencia significativa entre las dos primeras. Por otro lado, se determinó que todos los modelos con una o ninguna capa paralela adicional son significativamente diferentes a aquellas arquitecturas que sí las emplean. Por último, se determinó que existe una diferencia estadísticamente significativa entre 2 y 3 capas secuenciales. Esta prueba no encontró diferencias significativas entre los niveles de activación, a pesar de que el análisis de varianza muestre que sí.

Para complementar este análisis, en la tabla 5.3 se presentan los 8 modelos más eficientes para esta base de datos junto con la precisión media (para los 15 sujetos) y su desviación estándar.

Se observa que la convolución dominante en este caso es la estándar, sin embargo se aprecia una configuración con convolución separable en profundidad cuyo alto rendimiento podría atribuirse al número de capas paralelas y secuenciales (3 en ambos casos). Por otro lado, se observa que la mayoría de los modelos tienen 2 o 3 capas paralelas adicionales, confirmando que al usarlas aumenta la eficiencia del modelo. Finalmente, se observa que el número dominante de capas secuenciales es de 3, aunque también se muestran modelos menos complejos que ofrecen una alta precisión de clasificación.

En la figura 5.1 se presenta con mayor detalle la precisión obtenida por los modelos de la tabla 5.3 para cada sujeto. En este caso, son los primeros tres modelos los que alcanzan precisiones superiores al 90% para todos los sujetos, sin embargo, todas las arquitecturas sufren una caída en la precisión con los datos del sujeto 15, esto se atribuye a factores físicos y fisiológicos. Para

Los 8 modelos más eficientes (Ninapro DB1)							
Activación	Conexión	Convolución	Par.	Sec.	$\mu$	$\pm\sigma$	Params.
ReLU	Residuales	Estándar	3	3	0.9709	0.0142	880,914
Mish	Directas	Estándar	3	3	0.9629	0.0144	847,634
LReLU	Directas	Estándar	2	2	0.9609	0.0147	255,026
Swish	Highway	Estándar	3	2	0.9445	0.0228	133,426
ELU	Densas	Estándar	2	2	0.9445	0.0233	261,170
GELU	Densas	Estándar	0	3	0.9426	0.0210	99,346
GELU	Densas	Sep. Prof	3	3	0.9423	0.0185	129,398
LReLU	Highway	Estándar	0	3	0.9419	0.0228	71,826

Tabla 5.3: Precisión media y desviación estándar de los ocho modelos más eficientes. Los datos mostrados corresponde a 15 sujetos de la base Ninapro DB1.

observar la distribución de las precisiones, en la figura 5.2 se presenta el diagrama de cajas de los 8 modelos más eficientes. Los primeros 3 modelos presentan la menor desviación, y se observa con claridad el punto atípico correspondiente al sujeto mencionado.

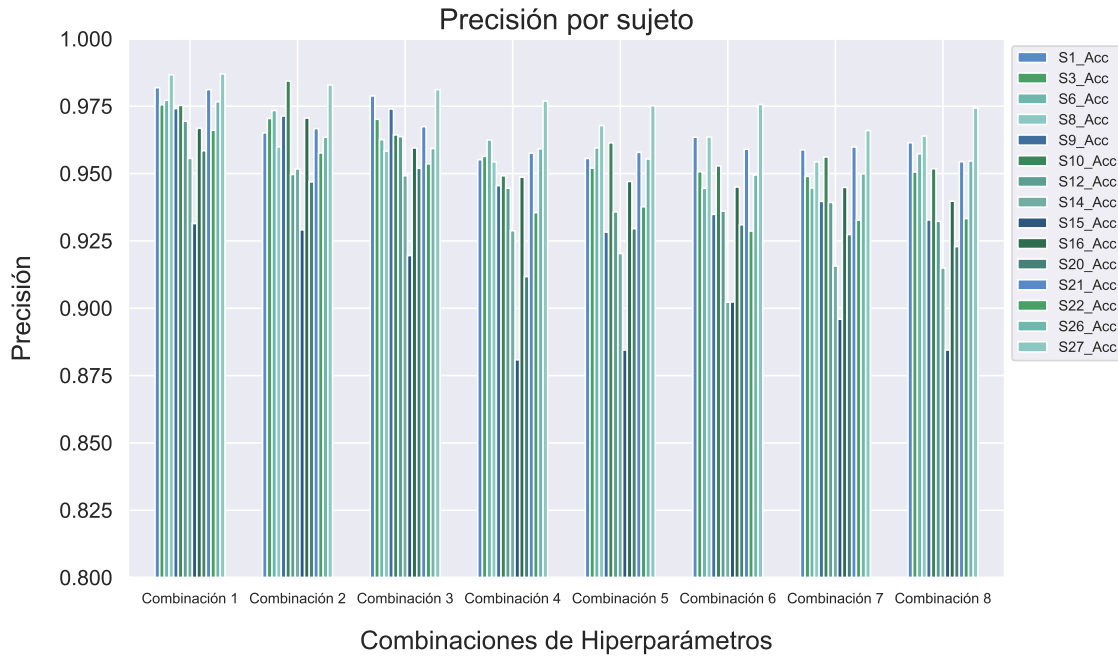


Figura 5.1: Precisiones obtenidas con los 8 modelos más eficientes para cada sujeto del subconjunto Ninapro DB1.

Por otra parte, en la tabla 5.4 se presentan los 8 modelos con la menor eficiencia. Los datos muestran que la convolución separable proporciona el menor rendimiento para esta tarea, seguida por la convolución dilatada. Igualmente, un bajo número de capas paralelas adicionales y de bloques secuenciales genera un bajo rendimiento, pues el número de capas dominante corresponde al mínimo para ambos factores. En la figura 5.3 se muestra el diagrama de cajas para los datos registrados por estos modelos. Se registraron precisiones mínimas del 77% aproximadamente, pero se debe

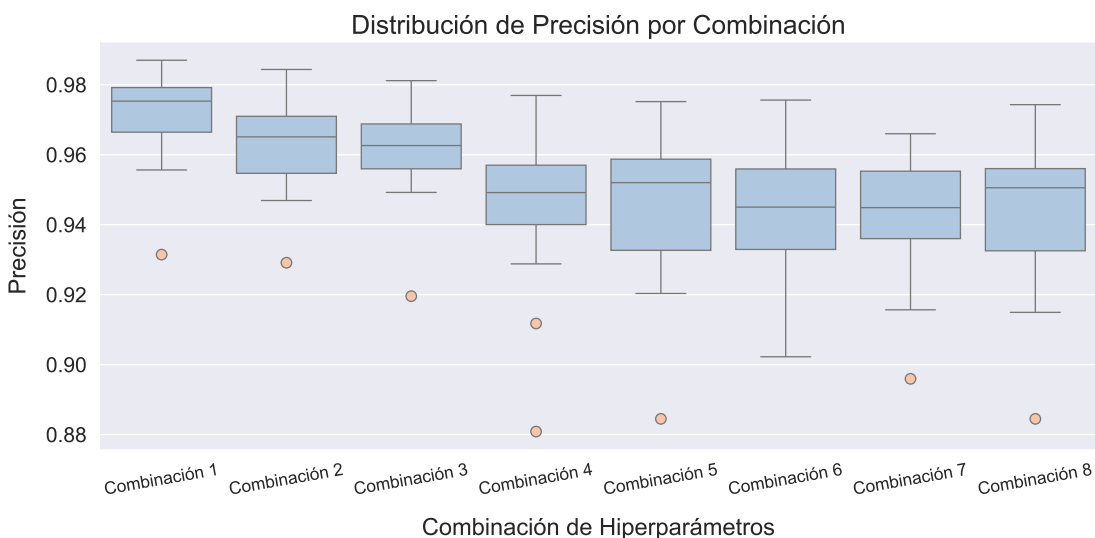


Figura 5.2: Diagrama de cajas de las precisiones obtenidas por los 8 modelos más eficientes evaluados en 15 sujetos de Ninapro DB1.

recordar que son sujetos sin amputaciones.

Los 8 modelos menos eficientes (Ninapro DB1)							
Activación	Conexión	Convolución	Par.	Sec.	$\mu$	$\pm\sigma$	Params.
LReLU	Densas	Sep. Prof	0	2	0.8548	0.0290	21,419
ReLU	Residuales	Sep. Prof	0	3	0.8599	0.0290	21,419
ELU	Directas	Sep. Prof	0	2	0.8603	0.0309	19,371
ReLU	Directas	Sep. Prof	1	2	0.8677	0.0247	27,268
Mish	Residuales	Dilatada	0	2	0.8900	0.0309	43,954
Swish	Densas	Dilatada	0	2	0.8907	0.0299	45,938
Swish	Directas	Dilatada	1	3	0.9022	0.0203	226,578
GELU	Highway	Dilatada	1	2	0.9051	0.0294	75,980

Tabla 5.4: Precisión media y desviación estándar de los ocho modelos menos eficientes. Ambos estadísticos se calcularon a lo largo de los 15 sujetos de Ninapro DB1.

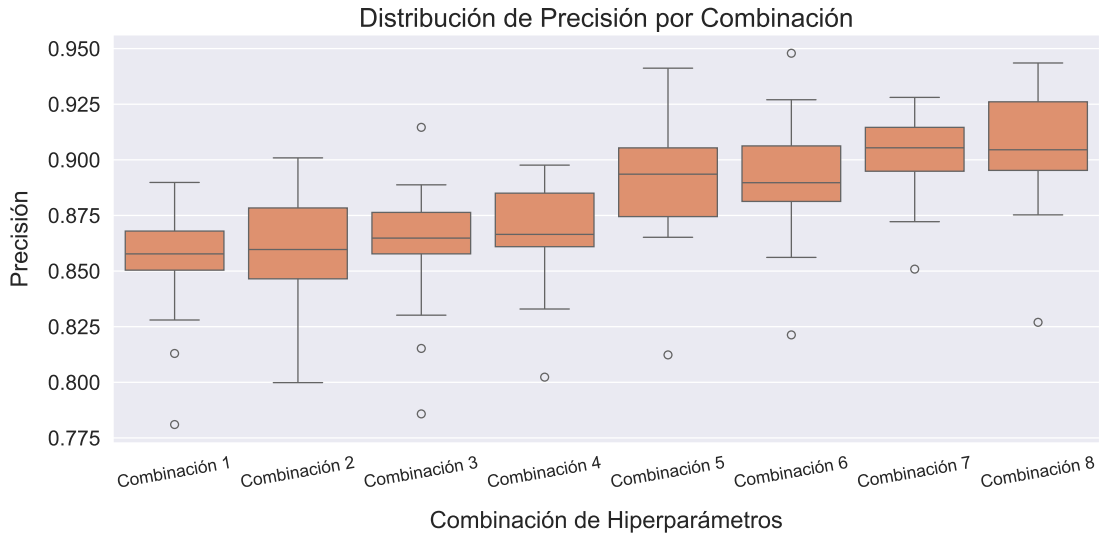


Figura 5.3: Diagrama de cajas que ilustra la distribución de datos para las 8 arquitecturas con menor eficiencia.

### 5.1.2. Base de datos 2

Los resultados presentados en esta sección se obtuvieron al evaluar las 30 arquitecturas en un subconjunto de 15 personas de la base de datos Ninapro DB2 (sujetos sin amputaciones). En cada experimento se registró la precisión obtenida para el conjunto de prueba. en la tabla 5.5 se presenta el análisis de varianza para estimar el efecto de cada hiperparámetro sobre el rendimiento del modelo. Se observa que todos los factores son estadísticamente significativos, sin embargo, los que tienen un mayor impacto son el tipo de convolución, el número de capas paralelas adicionales y finalmente el número de bloques secuenciales.

<b>Factor</b>	<b>SS</b>	<b>GdL</b>	<b>Estadístico <math>F</math></b>	<b><math>p</math>-valor</b>
Activacion	0.0738	5	9.74	$7,78 \times 10^{-9}$
Conexión	0.0644	3	14.18	$7,84 \times 10^{-9}$
Convolución	0.7377	2	243.44	$9,10 \times 10^{-72}$
Paralelas	0.3750	1	247.48	$1,66 \times 10^{-44}$
Secuenciales	0.1419	1	93.66	$3,36 \times 10^{-20}$
Residual	0.6621	437		

Tabla 5.5: Análisis de varianza realizado sobre los resultados obtenidos con la base Ninapro DB2. El nivel de significancia usado fue  $\alpha = 0,05$ . La suma de cuadrados es denotada por SS.

Para determinar las diferencias entre los grupos de los factores con mayor impacto se realizó la prueba de Tukey, cuyos resultados se muestran en la tabla 5.6. De acuerdo a la información de esta prueba, la convolución separable en profundidad tuvo un menor rendimiento comparado con la convolución dilatada y estándar, mientras que no hubo una diferencia significativa entre las dos últimas ( $p = 0,38$ ).

Por otra parte, la prueba de Tukey indica que los modelos con 0 capas paralelas adicionales

Factor	Grupo 1	Grupo 2	Diferencia media	IC 95% inferior	IC 95% superior	p-valor
<b>Convolución</b>	Dilatada	Sep. Prof.	-0.0828	-0.0977	-0.068	0.0
	Estándar	Sep. Prof.	-0.0914	-0.107	-0.0758	0.0
<b>Capas paralelas</b>	0	1	0.0525	0.0319	0.0731	0.0
	0	2	0.0699	0.0493	0.0905	0.0
	0	3	0.0939	0.074	0.1138	0.0
	1	3	0.0414	0.0208	0.062	0.0
	2	3	0.024	0.0034	0.0445	0.015
<b>Capas sec.</b>	2	3	0.0375	0.0251	0.0498	0.0

Tabla 5.6: Resultados de la prueba de Tukey para los factores con mayor impacto.

son significativamente diferentes a las arquitecturas que sí las emplean. De la misma forma, se determinó una diferencia entre los modelos con 1 y 3 capas en paralelo adicionales ( $p = 0$ ), y con 2 y 3 capas adicionales ( $p = 0,015$ ).

Cabe señalar que también existe una diferencia significativa entre 2 y 3 capas secuenciales como se puede ver en la última fila de la tabla (5.6). La prueba de Tukey no encontró diferencias entre grupos para el tipo de conexión y activación.

La precisión se promedió a lo largo de los sujetos para cada modelos, calculando también su desviación estándar. De este modo, se seleccionaron las 8 arquitecturas con mayor precisión media y las 8 menos eficientes de acuerdo a este criterio. en la tabla 5.7 se muestran las combinaciones de hiperparámetros correspondientes a los ocho mejores modelos junto a su precisión media normalizada y desviación estándar.

Los 8 modelos más eficientes (Ninapro DB2)							
Activacion	Conexión	Convolución	Par.	Sec.	$\mu$	$\pm\sigma$	Params.
Mish	Directas	Estándar	3	3	0.9902	0.0025	847,634
LReLU	Directas	Estándar	2	2	0.9831	0.0040	255,026
Mish	Densas	Dilatada	1	3	0.9770	0.0074	337,170
ReLU	Residuales	Estándar	3	3	0.9769	0.0115	880,914
LReLU	Residuales	Dilatada	3	2	0.9725	0.0073	437,650
GELU	Residuales	Dilatada	2	3	0.9696	0.0177	504,722
Swish	Directas	Dilatada	1	3	0.9682	0.0087	226,578
ELU	Highway	Dilatada	3	3	0.9662	0.0081	241,650

Tabla 5.7: Precisión media y desviación estándar de los ocho modelos más eficientes. Los datos mostrados corresponde a 15 sujetos de la base Ninapro DB2.

Se observa que la convolución estándar y dilatada son las que ofrecen un mayor rendimiento, justo como se determinó con la prueba post hoc. En cuanto al número de capas en paralelo adicionales y el número de capas secuenciales, el valor dominante es de tres, aunque se muestran arquitecturas con menor complejidad. Por otro lado, no se observa ninguna tendencia para la función de activación ni el tipo de conexión.

En la figura 5.4 se muestra la precisión que obtuvieron los modelos de la tabla 5.7 para cada

sujeto, la  $i$ -ésima combinación corresponde a a la  $i$ -ésima fila de la tabla.

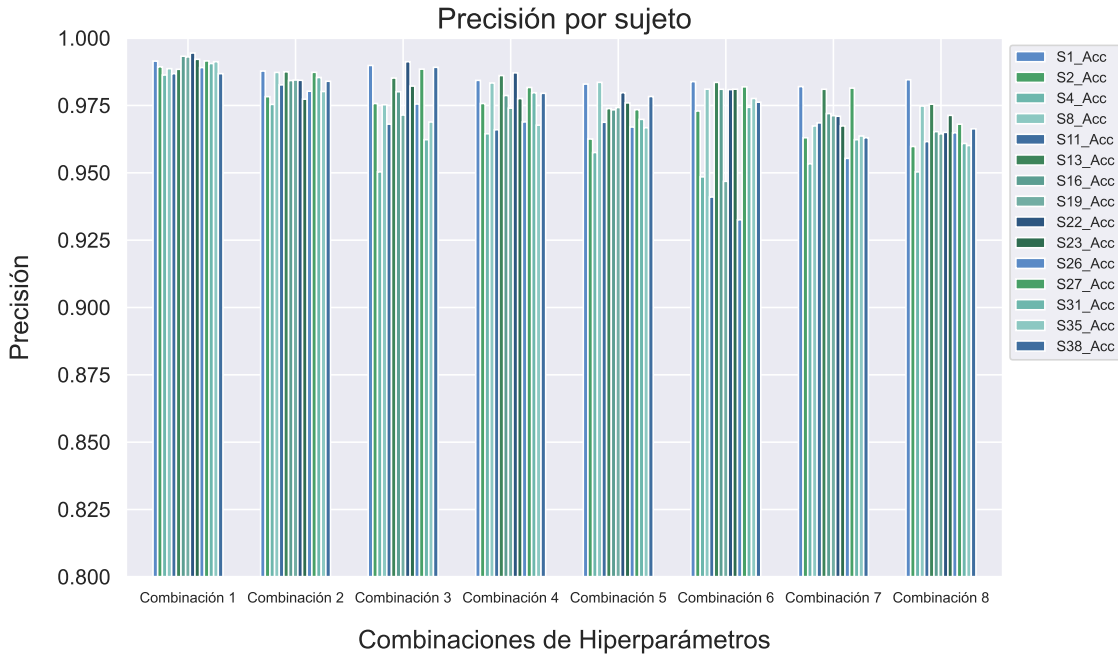


Figura 5.4: Precisiones para cada sujeto obtenidas con los modelos de la tabla 5.7.

Los primeros dos modelos son los más robustos, ya que presentan menor variabilidad en la precisión entre sujetos, además, el segundo modelo solo necesita de dos bloques secuenciales con dos capas paralelas adicionales cada uno. En efecto, la distribución y dispersión de las precisiones para los 8 modelos más eficientes se puede observar en la figura 5.5, donde se muestran gráficos de caja para los resultados obtenidos por cada arquitectura. Se aprecia un punto atípico que corresponde a una alta precisión para el sujeto 01 usando la arquitectura 8, esto podría deberse a los factores fisiológicos del mismo. Por otro lado, los primeros modelos son los que presentan la menor dispersión en general.

Por último, en la tabla 5.8 se presentan las combinaciones de hiperparámetros que generan los modelos con menor eficiencia. Los datos muestran que la convolución separable en profundidad es inferior para esta tarea en específico, justo como se mostró con la prueba de Tukey. De la misma forma, un bajo número de capas paralelas adicionales y de bloques secuenciales es responsable de un bajo rendimiento, pues el número de capas dominante es de 0 y 2, respectivamente. En la figura 5.6 se muestra el diagrama de cajas para los datos registrados por estos modelos. En este caso se tienen precisiones mínimas de 0.60 aproximadamente, sin embargo, los sujetos de esta base son sujetos sin amputaciones.

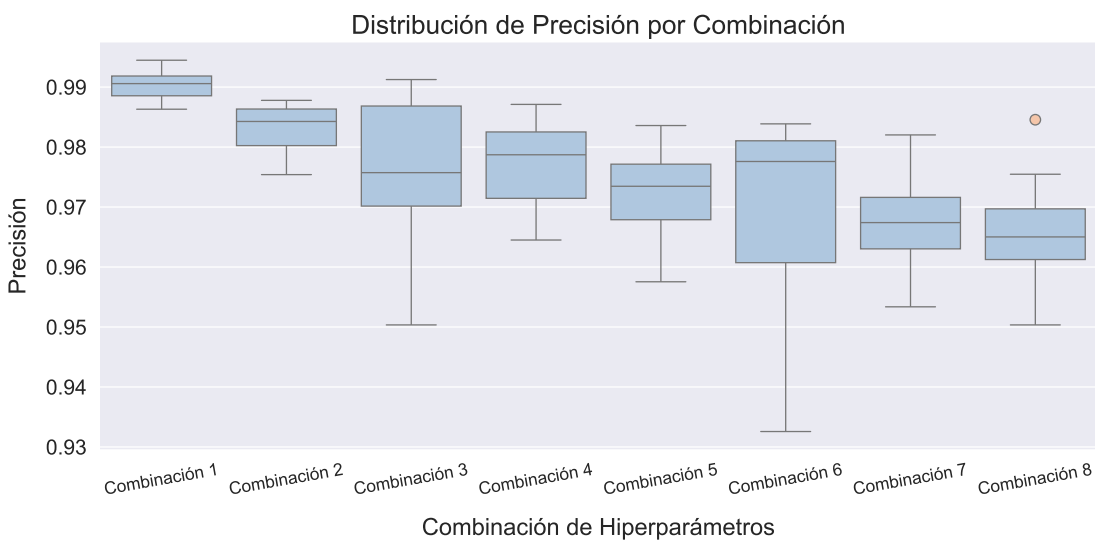


Figura 5.5: Diagrama de cajas de las precisiones obtenidas por los 8 modelos más eficientes.

Los 8 modelos menos eficientes (Ninapro DB2)							
Activación	Conexión	Convolución	Par.	Sec.	$\mu$	$\pm\sigma$	Params.
LReLU	Densas	Sep. Prof	0	2	0.7269	0.0582	21,419
ReLU	Residuales	Sep. Prof	0	3	0.7791	0.0619	21,419
ELU	Directas	Sep. Prof	0	2	0.8128	0.0579	19,371
Mish	Highway	Sep. Prof	2	2	0.8293	0.0441	31,101
LReLU	Highway	Sep. Prof	1	3	0.8533	0.0451	33,956
ReLU	Directas	Sep. Prof	1	2	0.8645	0.0373	27,268
Swish	Densas	Dilatada	0	2	0.8882	0.0326	45,938
Mish	Residuales	Dilatada	0	2	0.8923	0.0346	43,954

Tabla 5.8: Precisión media y desviación estándar de los ocho modelos menos eficientes. Ambos estadísticos se calcularon a lo largo de los 15 sujetos de Ninapro DB2.

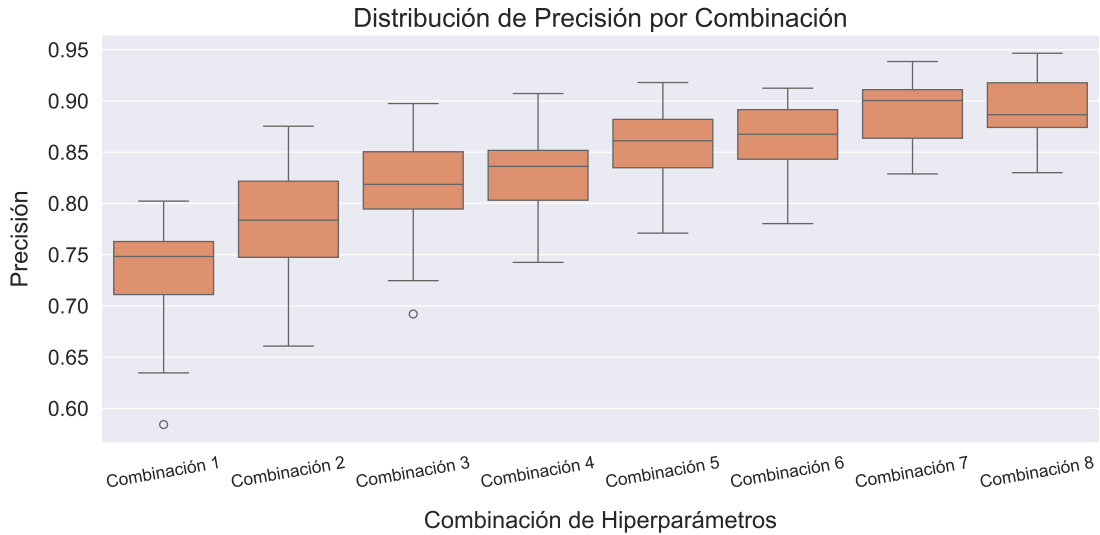


Figura 5.6: Diagrama de cajas que ilustra la distribución de datos para las 8 arquitecturas con menor eficiencia.

### 5.1.3. Base de datos 3

En esta sección se presentan los resultados obtenidos al evaluar los modelos de RNC en la base de datos correspondiente a los sujetos amputados. Se registró la precisión de cada una de las 30 arquitecturas evaluadas de forma independiente en los 11 sujetos, lo que resultó en un total de 330 experimentos para este conjunto de datos. Se detallan los resultados de los 8 modelos con mayor eficiencia en términos de precisión, además de incluir análisis estadísticos realizados sobre las métricas obtenidas en los experimentos. También se exponen los resultados de los 8 modelos con el menor rendimiento, con el objetivo de identificar posibles tendencias o patrones asociados a su desempeño.

Primero se muestra el análisis de varianza en la tabla 5.9 para estimar el efecto de los hiperparámetros sobre la precisión del modelo. Este análisis se realizó usando la precisión registrada de los 330 experimentos.

Factor	SS	GdL	Estadístico $F$	$p$ -valor
Activación	0.0605	5	1.07	$3,75 \times 10^{-1}$
Conexión	0.1184	3	3.50	$1,58 \times 10^{-2}$
Convolución	0.9589	2	42.53	$4,34 \times 10^{-17}$
Paralelas	0.4804	1	42.62	$2,64 \times 10^{-10}$
Secuenciales	0.1980	1	17.56	$3,61 \times 10^{-5}$
Residual	3.5733	317		

Tabla 5.9: Análisis de varianza realizado sobre los resultados obtenidos con la base Ninapro DB3. El nivel de significancia usado fue  $\alpha = 0,05$ . La suma de cuadrados es denotada por SS.

Del análisis de varianza se observa que los principales factores que tiene un impacto significativo (estadísticamente hablando) en la precisión de los modelos son el tipo de **convolución**, el número

de **capas paralelas adicionales** y el número de **capas secuenciales**. El tipo de conexión también es relevante aunque en menor medida en comparación a los factores ya mencionados, y el tipo de activación no tiene ningún impacto.

Para complementar el análisis anterior, también se realizó una prueba de Tukey para determinar las diferencias entre los distintos niveles de los factores con mayor impacto. en la tabla 5.10 se muestran las comparaciones entre niveles para las cuales se encontró una diferencia significativa ( $p < 0,05$ ). La convolución separable en profundidad tuvo un rendimiento menor en comparación a la convolución estándar y dilatada, sin embargo, no hubo diferencia significativa entre éstas dos últimas ( $p = 0,99 > 0,05$ ).

También se puede observar que las arquitecturas con 0 capas en paralelo fueron significativamente diferentes a todas las demás configuraciones ( $p < 0,05$  en todos los casos), al igual que la diferencia entre 1 y 3 capas adicionales en paralelo. Por otro lado, la comparación entre 1 y 2 capas no fue significativa ( $p = 0,85$ ), del mismo modo que la comparación entre 2 y 3 capas ( $p = 0,11$ ).

Por último, las arquitecturas con 2 y 3 capas secuenciales fueron significativamente diferentes ( $p = 0,0003$ ).

En cuanto al tipo de conexión, el ANOVA sugiere un posible impacto del mismo, pero la prueba de Tukey no encontró diferencias específicas entre pares de conexiones ( $p > 0,05$  en todos los casos).

Factor	Grupo 1	Grupo 2	Diferencia media	IC 95% inferior	IC 95% superior	p-valor
Convolución	Dilatada	Sep. Prof.	-0.1142	-0.1508	-0.0776	0.0
	Estándar	Sep. Prof.	-0.1147	-0.1532	-0.0762	0.0
Capas paralelas	0	1	0.0691	0.0203	0.1178	0.0017
	0	2	0.0849	0.0361	0.1336	0.0001
	0	3	0.1276	0.0805	0.1747	0.0
	1	3	0.0585	0.0098	0.1073	0.0112
Capas sec.	2	3	0.0510	0.0235	0.0785	0.0003

Tabla 5.10: Resultados de la prueba de Tukey para los factores con mayor impacto.

Es importante aclarar que la precisión se promedió a lo largo de los sujetos para cada uno de los modelos, reportando también su desviación estándar. De este modo, se seleccionaron las 8 arquitecturas más eficientes y las 8 menos eficientes. En la tabla 5.11 se muestran los hiperparámetros correspondientes a los ocho mejores modelos junto a su precisión media normalizada y desviación estándar, los cuales se calcularon a lo largo de los once sujetos para cada modelo de forma independiente.

Analizando la tabla (5.11) se puede observar que los tipos de convolución predominantes son la convolución estándar y la convolución dilatada, mientras que la mayoría de esos modelos usan 3 capas o bloques secuenciales. También se aprecia que el número de capas paralelas adicionales que predomina en los modelos es de 3, sin embargo, se exhiben arquitecturas con menor complejidad y alta eficiencia como lo es la RNC correspondiente a la combinación 2. Por otra parte, no se presenta una tendencia clara para los factores de activación y conexión.

Con el objetivo de analizar en mayor detalle el desempeño de los modelos presentados, la figura 5.7 muestra las precisiones obtenidas para cada sujeto. La combinación 1 corresponde a la primera fila de la tabla 5.11, la combinación 2 a la segunda fila, y así sucesivamente. Se observa que las combinaciones 1 y 2 generan los modelos más robustos, ya que presentan menor variabilidad en la precisión para todos los sujetos. En contraste, otros modelos experimentan una caída en la precisión con los datos de ciertos sujetos, lo que se atribuye al bajo porcentaje de extremidad

Los 8 modelos más eficientes (Ninapro DB3)							
Activación	Conexión	Convolución	Par.	Sec.	$\mu_{acc}$	$\pm\sigma_{acc}$	Params.
Mish	Directas	Estándar	3	3	0.9862	0.0084	847,634
LReLU	Directas	Estándar	2	2	0.9775	0.0116	255,026
Swish	Directas	Dilatada	1	3	0.9626	0.0191	226,578
GELU	Residuales	Dilatada	2	3	0.9608	0.0234	504,722
Mish	Densas	Dilatada	1	3	0.9608	0.0218	337,170
ReLU	Residuales	Estándar	3	3	0.9557	0.0376	880,914
LReLU	Residuales	Dilatada	3	2	0.9489	0.0405	437,650
ELU	Highway	Dilatada	3	3	0.9457	0.0312	241,650

Tabla 5.11: Precisión media y desviación estándar de los ocho modelos más eficientes. Los datos mostrados corresponde a 15 sujetos de la base Ninapro DB3.

remanente en esas personas.

Para visualizar la distribución de las precisiones obtenidas, la figura 5.8 presenta un gráfico de cajas de los ocho mejores modelos. En este, se pueden identificar puntos atípicos, los cuales corresponden al sujeto 7. De acuerdo con la descripción de la base de datos [8], este sujeto tenía un 0% de antebrazo remanente, lo que explica su comportamiento diferenciado en la clasificación. Aún así, los primeros dos modelos se muestran más robustos ante bajos porcentajes de extremidad restante.

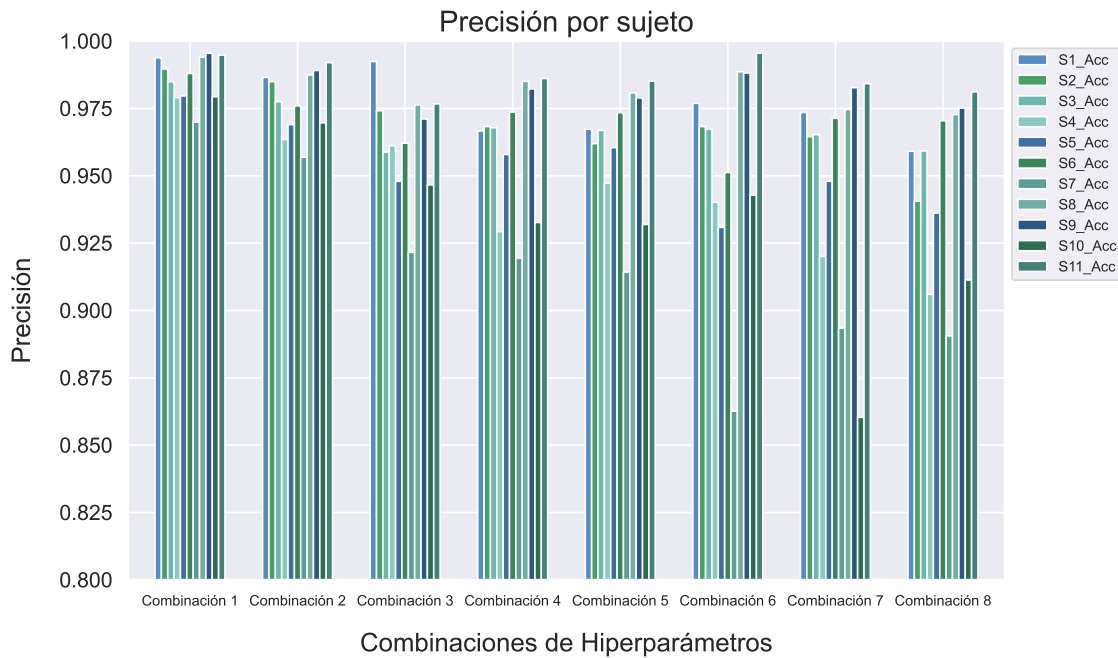


Figura 5.7: Precisiones para cada sujeto obtenidas con los 8 modelos más eficientes. La combinación  $i$  corresponde a la fila  $i$  de la tabla 5.11.

También se analizó la precisión media junto a su desviación estándar de los 8 modelos con

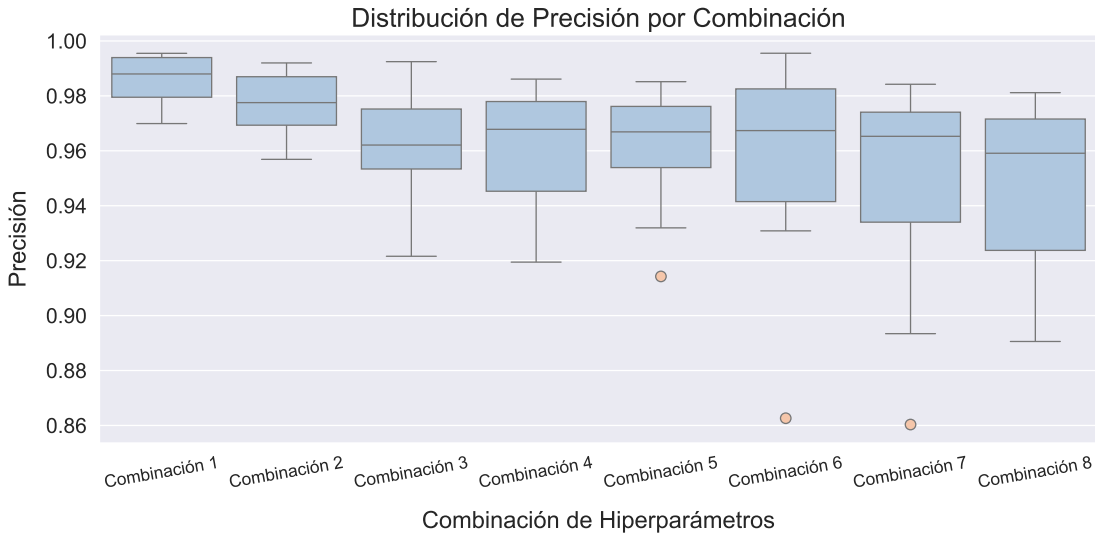


Figura 5.8: Diagrama de cajas que muestra la distribución de las precisiones obtenidas por los 8 modelos más eficientes. El punto atípico corresponde al sujeto 7 con 0% de antebrazo remanente.

menor eficiencia para observar cualquier tendencia existente que presente alguno de los factores. Estos datos se presentan en la tabla 5.12, donde se puede apreciar que el tipo de convolución dominante es la separable en profundidad, mientras que el número de capas paralelas adicionales es mayormente 0 o 1, justo como lo indica el análisis de varianza en conjunto con las pruebas de Tukey. La distribución de los datos por sujeto para estos modelos se presenta en la figura 5.9, donde se observa una mayor dispersión en las precisiones. Incluso se registraron precisiones extremadamente bajas (0,3 – 0,4) para varios modelos.

Los 8 modelos menos eficientes (Ninapro DB3)							
Activacion	Conexión	Convolución	Par.	Sec.	$\mu_{acc}$	$\pm\sigma_{acc}$	Params.
LReLU	Densas	Sep. Prof	0	2	0.6487	0.1908	21,419
ReLU	Residuales	Sep. Prof	0	3	0.7186	0.1784	21,419
ELU	Directas	Sep. Prof	0	2	0.7346	0.1813	19,371
Mish	Highway	Sep. Prof	2	2	0.7635	0.1741	31,101
LReLU	Highway	Sep. Prof	1	3	0.7953	0.1281	33,956
ReLU	Directas	Sep. Prof	1	2	0.8029	0.1360	27,268
Swish	Densas	Dilatada	0	2	0.8261	0.1290	45,938
Mish	Residuales	Dilatada	0	2	0.8368	0.1272	43,954

Tabla 5.12: Precisión media y desviación estándar de los ocho modelos menos eficientes. Ambos estadísticos se calcularon a lo largo de los 11 sujetos de Ninapro DB3.

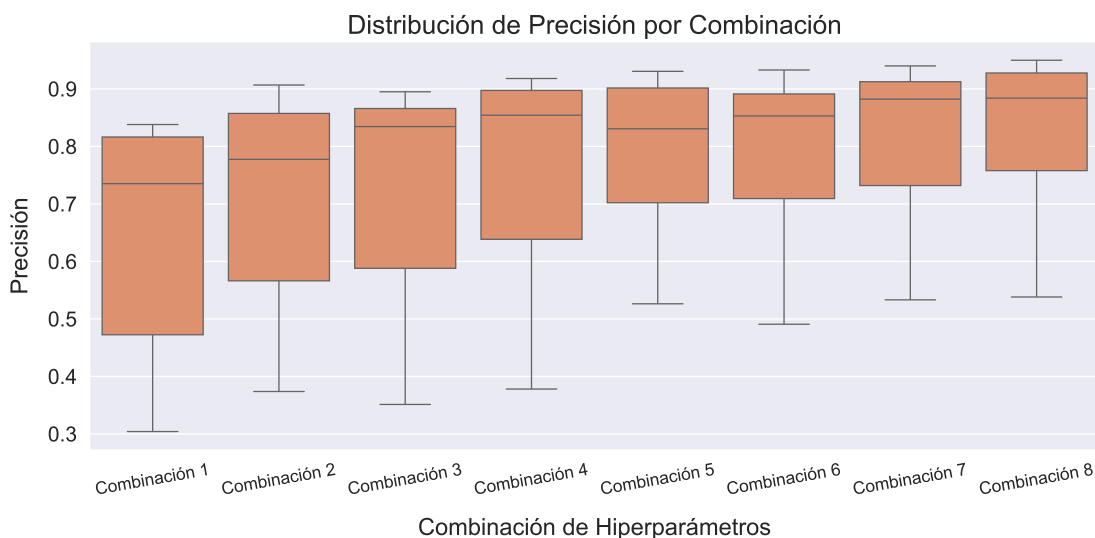


Figura 5.9: Diagrama de cajas que ilustra la distribución de datos para las 8 arquitecturas con menor eficiencia.

## 5.2. Comparación entre bases de datos

En esta sección se presentan las tendencias y diferencias encontradas en los análisis individuales de cada base de datos.

En cuanto al impacto de los hiperparámetros, los análisis de varianza (ANOVA) realizados en la sección anterior indican que el tipo de convolución es el factor más determinante en el rendimiento de los modelos, seguido por el número de capas paralelas adicionales y, en tercer lugar, el número de capas secuenciales. En contraste, la función de activación y el tipo de conexión no mostraron diferencias significativas en ninguna de las bases de datos.

Al analizar en detalle los factores significativos, la prueba de Tukey confirmó una diferencia estadísticamente significativa entre la convolución estándar/dilatada y la convolución separable en profundidad, siendo esta última la que obtuvo el menor rendimiento en las tres bases de datos. Sin embargo, en la base de datos 1 también se identificó una diferencia significativa entre la convolución estándar y la dilatada. En general, la convolución estándar demostró ser la más eficiente en los tres conjuntos de datos, seguida por la convolución dilatada, mientras que la convolución separable en profundidad obtuvo los peores resultados.

El número de capas paralelas adicionales fue el segundo factor con mayor impacto en el rendimiento de los modelos. La prueba estadística reveló una diferencia significativa entre utilizar tres capas paralelas adicionales y emplear solo una o ninguna. Este resultado también se puede corroborar al analizar las tablas de precisión media de los mejores y peores modelos, donde se observa que los modelos con mejor desempeño presentan, en su mayoría, dos o tres capas paralelas adicionales, mientras que aquellos con menor eficiencia suelen tener solo una o ninguna. Esta tendencia se mantiene constante en las tres bases de datos.

De manera similar, el número de capas secuenciales también mostró un impacto significativo en el desempeño. La prueba post-hoc confirmó que existe una diferencia estadísticamente significativa

entre utilizar dos y tres capas secuenciales en los tres conjuntos de datos. Además, las tablas de precisión media de los mejores y peores modelos reflejan que, en general, el uso de tres capas secuenciales conduce a un mejor rendimiento en comparación con solo dos.

Por otro lado, los 8 modelos más eficientes para las bases de datos DB2 y DB3 son los mismos, solo que están en un orden diferente (ver tablas 5.7 y 5.11). Para la base de datos DB1 algunos de esos modelos se mantuvieron entre los 8 mejores, sin embargo sufrieron una ligera caída de precisión, mientras que algunos fueron reemplazados por otros modelos que siguen la tendencia ya mencionada (convolución estándar, 2 o 3 capas en paralelo adicionales y 3 capas secuenciales).

En general, al analizar las tablas de los modelos con mayor y menor eficiencia se observa una relación entre el número de parámetros y el desempeño de los modelos: aquellos con una cantidad reducida de parámetros ( $\approx 19\text{k}$ – $45\text{k}$ ) presentaron de manera consistente los peores resultados en las tres bases de datos (por ejemplo, los modelos 10, 12 y 20). Por el contrario, los modelos con mayor número de parámetros ( $\approx 200\text{k}$ – $880\text{k}$ ), como los modelos 0, 3 y 29, se ubicaron entre los de mejor precisión en todas las bases. Este patrón sugiere que los modelos con muy baja complejidad no cuentan con la capacidad suficiente para generalizar adecuadamente, mientras que los de mayor tamaño ofrecen un mejor potencial de representación. No obstante, también se identificaron excepciones: algunos modelos intermedios ( $\approx 70\text{k}$ – $130\text{k}$ ) alcanzaron buenos resultados, y no todos los modelos grandes garantizaron un desempeño óptimo. Por lo tanto, aunque el número de parámetros constituye un buen indicador general de la capacidad del modelo, su efectividad depende también del diseño específico de la arquitectura.

Los tres modelos en común para las 3 bases de datos se reportan en la tabla 5.13, donde se expone su configuración de hiperparámetros.

Modelos en común para las 3 bases					
Modelo	Activación	Conexión	Convolución	Paralelas	Secuenciales
1	Mish	Directas	Estándar	3	3
2	LReLU	Directas	Estándar	2	2
3	ReLU	Residuales	Estándar	3	3

Tabla 5.13: Hiperparámetros de los 3 modelos comunes para las 3 bases de datos.

Por último, la figura 5.10 muestra la matriz de confusión promediada a lo largo de los sujetos de DB1. En este caso el modelo con la mayor precisión corresponde a la arquitectura de la fila 1 en la tabla 5.3, la cual tiene convolución estándar y el número máximo de capas paralelas y secuenciales. Las precisiones por movimiento varían de 0.9 a 1.0.

Para las bases de datos DB2 y DB3 el modelo con la mayor precisión es el mismo, y corresponde a la primera fila de las tablas 5.7 y 5.11. De nuevo se observa que la convolución es estándar y usa 3 capas paralelas adicionales y 3 bloques secuenciales. La matriz de confusión para DB3 se muestra en la figura 5.11. La precisión por movimiento para DB2 esté entre 0.97 y 1.0, mientras que para DB3 el rango es de 0.96 a 1.0.

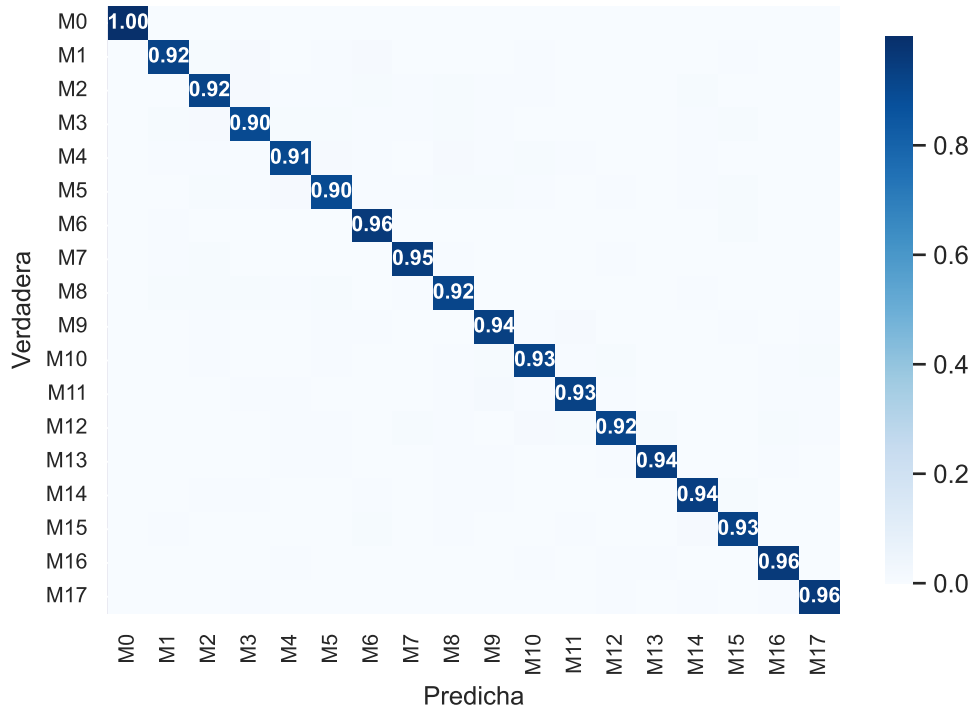


Figura 5.10: Matriz de confusión promedio (a lo largo de los sujetos) del mejor modelo para DB1

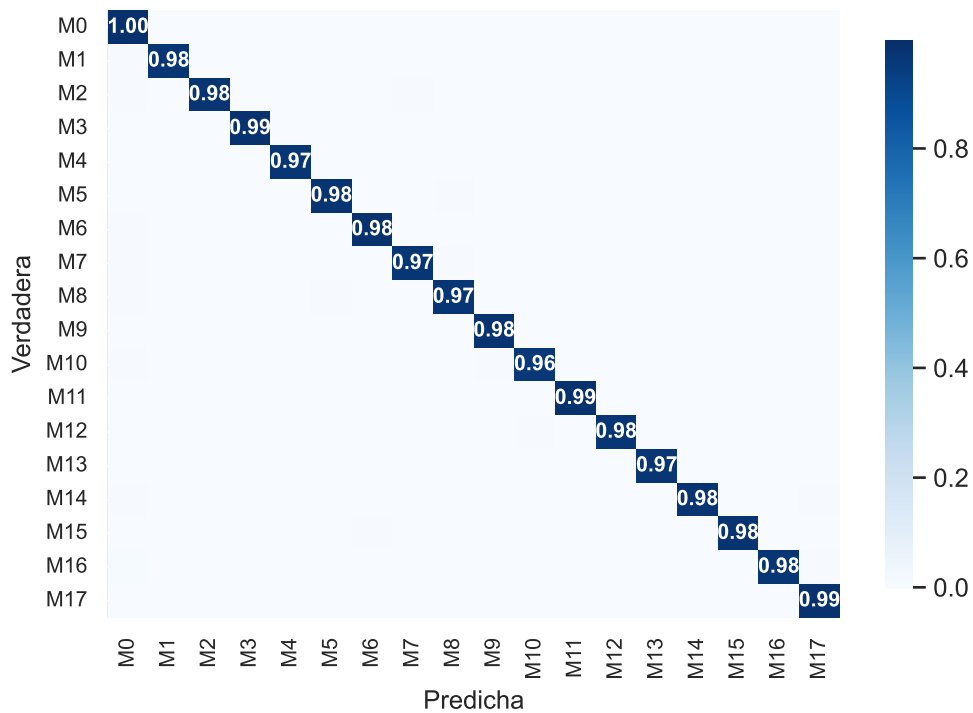


Figura 5.11: Matriz de confusión promedio (a lo largo de los sujetos) del mejor modelo para DB3

### 5.3. Tiempo de inferencia

En esta sección se presentan los resultados de la cuantificación del tiempo de inferencia de los 30 modelos, haciendo énfasis en las 8 arquitecturas con mayor eficiencia (en términos de precisión) y los 8 modelos con menor eficiencia para las bases de datos DB1 y DB3, ya que la forma de los datos de entrada no cambia entre DB2 y DB3. Medir el tiempo de inferencia es fundamental, ya que este estudio se realizó considerando que, en trabajos futuros, los modelos de RNC podrían implementarse en prótesis para aplicaciones reales.

De acuerdo con [36], el tiempo de inferencia puede estimarse mediante la ecuación (5.1):

$$T = \frac{1}{2}W + \frac{1}{2}V + T_e \quad (5.1)$$

donde  $W$  es el tamaño de la ventana utilizada en la segmentación,  $V$  es el solapamiento, y  $T_e$  es el tiempo de respuesta de la RNC, que abarca desde el preprocesamiento de la señal hasta la obtención de la predicción del modelo.

Para este análisis, se consideraron las 30 arquitecturas desarrolladas, con énfasis en los modelos de las tablas 5.3 y 5.4 con respecto a DB1, y los modelos presentados en las tablas 5.11 y 5.12 para DB3, y se midió el tiempo de inferencia utilizando tres procesadores: un **AMD Ryzen 7 5700G** (CPU 1), un **Intel Celeron 1017U** (CPU 2), y una **Raspberry Pi 4 modelo B** (CPU 3).

En la figura 5.12 se presentan los tiempos de inferencia para los 30 modelos evaluados con DB1. En el eje horizontal se encuentra (de forma codificada) la combinación de hiperparámetros correspondiente, donde la primera letra indica la activación, seguida por el tipo de conexión (N para indicar conexiones densas), tipo de convolución, número de capas paralelas y finalmente el número de capas secuenciales. Se observa que con el CPU 1 se obtienen tiempos menores a los 200 ms para todos los modelos, sin embargo, al ser un procesador de alto rendimiento su costo es mayor. Con los otros dos procesadores, especialmente el CPU 3, la mayoría de los modelos tiene tiempos por debajo del límite, lo cual sugiere que es posible usar dispositivos portátiles como la Raspberry para este tipo de aplicaciones.

En la tabla 5.14 se halla el tiempo de inferencia medio para los modelos más eficientes correspondientes a DB1, mientras que en la tabla 5.15 están los tiempos de los modelos con la menor eficiencia en términos de precisión.

Por otra parte, en la figura 5.13 se hallan los tiempos de inferencia obtenidos al evaluar con DB3. Con el CPU 1 algunos modelos tardan hasta más de 300 ms, sin embargo, también hay modelos como el MDE33 que está en el límite y tiene una alta precisión (superior al 95 %). Con

Tiempo medio de los 8 modelos más eficientes para DB1 (ms)								
Activ	Cnx	Conv	Par	Sec	CPU 1	CPU 2	CPU 3	FLOPS
ReLU	Resid	Estd	3	3	182,87 ± 0,14	216,22 ± 10,10	207,68 ± 5,69	3.47 × 10 <sup>8</sup>
Mish	Direct	Estd	3	3	182,00 ± 0,80	198,45 ± 8,01	192,90 ± 2,11	50.6 × 10 <sup>7</sup>
LReLU	Direct	Estd	2	2	181,62 ± 0,49	190,70 ± 5,45	188,51 ± 2,75	24.3 × 10 <sup>7</sup>
Swish	Highway	Estd	3	2	182,29 ± 0,92	196,40 ± 6,80	194,04 ± 5,69	4.89 × 10 <sup>7</sup>
ELU	Densas	Estd	2	2	181,96 ± 0,50	198,50 ± 6,97	193,66 ± 4,08	9.99 × 10 <sup>7</sup>
GELU	Densas	Estd	0	3	181,84 ± 0,52	192,68 ± 8,13	192,60 ± 4,19	3.54 × 10 <sup>7</sup>
GELU	Densas	SP	3	3	185,72 ± 1,68	289,94 ± 22,99	259,82 ± 16,13	4.73 × 10 <sup>7</sup>
LReLU	Highway	Estd	0	3	181,89 ± 0,57	191,45 ± 4,93	191,27 ± 4,23	2.43 × 10 <sup>7</sup>

Tabla 5.14: Precisión media y desviación estándar de los ocho modelos más eficientes. Los datos mostrados corresponde a 15 sujetos de la base Ninapro DB1.

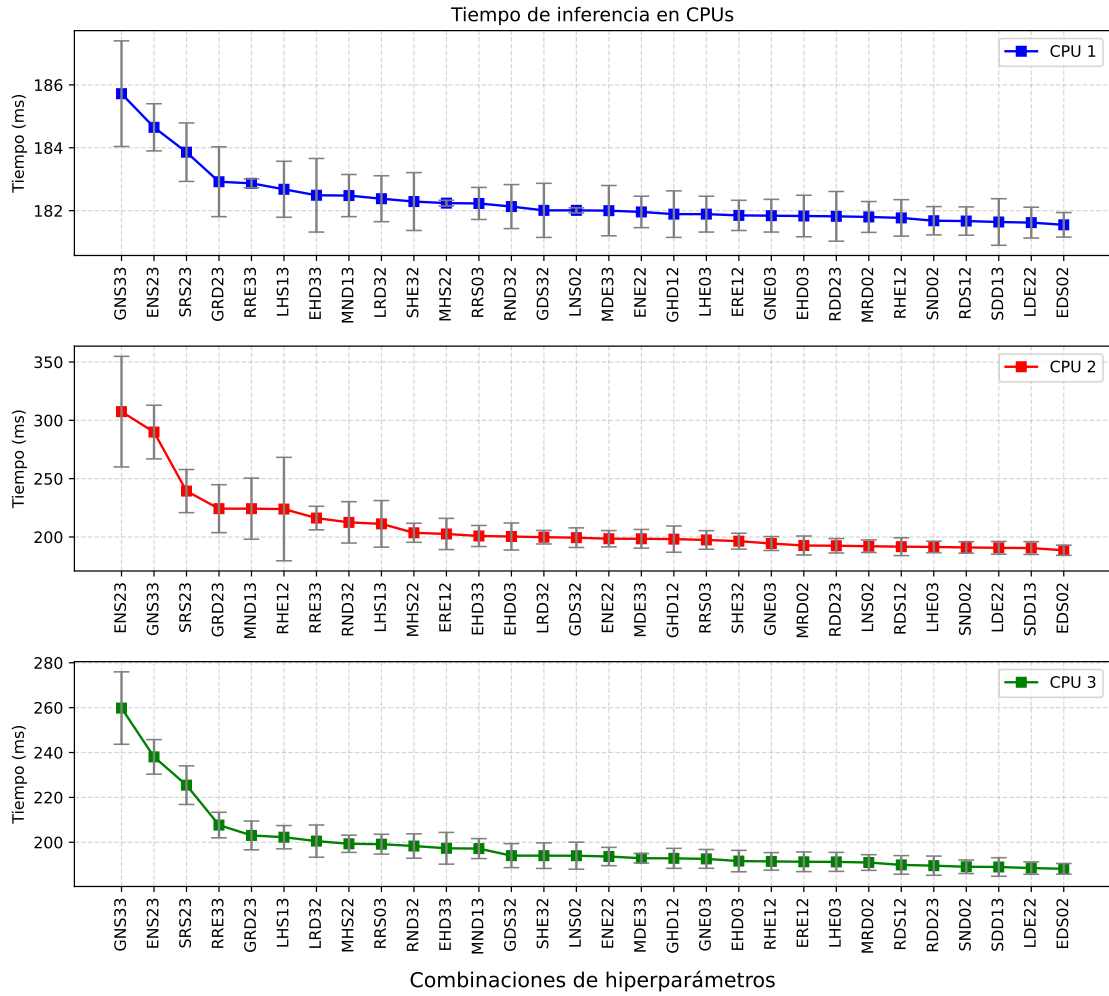


Figura 5.12: Tiempo de inferencia de las 30 arquitecturas evaluadas con DB1. El tiempo se midió con 3 procesadores. En el eje horizontal se indican de forma codificada los hiperparámetros del modelo.

Tiempo medio de los 8 modelos menos eficientes para DB1 (ms)								
Activ	Cnx	Conv	Par	Sec	CPU 1	CPU 2	CPU 3	FLOPS
LReLU	Densas	SP	0	2	182,01 ± 0,08	192,08 ± 5,40	194,02 ± 6,02	4.14×10 <sup>6</sup>
ReLU	Resid	SP	0	3	182,23 ± 0,51	197,45 ± 7,96	199,16 ± 4,42	4.11×10 <sup>6</sup>
ELU	Direct	SP	0	2	181,55 ± 0,39	188,59 ± 4,35	188,18 ± 2,37	3.72×10 <sup>5</sup>
ReLU	Direct	SP	1	2	181,67 ± 0,45	191,69 ± 7,64	189,92 ± 4,15	1.00×10 <sup>6</sup>
Mish	Resid	Dlt	0	2	181,80 ± 0,49	192,68 ± 8,13	190,99 ± 3,49	1.32×10 <sup>7</sup>
Swish	Densas	Dlt	0	2	181,68 ± 0,45	191,04 ± 4,93	189,09 ± 3,02	1.40×10 <sup>7</sup>
Swish	Direct	Dlt	1	3	181,64 ± 0,74	190,50 ± 5,50	189,00 ± 4,14	1.55×10 <sup>7</sup>
GELU	Highway	Dlt	1	2	181,89 ± 0,74	198,17 ± 11,24	192,85 ± 4,45	2.61×10 <sup>7</sup>

Tabla 5.15: Precisión media y desviación estándar de los ocho modelos menos eficientes. Ambos estadísticos se calcularon a lo largo de los 15 sujetos de Ninapro DB1.

**Resultados y discusión**  
5.3 Tiempo de inferencia

Tiempo medio de los 8 modelos más eficientes para DB3 (ms)								
Activ	Cnx	Conv	Par	Sec	CPU 1	CPU 2	CPU 3	FLOPS
Mish	Direct	Estd	3	3	199,13 ± 0,32	696,14 ± 220,87	200,38 ± 0,62	1.26×10 <sup>9</sup>
LReLU	Direct	Estd	2	2	192,85 ± 0,18	321,37 ± 39,52	298,19 ± 17,88	5.84×10 <sup>8</sup>
Swish	Direct	Dlt	1	3	189,93 ± 0,20	294,79 ± 25,03	278,81 ± 10,93	3.46×10 <sup>8</sup>
GELU	Resid	Dlt	2	3	217,48 ± 0,58	1351,38 ± 358,29	2795,29 ± 504,62	4.74×10 <sup>9</sup>
Mish	Densas	Dlt	1	3	211,73 ± 0,53	944,98 ± 90,31	793,47 ± 182,14	3.13×10 <sup>9</sup>
ReLU	Resid	Estd	3	3	224,76 ± 1,31	1629,86 ± 208,40	981,47 ± 68,40	8.34×10 <sup>9</sup>
LReLU	Resid	Dlt	3	2	209,33 ± 0,85	952,70 ± 78,32	628,85 ± 13,70	4.09×10 <sup>9</sup>
ELU	Highway	Dlt	3	3	212,29 ± 0,66	809,79 ± 135,62	594,47 ± 29,46	2.20×10 <sup>9</sup>

Tabla 5.16: Precisión media y desviación estándar de los ocho modelos más eficientes. Ambos estadísticos se calcularon a lo largo de los once sujetos.

Tiempo medio de los 8 modelos menos eficientes para DB3 (ms)								
Activ	Cnx	Conv	Par	Sec	CPU 1	CPU 2	CPU 3	FLOPS
LReLU	Densas	SP	0	2	197,71 ± 0,62	364,71 ± 24,27	359,33 ± 28,14	9.89×10 <sup>7</sup>
ReLU	Resid	SP	0	3	201,20 ± 0,20	487,91 ± 61,31	487,53 ± 82,10	9.81×10 <sup>7</sup>
ELU	Direct	SP	0	2	194,82 ± 0,42	265,12 ± 19,43	300,76 ± 3,20	9.19×10 <sup>6</sup>
Mish	Highway	SP	2	2	212,70 ± 1,02	1073,90 ± 476,68	818,32 ± 160,19	1.90×10 <sup>8</sup>
LReLU	Highway	SP	1	3	212,56 ± 0,93	818,66 ± 146,54	793,47 ± 182,14	2.16×10 <sup>8</sup>
ReLU	Direct	SP	1	2	200,38 ± 0,62	332,30 ± 34,31	395,34 ± 7,87	2.46×10 <sup>7</sup>
Swish	Densas	Dlt	0	2	191,70 ± 0,67	355,10 ± 29,98	323,99 ± 41,93	3.36×10 <sup>8</sup>
Mish	Resid	Dlt	0	2	195,57 ± 0,60	398,60 ± 30,37	373,18 ± 18,96	3.16×10 <sup>8</sup>

Tabla 5.17: Precisión media y desviación estándar de los ocho modelos menos eficientes. Ambos estadísticos se calcularon a lo largo de los once sujetos.

los otros procesadores se obtuvieron tiempos extremadamente altos, pero con Raspberry se logró un tiempo límite de 200 ms aproximadamente con el mejor modelo para este conjunto de datos, reforzando la idea de su posible aplicación en prótesis controladas por señales EMG. en la tabla 5.16 se presenta el tiempo para los mejores modelos de DB3, y en la tabla 5.17 están los tiempos para los modelos con la menor eficiencia de esta misma base de datos.

Es crucial que el tiempo de inferencia sea menor a 200 ms, dado que este es el tiempo de respuesta del cuerpo humano.

Se observa que todos los modelos de DB1 presentan tiempos de inferencia menores a los 200 ms al utilizar el CPU 1. En contraste, con el CPU 2 y CPU 3, únicamente dos RNCs superan significativamente este umbral; ambos modelos comparten la característica de tener el número máximo de capas paralelas y secuenciales, además de contar con conexiones densas y residuales, respectivamente. Cabe destacar que otro modelo con la misma cantidad de capas (3 secuenciales y 3 paralelas) pero con conexiones directas, no supera los 200 ms con el CPU 3. Esto podría ser un indicio de que el tipo de conexión es uno de los factores más influyentes en el tiempo de inferencia. Asimismo, al reducir el número de capas paralelas o secuenciales, no se observó una disminución significativa en el tiempo de inferencia en los modelos de DB1, al menos con el CPU 1. En el caso del CPU 2 y del CPU 3, la diferencia entre los modelos más eficientes y menos eficientes es en general ligeramente mayor. Es importante recordar que los datos de entrada de esta base son matrices de 10×20.

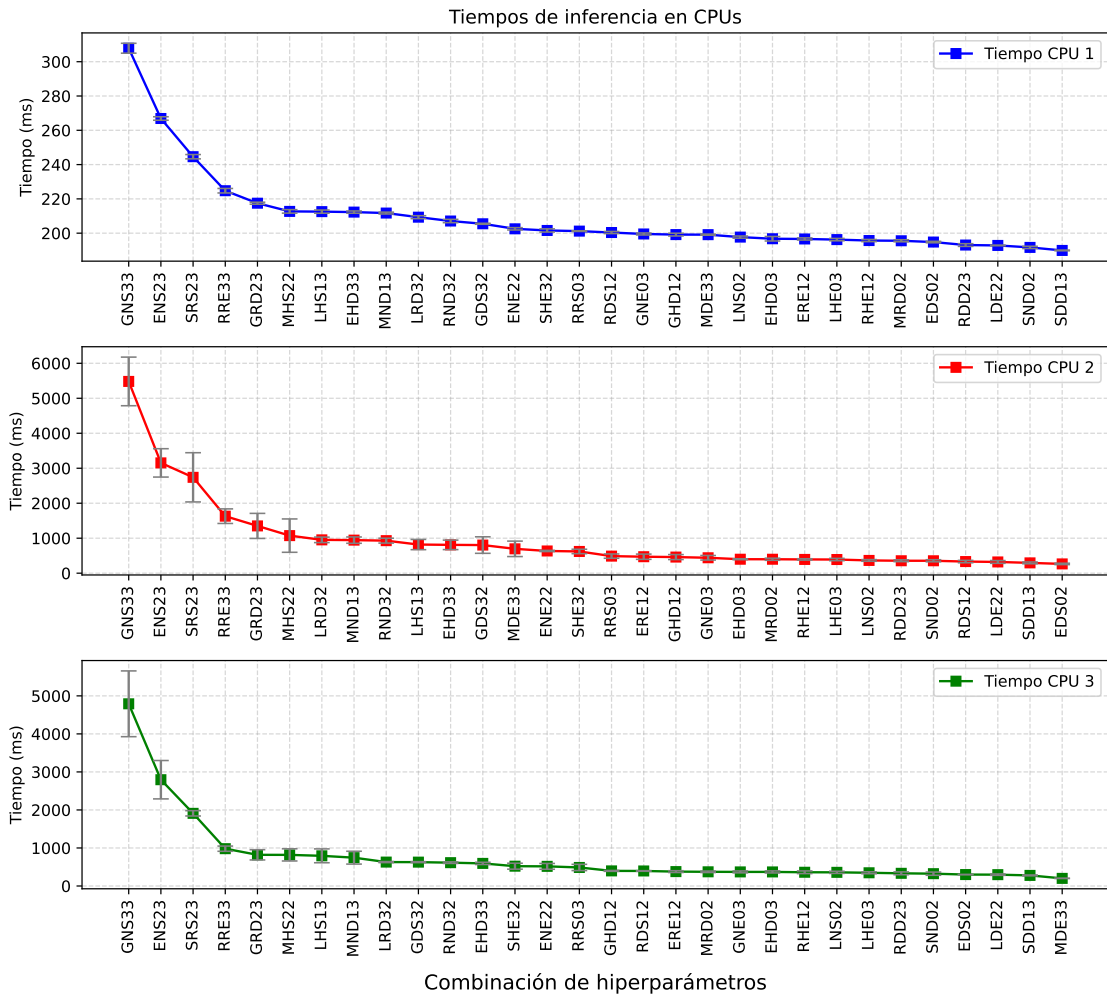


Figura 5.13: Tiempo de inferencia de las 30 arquitecturas evaluadas con DB3. El tiempo se midió con 3 procesadores. En el eje horizontal se indican de forma codificada los hiperparámetros del modelo.

Considerando ahora las RNCs de DB3, en la tabla 5.16 se muestra que solo los tres primeros modelos logran un tiempo de inferencia menor a 200 ms con el CPU 1, y todos ellos utilizan conexiones directas. También se observa que existen arquitecturas con una o dos capas paralelas adicionales, pero con conexiones residuales, densas o highway, que superan el límite de tiempo establecido. Incluso algunos modelos que no incorporan capas paralelas adicionales y que solo cuentan con dos capas secuenciales no cumplen con la condición de tiempo, como también se indica en la tabla 5.17. Esto refuerza la evidencia de que el tipo de conexión es un factor más determinante en el tiempo de inferencia que el número de capas paralelas o secuenciales.

Con el CPU 2, todos los modelos evaluados en DB3 exceden los 200 ms, lo cual indica que este procesador no es adecuado para aplicaciones en tiempo real con esta base de datos. Sin embargo, con la Raspberry se alcanzó un tiempo de  $200,38 \pm 0,62$  para el mejor modelo. Cabe recordar que, para DB3, las entradas tienen una dimensión de  $12 \times 400$ .

Desde esta perspectiva del tiempo de inferencia, es posible aprovechar las ventajas del diseño experimental implementado. En particular, se identificó que la función de activación y el tipo de conexión no generaron diferencias significativas en el desempeño, por lo que es viable optar por conexiones directas y funciones de activación computacionalmente eficientes, con el fin de reducir el tiempo de inferencia sin comprometer la precisión del modelo.

También se registró la cantidad de operaciones de punto flotante (FLOPs) de cada modelo como una medida de su complejidad computacional. Al comparar los modelos más y menos eficientes para ambas bases de datos, se observa que, en general, los de menor eficiencia presentan un menor número de FLOPs. Sin embargo, esta diferencia no se refleja claramente en los tiempos de inferencia con el CPU 1, ya que todos los modelos muestran tiempos del mismo orden de magnitud, a pesar de que algunas RNCs difieren en millones de FLOPs. En el caso del CPU 2, la relación tampoco es consistente o resulta poco significativa. Cabe destacar que las entradas de DB3 son 24 veces más grandes que las de DB1 ( $12 \times 400 / 10 \times 20 = 24$ ), y que, en consecuencia, el número de FLOPs de un mismo modelo al ser evaluado con entradas de DB3 es aproximadamente 24 veces mayor que con entradas de DB1. Esto confirma que el tamaño de entrada tiene un impacto directo sobre la complejidad computacional del modelo, aunque no siempre se traduzca proporcionalmente en el tiempo de inferencia medido.

## Capítulo 6

# Conclusión

Se logró evaluar el impacto del diseño factorial en la optimización de hiperparámetros para mejorar la precisión en la clasificación de señales EMG con redes neuronales convolucionales. Los resultados muestran que el tipo de convolución es el factor con mayor influencia en el desempeño de los modelos, seguido por el número de capas paralelas adicionales y el número de capas secuenciales. En particular, la convolución estándar demostró ser la opción más eficiente, mientras que la convolución separable en profundidad obtuvo el menor rendimiento en todas las bases de datos. Asimismo, se confirmó que el uso de un mayor número de capas paralelas y secuenciales contribuye significativamente a una mejor precisión en la clasificación.

Estos hallazgos indican que la aplicación de un diseño factorial permite identificar configuraciones óptimas de hiperparámetros, confirmando la hipótesis planteada de que esta metodología mejora la precisión de los modelos de RNC. Además, se observó que las tendencias encontradas fueron consistentes entre las tres bases de datos evaluadas, lo que sugiere la robustez de los resultados obtenidos.

Adicionalmente, se realizó un análisis del tiempo de inferencia de las 30 arquitecturas usadas, con énfasis en los 8 modelos con mayor y menor eficiencia utilizando tres procesadores distintos. Los resultados muestran que, con un procesador de alto rendimiento, el tiempo de inferencia cumple con el requisito de ser menor a 200 ms en todos los casos evaluados con DB1, y con algunos modelos evaluados con DB3, especialmente con los 3 más eficientes. Sin embargo, en procesadores de menor capacidad, el tiempo de inferencia aumenta considerablemente, superando en algunos casos 1 segundo. Aunque también se tiene evidencia de que con dispositivos como Raspberry se puede alcanzar un equilibrio entre el tiempo de inferencia y la precisión del modelo. Esto sugiere que la implementación en dispositivos de bajo consumo energético puede ser posible, pero se requerirá optimización adicional para garantizar tiempos de respuesta adecuados en aplicaciones en tiempo real.

Como limitaciones del presente estudio, se considera que la evaluación se centró en arquitecturas específicas de RNC sin explorar otros enfoques como modelos híbridos o técnicas de aprendizaje profundo más avanzadas.

La principal dirección a seguir es profundizar en la búsqueda de la arquitectura de red más adecuada para la clasificación de señales EMG. Esto puede lograrse explorando nuevos diseños de modelos, incorporando diferentes tipos de conexiones y evaluando variaciones estructurales fundamentadas en los hallazgos obtenidos en esta tesis. De manera complementaria, un paso relevante será implementar los modelos en hardware embebido, con el fin de analizar su desempeño en escenarios de tiempo real, tales como el control de prótesis. Finalmente, como mejora metodológica, sería valioso extender el estudio hacia bases de datos más diversas o incluso datos adquiridos en tiempo real, lo que permitiría evaluar con mayor solidez la capacidad de generalización de los modelos.



# Bibliografía

- [1] Victor Alvarado Castillo, Josué Sánchez Flores, Jhon Carlos Gómez, Efraín Chihuan Huayta, and Celso De La Cruz Casaño. Adquisición de señales semg con electrodos secos para el control de movimiento de dedos en una prótesis robótica fabricada en una impresora 3d. *Ingeniare. Revista chilena de ingeniería*, 27(3):522–536, 2019.
- [2] Laith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, Mar 2021.
- [3] Danel Draguljić Angela Dean, Daniel Voss. *Design and Analysis of Experiments*. Springer Cham, 2 edition, 2017.
- [4] Randall Tobias Anthony Atkinson, Alexander Donev. *Optimum Experimental Designs, with SAS*. Oxford Statistical Science Series. Oxford University Press, USA, 2007.
- [5] Jiju Antony. *Design of Experiments for Engineers and Scientists*. Elsevier, 3 edition, 2023.
- [6] Hassan Ashraf, Asim Waris, Syed Omer Gilani, Uzma Shafiq, Javaid Iqbal, Ernest Nlandu Kamavuako, Yaakoub Berrouche, Olivier Brüls, Mohamed Boutayamou, and Imran Khan Niazi. Optimizing the performance of convolutional neural network for enhanced gesture recognition using semg. *Scientific Reports*, 14(1):2020, Jan 2024.
- [7] Ali Raza Asif, Asim Waris, Syed Omer Gilani, Mohsin Jamil, Hassan Ashraf, Muhammad Shafique, and Imran Khan Niazi. Performance evaluation of convolutional neural network for hand gesture recognition using EMG. *Sensors (Basel)*, 20(6):1642, March 2020.
- [8] Manfredo Atzori, Arjan Gijbarts, Claudio Castellini, Barbara Caputo, Anne-Gabrielle Mittaz Hager, Simone Elsig, Giorgio Giatsidis, Franco Bassetto, and Henning Müller. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, 1(1):140053, Dec 2014.
- [9] Marcos Aviles, José Manuel Alvarez-Alvarado, Jose-Billerman Robles-Ocampo, Perla Yazmín Sevilla-Camacho, and Juvenal Rodríguez-Reséndiz. Optimizing rnns for emg signal classification: A novel strategy using grey wolf optimization. *Bioengineering*, 11(1), 2024.
- [10] Marcos Aviles, Juvenal Rodríguez-Reséndiz, and Danjela Ibrahimi. Optimizing emg classification through metaheuristic algorithms. *Technologies*, 11(4), 2023.
- [11] Dianchun Bai, Tie Liu, Xinghua Han, and Hongyu Yi. Application research on optimization algorithm of semg gesture recognition based on light cnn+lstm model. *Cyborg and Bionic Systems*, 2021, 2021.

- [12] A. Bakiya, V. Vetrivel, K. Kamalanand, and A. Anitha. Enhancing emg signal classification using convolution neural network optimized with fractional order bat algorithm. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 16(4):372–383, Dec 2024.
- [13] Lorena Isabel Barona López, Francis M. Ferri, Jonathan Zea, Ángel Leonardo Valdivieso Caraguay, and Marco E. Benalcázar. Cnn-lstm and post-processing for emg-based hand gesture recognition. *Intelligent Systems with Applications*, 22:200352, 2024.
- [14] Jay H. Beder. *Linear Models and Design*. Springer Cham, 2022.
- [15] Katherine Allen-Moyer Bradley Jones and Peter Goos. A-optimal versus d-optimal design of screening experiments. *Journal of Quality Technology*, 53(4):369–382, 2021.
- [16] François Chollet. *Deep Learning with Python*. Manning Publications, 2 edition, 2017.
- [17] Eleanor Criswell. *Cram’s introduction to surface electromyography*. Jones & Bartlett Learning, Massachusetts, 2 edition, 2011.
- [18] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.
- [19] Thippa Reddy Gadekallu, Mamoun Alazab, Rajesh Kaluri, Praveen Kumar Reddy Maddikunta, Sweta Bhattacharya, Kuruva Lakshmana, and Parimala M. Hand gesture classification using a novel cnn-crow search algorithm. *Complex & Intelligent Systems*, 7(4):1855–1868, Aug 2021.
- [20] L. Gila, A. Malanda, I. Rodríguez Carreño, J. Rodríguez Falces, and J. Navallas. Métodos de procesamiento y análisis de señales electromiográficas. *Anales del Sistema Sanitario de Navarra*, 32(Supl. 3):27–43, 2009.
- [21] Alessandro Grattarola, Marta C. Mora, Joaquín Cerdá-Boluda, and José V. García Ortiz. Grasp pattern recognition using surface electromyography signals and bayesian-optimized support vector machines for low-cost hand prostheses. *Applied Sciences*, 15(3), 2025.
- [22] John Edward Hall. *Textbook of Medical Physiology*. Elsevier, Philadelphia, 13 edition, 2016.
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [24] H. Hermens, B. Freriks, D. Merletti, J. Stegeman, G. Blok, C. Rau, C. Disselhorst-Klug, and G. Hägg. *European Recommendations for Surface ElectroMyoGraphy, Results of the SENIAM Project*. Roessingh Research and Development, Enschede, The Netherlands, 1999.
- [25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.
- [26] B. Hudgins, P. Parker, and R.N. Scott. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, 40(1):82–94, 1993.
- [27] Ameya D. Jagtap and George Em Karniadakis. How important are activation functions in regression and classification? a survey, performance comparison, and future directions. *Journal of Machine Learning for Modeling and Computing*, 4(1):21–75, 2023.

- [28] Jun-Gi Jang, Chun Quan, Hyun Dong Lee, and U. Kang. Falcon: lightweight and accurate convolution based on depthwise separable convolution. *Knowledge and Information Systems*, 65(5):2225–2249, May 2023.
- [29] Xiu Kan, Dan Yang, Le Cao, Huisheng Shu, Yuanyuan Li, Wei Yao, and Xiaofeng Zhang. A novel pso-based optimized lightweight convolution neural network for movements recognizing from multichannel surface electromyogram. *Complexity*, 2020(1):6642463, 2020.
- [30] Rami N. Khushaba and Kianoush Nazarpour. Decoding hd-emg signals for myoelectric control - how small can the analysis window size be? *IEEE Robotics and Automation Letters*, 6(4):8569–8574, 2021.
- [31] Peter Konrad. The abc of emg. *A practical introduction to kinesiological electromyography*, 1, 01 2005.
- [32] Peter Konrad. The abc of emg. *A practical introduction to kinesiological electromyography*, 1, 01 2005.
- [33] Moez Krichen. Convolutional neural networks: A survey. *Computers*, 12(8), 2023.
- [34] Frank Kulwa, Oluwarotimi Williams Samuel, Mojisola Grace Asogbon, Olumide Olayinka Obe, and Guanglin Li. Analyzing the impact of varied window hyper-parameters on deep cnn for semg based motion intent classification. In *2022 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT)*, pages 81–86, 2022.
- [35] Frank Kulwa, Haoshi Zhang, Oluwarotimi Williams Samuel, Mojisola Grace Asogbon, Erik Scheme, Rami Khushaba, Alistair A. McEwan, and Guanglin Li. A multidataset characterization of window-based hyperparameters for deep cnn-driven semg pattern recognition. *IEEE Transactions on Human-Machine Systems*, 54(1):131–142, 2024.
- [36] Frank Kulwa, Haoshi Zhang, Oluwarotimi Williams Samuel, Mojisola Grace Asogbon, Erik Scheme, Rami Khushaba, Alistair A. McEwan, and Guanglin Li. A multidataset characterization of window-based hyperparameters for deep cnn-driven semg pattern recognition. *IEEE Transactions on Human-Machine Systems*, 54(1):131–142, 2024.
- [37] Gakuto Kurata, Bhuvana Ramabhadran, George Saon, and Abhinav Sethy. Language modeling with highway lstm. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–251, 2017.
- [38] Iris Kyranou, Sethu Vijayakumar, and Mustafa Suphi Erden. Causes of performance degradation in non-invasive electromyographic pattern recognition in upper limb prostheses. *Frontiers in Neurorobotics*, 12, 2018.
- [39] Christopher Ward Lauralee Sherwood. *Human Physiology From Cells to Systems*. Nelson College Indigenous, 4 edition, 2018.
- [40] Xiu Ting Liu, Sen Lin, and Jun Yang. Optimal trial number for d-optimal designs based on efficiency-cost ratio analysis. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 54–58, 2014.
- [41] Julie K. Silver Lyn D Weiss, Jay M. Weiss. *Easy EMG: A Guide to Performing Nerve Conduction Studies and Electromyography*. Elsevier, 3 edition, 2022.
- [42] V. Mendez, L. Pollina, F. Artoni, and S. Micera. Deep learning with convolutional neural network for proportional control of finger movements from surface emg recordings. In *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 1074–1078, 2021.

- [43] Andrew W. Michell. *Understanding EMG*. Oxford University Press, Oxford, 2013.
- [44] Douglas C. Montgomery. *Design and Analysis of Experiments, 9th Edition*. Wiley, 9 edition, 2017.
- [45] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 1 edition, 2015.
- [46] Leonard C. Onyiah. *Design and Analysis of Experiments (Classical and Regression Approaches with SAS)*. Chapman and Hall/CRC, 1 edition, 2008.
- [47] Tayyip Ozcan and Alper Basturk. Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition. *Neural Computing and Applications*, 31(12):8955–8970, Dec 2019.
- [48] Mehmet Akif Ozdemir, Deniz Hande Kisa, Onan Guren, Aytug Onan, and Aydin Akan. Emg based hand gesture recognition using deep learning. In *2020 Medical Technologies Congress (TIPTEKNO)*, pages 1–4, 2020.
- [49] Elahe Rahimian, Soheil Zabihi, S. Farokh Atashzar, Amir Asif, and Arash Mohammadi. Semg-based hand gesture recognition via dilated convolutional neural networks. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5, 2019.
- [50] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions, 2018.
- [51] H. Romo Romero, J. Realpe Chamorro, and P. Jojoa Gómez. Análisis de señales emg superficiales y su aplicación en control de prótesis de mano. *Universidad Nacional de Colombia - Sede Medellín*, 2007.
- [52] Robert V Schulte, Marijke Zondag, Jaap H Buurke, and Erik C Prinsen. Multi-day EMG-based knee joint torque estimation using hybrid neuromusculoskeletal modelling and convolutional neural networks. *Front. Robot. AI*, 9:869476, April 2022.
- [53] Kasra Sehat, Seyed Mohammadreza Shokouhyan, Nada K. Abdallah, and Kinda Khalaf. Deep network optimization using a genetic algorithm for recognizing hand gestures via emg signals. *Preprints*, January 2023.
- [54] Eren Can Seyrek and Murat Uysal. A comparative analysis of various activation functions and optimizers in a convolutional neural network for hyperspectral image classification. *Multimedia Tools and Applications*, 83(18):53785–53816, May 2024.
- [55] Ronghua Shang, Jianghai He, Jiaming Wang, Kaiming Xu, Licheng Jiao, and Rustam Stolkin. Dense connection and depthwise separable convolution based cnn for polarimetric sar image classification. *Knowledge-Based Systems*, 194:105542, 2020.
- [56] Shafaa Mahmood Shnawa and Firas Sabar Miften. Using a convolutional neural network features to EMG signals classification with continuous wavelet transformation and LS-SVM. *Journal of Education for Pure Science*, 12(2):284–296, 2022.
- [57] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015.
- [58] Tomasz Szandala. *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*, pages 203–224. Springer Singapore, Singapore, 2021.
- [59] Simon Tam, Mounir Boukadoum, Alexandre Campeau-Lecours, and Benoit Gosselin. A fully embedded adaptive real-time hand gesture classifier leveraging HD-sEMG and deep learning. *IEEE Trans. Biomed. Circuits Syst.*, 14(2):232–243, April 2020.

- [60] Simon Tam, Mounir Boukadoum, Alexandre Campeau-Lecours, and Benoit Gosselin. Intuitive real-time control strategy for high-density myoelectric hand prosthesis using deep and transfer learning. *Scientific Reports*, 11(1):11275, May 2021.
- [61] David Tinoco-Varela, Jose Amado Ferrer-Varela, Raúl Dalí Cruz-Morales, and Erick Axel Padilla-García. Design and implementation of a prosthesis system controlled by electromyographic signals means, characterized with artificial neural networks. *Micromachines*, 13(10), 2022.
- [62] Triwiyanto Triwiyanto, I. Putu Alit Pawana, and Mauridhi Hery Purnomo. An improved performance of deep learning based on convolution neural network to classify the hand motion by evaluating hyper parameter. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(7):1678–1688, 2020.
- [63] J. Wagner, Jonghwa Kim, and E. Andre. From physiological signals to emotions: Implementing and comparing selected methods for feature extraction and classification. In *2005 IEEE International Conference on Multimedia and Expo*, pages 940–943, 2005.
- [64] Yingying Wang, Yibin Li, Yong Song, and Xuewen Rong. The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences*, 10(5), 2020.
- [65] Rui Xia, Yumpeng Gao, Yanqing Zhu, Dexi Gu, and Jiangzhao Wang. An attention-based wide and deep cnn with dilated convolutions for detecting electricity theft considering imbalanced data. *Electric Power Systems Research*, 214:108886, 2023.
- [66] Baao Xie, James Meng, Baihua Li, and Andy Harland. Biosignal-based transferable attention bi-convgru deep network for hand-gesture recognition towards online upper-limb prosthesis control. *Computer Methods and Programs in Biomedicine*, 224:106999, 2022.
- [67] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, Aug 2018.
- [68] Aaron J Young, Lauren H Smith, Elliott J Rouse, and Levi J Hargrove. Classification of simultaneous movements using surface EMG pattern recognition. *IEEE Trans. Biomed. Eng.*, 60(5):1250–1258, May 2013.
- [69] Guangjie Yu, Ziting Deng, Zhenchen Bao, Yue Zhang, and Bingwei He. Gesture classification in electromyography signals for real-time prosthetic hand control using a convolutional neural network-enhanced channel attention model. *Bioengineering*, 10(11), 2023.
- [70] Mirza Mumtaz Zahoor, Saddam Hussain Khan, Tahani Jaser Alahmadi, Tariq Alsahfi, Al-noud S. Al Mazroa, Hesham A. Sakr, Saeed Alqahtani, Abdullah Albanyan, and Bader Khalid Alshemaimri. Brain tumor mri classification using a novel deep residual and regional cnn. *Bio-medicines*, 12(7), 2024.
- [71] Xiaohu Zhang, Yuexian Zou, and Wei Shi. Dilated convolution neural network with leakyrelu for environmental sound classification. In *2017 22nd International Conference on Digital Signal Processing (DSP)*, pages 1–5, 2017.