

# Trabajo de Fin de Grado

## Ingeniería de las Tecnologías Industriales

### Eficiencia del Deep Learning para el modelado de la fuerza de reacción durante la marcha

Autor: Carlos Ruiz Mayo

Tutora: Juana María Mayo Núñez

**Dpto. Ingeniería Mecánica y Fabricación**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2021





Trabajo de Fin de Grado  
Ingeniería de las Tecnologías Industriales

# **Eficiencia del Deep Learning para el modelado de la fuerza de reacción durante la marcha**

Autor:  
Carlos Ruiz Mayo

Tutor:  
Juana María Mayo Núñez  
Catedrática de Universidad

Dpto. de Ingeniería Mecánica y Fabricación  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla  
Sevilla, 2021



Trabajo de Fin de Grado: Eficiencia del Deep Learning para el modelado de la fuerza de reacción durante la marcha

Autor: Carlos Ruiz Mayo

Tutora: Juana María Mayo Núñez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2021

El Secretario del Tribunal



*A mi familia*

*A mis amigos*





# Resumen

---

La medición de las fuerzas de reacción con el suelo durante la marcha humana de un paciente es una herramienta esencial para la disminución, prevención y rehabilitación de lesiones. El objetivo de este trabajo ha sido el de evaluar la eficiencia del Deep Learning implementado en Matlab 2021a para el modelado de las fuerzas de reacción con el suelo a partir de mediciones de los ángulos de flexión de la rodilla, del tobillo y de la cadera. Las mediciones fueron proporcionadas por el laboratorio de biomecánica de la Universidad de Sevilla, el cual desarrolló un ensayo basado en el protocolo Plug-in-Gait (PiG), el cual está limitado a la medición en un espacio cerrado sobre placas de fuerza. Los resultados del estudio mostraron resultados superiores para el entrenamiento de redes Long-Short-Term-Memory (LSTM) de varias capas. Por ende, la simulación en Deep Learning basada en estructuras LSTM es una herramienta prometedora para el modelado de la dinámica de la marcha eliminando las limitaciones de laboratorio.



# Abstract

---

The measurement of ground reaction forces during human gait is an essential tool for injury reduction, prevention and rehabilitation. The aim of this project has been to evaluate the efficiency of Deep Learning, implemented in Matlab 2021a, for the modeling of ground reaction forces from measurements of knee, ankle and hip flexion angles. The database was taken from the biomechanics laboratory of the University of Seville, which developed a Plug-in-Gait (PiG) based experiment, which is limited to measurement in an indoor space on force plates. The results of the study showed superior results for the training of multi-layer Long-Short-Term-Memory (LSTM) networks. Thus, Deep Learning simulation based on LSTM structures is a promising tool for modeling gait dynamics by eliminating laboratory limitations.



---

<b>Resumen</b>	<b>9</b>
<b>Abstract</b>	<b>11</b>
<b>Índice</b>	<b>13</b>
<b>Índice de Tablas</b>	<b>15</b>
<b>Índice de Figuras</b>	<b>17</b>
<b>1 Estado del Arte del Deep Learning</b>	<b>1</b>
1.1 <i>Inteligencia Artificial</i>	1
1.1.1 Tipos de IA	1
1.1.2 Ventajas y desventajas de la IA	2
1.2 <i>Machine Learning</i>	2
1.2.1 Métodos de aprendizaje	2
1.3 <i>Redes Neuronales</i>	3
1.3.1 Perceptrón	3
1.3.2 Capa neuronal	4
1.3.3 Mecanismo de Backpropagation	4
1.3.4 Tipos de RN	5
1.4 <i>Deep Learning</i>	6
<b>2 Introducción a la Bioingeniería</b>	<b>9</b>
2.1 <i>Nociones sobre Anatomía</i>	9
2.2 <i>Ciclo de la Marcha</i>	12
2.2.1 Deep Learning en la Bioingeniería	14
<b>3 Modelo y Toma de Datos</b>	<b>11</b>
3.1 <i>Protocolo utilizado</i>	11
3.1.1 Reconstrucción de la pelvis	11
3.1.2 Reconstrucción del fémur	12
3.1.3 Reconstrucción de la tibia	13
3.1.4 Reconstrucción del pie	14
3.2 <i>Toma de datos</i>	14
<b>4 Aplicación del Deep Learning</b>	<b>17</b>
4.1 <i>Clasificación del problema</i>	17
4.2 <i>Preprocesado de datos</i>	18
4.3 <i>Entrenamiento</i>	19
4.3.1 Red LSTM	20
4.3.2 Red Profunda LSTM	21
4.3.3 App 'Neural Net Fitting'	21
<b>5 Resultados</b>	<b>23</b>
5.1 <i>Red LSTM</i>	23
5.2 <i>Red Profunda LSTM</i>	27
5.3 <i>App 'Neural Net Fitting'</i>	29
5.4 <i>Influencia de los datos de entrenamiento</i>	32

<b>6</b>	<b>Conclusión</b>	<b>37</b>
<b>7</b>	<b>Código LSTM</b>	<b>39</b>
<b>8</b>	<b>Código LSTM Profundo</b>	<b>41</b>
<b>9</b>	<b>Código 'Neural Net Fitting'</b>	<b>43</b>
	<b>Referencias</b>	<b>47</b>
	<b>Glosario</b>	<b>51</b>

# ÍNDICE DE TABLAS

---

Tabla 2-1 Ejes de coordenadas de la pelvis.	12
Tabla 2-2 Ejes de coordenadas del fémur.	12
Tabla 2-3 Ejes de coordenadas de la tibia.	14
Tabla 2-4 Ejes de coordenadas del pie.	14
Tabla 2-5 Información de sujetos experimentados.	15





# ÍNDICE DE FIGURAS

---

Figura 1.1 DL dentro de la Inteligencia Artificial [1].	1
Figura 1.1.2 Clasificación de los métodos de aprendizaje más comunes [6].	3
Figura 1.1.3 Evolución de un sistema neuronal [8].	3
Figura 1.1.4 Estructura de un perceptrón simple [9].	4
Figura 1.1.5 Arquitectura de una RN para el reconocimiento facial [11].	4
Figura 1.1.6 Arquitectura de una CNN para el reconocimiento de imágenes [14].	5
Figura 1.7 Arquitectura de una RNN [13].	5
Figura 1.8 Arquitectura de una RBF [15].	6
Figura 1.9 Comparación entre ML y DL para la clasificación de datos [20].	7
Figura 2.1 Posición anatómica [22].	10
Figura 2.2 Planos anatómicos [22].	10
Figura 2.3 Movimientos relativos en el plano sagital [24].	11
Figura 2.4 Movimientos relativos en el plano frontal [23].	11
Figura 2.5 Tipos de pisada durante la marcha [25].	12
Figura 2.6 Esquema de las distintas fases del ciclo de la marcha. [26]	12
Figura 2.7 Esquema de las distintas sub-fases del ciclo de la marcha [27].	13
Figura 3.1 Ubicación de los marcadores de la pelvis.	12
Figura 3.2 Ubicación de los marcadores del fémur, KNE y THI.	13
Figura 3.3 Ubicación del marcador de la tibia, TIB, y del tobillo, ANK	13
Figura 3.4 Ubicación de los marcadores del pie, HEE y TOE.	14
Figura 4.1 Fuerza experimental de reacción con el suelo en dirección longitudinal del sujeto 1, prueba 1.	17
Figura 4.2 Ángulos experimentales de flexión para la cadera, rodilla y tobillo del sujeto 1, prueba 1.	18
Figura 4.3 Ángulos experimentales de flexión tras el pre procesado 1 del sujeto 1, prueba 1.	18
Figura 4.4 Fuerza experimental de reacción con el suelo tras el pre procesado 1 del sujeto 1, prueba 1.	19
Figura 4.5 Fuerza experimental de reacción con el suelo tras el pre procesado 2 del sujeto 1, prueba 1.	19
Figura 4.6 Estructura de una red LSTM para regresión.	20
Figura 4.7 Funcionamiento del descarte cuando vale un 50 % [36].	21
Figura 4.8 Estructura de la red de la aplicación ‘Neural Net Fitting’	22
Figura 5.1 Comparación de los valores de RMSE al variar la tasa de aprendizaje y los <i>Epoch</i> entre 60 y 2000.	24
Figura 5.2 Comparación de los valores de RMSE al variar la tasa de aprendizaje y los <i>Epoch</i> entre 125 y 200.	24
Figura 5.3 Valores de RMSE al variar el número de neuronas ocultas LSTM entre 50 y 300.	25
Figura 5.4 Valores de RMSE al variar el número de neuronas ocultas LSTM entre 100 y 200.	25

Figura 5.5 Valores de RMSE al variar el porcentaje de descarte.	26
Figura 5.6 Comparación de las GRF experimentales y las simuladas mediante la red LSTM.	27
Figura 5.7 Comparación de los valores de RMSE al aumentar el número de capas ocultas LSTM.	28
Figura 5.8 Comparación de los valores de RMSE al variar la tasa de aprendizaje y los <i>Epoch</i> .	28
Figura 5.9 Comparación de las GRF experimentales y las simuladas mediante la red LSTM de 3 capas.	29
Figura 5.10 Valores de RMSE al variar el número de neuronas y el algoritmo de entrenamiento.	30
Figura 5.11 Comparación de las GRF experimentales y las simuladas mediante el algoritmo de SCG.	30
Figura 5.12 Comparación de las GRF experimentales y las simuladas mediante el algoritmo de LM.	31
Figura 5.13 Comparación de las GRF experimentales y las simuladas mediante el algoritmo de BR.	31
Figura 5.14 Variación del RMSE al variar el tamaño del lote de entrenamiento.	32
Figura 5.15 Variación del RMSE al variar el pre procesado del lote de entrenamiento.	33
Figura 5.16 Comparación de las GRF experimentales y las simuladas a partir de los datos sin pre procesar.	34
Figura 5.17 Comparación de GRF experimental y simulado a partir de los datos con el pre procesado 1.	34
Figura 5.18 Evolución del RMSE al variar los ángulos de entrada como datos de entrenamiento.	35
Figura 5.19 Comparación de GRF experimental y simulado a partir del ángulo de flexión del tobillo.	35
Figura 5.20 Comparación de GRF experimental y simulado a partir del ángulo de flexión de la rodilla.	36
Figura 5.21 Comparación de GRF experimental y simulado a partir del ángulo de flexión de la cadera.	36





# 1 ESTADO DEL ARTE DEL DEEP LEARNING

**D**urante este trabajo se estimarán las fuerzas de reacción vertical del pie con el suelo durante el transcurso de la marcha humana con la ayuda de las técnicas de Deep Learning. Para ello, se utilizó los datos proporcionados por Joaquín Ojeda Granja y sus colegas, que midieron dichas fuerzas en un laboratorio tal y como se explicará en el capítulo 3. En el capítulo 2 se introducirá algunos conceptos básicos de la bioingeniería que son necesarios para entender el documento. En el capítulo 4 se desarrollará cómo se ha implementado las técnicas del Deep Learning para la predicción de las fuerzas sin la necesidad de un estudio de laboratorio, a partir de la base de datos proporcionada en el capítulo 3. Finalmente, en el capítulo 5 se comentará la precisión de los resultados obtenidos en el modelado y se compararán las distintas técnicas aplicadas entre ellas en el capítulo 6.

Para comprender que es el Deep Learning primero hay que profundizar en los términos generales que engloban a dicha técnica.

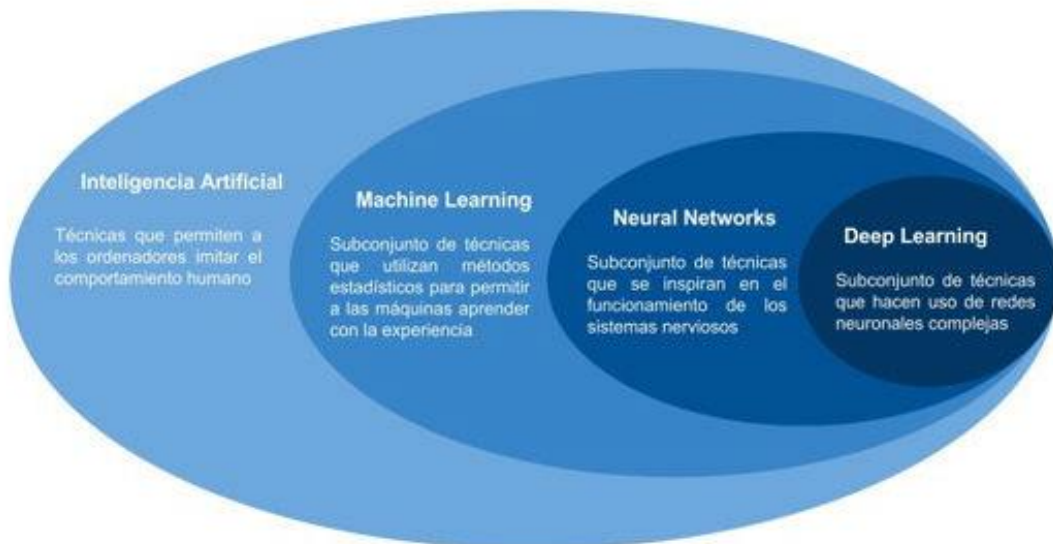


Figura 1.1 DL dentro de la Inteligencia Artificial [1].

## 1.1 Inteligencia Artificial

La inteligencia artificial (IA) es el conjunto de herramientas y técnicas que permiten a una máquina realizar procesos automáticos basándose en el aprendizaje, razonamiento, creatividad y la capacidad de toma de decisiones, simulando el comportamiento de un ser humano [2].

La IA permite a sistemas tecnológicos relacionarse con su entorno y reaccionar a imprevistos, además de resolver problemas, autocorregirse y adaptarse a un fin específico.

### 1.1.1 Tipos de IA

Stuart Russell y Peter Norvig diferencian la IA en los siguientes tipos [3]:

- Sistemas que piensan como humanos: Automatizan actividades como la toma de decisiones, la resolución de problemas y el aprendizaje. Buscan imitar el funcionamiento del sistema nervioso humano mediante redes neuronales.

- Sistemas que actúan como humanos: Se trata de computadoras que realizan tareas de forma similar a como lo hacen las personas y de forma más eficiente. Es el caso de los robots y androides.
- Sistemas que piensan como racionalmente: Tratan de imitar el pensamiento lógico racional humano. Es decir, se trata de investigar cómo lograr que las máquinas perciban, razonen y actúen en consecuencia.
- Sistemas que actúan racionalmente: Aquellos capaces de percibir el entorno, que tratan de imitar de forma racional el comportamiento humano y actuar en consecuencia.

### 1.1.2 Ventajas y desventajas de la IA

Esta técnica se ha popularizado enormemente en las últimas décadas gracias a su capacidad de reducir errores y de aumentar el grado de precisión, la posibilidad de sustituir la mano de obra humana en trabajos difíciles o peligrosos para la salud, la gran aplicación en la vida cotidiana y la disminución de tiempo por su alta velocidad de trabajo y dado que no requieren ni pausas ni descansos [4].

Sin embargo, el debate está a la orden del día, ya que, a pesar de sus claras ventajas, dichas máquinas, además de ser muy costosas y no contar con la capacidad creativa o imaginación, podrían conllevar a un proceso de desempleo a gran escala.

## 1.2 Machine Learning

El Machine Learning (ML) o Aprendizaje Automático es una disciplina del campo de la Inteligencia Artificial que, a través de algoritmos, dota a los ordenadores con la capacidad para identificar patrones en datos masivos para hacer predicciones. Permitiendo al computador todas las características que se han comentado en el apartado de la IA. De esta manera se consigue que las máquinas aprendan automáticamente sin necesidad de ser programadas [5].

Esta técnica nace en 1959, pero no es hasta la aparición del Big Data, el análisis masivo de datos, que esta técnica cobra importancia.

Dependiendo de cómo se produce la implementación de este aprendizaje automático se distinguen varias categorías:

### 1.2.1 Métodos de aprendizaje

- Aprendizaje Supervisado:

Son aquellos modelos que aprenden relaciones entre unas entradas y unas salidas conocidas de antemano. Se subclasifica en modelos de clasificación y de regresión, aquellos que la salida es un valor continuo [6].

- Aprendizaje No Supervisado:

A diferencia del método anterior, no es necesario conocer la salida de los datos de entrenamiento, ya que el objetivo del estudio es conocer estructuralmente los datos disponibles, con el fin de encontrar patrones que permitan ordenarlos de alguna manera (clustering) [6].

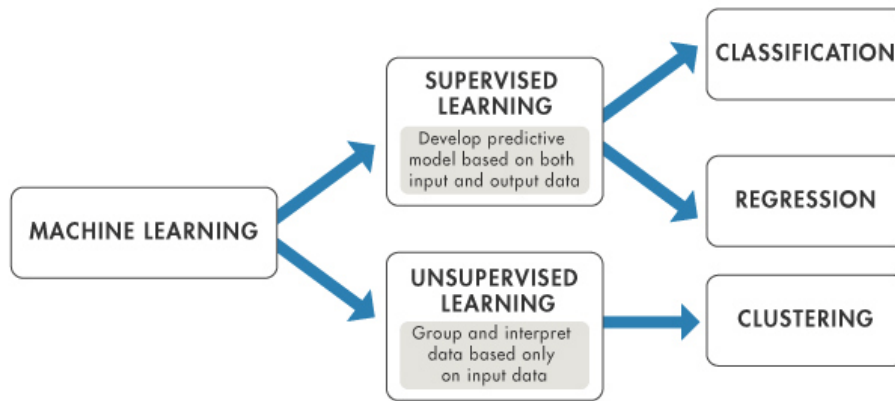


Figura 1.2 Clasificación de los métodos de aprendizaje más comunes [6].

- **Por Refuerzo:**

Es el método menos utilizado de los tres. En este caso el algoritmo aprende en base al proceso prueba y error, recompensando las decisiones correctas. Es decir, mejora la respuesta mediante un mecanismo de retroalimentación [6].

### 1.3 Redes Neuronales

Las Redes Neuronales (RN) o Neural Network es un tipo de ML que busca simular la forma en la que las neuronas actúan en el momento de toma de decisiones en un ser humano. Se basa en la estructura del perceptrón de Rosenblatt y en el mecanismo de backpropagation de Rummelhart, que permite a la neurona aprender, generalizar y abstraer características de los datos de entrenamiento [7].

Dicha red se forma por los siguientes elementos más simples:

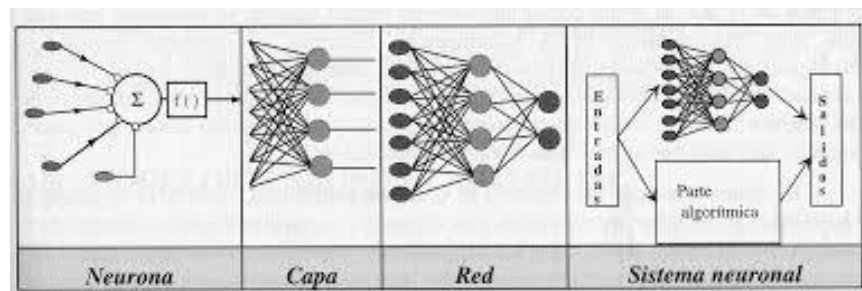


Figura 1.3 Evolución de un sistema neuronal [8].

#### 1.3.1 Perceptrón

Es la red neuronal más básica que existe para el aprendizaje supervisado, también se le conoce como neurona.

Como se puede observar en la figura 1.4, el perceptrón suma todas las entradas, ponderadas con su peso correspondiente, y almacena el resultado en una función de activación que genera la salida.

Para entrenar el sistema se genera un bucle en el cual se establecen los pesos sinápticos y el umbral que consigan que la entrada se ajuste de mejor forma a la salida. Sin embargo, este modelo de perceptrón simple sólo es capaz de adaptarse para funciones lineales [9].

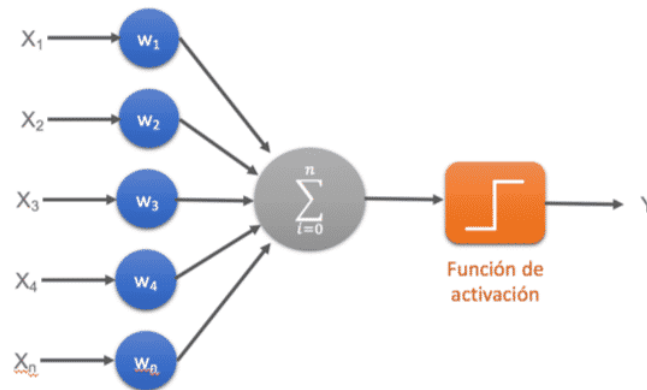


Figura 1.4 Estructura de un perceptrón simple [9].

### 1.3.2 Capa neuronal

Se conoce como capa neuronal a la agrupación de un conjunto de perceptrones que trabajan a la vez. Dichas capas suelen trabajar en sistemas multicapas donde se relacionan los perceptrones de las distintas fases en paralelo, siendo capaz de adaptarse para funciones no lineales (Figura 1.5).

Cuando se habla de capas neuronales se suele distinguir entre las capas de entrada, ocultas y de salida. La capa de entrada es aquella que recibe los datos del exterior, las ocultas no tienen conexión con el exterior, sino que están rodeadas por otras ocultas o las de salida o entrada, y la de salida proporciona la solución de la RN [10].

A mayor número de capas ocultas se dice que la RN es más profunda.

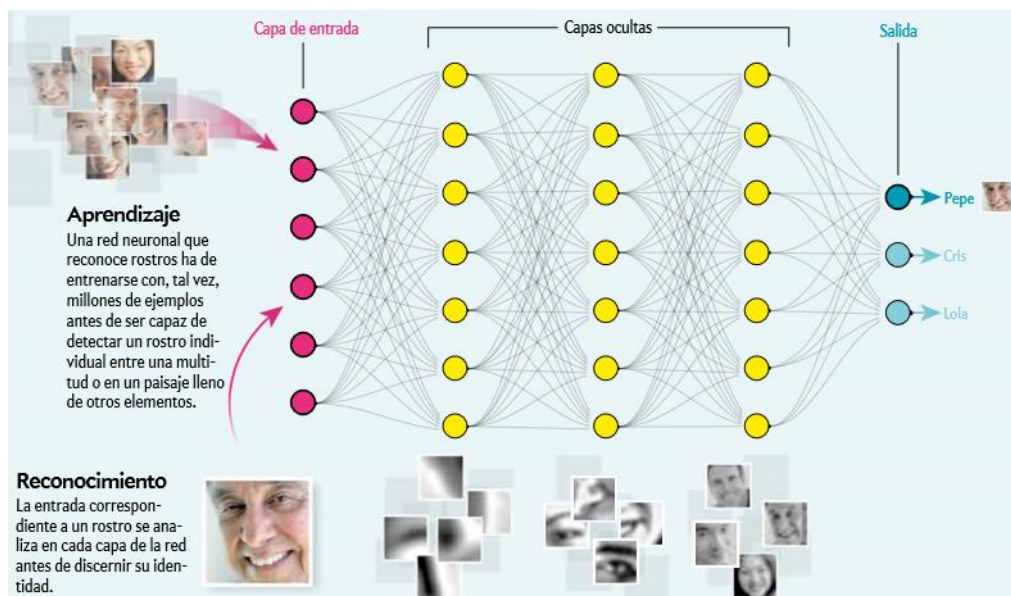


Figura 1.5 Arquitectura de una RN para el reconocimiento facial [11].

### 1.3.3 Mecanismo de Backpropagation

El mecanismo de Rumelhart consiste en un sistema de entrenamiento de RN con aprendizaje supervisado. En este entrenamiento se establecen unos valores aleatorios iniciales de los pesos sinápticos y mediante el uso de la



RN se genera una salida que se compara con el valor real. A partir de dicha comparación, se establece el error obtenido en la salida y se propaga hacia las capas anteriores, realizando pequeñas correcciones en los pesos para minimizar el error. Tras la modificación de los pesos, se obtiene una nueva salida y se repite de mecanismo. Dicho proceso se repite hasta que el error no influya en la toma de decisiones de la RN [12].

Dado que los pesos iniciales se establecen aleatoriamente, es común que al entrenar varias veces la misma RN se obtengan distintos resultados para los pesos sinápticos finales.

### 1.3.4 Tipos de RN

- **RN Convolutiva (CNN):**

Es un caso particular en el que las neuronas no están conectadas con todas las de la capa siguiente, sino que solamente lo hace con algunas de ellas, reduciendo el nivel de complejidad del sistema y el costo computacional. La selección de que perceptrones de las capas ocultas se deben relacionar con cual se establece a partir de la entrada del sistema y del tipo de la capa oculta [13].

Entre las capas ocultas se distinguen aquellas que modifican los datos, con el objetivo de aprender características específicas de ellos. Entre ellas, se encuentran las de convolución, que activan características específicas de los datos, las de unidad lineal rectificada (ReLU), que determinan que características deben continuar a la siguiente capa, y las de pooling, que reducen el número de parámetros que la red necesita aprender [14].

Dadas sus características, es una red principalmente utilizada para la clasificación e identificación a partir de imágenes, como se puede ver en la figura 3.6 para la identificación de un vehículo a partir de una imagen.

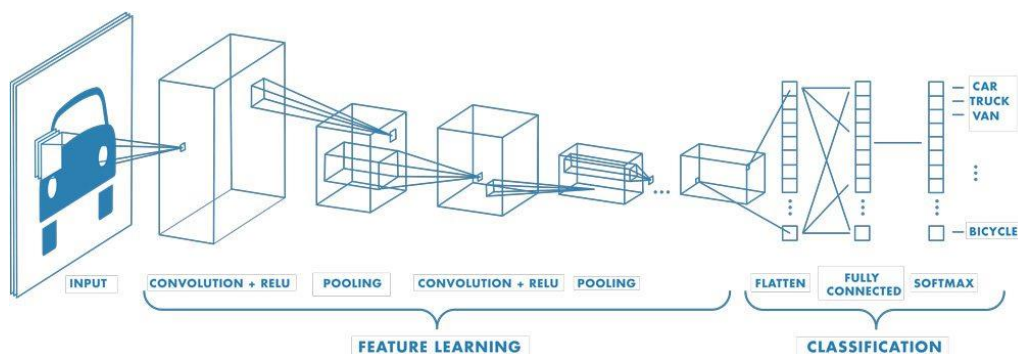


Figura 1.1.6 Arquitectura de una CNN para el reconocimiento de imágenes [14].

- **RN Recurrente (RNN):**

Este tipo de RN no trabaja con capas paralelas, sino que las neuronas se pueden distribuir aleatoriamente, permitiendo la realimentación si fuera necesario. De esta manera se aporta temporalidad al problema y memoria a la red [13].

La más famosa de ellas es la Long-Short Term Memory (LSTM) que permite el aprendizaje a largo plazo.

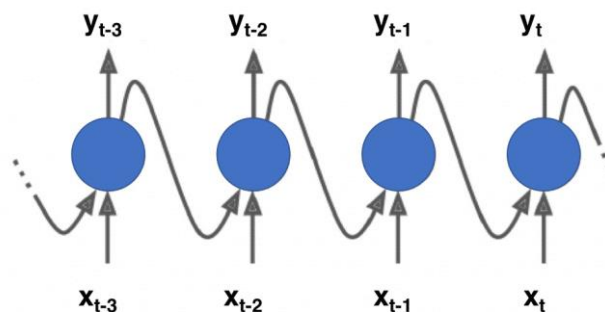


Figura 1.7 Arquitectura de una RNN [13].

- **RN de Base Radial (RBF):**

Estas redes calculan la salida en función de la distancia a un punto denominado centro. La salida es una combinación lineal de las funciones de activación radiales utilizadas por las neuronas individuales.

Las redes de base radial tienen la ventaja de que no presentan mínimos locales donde la retropropagación (*backpropagation*) pueda quedarse bloqueada [13].

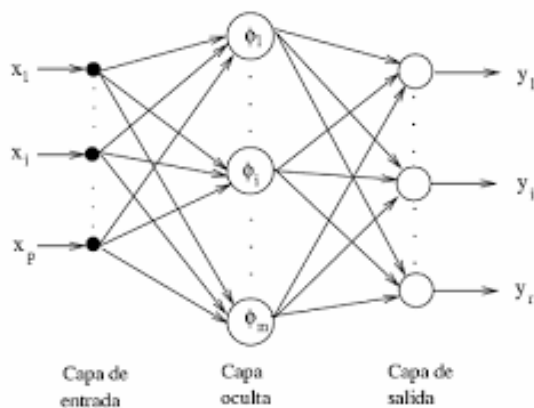


Figura 1.8 Arquitectura de una RBF [15].

## 1.4 Deep Learning

Aunque el nacimiento del Deep Learning se remonta a los años 50 del siglo XX con la aparición de la Inteligencia Artificial, el nombre actual aparece por primera vez en 1974 de mano de Paul Werbos, describiendo el proceso de entrenamiento de una red neuronal a través de la retro propagación. Sin embargo, el auge de esta técnica surge hace poco más de una década con la mejora de los procesos computacionales [16].

Para entender el origen del Deep Learning se pueden establecer tres etapas principales [17]:

- **Etapla Cibernética (1940-1960):** Fue motivada por los estudios sobre el aprendizaje biológico de McCulloch&Pitts (1943) y Hebb (1949) y continuada por el Perceptrón Rosenblatt (1958), que permitía el entrenamiento de una sola neurona.
- **Etapla de Conexionismo (1980-1955):** Surgió el concepto de “*backpropagation*” con Rummelhart et al (1986), técnica que se utiliza en la actualidad para el entrenamiento de redes neuronales, permitiendo calcular los pesos de las neuronas correspondientes a las distintas capas de estas.
- **Etapla como Deep Learning (2006):** Aunque los modelos actuales están basados en el cerebro biológico, no están diseñados para realizar una representación realista. Es decir, se aleja de la perspectiva neurocientífica y busca un mecanismo de aprendizaje a través de múltiples niveles de composición.

El DL o Aprendizaje Profundo es un subcampo dentro de las RN, que se caracteriza por una gran profundidad de red, consiguiendo el aprendizaje de elementos más complejos [18]. Sin embargo, lo que realmente caracteriza al DL es el procesamiento de datos de forma jerárquica, es decir, la utilización del aprendizaje por capas para empezar extrayendo los datos más genéricos e ir particularizando hasta el más específico [19].

Este sistema permite abarcar gran variedad de estudios, desde el procesamiento de imágenes, de audio, de texto o de secuencias temporales.

Además, este subcampo puede dotarse de cualquier tipo de Red Neuronal o método de aprendizaje comentado anteriormente.

En la figura 1.9 se puede observar la diferencia entre el procesamiento de un algoritmo de ML y DL para la identificación de si la fotografía proporcionada es o no un automóvil. El ML actúa como un árbol de decisión, evaluando características concretas que previamente se han extraído de una base de datos. Por otro lado, el DL procesa la imagen mediante capas neuronales extrayendo características de ella y clasificándola.

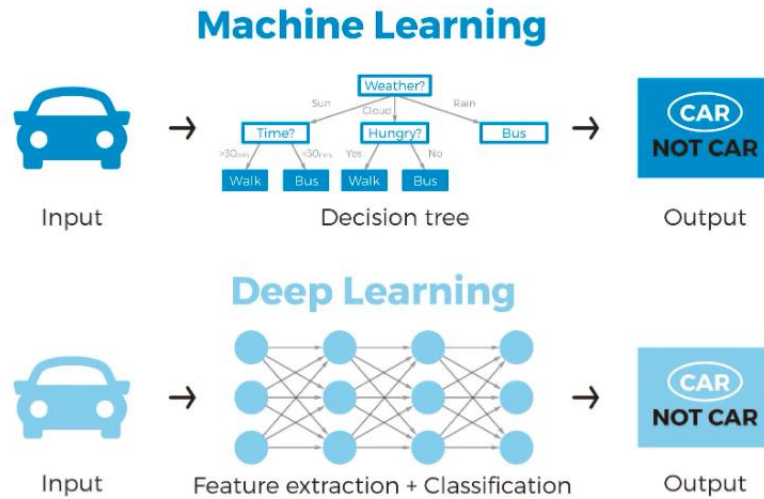


Figura 1.9 Comparación entre ML y DL para la clasificación de datos [20].



## 2 INTRODUCCIÓN A LA BIOINGENIERÍA

La Real Academia Española define la biomecánica como el estudio de la aplicación de las leyes de la mecánica a la estructura y el movimiento de los seres vivos. La mecánica es una rama de la física que se ocupa de la descripción del movimiento y de cómo las fuerzas crean el movimiento. Las fuerzas que actúan sobre el ser vivo pueden ser causantes de lesiones, por lo que la biomecánica proporciona herramientas suficientes y necesarias para entender cómo se producen estos movimientos y como mejorarlos o realizarlos de un modo más seguro para el cuerpo, con el objetivo de reducir o tratar dichas lesiones [21].

Durante este trabajo se estimarán las fuerzas de reacción vertical del pie con el suelo durante el transcurso de la marcha humana con la ayuda de las técnicas de Deep Learning. Pero para entender dicho proceso primero es necesario conocer algunos conceptos básicos de la bioingeniería.

### 2.1 Nociones sobre Anatomía

En primer lugar, es necesario definir los planos anatómicos que dividen el cuerpo humano, para ello se establece una posición de referencia llamada posición anatómica. En esta posición el sujeto se encuentra erguido, con los brazos estirados y las palmas de las manos mirando hacia delante (Figura 2.1). Los planos anatómicos que se definen son los siguientes (Figura 2.2) [22]:

- Plano sagital o lateral: Divide al cuerpo verticalmente en dos partes asimétricas, la parte derecha e izquierda.
- Plano coronal o frontal: Divide al cuerpo verticalmente, pero esta vez divide al cuerpo en la mitad anterior y posterior.
- Plano horizontal, transversal o axial: Divide al cuerpo horizontalmente, en las mitades superior e inferior, estableciendo el centro en la pelvis.

Además, la anatomía también establece una terminología especial para las principales rotaciones de los huesos en las articulaciones. La “flexión” se especifica como el giro de la articulación que provoca una disminución del ángulo articular en el plano sagital, mientras que la “extensión” aumenta ese ángulo, además, se añade el prefijo “hiper” a dados movimientos hasta el extremo. (Figura 2.3) Por otro lado, el movimiento en el plano frontal que aleja un segmento de la línea media se denomina “abducción”, determinando como “aducción” al movimiento que lo acerca [23]. (Figura 2.4)

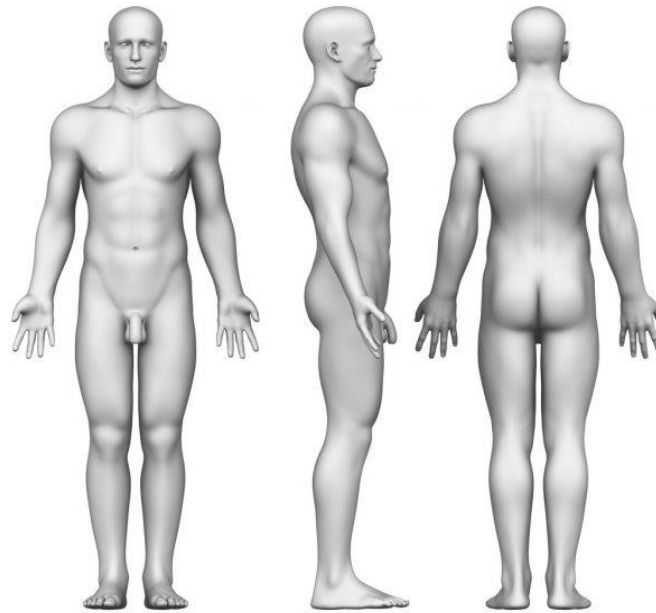


Figura 2.1 Posición anatómica [22].

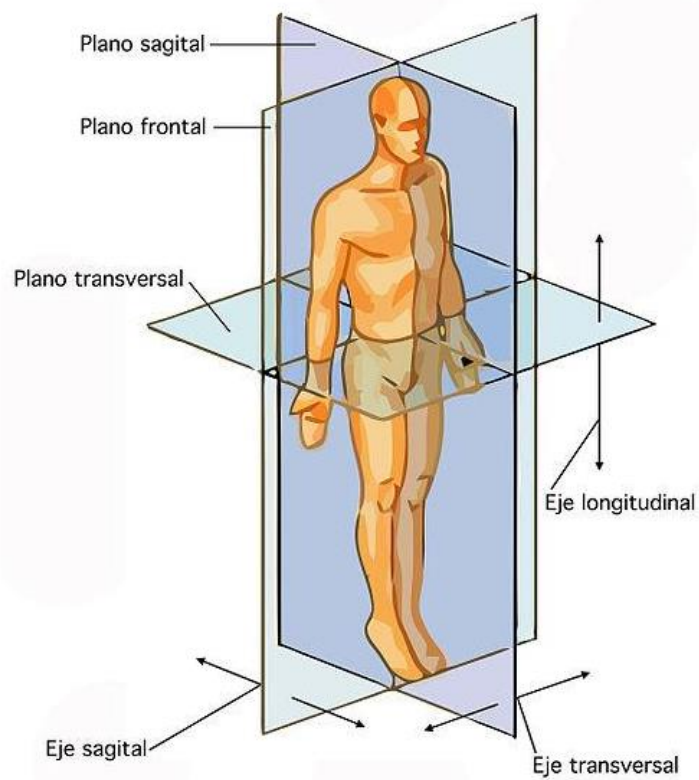


Figura 2.2 Planos anatómicos [22].

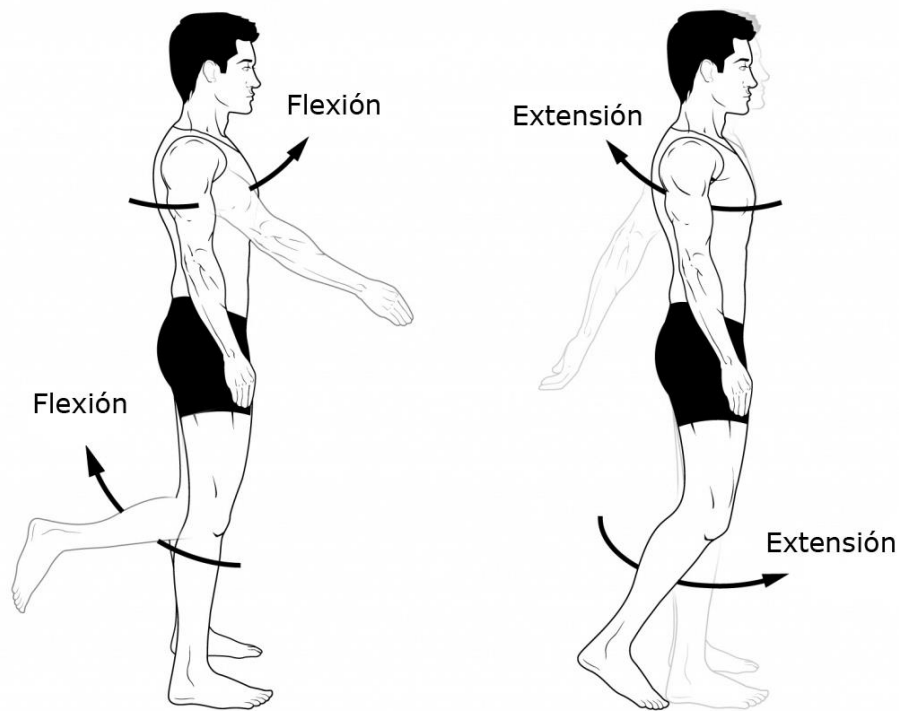


Figura 2.3 Movimientos relativos en el plano sagital [24].

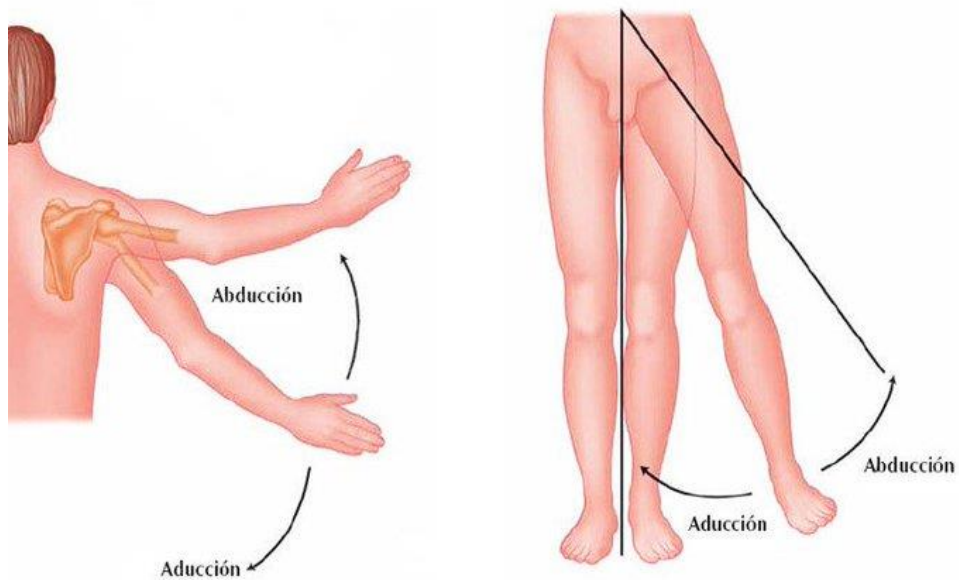


Figura 2.4 Movimientos relativos en el plano frontal [23].

Finamente, durante el proceso de una marcha normal, en la fase de apoyo del pie, dependiendo de donde se distribuyan en la planta del pie las fuerzas generadas con el suelo, se distinguen tres tipos de pisada. Si el pie golpea al suelo con la superficie lateral interior se denomina “pronación”, si lo hace con la exterior, “supinación” y si se realiza con el centro de la planta del pie se conoce como “neutro”) [25]. (Figura 2.5)



Figura 2.5 Tipos de pisada durante la marcha [25].

## 2.2 Ciclo de la Marcha

La marcha humana se puede definir como una serie de movimientos alternos de las extremidades inferiores en un movimiento rítmico que resulta en la progresión hacia adelante del cuerpo con un gasto energético mínimo. Esta serie de movimientos se puede considerar como una actividad cíclica, donde el ciclo de la marcha se establece como el periodo de tiempo mínimo entre los mismos actos. Usualmente se establece el inicio del ciclo cuando el talón entra en contacto por primera vez con el suelo [26].

Durante un ciclo de marcha cada pie experimentará dos etapas, la fase de apoyo de la marcha y la de oscilación. La etapa de apoyo comienza cuando el talón apoya con el suelo y finaliza cuando el mismo abandona el suelo. La etapa de oscilación ocupa desde el fin de la etapa anterior hasta un nuevo contacto con el suelo. (Figura 2.6)

A lo largo del ciclo se distingue una etapa de soporte único, cuando un solo pie está en contacto con el suelo, y de doble soporte, cuando ambos pies están apoyados.

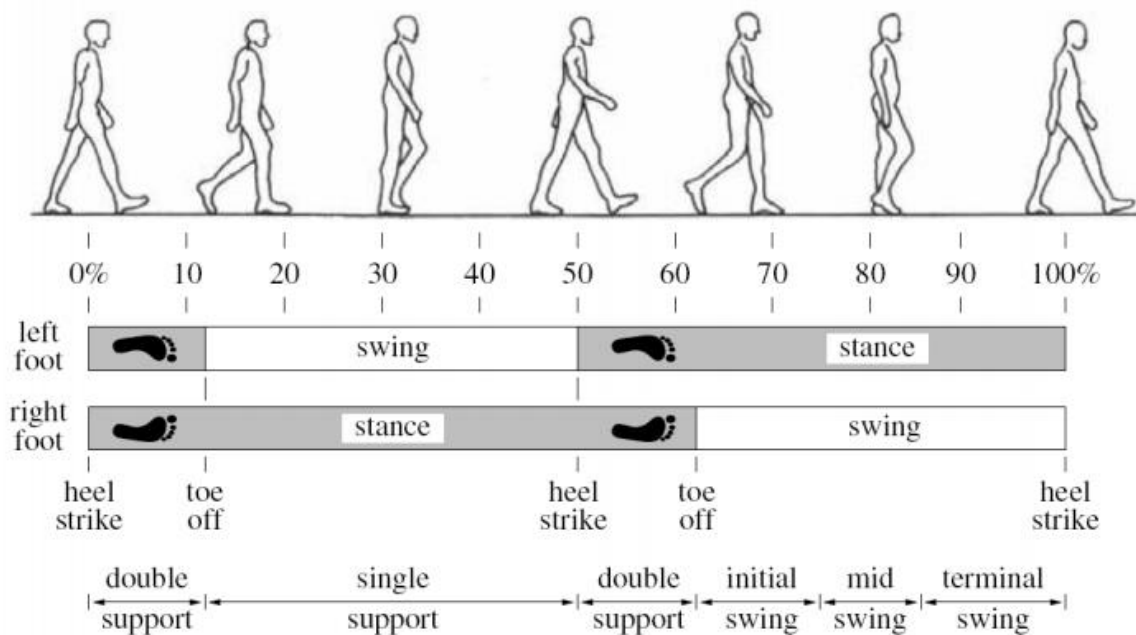


Figura 2.6 Esquema de las distintas fases del ciclo de la marcha. [26]



La fase de apoyo se constituye por el momento de contacto del talón (Figura 2.7-a) con el suelo seguido del apoyo plantar (Figura 2.7-b), cuando la planta entera está en contacto con el suelo y por delante del eje frontal, es en estas etapas cuando el peso se traslada al pie nuevo de apoyo [27]. Continuado del apoyo medio de la planta del pie (Figura 2.7-c), estando ella en el plano sagital. Tras ello se produce la elevación del talón (Figura 2.7-d) y finalmente el despegue de los dedos del suelo (Figura 2.7-e) [28].

La fase de balanceo, sin embargo, se subdivide en tres etapas. La etapa inicial donde se produce una aceleración rápida del extremo de la pierna tras elevarse los dedos del suelo (Figura 1-7-f). El balanceo medio, donde la pierna en balanceo sobrepasa a la pierna en apoyo (Figura 2.7-g). Y el balanceo terminal, produciéndose una desaceleración de la pierna balanceada preparándose para apoyar (Figura 2.7-h) [27].

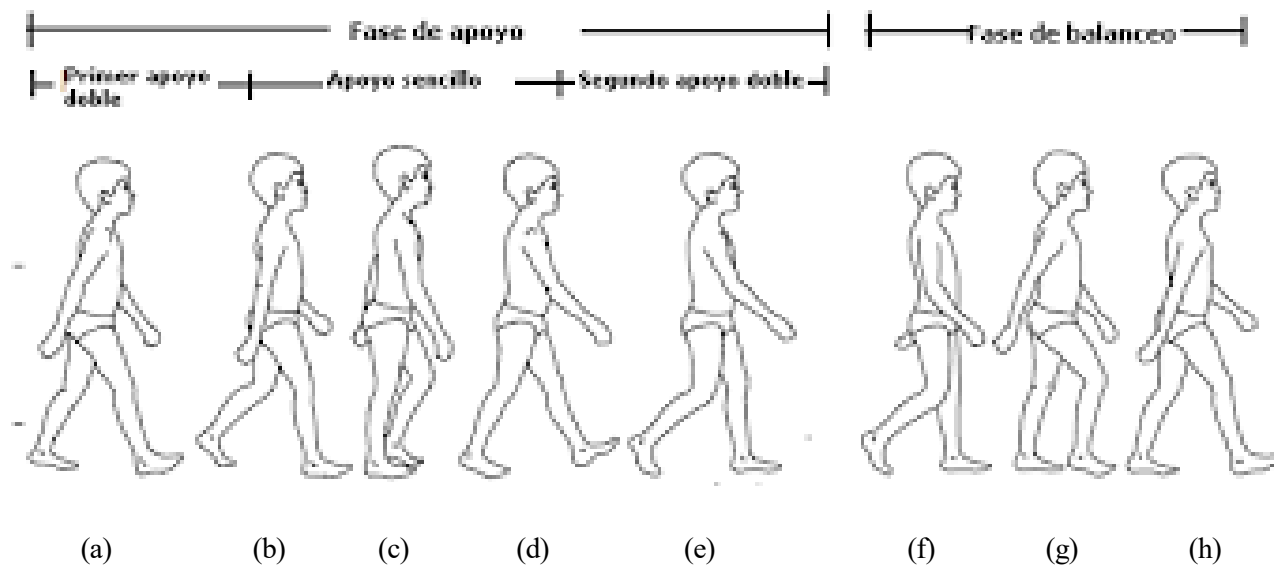


Figura 2.7 Esquema de las distintas sub-fases del ciclo de la marcha [27].

### 2.2.1 Dinámica Inversa

El objetivo de la dinámica inversa en el estudio de la marcha es determinar las fuerzas de reacción en las articulaciones durante el movimiento. Para ello, se parte del conocimiento de la cinemática del sujeto y se plantea un sistema de barras articuladas, donde cada barra representa a un segmento del cuerpo [28].

Para resolver el análisis dinámico inverso existen dos procedimientos. El primero de ellos, consiste en plantear directamente las ecuaciones de la dinámica, conocido el movimiento del cuerpo durante el movimiento. El segundo, aprovecha las fuerzas de reacción con el suelo y la cinemática del sujeto para plantear un sistema de ecuaciones dinámicas sobredeterminado, empezando desde el suelo hasta llegar a la cabeza [28].

Sin embargo, la diferencia entre la dinámica real y la calculada mediante el análisis dinámico es destacable, lo que se considera como el problema fundamental de la dinámica inversa. Esta diferencia se crea tanto por problemas del modelado, como son la suposición de los segmentos corporales como sólidos o la suposición de que las articulaciones se comportan como pares de revolución o esféricos, como por errores de la medición cinemática, como es el ruido de las señales. Es por la inexactitud de los datos cinemáticos que los valores de las fuerzas inerciales difieren significativamente de la realidad, debido a que dicha fuerza considera la doble derivación numérica de la posición [28].

Dado el primer procedimiento del análisis dinámico se puede descartar por la inconsistencia de los datos cinemáticos, nace la necesidad de conocer o estimar las fuerzas de reacción con el suelo para poder completar el segundo de ellos. La medición de dichas fuerzas es comúnmente realiza mediante placas de fuerzas, pero dada a su limitación a entornos cerrados surge la necesidad de un método de estimación más versátil y es por ello que los estudios de la marcha humana mediante técnicas de aprendizaje profundo están en auge.

### 2.2.2 Deep Learning en la Bioingeniería

En relación al problema expuesto en este trabajo, existen en el estado del arte algunas aproximaciones basadas en la aplicación de las redes neuronales para la predicción de la cinemática y la dinámica de la marcha humana. Este trabajo consta de la predicción de las fuerzas verticales con el suelo a partir de los ángulos de flexión de la rodilla, del tobillo y de la cadera medidos experimentalmente mediante captadores de movimiento. Sin embargo, se ha realizado una búsqueda de los trabajos realizados en esta área, dado que muchos de ellos podrían ser útiles para mejorar el proyecto.

En 2020, R. Johnson realizó un modelo de predicción de las fuerzas de reacción y los momentos a partir de las mediciones de las aceleraciones de cinco sensores portátiles colocados en la pelvis y en el exterior del muslo y de la tibia. El modelo presentó un sistema de análisis y de procesado de imágenes para las gráficas de las aceleraciones en el tiempo [29].

En 2021, J. Conte analizó la eficiencia de las redes neuronales profundas para estimar los ángulos de las articulaciones en el transcurso de la marcha humana. La información se aportaba al sistema a partir de una IMU (Unidad de medición inercial) colocada en el pie. El documento comparó las redes neuronales LSTM, GRNN ('Generalized regression neural networks') y NARX ('Nonlinear autoregressive networks with exogenous inputs') a partir de los RMSE cometidos, y concluyó destacando la precisión de la red LSTM para el modelado de la marcha [30].

En 2021, D. Sharma estimó las fuerzas de reacción vertical con el suelo a partir de un GPS (Sistema de posicionamiento global) usando Machine Learning. Los sistemas utilizados por el estudio fueron una mochila equipada de un sistema GPS y unas plantillas instrumentadas, los cuales permitieron obtener 27 medidas que sirvieron para entrenar al programa. El estudio utilizó las redes neuronales LSTM a parte de las técnicas de Machine Learning. El ensayo destacó un error RMSE menor para las técnicas de ML, pero también destacó la incapacidad de estas técnicas para detectar nuevos patrones en las curvas de fuerza comparándolo con las LSTM [31].

# 3 MODELO Y TOMA DE DATOS

---

EN este capítulo se va a desarrollar procedimiento que siguió el laboratorio de biomecánica de la Universidad de Sevilla para realizar el ensayo de la marcha humana, con el objetivo de evaluar las fuerzas de reacción del suelo y los momentos en las articulaciones mediante plataformas de fuerza. Durante el ensayo se midió los ángulos de rotación de las articulaciones gracias a un sistema de captura de movimiento en 3D de la empresa Vicon. El ensayo consiste en la medición de los movimientos de los marcadores y las fuerzas de reacción con el suelo y a partir de los datos de los marcadores se obtiene por cinemática inversa los ángulos de las articulaciones. Este capítulo es un extracto y traducción de [32] con el objetivo de que el documento sea completo y auto comprensible.

## 3.1 Protocolo utilizado

El montaje experimental consistió en un modelo de sistemas multicuerpo, donde las distintas partes que constituyen el cuerpo humano se simularon como cuerpos rígidos, desatendiendo las deformaciones elásticas producidas, al ser estas despreciables para la mayoría de los movimientos humanos. En dicho modelado se supone que cada segmento tendrá su masa localizada en su centro de gravedad, que las coordenadas locales del centro de gravedad permanecen fijas durante el movimiento y que el momento de inercia respecto de su centro de masa es considerado constante durante todo el movimiento [33].

El protocolo utilizado para la reconstrucción del movimiento se basó en el software comercial Plug-in-Gait (PiG - Vicon Motion Systems, Oxford, UK), el cual nació a raíz del modelo Newington. Este trabajo, se centrará únicamente en el tren inferior, componiéndose el modelo de 7 segmentos; dos pies, dos fémures, dos tibias y la cadera.

Las coordenadas de los segmentos se definieron usando 16 marcadores; cuatro para la pelvis (Figura 3.1), dos para cada fémur (Figura 3.2), 1 para cada tibia (Figura 3.3), uno para cada tobillo y dos para cada pie (Figura 3.4).

### 3.1.1 Reconstrucción de la pelvis

El segmento de la pelvis se crea a partir de los marcadores fijados al sujeto, LASI, RASI, LPSI y RPSI, y un punto virtual, PSI, que se define entre RPSI y LPSI. Dados los marcadores RASI, LASI y el punto PSI se crea el plano que contiene a dichos puntos. (Figura 3.1 y Tabla 3-1)

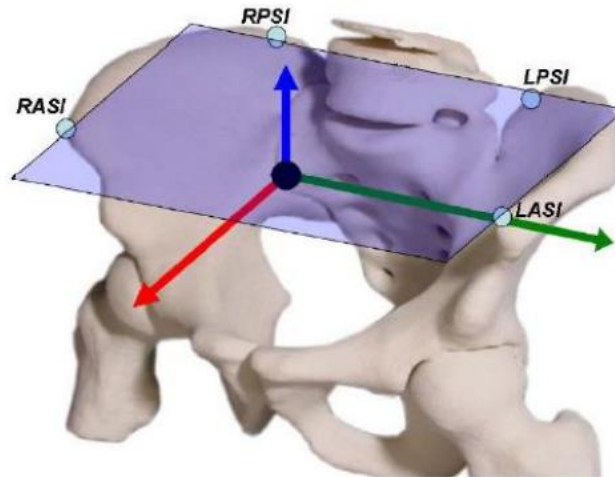


Figura 3.1 Ubicación de los marcadores de la pelvis.

X	Perpendicular al eje Y, Z
Y	Desde RASI hasta LASI
Z	Normal al plano formado por RASI, LASI y LPSI

Tabla 3-1 Ejes de coordenadas de la pelvis.

Además, siguiendo el modelo de Newington, se aplica el procedimiento predictivo para el cálculo de la posición del centro de la cadera, HJC. Por lo que sistemas de coordenadas local de la cadera será igual que el de la figura 3.1 pero con el origen trasladado hasta el punto medio de los centros de las caderas.

### 3.1.2 Reconstrucción del fémur

A partir de las coordenadas del centro de la cadera y de los marcadores, KNE y THI, se puede calcular la posición de los centros de las rodillas, KJC. El procedimiento para calcular dicho punto se basa en imponer tres condiciones que deben cumplirse simultáneamente. La primera es que el punto está en el plano comprendido por HJC, THI y KNE. La segunda, que el punto se encuentra a una distancia igual a la mitad del ancho de la rodilla del marcador KNE. Y, por último, que el vector KNE-KJC es perpendicular al vector HJC-KJC. (Figura 3.2 y Tabla 3-2)

X	Normal al plano formado por HJC, THI y KNEE
Y	Perpendicular a X, Z
Z	De HJC a KJC

Tabla 3-2 Ejes de coordenadas del fémur.

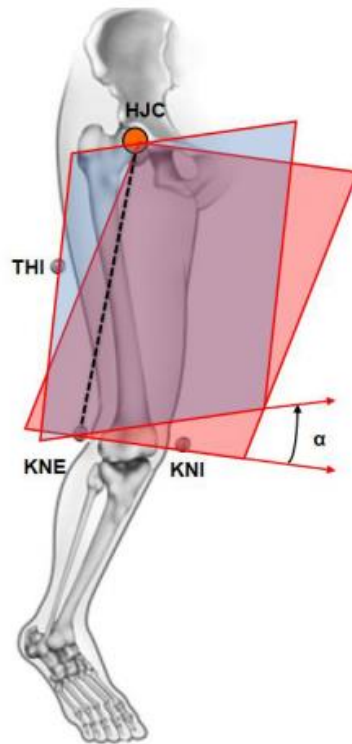


Figura 3.2 Ubicación de los marcadores del fémur, KNE y THI.

### 3.1.3 Reconstrucción de la tibia

Para la tibia se procede de una manera análoga que, para el fémur, donde a partir del punto virtual de KJC, y de los marcadores de ANK y TIB, se puede calcular el centro del tobillo, AJC. Donde AJC debe pertenecer al plano formado por TIB, ANK y KJC, encontrarse a una distancia igual al radio del tobillo del marcador ANK y que el vector AJC-KJC sea perpendicular al AJC-ANK. (Figura 3.3 y Tabla 3-3)

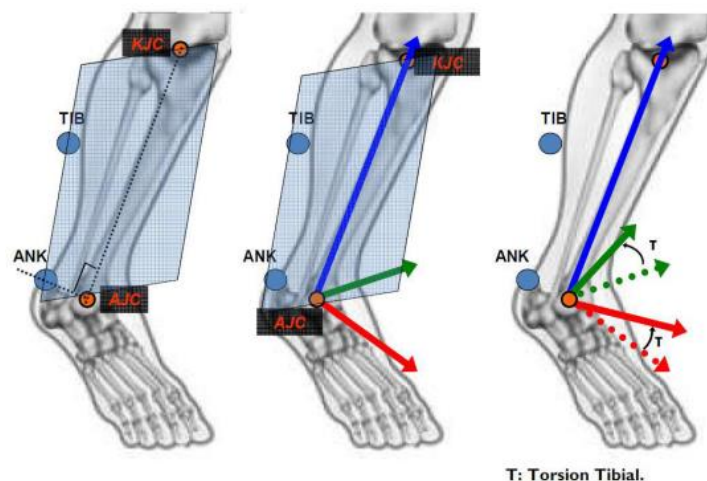


Figura 3.3 Ubicación del marcador de la tibia, TIB, y del tobillo, ANK

X	Normal al plano formado por KJC, TIB y ANK
Y	Perpendicular a X, Z
Z	Desde AJC a KJC

Tabla 3-3 Ejes de coordenadas de la tibia.

### 3.1.4 Reconstrucción del pie

Finalmente, para la reconstrucción del pie se utiliza un procedimiento semejante a los anteriores a partir de los marcadores, HEE y TOE, y del centro del tobillo, AJC. (Figura 3.4 y Tabla 2-4)

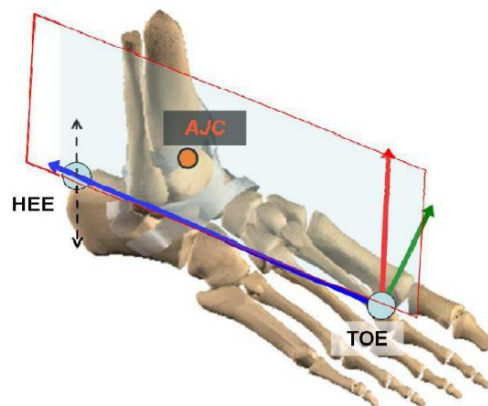


Figura 3.4 Ubicación de los marcadores del pie, HEE y TOE.

X	Perpendicular a Y, Z
Y	Normal al plano definido por HEE, TOE y AJC
Z	De TOE a HEE

Tabla 3-4 Ejes de coordenadas del pie.

## 3.2 Toma de datos

El ensayo se realizó para 14 sujetos adultos sanos, obtenido una base de datos total de 143 ciclos de marcha de distintos experimentos, la realización del número de pasadas de cada sujeto se desarrolla en la tabla 2-5. Durante el experimento, los sujetos caminaron sobre una superficie plana de 12 metros de longitud en condiciones normales y sin perturbaciones. Al mismo tiempo, la superficie de trabajo, de dimensiones 1.5x1.5x2 metros, se colocó en el centro de la sala con el objetivo de que el individuo tuviera tiempo suficiente para alcanzar su velocidad normal de marcha y para frenar tras sobrepasar las placas. Cabe recalcar que todos los datos de ensayo se han obtenido de la base de datos del departamento de Ingeniería mecánica y fabricación de la Universidad de Sevilla.

Se sabe que el análisis de movimiento con cámaras de infrarrojos lleva asociados distintas fuentes de error en su proceso de reconstrucción del esqueleto, entre ellos, la más importante es la del movimiento relativo de la piel con el hueso (Soft Tissue Artifact, STA). Este error se ha demostrado que se puede asumir del orden de 1/3 cm

durante la marcha [32]. No obstante, la única manera de eliminar estos errores sería fijar los marcadores directamente a los huesos mediante clavijas óseas, una técnica invasiva. Otros errores comunes son los de instrumentación y los causados por la mala colocación de los marcadores por parte del examinador.

A continuación, se presenta en la tabla las principales características físicas de los sujetos ensayados, así como el número de pruebas o pasadas por individuo.

<b>Sujeto</b>	<b>Peso (kg)</b>	<b>Altura (cm)</b>	<b>Número de pasadas</b>	<b>Longitud Fémur (cm)</b>	<b>Longitud Tíbia (cm)</b>	<b>Ancho de Cadera (cm)</b>
<b>1</b>	65.40	1770	7	40.8	42.1	13.2
<b>2</b>	61.51	1680	9	36.3	39.0	18.4
<b>3</b>	84.6	1740	8	46.1	41.0	26.0
<b>4</b>	63.66	1805	20	42.2	42.7	12.4
<b>5</b>	59.00	1740	9	39.7	39.8	15.1
<b>6</b>	73.50	1650	4	36.3	38.4	17.1
<b>7</b>	76.01	1890	6	55.0	45.3	24.0
<b>8</b>	83.60	1830	9	40.1	43.1	15.6
<b>9</b>	68.30	1860	15	42.9	44.0	13.9
<b>10</b>	87.30	1890	8	41.2	46.9	18.8
<b>11</b>	96.28	1790	7	38.4	44.2	21.3
<b>12</b>	78.10	1825	16	40.5	46.5	15.9
<b>13</b>	68.30	1860	18	40.9	41.3	11.8
<b>14</b>	63.13	1770	7	41.3	42.5	16.3

Tabla 3-5 Información de sujetos experimentados.





## 4 APLICACIÓN DEL DEEP LEARNING

EN este capítulo se va a implementar diversas técnicas de Deep Learning en el software de Matlab 2021a para simular las fuerzas de reacción con el suelo (GRF) a partir de la base de datos de las medidas de los ángulos de las articulaciones del modelo del capítulo 3.

La necesidad de este estudio nace ya que, en la dinámica inversa, las fuerzas de reacción con el suelo se encuentran completamente determinadas por los desplazamientos de los segmentos durante la fase de mono apoyo. Sin embargo, para calcular dichas fuerzas hay que derivar doblemente las posiciones, lo cual puede dar lugar a grandes errores. Además, en la fase de biapoyo el sistema queda indeterminado.

### 4.1 Clasificación del problema

El estudio consistirá en la predicción de las fuerzas de reacción con el suelo en dirección del eje longitudinal a partir de los ángulos de flexión del tobillo, de la rodilla y de la cadera. Se puede ver un ejemplo de representación de ambos respecto al tiempo a lo largo de un ciclo de marcha en las figuras 4.1 y 4.2.

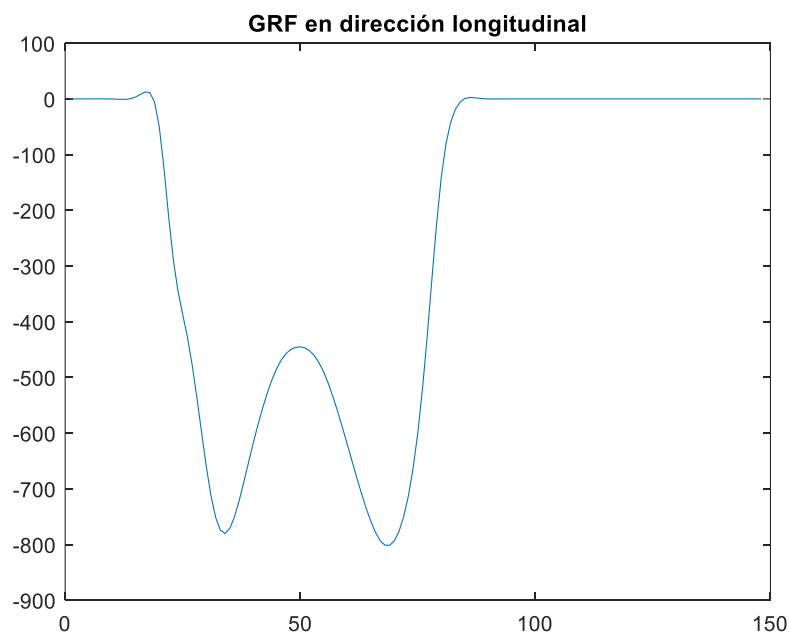


Figura 4.1 Fuerza experimental de reacción con el suelo en dirección longitudinal del sujeto 1, prueba 1.

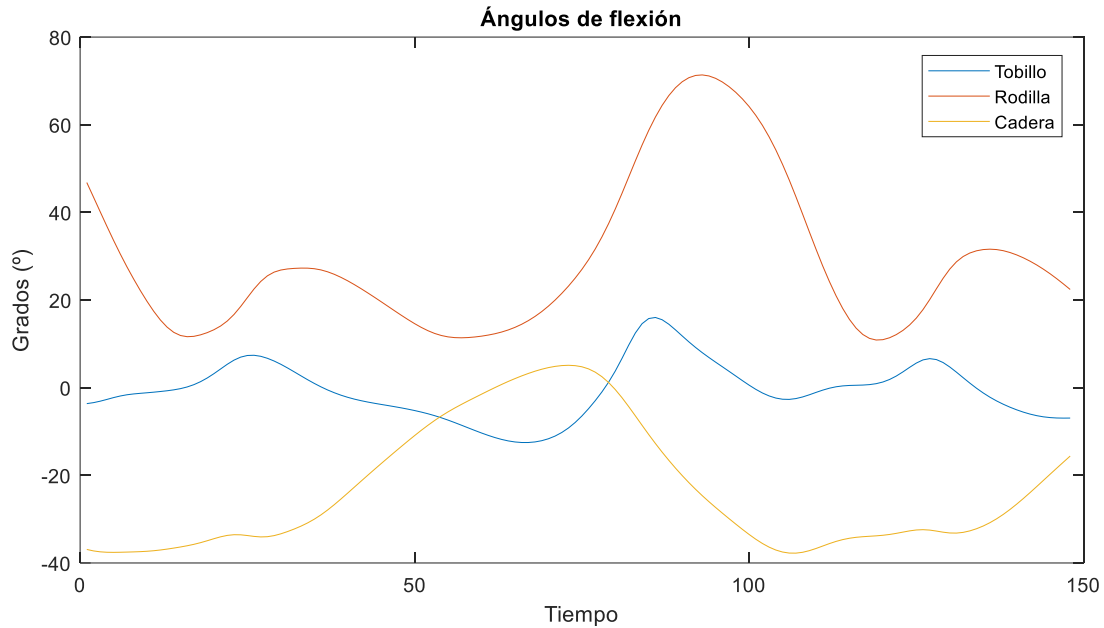


Figura 4.2 Ángulos experimentales de flexión para la cadera, rodilla y tobillo del sujeto 1, prueba 1.

En primer lugar, dado que las entradas y salidas de la base de datos están disponibles, el problema se puede clasificar en un sistema de aprendizaje supervisado. Y como, además, el resultado será el de obtener un valor continuo, se puede subclasificar en un problema de regresión.

Por otro lado, los datos tratan de series temporales donde los valores pasados son influyentes en los valores siguientes, es por ello, que es aconsejable la utilización de algún tipo de red neuronal con retroalimentación, como la RNN.

## 4.2 Preprocesado de datos

Con el objetivo de optimizar el proceso de aprendizaje, se han procesado los datos tanto de entrada como de salida antes de entrar la red de entrenamiento. Las modificaciones fueron las siguientes:

1. La primera fue normalizar la señal al rango  $[0,100]$ , representando el porcentaje del ciclo de la marcha.

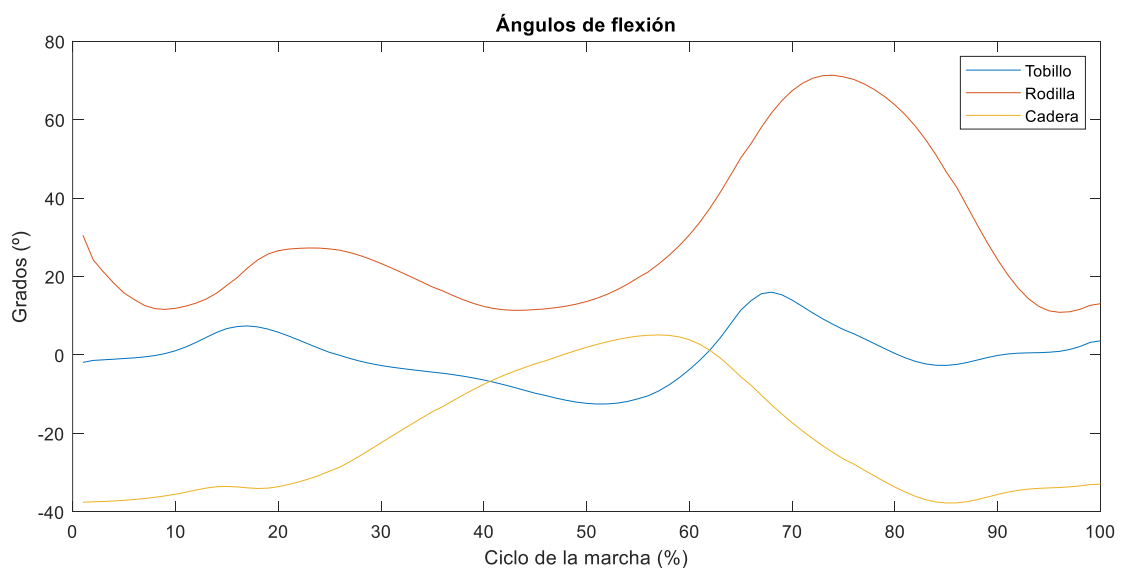


Figura 4.3 Ángulos experimentales de flexión tras el pre procesado 1 del sujeto 1, prueba 1.

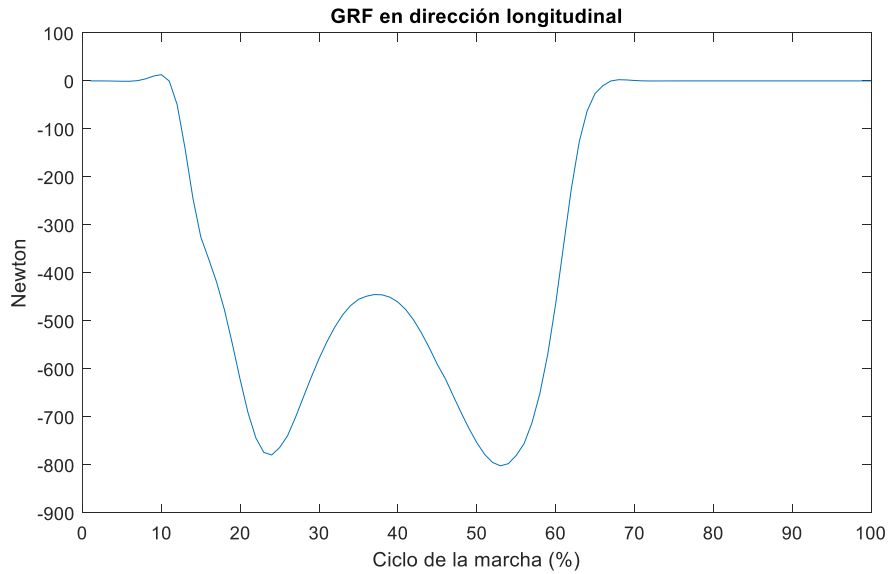


Figura 4.4 Fuerza experimental de reacción con el suelo tras el pre procesado 1 del sujeto 1, prueba 1.

2. Normalizar las fuerzas de reacción con respecto al peso de cada sujeto.

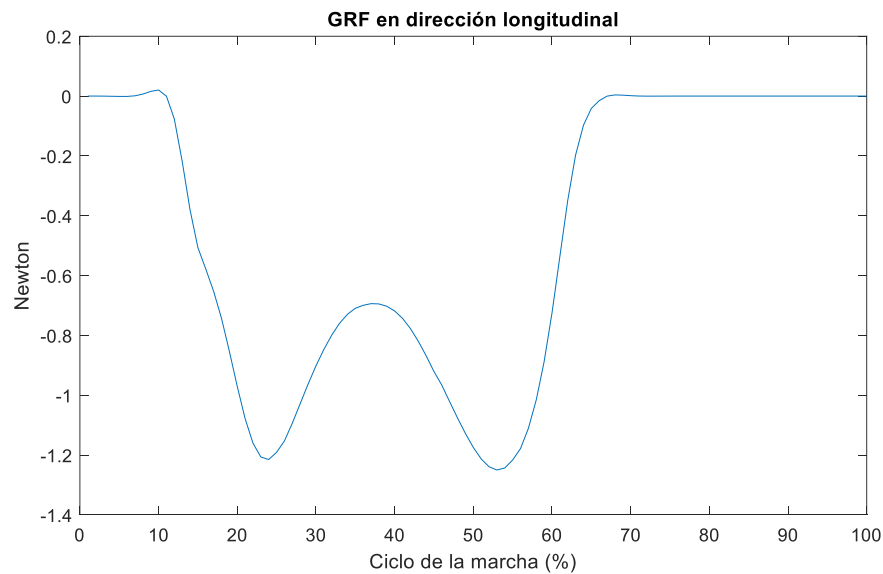


Figura 4.5 Fuerza experimental de reacción con el suelo tras el pre procesado 2 del sujeto 1, prueba 1.

### 4.3 Entrenamiento

Las Redes Neuronales analizadas fueron implementadas en Matlab. Se entrenaron para modelar la relación entre las fuerzas de reacción ejercidas con el suelo y los ángulos de flexión del tobillo, rodilla y cadera ejercidos durante un ciclo de marcha.

Como herramienta para comparar la validez de las RN entre ellas y valorar su parecido con la realidad, se utilizará el RMSE, error cuadrático medio, que representa la diferencia cuadrática entre los resultados y los objetivos. Por lo tanto, menores valores del error son mejores predicciones. Se calcula de la siguiente forma:

$$RMSE = \sqrt{\frac{1}{L} \sum_{n=1}^L (y(n) - \hat{y}(n))^2}$$

Siendo  $L$  el tamaño de la señal,  $y(n)$  las GRF medidas experimentalmente y  $\hat{y}(n)$  el valor estimado por la RN.

### 4.3.1 Red LSTM

La red LSTM fue diseñada con una capa de entrada en secuencia, capas ocultas de LSTM, una capa totalmente conectada y una capa de salida de regresión (Figura 4.6). Para el entrenamiento, se estableció el número de capas ocultas de forma iterativa hasta conseguir el valor que establecía el menor RMSE, estas capas ocultas corresponden a la cantidad de información recordada entre los pasos de tiempo [34].

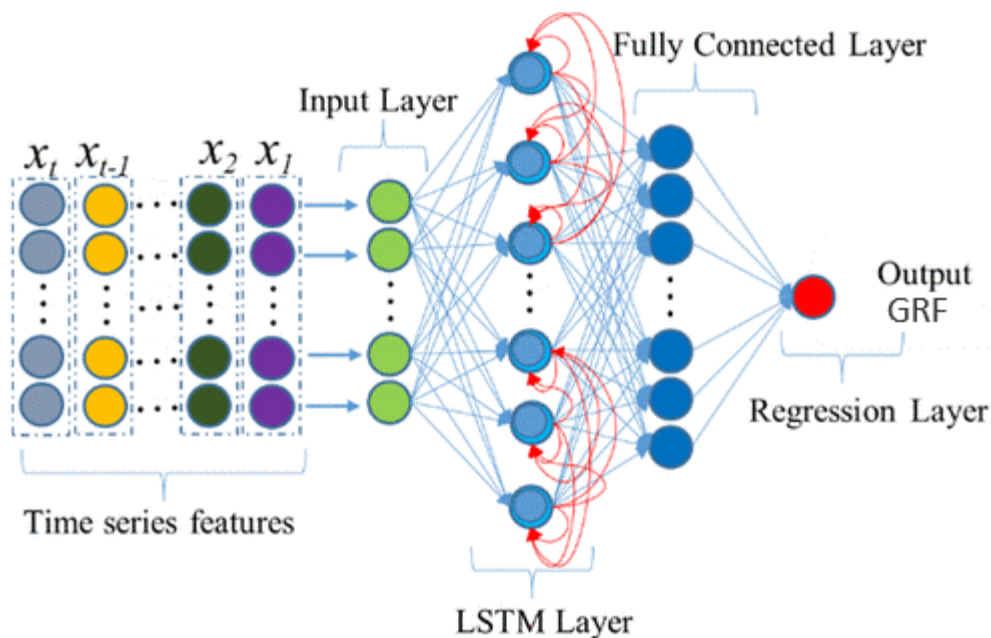


Figura 4.6 Estructura de una red LSTM para regresión.

Además, se ha aplicado la técnica del *'dropout'* o descarte después de cada capa con el objetivo de evitar el sobreajuste, es decir, que la red memorice los resultados para una entrada y no sea capaz de generalizar el problema. Esta técnica consiste en desactivar aleatoriamente un porcentaje de las neuronas de cada capa oculta. Sin embargo, durante la predicción de nuevos valores todas las neuronas estarán activas [35].

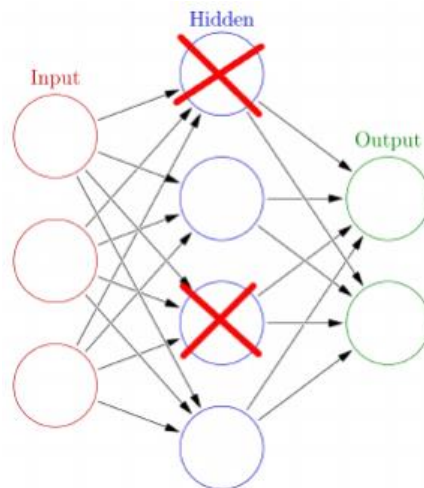


Figura 4.7 Funcionamiento del descarte cuando vale un 50 % [36].

Otra forma para controlar el sobreajuste es el uso de regularizadores, estos tratan de penalizar los pesos y umbrales de valores absolutos muy altos sumándole a la función de error otra función que dependan de dichos pesos [37].

Dentro del mecanismo de *Back Propagation* existen varios métodos de optimización de los parámetros, en este estudio se ha aplicado el Adam.

Asimismo, el número de *Epoch* define el número de veces que el algoritmo de aprendizaje funcionará en todo el conjunto de los datos de entrenamiento. Por otro lado, un lote o *mini Batch size* define el número de muestras que se analizarán antes de actualizar los parámetros internos del modelo [38]. Es por ello por lo que se ha definido el tamaño del lote durante el entrenamiento como el tamaño de cada prueba normalizada, ya que conviene que los parámetros se modifiquen una vez analizado un ciclo completo.

Otro parámetro que se debe controlar es la tasa de aprendizaje. Esta controla la rapidez con la que el modelo se adapta al sistema, es por ello por lo que para valores más pequeños de la tasa de aprendizaje será necesario mayor número de *Epoch*. De esta manera, si la tasa es muy pequeña, el entrenamiento puede ser demasiado lento, mientras que, si es muy alta se puede obtener un valor subóptimo del problema [39].

Por último, se ha especificado que los datos de entrenamientos no se barajen en ningún momento durante el entrenamiento con el objetivo de no perder la secuencia temporal de los datos.

#### 4.3.2 Red Profunda LSTM

Con el fin de aumentar el rendimiento de la estructura LSTM del apartado anterior, se utilizó la misma red LSTM, pero añadiendo capas LSTM adicionales a la salida de la capa LSTM, justo antes de la capa totalmente conectada. De esta manera se consigue una red neuronal más profunda.

Los demás parámetros se han especificado análogamente al apartado 4.3.2.

#### 4.3.3 App 'Neural Net Fitting'

Otro método estudiado para entrenar la red fue aprovechar la aplicación de 'Neural Net Fitting' que proporciona Matlab. Esta herramienta permite relacionar los valores de los datos de entrenamiento con los datos objetivos mediante una red neuronal feed-forward de dos capas con neuronas ocultas sigmoideas y neuronas de salida lineales.

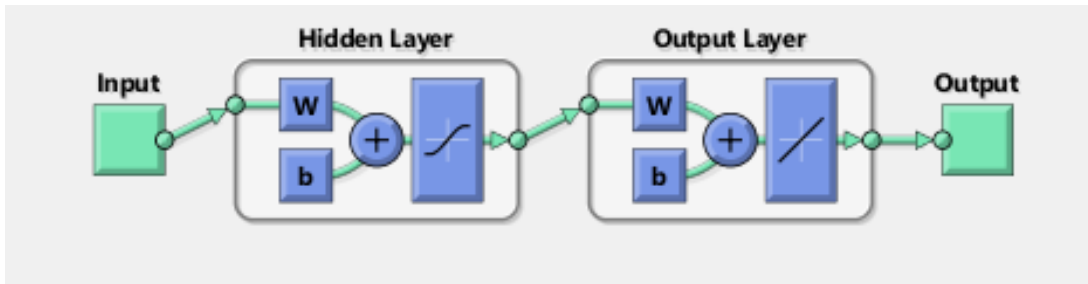


Figura 4.8 Estructura de la red de la aplicación 'Neural Net Fitting'

Para el entrenamiento de la red se ha establecido mediante un proceso iterativo el número de las capas ocultas del mismo modo que el tipo de algoritmo de entrenamiento, entre los que proporciona Matlab. Para ello se comparará los valores del error cuadrático medio entre las entradas y la salida.

Por otra parte, los datos de entrenamiento se dividen automática y aleatoriamente entre datos de entrenamiento, de validación y de testado, en un 70 %, 15 % y 15 % respectivamente para la distribución predeterminada. Los datos de entrenamiento serán aquellos que se utilicen para el entrenamiento de la red, los de validación servirán para medir la generalización de la red y detendrán el entrenamiento cuando esta generalización deje de mejorar, y los de testado que no afectarán en el entrenamiento y otorgarán una imagen real de la eficacia de la red.

Además, la aplicación proporciona tres algoritmos de entrenamiento diferentes, el de Levenberg-Marquardt, LM, el de Bayesian Regularization, BR, y el de Scaled Conjugated Gradient, SCG.

## 5 RESULTADOS

EL objetivo de este trabajo ha sido evaluar la eficiencia del Deep Learning para la predicción de la GRF en pacientes adultos y sanos a partir de los ángulos de flexión del tobillo, rodilla y de la cadera. Para ello, en este capítulo se va a comparar los resultados obtenidos según las redes de aprendizaje seleccionadas, los datos de entrenamientos estudiados y los parámetros de entrenamiento utilizados.

En primer lugar, se ha optimizado cada parámetro de la red de entrenamiento para poder compararlas entre ellas, y posteriormente, se ha observado la influencia de los datos de entrenamiento en dicha red.

### 5.1 Red LSTM

La estructura de la red LSTM depende del número de neuronas ocultas y del valor de la técnica ‘*dropout*’. Pero, además, el entrenamiento de la red dependerá a su vez del tamaño del lote, el tamaño del ‘*Epoch*’ y de la tasa de aprendizaje. Inicialmente se ha partido con los siguientes valores aleatorios con los que poder comenzar un proceso iterativo:

- Capas ocultas: 100
- *Dropout*: 0.2
- *Epoch*: 60
- Tamaño del lote: 100
- Tasa de aprendizaje inicial: 0.001

Para comparar los valores entre ellos, se aplicó el entrenamiento sobre los datos con preprocesado 1 y 2, de los sujetos 1 hasta el 13, estableciendo los datos del sujeto 14 para calcular el RMSE sobre datos no influyentes en el proceso de entrenamiento.

Se comenzó modificando la tasa de aprendizaje junto con el número de ‘*Epoch*’ dado que el último permite al primero completar su función.

En la figura 5.1 se puede observar que el RMSE alcanza mínimos para unos valores de la tasa de aprendizaje inicial de 0.001 y 0.01, para el primer valor esto ocurre cuando el número de ‘*Epoch*’ alcanza los 1000, un valor muy alto que provoca un entrenamiento más lento, mientras que, para el segundo valor, tiene un valor de 150.

A partir de la figura 5.1 se obtienen unos valores referencia tanto de la tasa de aprendizaje inicial como del número de ‘*Epoch*’, los cuales se concretan su valor final en la figura 5.2. Se pueden observar en ambas figuras que, aunque lo intuitivo sería que, a mayor número de iteraciones, ‘*Epoch*’, mejor fuera la precisión del método, esto no pasa en la realidad, ya que el sistema empieza a sobre ajustarse y a empeorar los resultados. Además, habría que destacar que los valores de los errores obtenidos son valores de referencia, dado que entre dos entrenamientos con parámetros iguales se obtienen errores diferentes. Esto pasa debido que el entrenamiento comienza con valores de pesos aleatorios los cuales difieren entre entrenamientos y por tanto, la redes neuronales convergen a resultados distintos. Es por ello por lo que cualquier valor de la tasa de aprendizaje inicial entre 0.01 y 0.001 y unas *Epoch* entre 150 y 175 se han tomado como resultados semejantes.

Se ha tomado como valores finales:

- *Epoch*: 150
- Tasa de aprendizaje inicial: 0.01

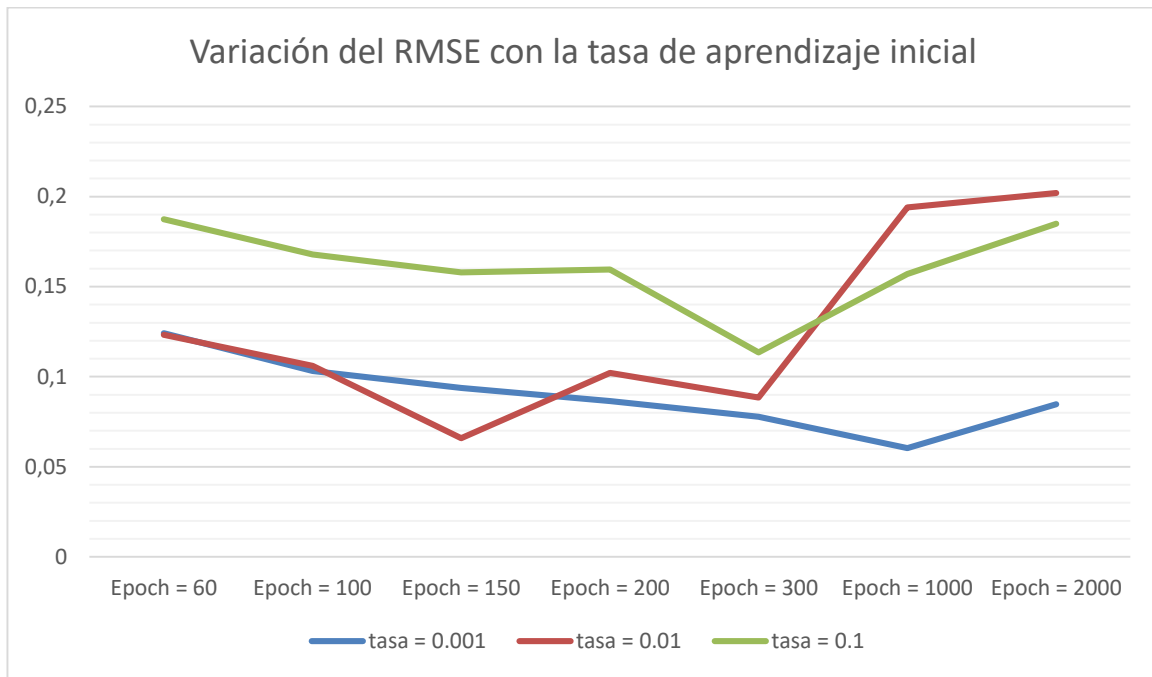


Figura 5.1 Comparación de los valores de RMSE al variar la tasa de aprendizaje y los Epoch entre 60 y 2000.

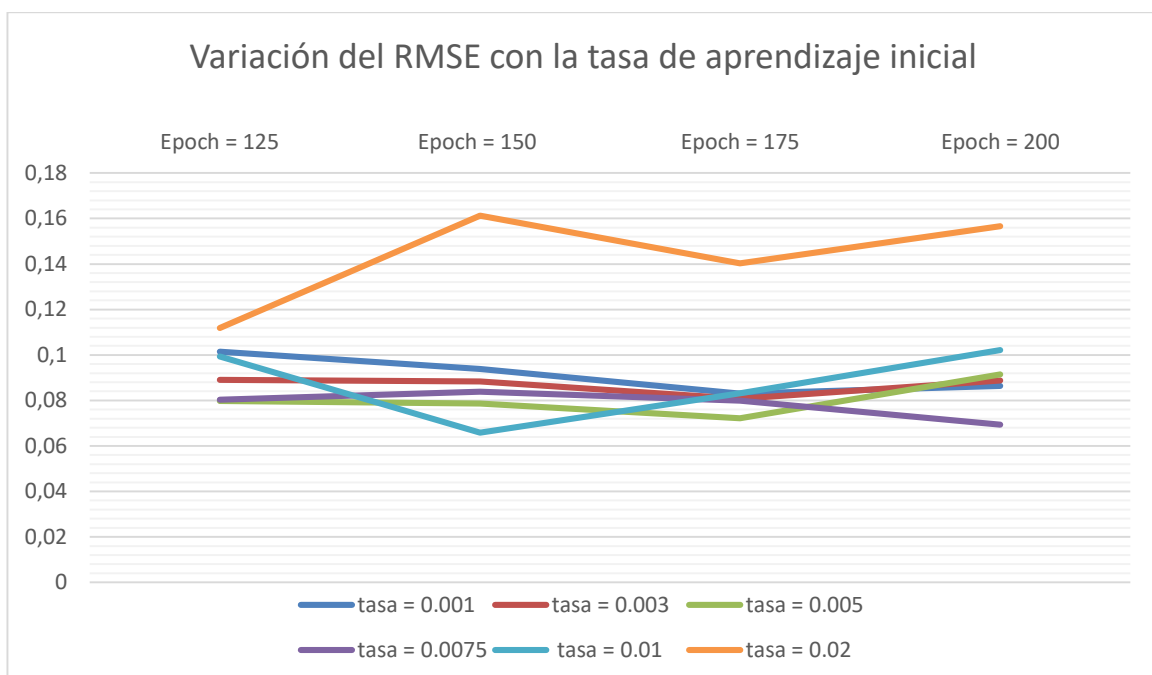


Figura 5.2 Comparación de los valores de RMSE al variar la tasa de aprendizaje y los Epoch entre 125 y 200.

A continuación, se han variado el número de las neuronas de la capa oculta LSTM. Para ello se ha entrenado cada configuración de la estructura de la red 3 veces para evitar valores de RMSE inusuales o improbables en la configuración final y para poder evaluar con mayor criterio los valores obtenidos.

En la figura 5.3 se puede observar como un gran aumento del número de neuronas de la capa oculta LSTM puede conllevar a un peor modelado del problema, provocando un sobreajuste del sistema, y como un número pequeño de neuronas sufre un resultado subóptimo del modelado.

Obtenidos los resultados de la figura 5.3, se ha detallado la figura 5.4, de la cual se observa valores de RMSE muy parejos para todo el rango del número de neuronas ocultas.



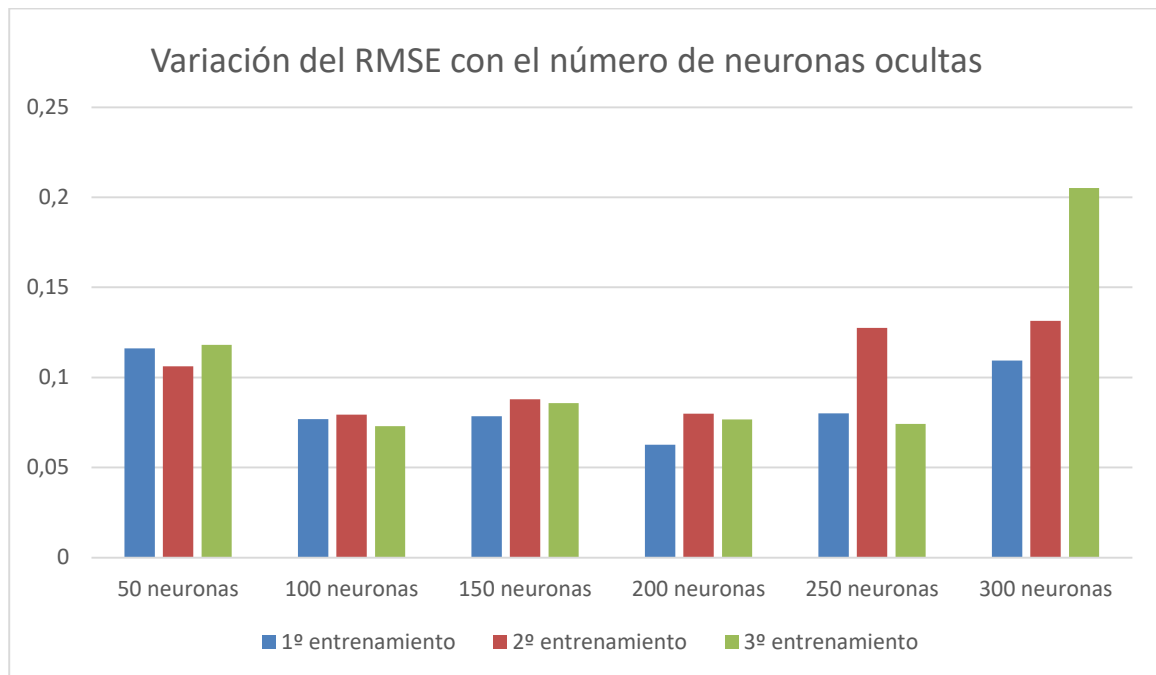


Figura 5.3 Valores de RMSE al variar el número de neuronas ocultas LSTM entre 50 y 300.

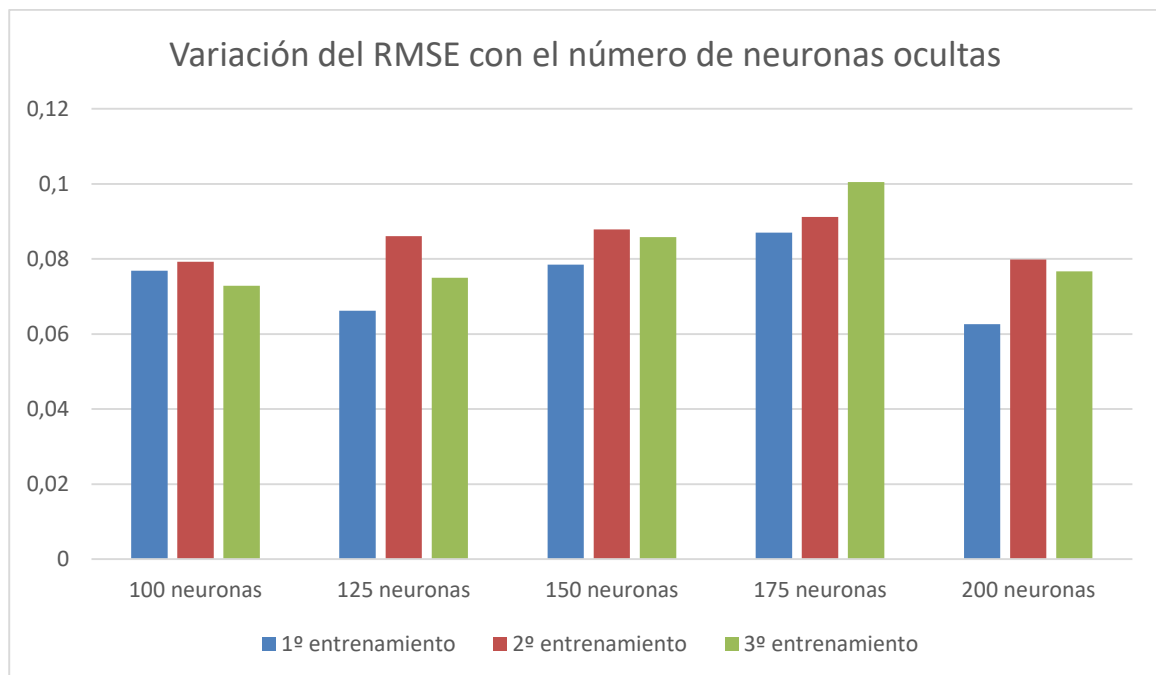


Figura 5.4 Valores de RMSE al variar el número de neuronas ocultas LSTM entre 100 y 200.

Se ha tomado como valores finales:

- Número de neuronas de la capa oculta LSTM: 200

Finalmente, se ha comparado el efecto del mecanismo *'Dropout'*. Según la figura 5.5, el mecanismo no produce una mejoría significativa del modelado, aunque en algunos casos los resultados sean mejores. Sin embargo, se ha decidido mantenerlo en un 20 % ya que ambas opciones parecen ser válidas.

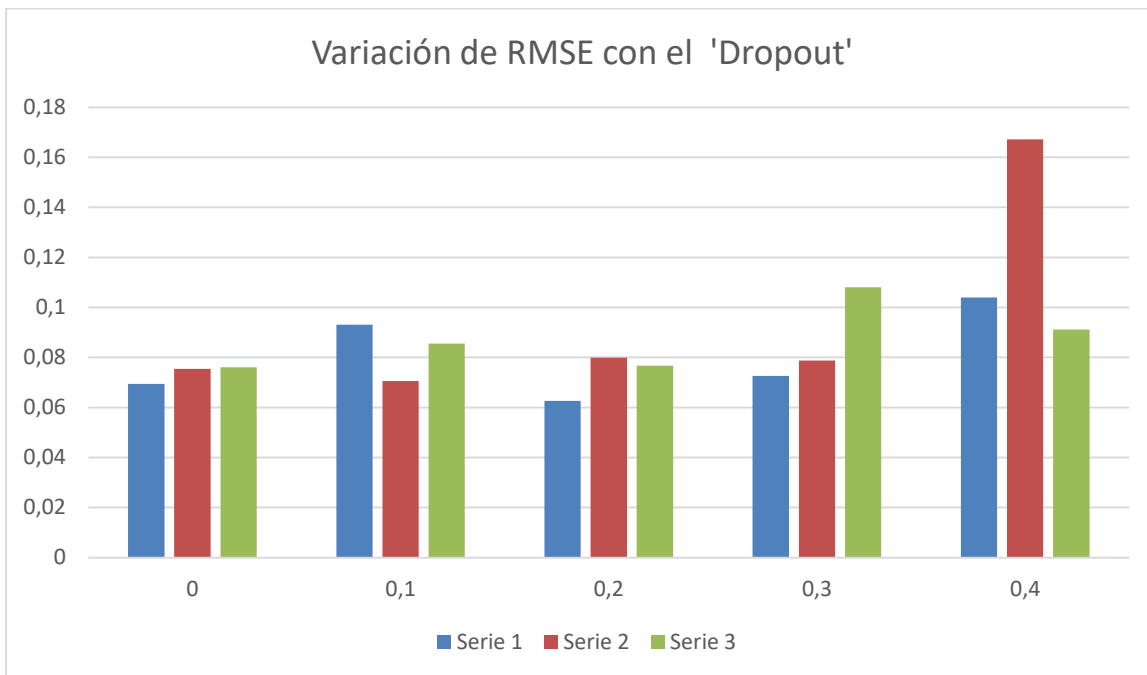


Figura 5.5 Valores de RMSE al variar el porcentaje de descarte.

Tras la optimización de los parámetros y la realización de varios entrenamientos, se ha conseguido un RMSE mínimo de 0.626. En la figura 5.6 se puede observar una representación de las GRF medidas experimentalmente frente las calculadas mediante la red LSTM en Matlab para el sujeto 14.

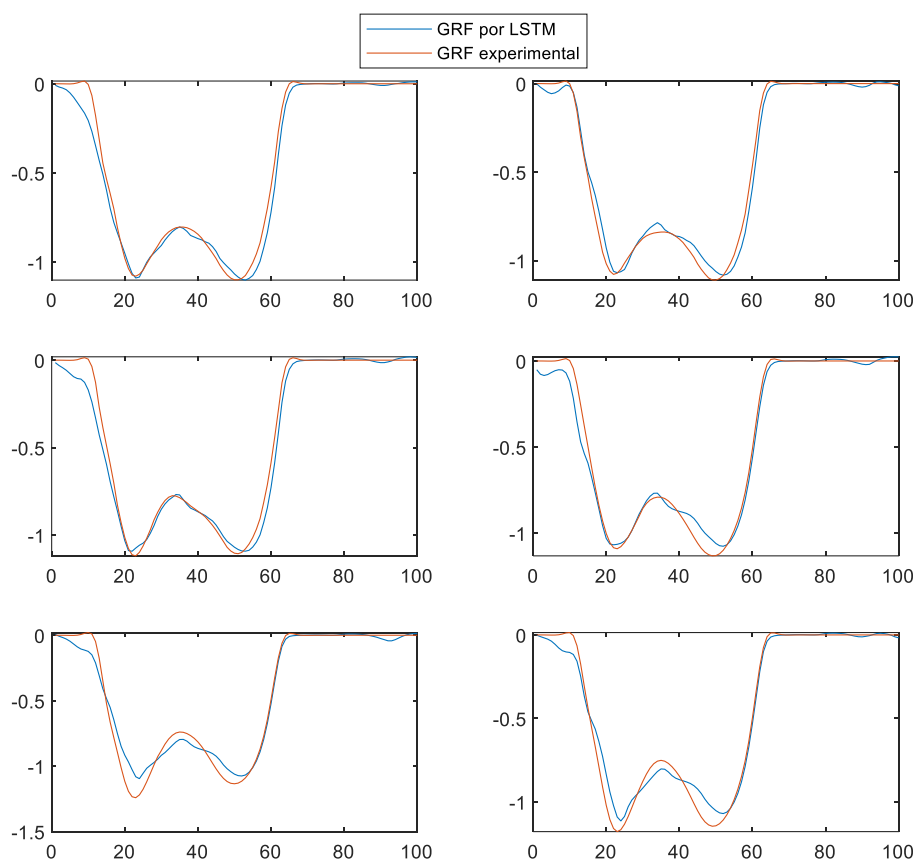


Figura 5.6 Comparación de las GRF experimentales y las simuladas mediante la red LSTM.

## 5.2 Red Profunda LSTM

En este apartado se va a estudiar la influencia del número de capas LSTM y del número de las neuronas ocultas de estas, partiendo de la estructura y entrenamiento de la red del apartado 5.2. Dado que el número de combinaciones capas-neuronas es doblemente infinito, se ha comenzado por determinar el número de capas óptimas si todas tienen el mismo número de neuronas, 200, posteriormente, se ha optimizado el número de neuronas por capas y finalmente, se ha comparado la influencia de las opciones de entrenamiento.

Tras la figura 5.7 se observa una disminución significativa del RMSE cuando la red la forman 3 capas LSTM, esta mejoría se mantiene al aumentar las capas, aunque se vuelve más inestable el proceso y más lento. Por lo tanto, se optó por tomar 3 capas.

Entrenando para todas las combinaciones posibles de valores entre 100, 150 y 200 neuronas para las 3 capas LSTM, se obtuvo que el mínimo valor de RMSE se daba en la red cuya estructura la formaba 100 neuronas ocultas en la primera capa, 150 en la segunda y 100 en la tercera. Dicho error fue de 0.041, un 65.5 % del error cometido tras el entrenamiento del apartado 5.1. Cabe destacar que, para la combinación de neuronas en las capas de la forma: 100-100-200, el error cometido era de 0.0485, muy cercano al escogido como óptimo.

En último lugar, se ha modificado las opciones de entrenamiento de la red profunda. El primer parámetro que se modificó fue la tasa de aprendizaje inicial junto al número de *Epoch*, al igual que se hizo en el apartado 5.1. En la figura 5.8 se puede observar la importancia de la tasa de aprendizaje, donde una elección de la tasa de 0.1 dispara el error cometido en el modelado. Sin embargo, no se perciben mucha diferencia entre los errores cuando la tasa es de 0.01 y 0.001, por consiguiente, se ha optado por no variar las opciones de entrenamiento y tomar la simulación con 0.01, obteniendo un RMSE mínimo de 0.041.

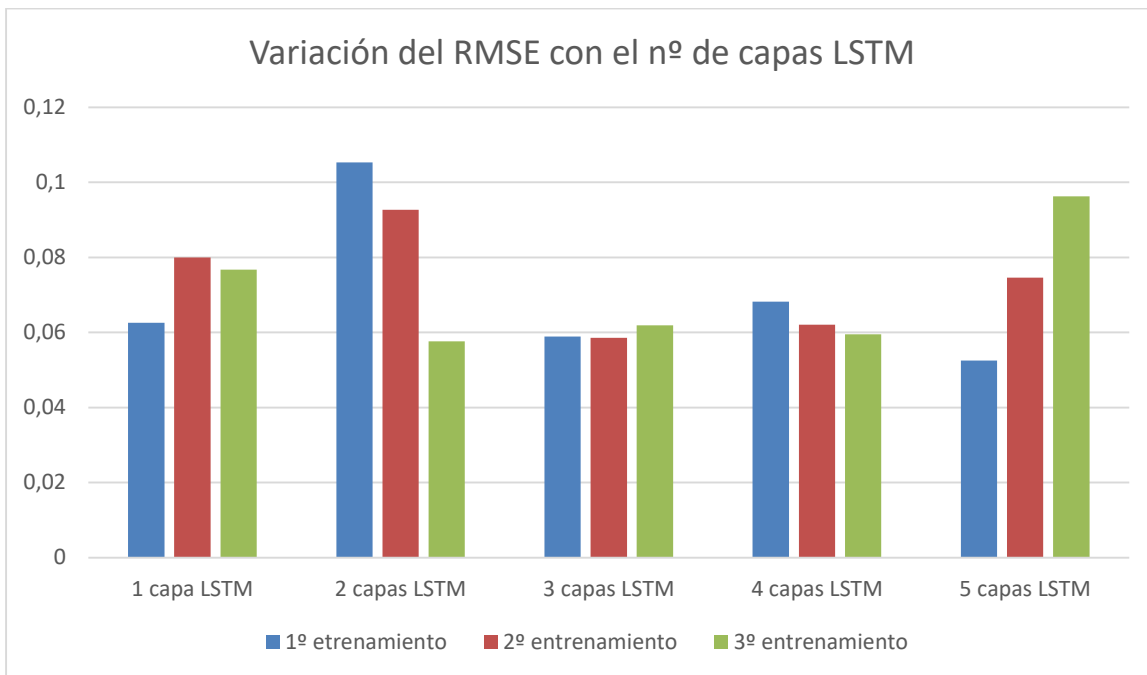


Figura 5.7 Comparación de los valores de RMSE al aumentar el número de capas ocultas LSTM.

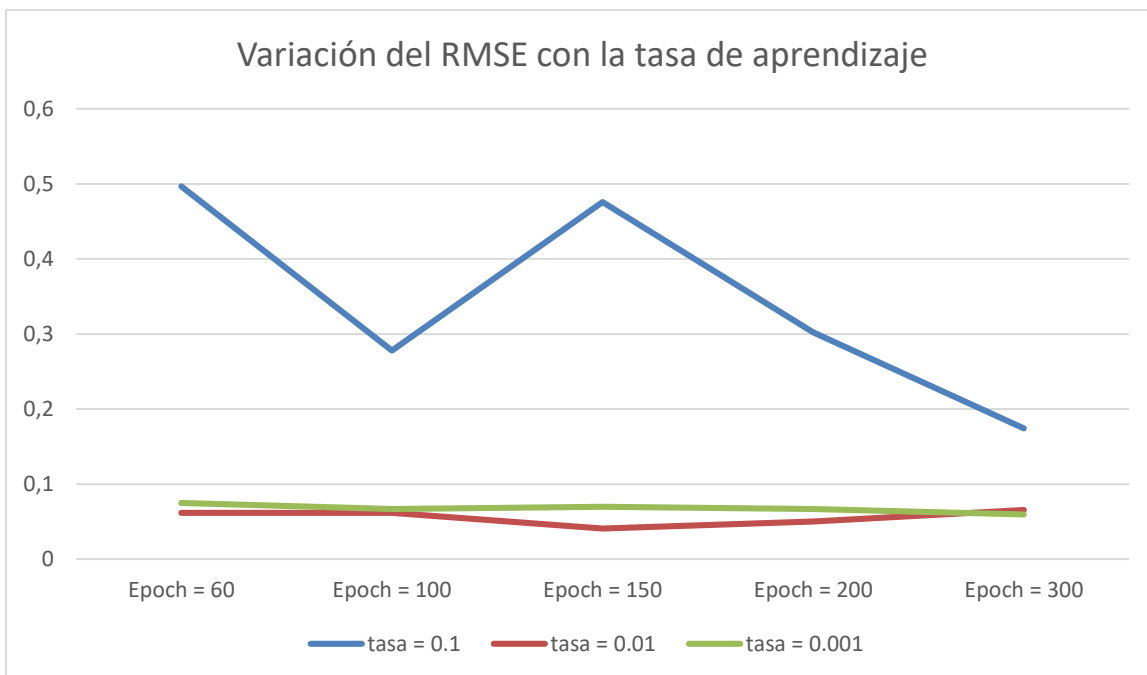


Figura 5.8 Comparación de los valores de RMSE al variar la tasa de aprendizaje y los *Epoch*.

En la figura 5.9 se puede observar una representación conjunta de las GRF medidas experimentalmente y las calculadas mediante la red profunda LSTM de 3 capas en Matlab para el sujeto 14. Se pueden distinguir mejoras significativas en comparación a la representación de la figura 5.6 para la red LSTM simple.

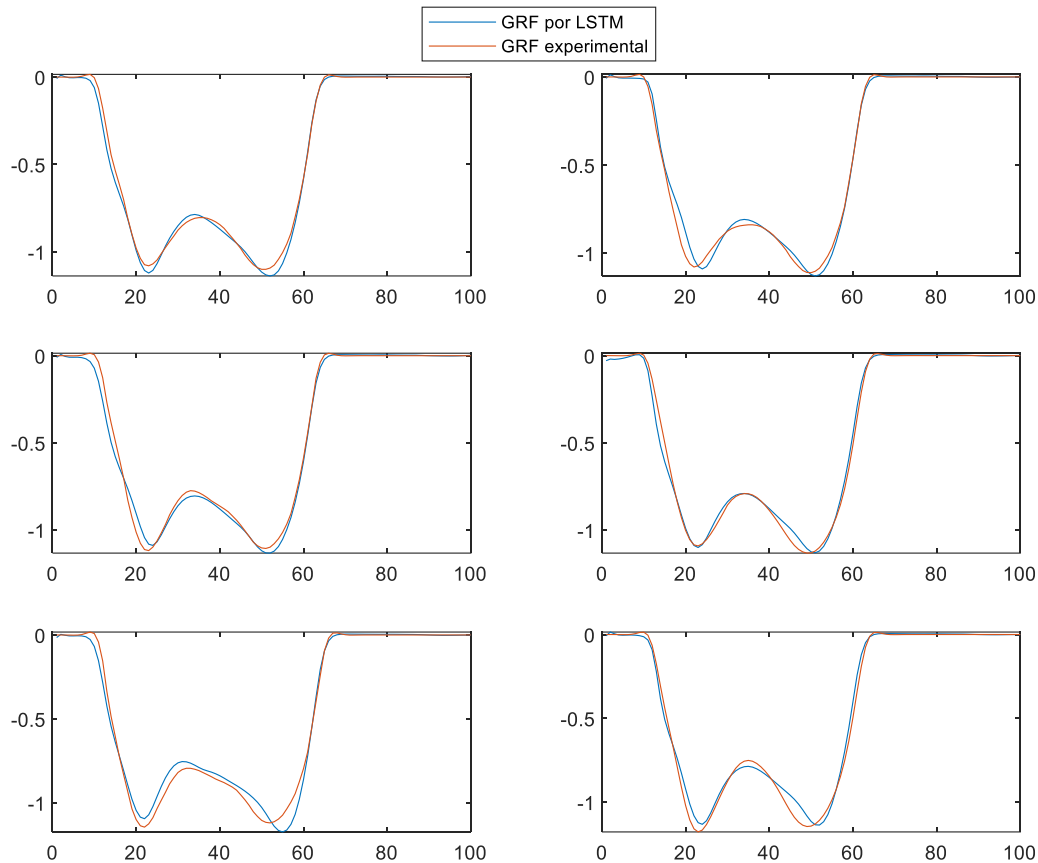


Figura 5.9 Comparación de las GRF experimentales y las simuladas mediante la red LSTM de 3 capas.

### 5.3 App 'Neural Net Fitting'

Para poder comparar los errores obtenidos con los de los apartados anteriores, se ha establecido como datos de entrada del entrenamiento los datos experimentales de los sujetos desde el 1 al 13, dejando el sujeto 14 para comparación.

Dado que el software obliga a dividir los datos aleatoriamente, reservando datos para el testeo, y en el estudio se ha querido testear respecto a los datos del sujeto 14, se especificó el menor porcentaje permitido, 5 %, para dichos datos de testeo. Mientras que, se estableció un inicialmente un 20 % para los datos de validación y un 75 % para los de entrenamiento.

Para comenzar el proceso iterativo, se ha comenzado comparando los resultados para los distintos valores de las neuronas de la capa oculta cuando se usa el algoritmo de Levenberg-Marquardt, el de Bayesian Regularization o el de Scaled Conjugated Gradient. (Figura 5.10) Para la toma de resultados, se ha entrenado la red dos veces y se ha anotado el menor valor de RMSE.

En la figura 5.10 se observa que los valores de RMSE son muy parejos para los tres algoritmos y para cualquier número de neuronas, y que dichos errores son muy superiores a los obtenidos en los apartados 5.1 y 5.2. Lo cual justifica una mayor distorsión de las GRF calculadas gracias a este software y las medidas experimentalmente. (Figura 5.11) Las figuras 5.11, 5.12 y 5.13 muestran la mala representación de los modelos para todas las combinaciones de neuronas ocultas para los algoritmos de SCG, LM y BR respectivamente, donde las pruebas experimentales salen representadas en rojo.

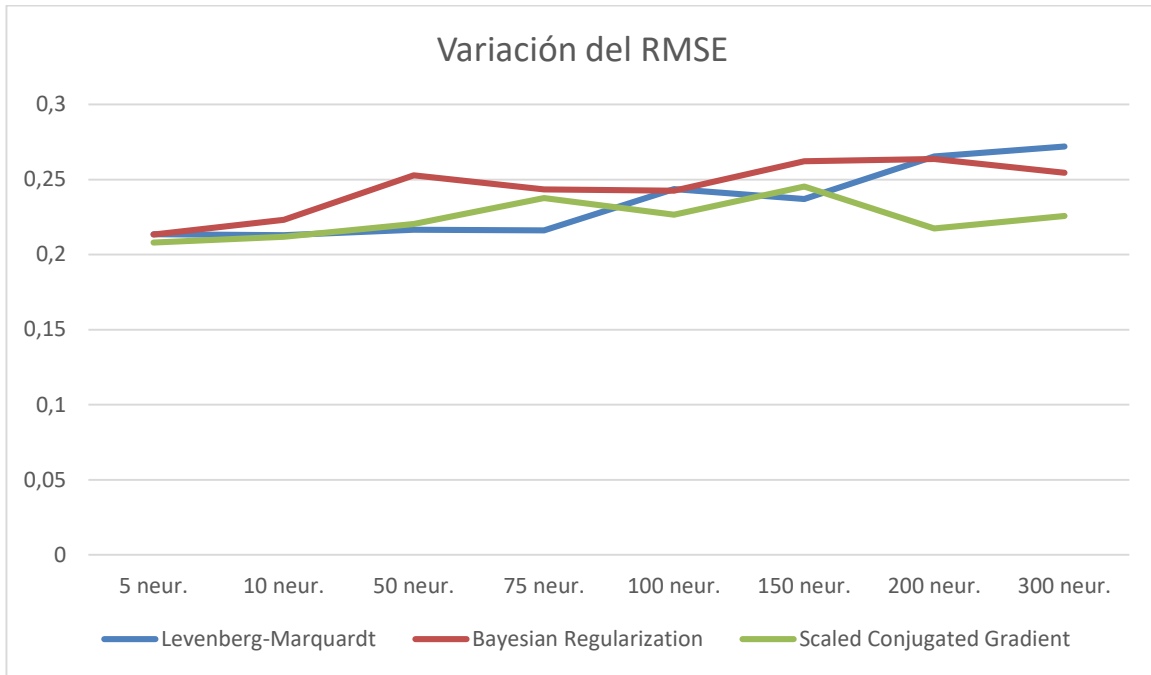


Figura 5.10 Valores de RMSE al variar el número de neuronas y el algoritmo de entrenamiento.

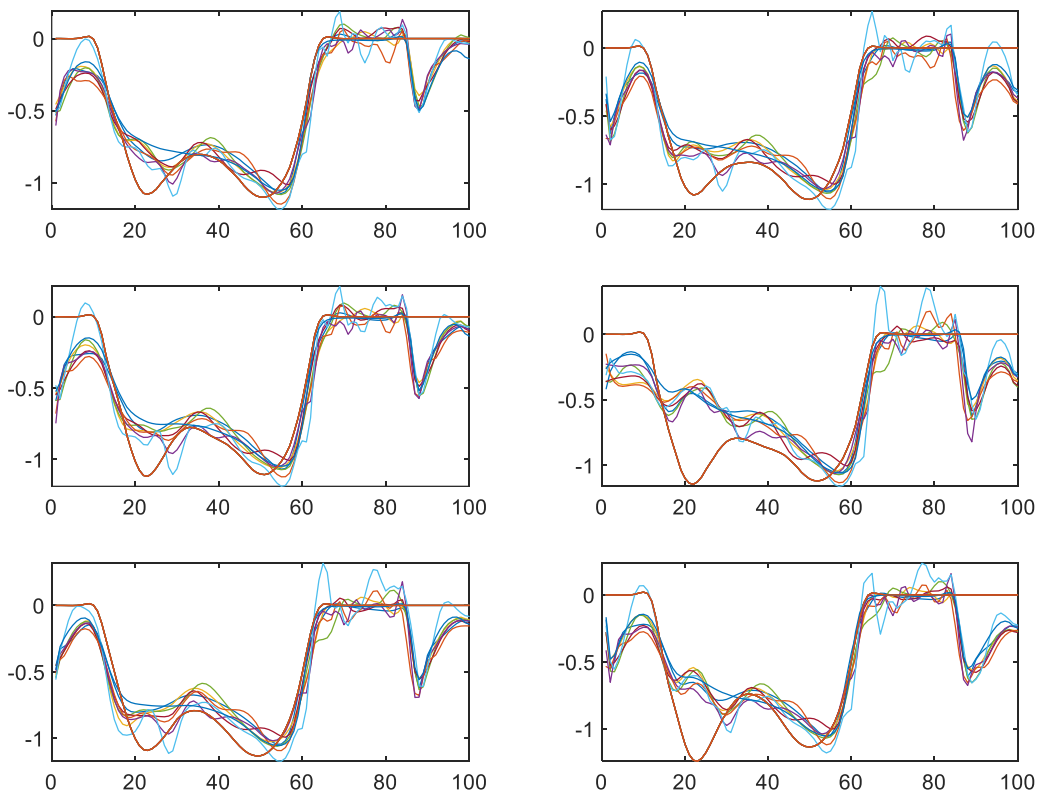


Figura 5.11 Comparación de las GRF experimentales y las simuladas mediante el algoritmo de SCG.

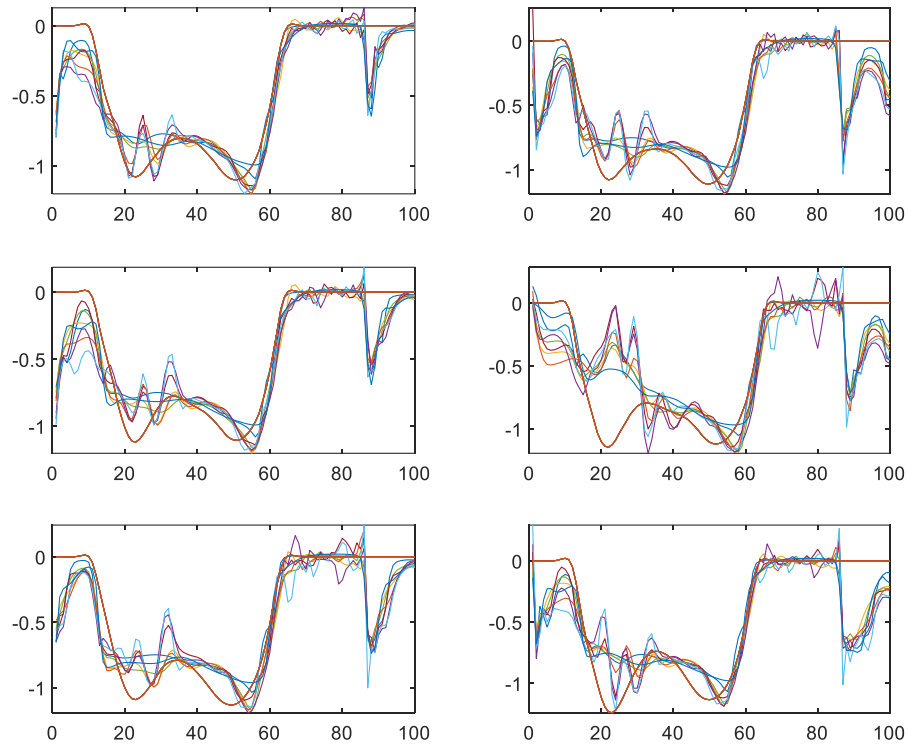


Figura 5.12 Comparación de las GRF experimentales y las simuladas mediante el algoritmo de LM.

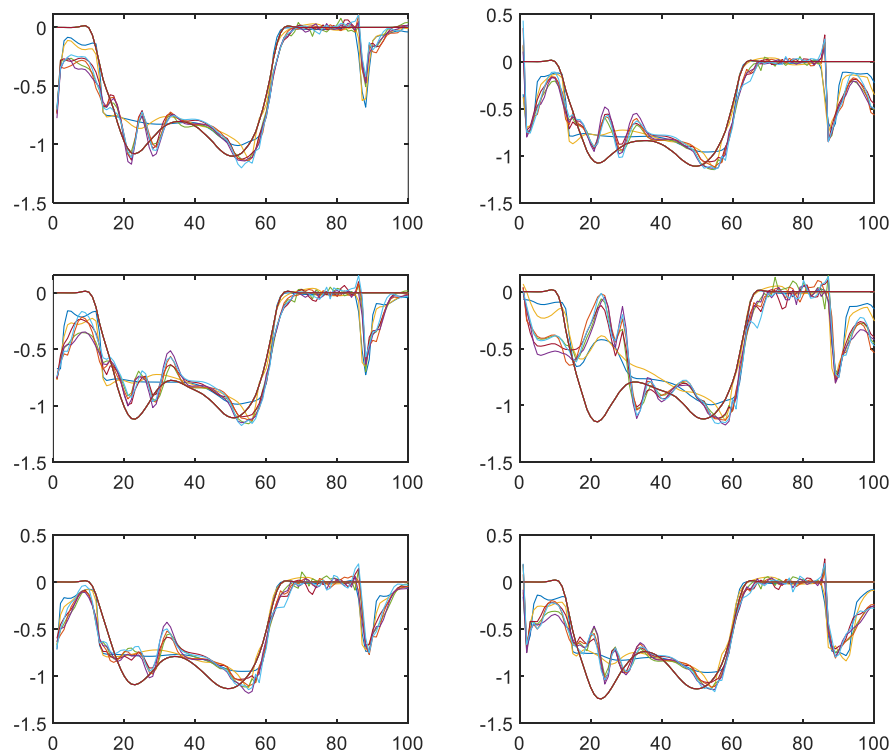


Figura 5.13 Comparación de las GRF experimentales y las simuladas mediante el algoritmo de BR.

## 5.4 Influencia de los datos de entrenamiento

En los apartados anteriores se ha mostrado la precisión de distintas RN para la predicción de las fuerzas de reacción con el suelo sin que los ensayos del sujeto analizado participen en el entrenamiento y con datos pre procesados. En este apartado se analizará la influencia de la selección de los datos de entrenamiento en la eficacia de la predicción del problema.

Dado que los mejores resultados se obtuvieron para las redes LSTM de 3 capas, se trabajará con ella en el estudio, con la estructura y opciones de entrenamiento establecidas como óptimas en el apartado 5.2.

En primer lugar, se tomó como datos de entrenamiento todos los ensayos menos uno de cada uno de los 14 sujetos. Los datos que no fueron utilizados para entrenar la red se utilizaron como datos de testeo, para calcular el RMSE y compararlos con los resultados de los apartados anteriores. Tras el nuevo entrenamiento de la RN del apartado 5.2, se obtuvo un valor mínimo de RMSE de 0.0293, mejorando en un 28.54 % el error cometido cuando el error se media para un sujeto que no había influenciado el entrenamiento.

En segundo lugar, se redujo el número de pruebas que forman los datos de entrenamiento con el objetivo de visualizar como varía el RMSE al variar el tamaño del lote de entrenamiento. De esa manera, se realizó 13 modelados, el primero desde el sujeto 1 al 13, el segundo del 1 al 12 y así sucesivamente hasta que el lote solamente fueran las pruebas del sujeto 1. Además, con el fin de que las comparaciones sean lo más parejas posibles, se ha establecido como lote de testeo las pruebas del sujeto 14 para todos los modelados. (Figura 5.14)

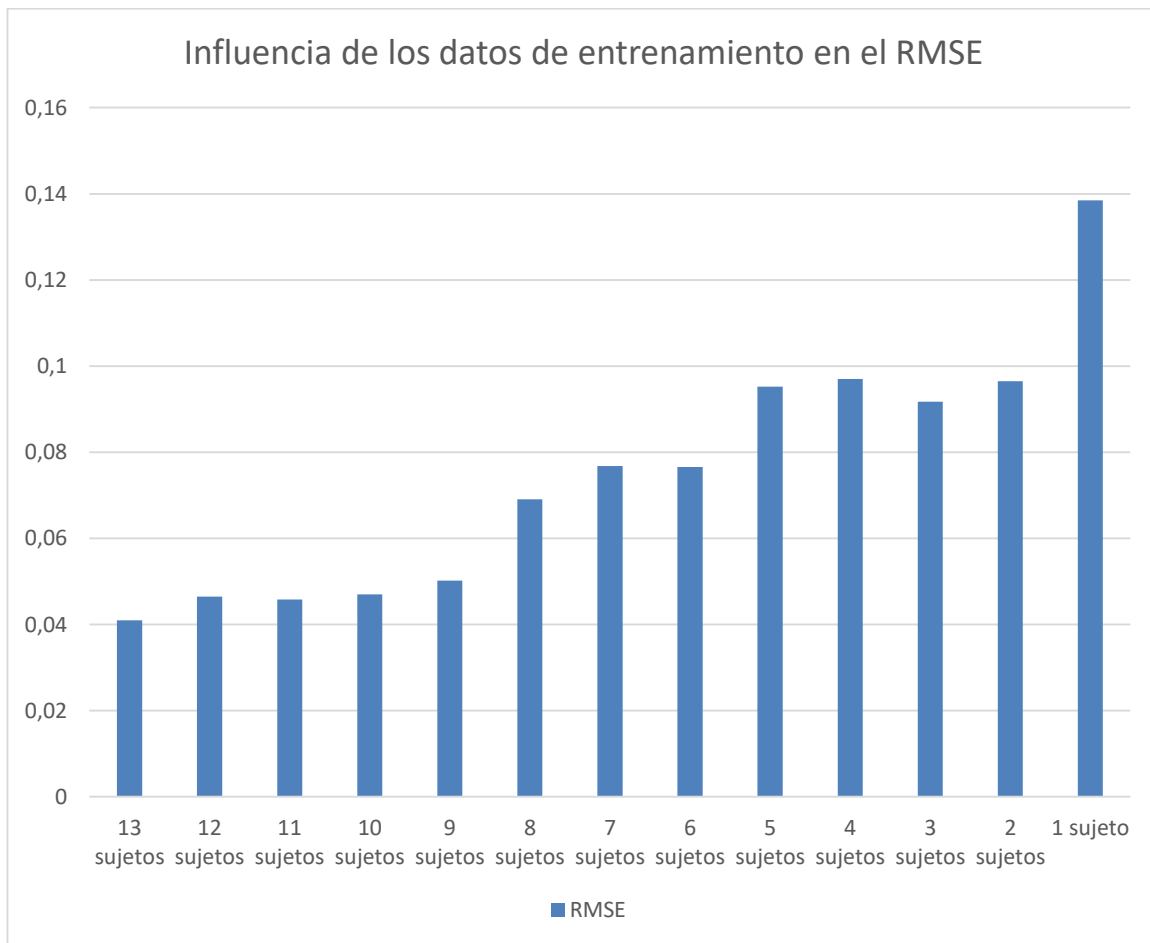


Figura 5.14 Variación del RMSE al variar el tamaño del lote de entrenamiento.

En la figura 5.14 se puede observar que como era de esperar, a menor número de pruebas menor eficacia de la



red, disparándose el error cuando solamente se entrena un sujeto. Además, parece ser que el sistema deja de mejorar considerablemente al llegar a los 9 sujetos de entrenamiento, lo que deja pensar que el RMSE tiene una asíntota alrededor de 0.04. Por lo que, aunque se amplíe la base de datos, no parece que el error mejorase significativamente. Sin embargo, esto es sólo una suposición, la cual únicamente podría comprobarse ampliando los datos de entrenamiento.

Por otro lado, se ha comparado los modelados obtenidos al entrenar al modelo con datos sin procesar o semi procesados. Con el fin de poder comparar los errores de cada simulación, dado que el RMSE no está expresado en tanto por ciento, se dividirá este entre el peso del sujeto para los casos en los que los datos de entrenamiento no estén normalizados con la masa. Los pre procesados aplicados son los desarrollados en el apartado 4.2. (Figura 5.15)

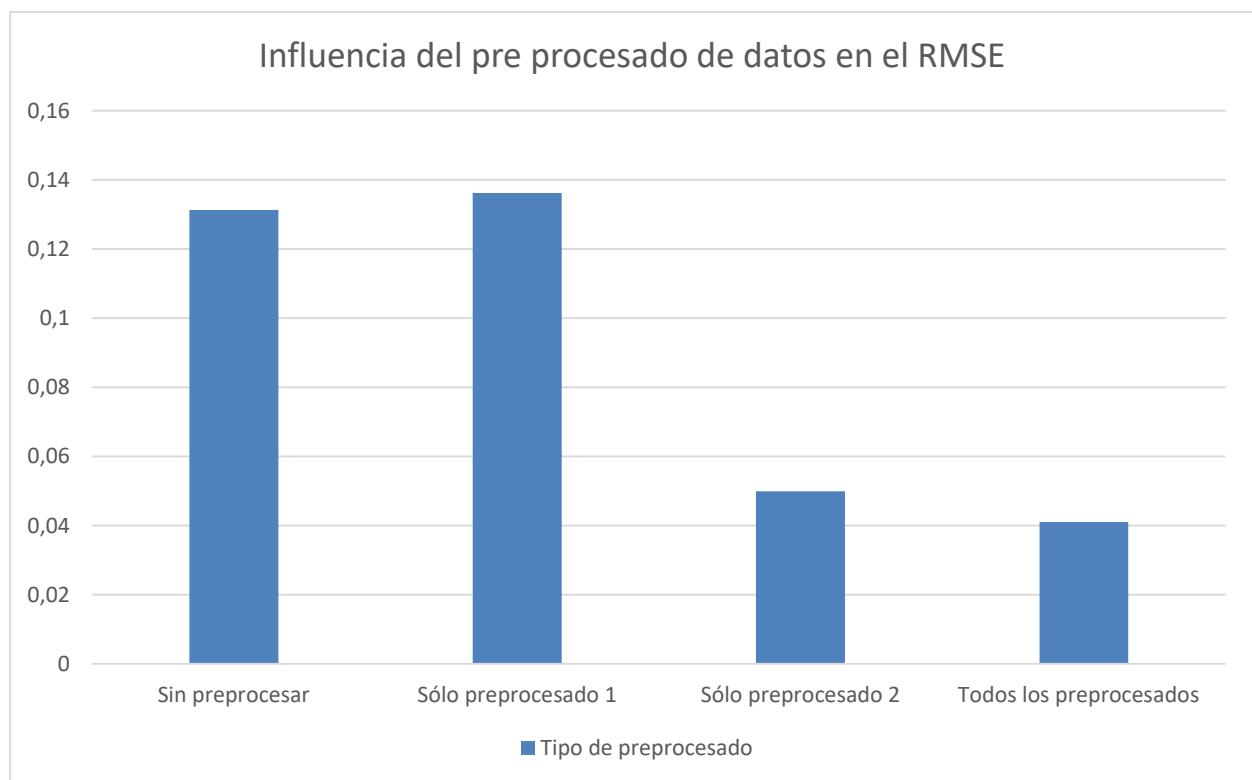


Figura 5.15 Variación del RMSE al variar el pre procesado del lote de entrenamiento.

En la figura 5.15 se muestra claramente la importancia de pre procesar todo el conjunto de datos respecto a la masa de cada sujeto, ya que esto disminuye bruscamente el error cometido. Mientras que, demuestra que el pre procesado del tipo 1 no es influyente en el RMSE, obteniéndose mismos valores de este cuando se aplica el pre procesado y cuando no. Además, en las figuras 5.16 y 5.17 se comparan gráficamente las GRF calculadas para las redes que han sido entrenadas sin normalizar frente la mas, observándose que el sistema es capaz de predecir la amplitud de la fuerza, pero no la forma que esta tendrá, siendo una información inútil para el estudio biomecánico de la GRF. Cabe destacar que para el entrenamiento de las redes sin la normalización con el peso se ha variado la tasa de aprendizaje inicial para acelerar el proceso de entrenamiento, ya que este aumentaba muchísimo dada la dispersión de los valores.

Por último, hasta este momento todos los entrenamientos se han realizado a partir de los ángulos de flexión tobillo, rodilla y cadera conjuntamente. En este último modelado se va a analizar la capacidad de la red para predecir la GRF a partir de uno o dos de ellos solamente. (Figura 5.18)

La figura 5.18 muestra la importancia del ángulo de flexión de la rodilla en el entrenamiento, frente al del tobillo y la cadera, mejorando los resultados cuando al ángulo de la rodilla se le suma el del tobillo. Además, en las figuras 5.19, 5.20 y 5.21 se puede observar la buena aproximación de los resultados a partir cada uno de los ángulos.

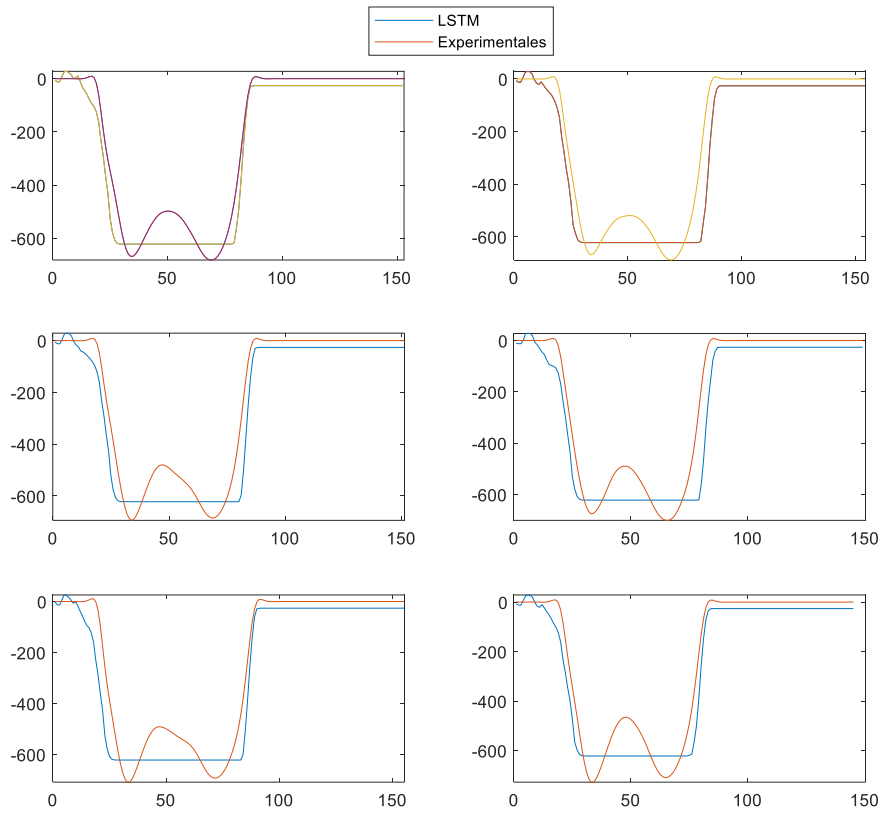


Figura 5.16 Comparación de las GRF experimentales y las simuladas a partir de los datos sin pre procesar.

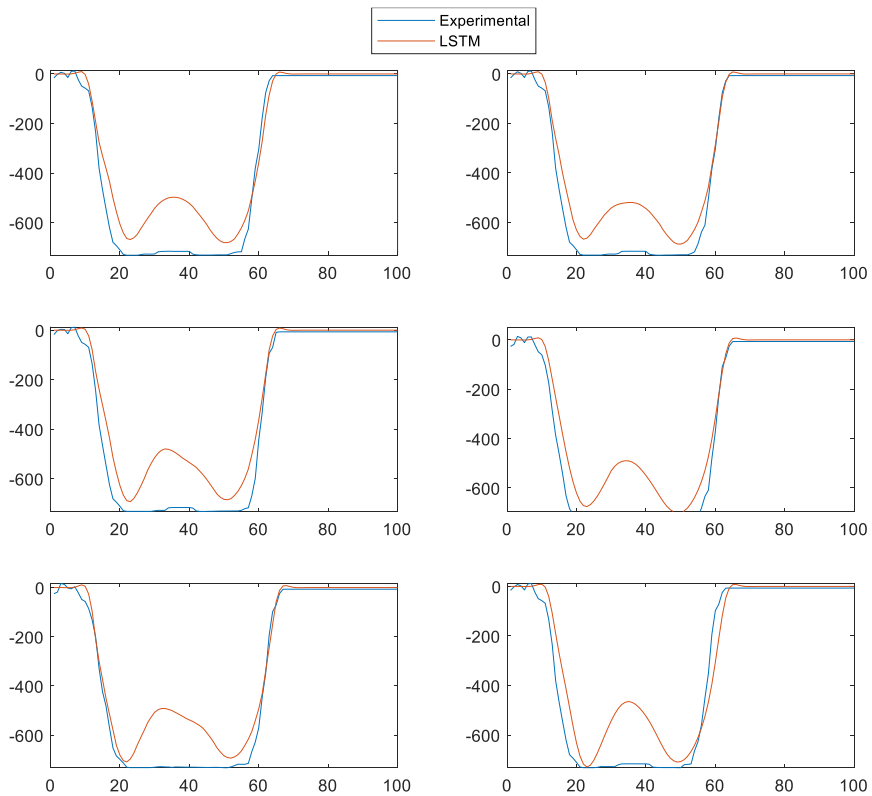


Figura 5.17 Comparación de GRF experimental y simulado a partir de los datos con el pre procesado 1.

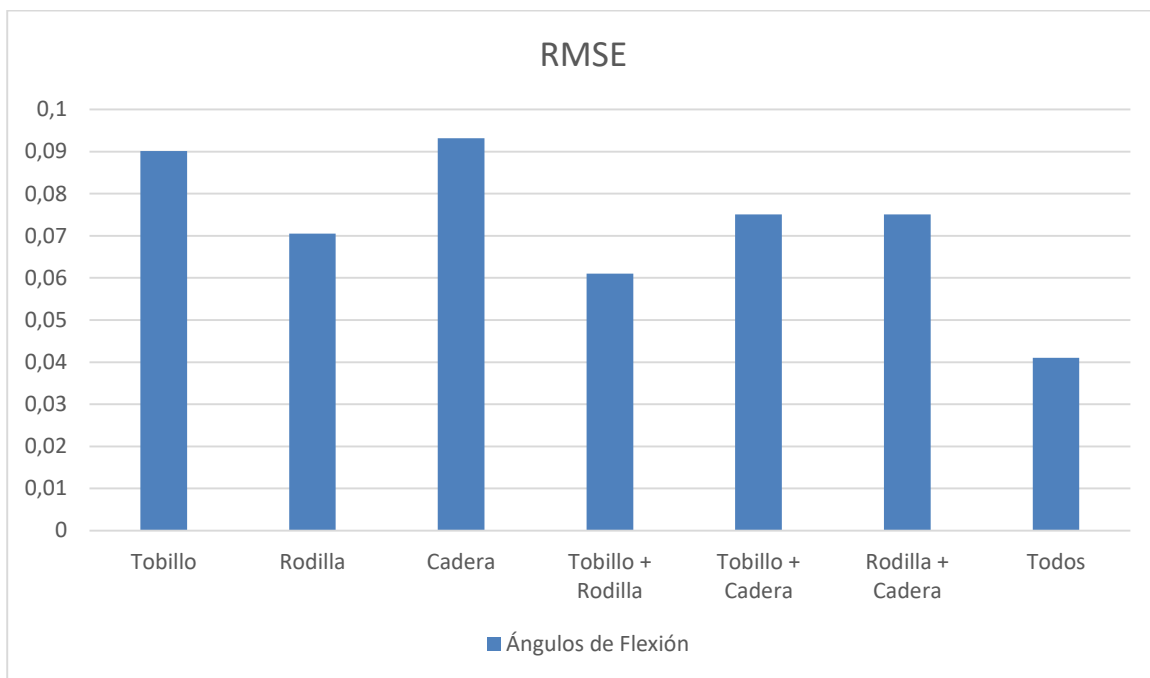


Figura 5.18 Evolución del RMSE al variar los ángulos de entrada como datos de entrenamiento.

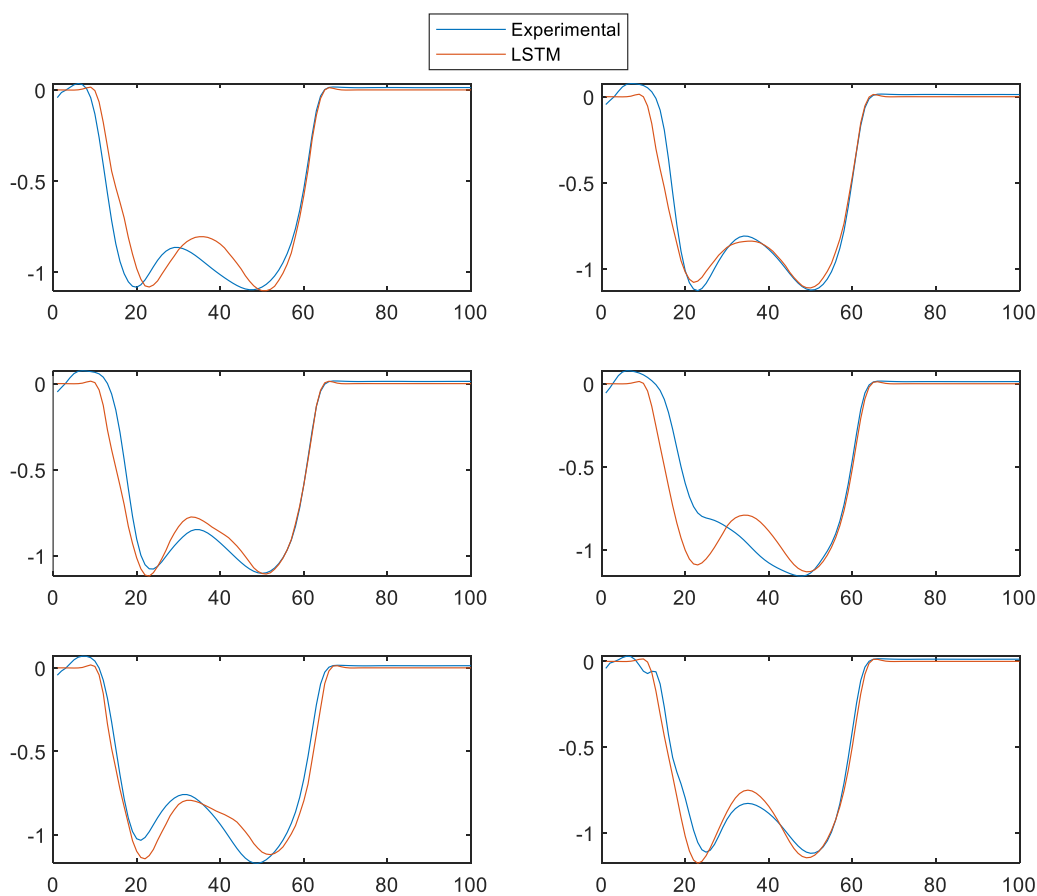


Figura 5.19 Comparación de GRF experimental y simulado a partir del ángulo de flexión del tobillo.

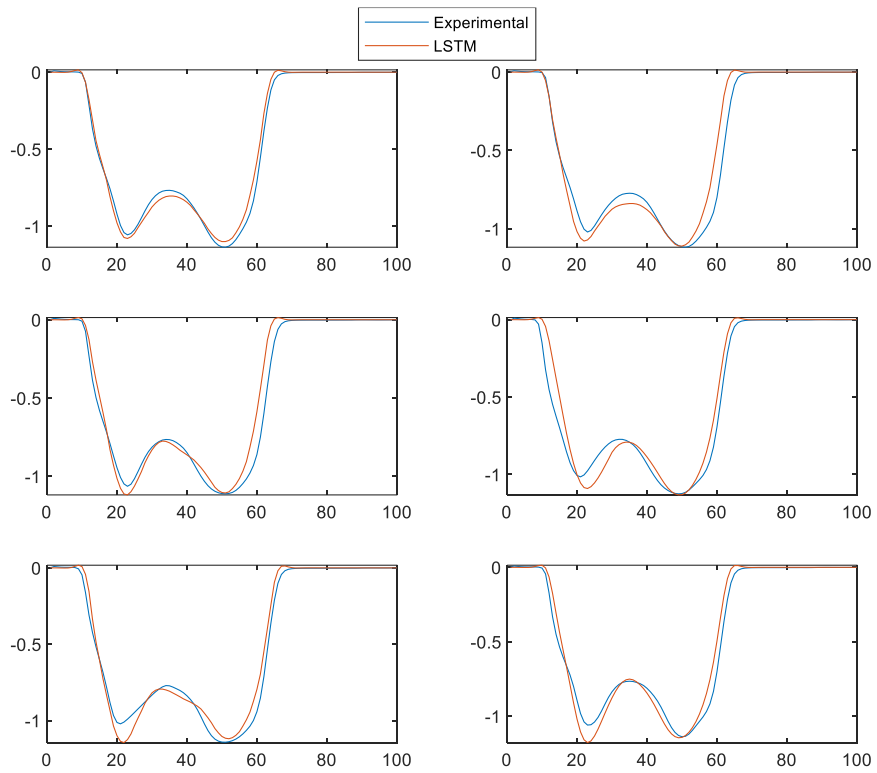


Figura 5.20 Comparación de GRF experimental y simulado a partir del ángulo de flexión de la rodilla.

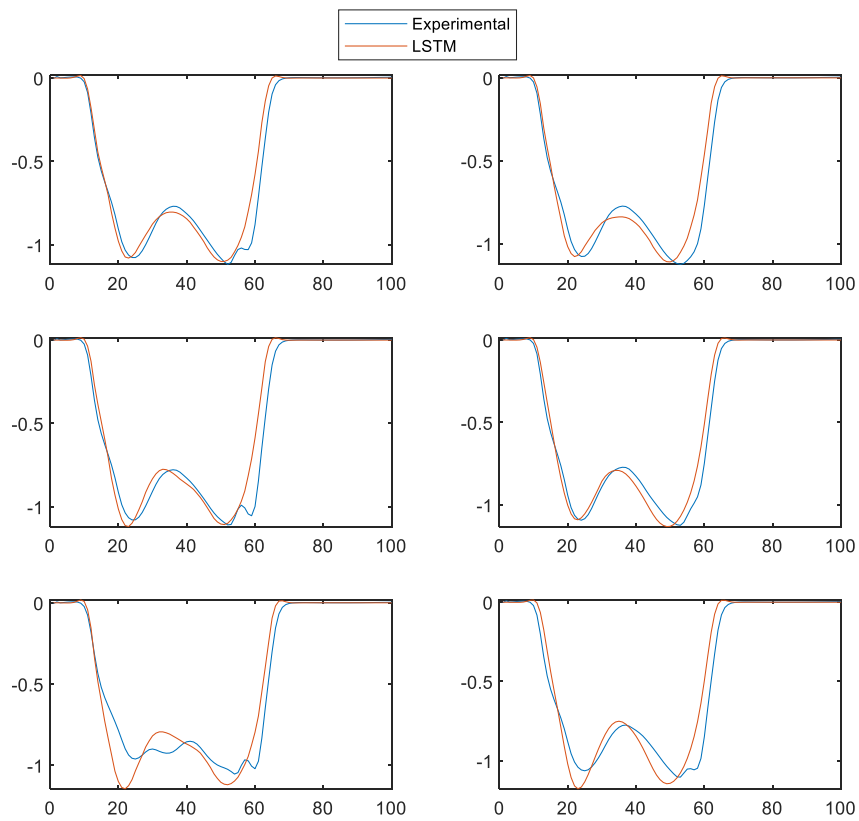


Figura 5.21 Comparación de GRF experimental y simulado a partir del ángulo de flexión de la cadera.

## 6 CONCLUSIÓN

EL objetivo principal de este estudio ha sido analizar la capacidad de las redes neuronales de Deep Learning, implementadas en Matlab 2021a, para modelar las fuerzas de reacción vertical con el suelo de un sujeto sano durante una marcha normal a partir de los ángulos de flexión del tobillo, cadera y rodilla.

Se realizó una comparación de tres enfoques diferentes de redes neuronales y se utilizó el RMSE para evaluar la precisión de la estimación de la GRF. Las técnicas implementadas fueron las redes LSTM simple y profunda, las cuales destacan por su capacidad de retener información en el tiempo por su memoria a corto plazo, y la aplicación 'Neural Net Fitting' de Matlab 2021a, que aporta un mapeado entre datos y salidas sin tener en cuenta el carácter temporal del problema.

Los resultados obtenidos en el trabajo muestran que la LSTM de 3 capas ocultas mejoraba considerablemente los resultados de la LSTM simple y las técnicas proporcionadas por la aplicación de 'Neural Net Fitting'. Siendo el error menor calculado cometido de 0.041. Para optimizar la predicción de las redes LSTM simple y profunda, se empezó comparando los errores cometidos al variar las opciones de entrenamiento de la red y se continuó modificando la estructura de la red, mientras que para las redes de la aplicación 'Neural Net Fitting' se modificó el número de neuronas ocultas y el método de optimización realizado.

Además, con el objetivo de minimizar el error cometido se distinguieron distintas combinaciones de entradas a la red de entrenamiento. En primer lugar, se compararon las redes cuando se entrenaban a partir de pruebas del mismo sujeto que el de las pruebas utilizadas para la predicción y cálculo del RMSE, y cuando ninguna de las pruebas del sujeto de testeo formaba parte de la base de datos del modelo, lo cual mostró una mejoría del RMSE de 0.041 al 0.0293 para el primer caso. Por otro lado, se determinó que cuando aumenta el tamaño del lote de datos de entrenamiento, aumenta la precisión de la predicción. No obstante, los resultados mostraron como la curva de mejora se frena al llegar a valores próximos a los utilizados en el trabajo. Finalmente, con el objetivo de simplificar el modelo, se varió el número de los ángulos utilizados como unidad de entrada, probando la red para cuando solamente se entrena con uno o dos de los ángulos de flexión de la rodilla, tobillo o cadera. La simulación concluyó en una empeora de los errores cometidos y marcó al ángulo de la rodilla como el dato de entrada más significativo, es decir, el que obtenía mejores resultados para cuando se entrenaba en solitario.

En resumen, los resultados obtenidos durante el estudio mostraron a las redes LSTM profundas como una buena aproximación para el cálculo de las fuerzas de reacción con el suelo. También muestra que los mejores resultados se consiguen cuando el sujeto a predecir es previamente ensayado y la red utilizada se entrena con sus datos.

Como mejoras para el futuro destaca que, la base de datos utilizada fue obtenida a partir de un ensayo de laboratorio con marcadores, los cuales están asociados a errores experimentales como el STA. Es por ese motivo que sería recomendable entrenar la red neuronal a partir de datos medidos mediante marcadores fijados al hueso para eliminar gran parte de los errores cometidos y restar errores al GRF simulado. Otra mejora a tener en cuenta sería la utilización de sistemas de medición de ángulos que puedan ser utilizados fácilmente en el exterior y que no exijan la utilización de cámaras de infrarrojos.

Cabe destacar que el estudio solamente se ha visto afectado por la masa del sujeto, es por ello, que futuros estudios podrían centrarse en la normalización de los datos a partir de parámetros como la altura del sujeto o la longitud de la tibia o del muslo.



## 7 CÓDIGO LSTM

```

% Datos de entrada como celdas

XTrain = CellerJA;
YTrain = CellGz;

% Estructura de la red

numFeatures = 3;% Número de entradas (3=ángulos de rodilla, tobilla y cadera)
numHiddenUnits = 200; % Número neuronas ocultas
numResponses = size(YTrain{1},1); % Número de salidas (1)

layers = [ ...
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits, 'OutputMode', 'sequence')
    dropoutLayer(0.2)
    fullyConnectedLayer(numResponses)
    regressionLayer];

% entrenamiento

maxEpochs = 150;
miniBatchSize = 100;

options = trainingOptions('adam', ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'InitialLearnRate',0.01, ... % Tasa de aprendizaje inicial
    'GradientThreshold',1, ...
    'Shuffle','never', ...
    'Plots','training-progress',...
    'Verbose',0);

net = trainNetwork(XTrain,YTrain,layers,options);

% testado
% Cálculo del RMSE

Xtest = CellerJAtest;
Ytest = CellGztest;

YPred = predict(net,Xtest, 'MiniBatchSize',1);
RESTA = 0;
for j=1:1:7 % tamaño de la celda de testado
for i=1:1:100 % tamaño de las pruebas
    a = Ytest{1,j};
    b = YPred{j,1};
    RESTA = RESTA + (a(1,i)-b(1,i)).^2;
end
end
RMSE = sqrt((1/(100*j))*(RESTA))

```





## 8 CÓDIGO LSTM PROFUNDO

```

XTrain = CelleJA;
YTrain = CellGz;

% Estructura de la red

numFeatures = 1;
numHiddenUnits1 = 100; % Número neuronas ocultas de la capa 1
numHiddenUnits2 = 150; % Número neuronas ocultas de la capa 2
numHiddenUnits3 = 100; % Número neuronas ocultas de la capa 3
numResponses = size(YTrain{1},1);

maxEpochs = 200;
miniBatchSize = 100;

layers = [ ...
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits1,'OutputMode','sequence') % capa 1
    dropoutLayer(0.2)
    lstmLayer(numHiddenUnits2,'OutputMode','sequence') % capa 2
    dropoutLayer(0.2)
    lstmLayer(numHiddenUnits3,'OutputMode','sequence') % capa 3
    dropoutLayer(0.2)
    fullyConnectedLayer(numResponses)
    regressionLayer];

% entrenamiento

options = trainingOptions('adam', ...
    'MaxEpochs',maxEpochs, ...
    'MiniBatchSize',miniBatchSize, ...
    'InitialLearnRate',0.01, ...
    'GradientThreshold',1, ...
    'Shuffle','never', ...
    'Plots','training-progress',...
    'Verbose',0);

net = trainNetwork(XTrain,YTrain,layers,options);

% testado

Xtest = CelleJAtest;
Ytest = CellGztest;

YPred = predict(net2,Xtest,'MiniBatchSize',1);
RESTA = 0;
for j=1:1:7
for i=1:1:100
    a = Ytest{1,j};
    b = YPred{j,1};
    RESTA = RESTA + (a(1,i)-b(1,i)).^2;
end
end
RMSE = sqrt((1/(100*j))*(RESTA))

```



## 9 CÓDIGO 'NEURAL NET FITTING'

```

function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Auto-generated by MATLAB, 03-Jul-2021 17:16:31.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 3xQ matrix, input #1
% and returns:
%   y = 1xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [-16.7113177200521;-11.0702458962138;-48.0552788025284];
x1_step1.gain = [0.0187924840377649;0.0219481611250204;0.0300725564381793];
x1_step1.ymin = -1;

% Layer 1
b1 = [-5.9230049971432592315;-5.6305837520686097264;5.5909746370503006219;-
5.4275080921500125086;-
5.2082236221227899975;5.1319035200288967147;4.9676446101004092526;-
4.5128381849942638127;-4.6391306617070124929;-
4.1177131124097723358;4.4017064469838418361;-4.2394892691886463254;-
3.9544891195534654926;3.2980491206038378138;-
3.6817093393802950985;3.3690866695649117624;3.4427361457075216222;-
3.0467561648587797585;-3.6908385456250196732;-2.7168598852934664478;-
2.2208633613282038333;2.4339240503105670221;-2.295226432094298108;-
2.0578407002016327887;-2.3524822994036447277;-2.4029069229465322977;-
1.9956959406664556145;2.4892323590940290678;-
1.084858294195199635;1.0271591848461754104;-
1.7237093726928991444;1.31774303050624364;1.414801034970631255;0.793499614878
35540819;-1.0306549136573823766;-0.53398548765825049056;-
1.6820553058905831634;-0.15944949538106931164;0.42286931404452826611;-
0.16014335537629503792;-
1.6314788864568341431;0.38139465247769555045;1.6642467043946345129;-
1.5387855986867000624;-1.5113273042407258107;-2.2189913076036553718;-
0.94725268171281884477;0.82058993688656545196;2.1895729625773712179;2.0932189
72238783504;-
2.1930632189183283387;1.7834476892808421855;2.3175864334513471832;-
2.4088916330577592184;-1.5240872377827856532;-
3.1132820813431871798;4.464680323132008688;-
3.6569133432706113496;2.9345680843556238848;-
3.1104258455043765075;3.1435126262014829734;-
2.2959601575010504249;4.5396558259314661044;3.4869396331954209955;4.367795668
6047387791;2.9331138215007199754;4.6008531761351720846;-
7.8293406098582964958;-3.6594361735757461318;-
5.2401668810325894299;5.2342358724329791642;-
5.2759844458532949574;5.2592978605710243301;-
6.9883175683832901015;5.418899557015434354];
IW1_1 = [5.1890146374433072651 -2.4655953903439304931 -
1.3532176594199889497;3.5934571709482741042 -2.7035692032064848611 -
3.9772828468801968427;-3.6732790108262749484 -2.4099323122405680309
3.9448082995837490117;3.9488387188240441361 1.1086249234764098404
4.2457613326909138252;5.8101726141469764997 -1.1999766308925623459 -

```

0.42801967217801967225;-5.5125756516814270469	-2.0556131175933671784	-
0.160711207338075035;-3.1013693469000567227	4.5187149594021196464	-
2.1917154727476391152;0.17268282341306742667	0.45390752545407120344	-
6.2347498189107213307;5.6481754681699918308	0.61627185069667222983	-
1.6073127274719822832;3.7923027843161793982	3.5681289608948087988	-
3.3060901941357232481;-4.2274250715942791601	-2.4115795832383279951	-
3.3085559401249207134;5.004964427779786007	2.7887059651606649346	-
1.1624713328380922306;4.571533548602269903	2.6208459857993866571	-
2.6567545323159378334;-0.24117505562096908101	-4.6958573732852491034	-
3.8982555656296167257;3.8144450344107503348	0.85153201148489265915	-
4.5155082906814918431;-2.9394029303572204448	-3.8867218455756922246	-
3.3279489952933869468;-0.71430895422446860632	-4.3008449278100640001	-
3.7423638432035537527;3.996806228031656083	0.34037812799847771972	-
4.4080575560384360756;4.2322120243177119292	3.3972312076521649082	-
1.1823600979816364376;3.4647706274519434189	3.0641883014769581983	-
3.8517399618848222609;1.3060850616309800376	-5.709886025199887527	-
2.1079724133355055571;-5.633326914946178654	0.88634620327244251214	-
1.6974135272528196872;4.4249622810134159323	-0.39159247388153572977	-
3.9642958916114139356;3.1254375965729526143	-5.5449192975000860528	-
4.5112720095404341691;2.3383558321280366599	-0.36759734908520863561	-
6.8081304457098239524;2.82778463190850049	-3.4448453720828071667	-
3.353992286199612316;3.7323753264689272235	4.1599979066148451423	-
1.9490217539483636333;-2.1701070558456638793	-2.5770887404111393337	-
5.4261533110161090931;4.2500212985065566329	0.29017860353985208688	-
4.2336143667721684025;-4.0989533814023744185	-4.191277517063208613	-
0.92047533486253940715;5.3060873610130538935	2.67847989616682014	-
0.77505711178618863499;-5.120362903393689713	-2.357543930820128697	-
1.9333342310344907578;-4.1205263595835228685	2.3991172042061381653	-
3.5868632976636654064;-4.608433771925287914	2.0192267162565360472	-
3.2533646173651664668;-4.2546023585808701029	2.1445983530433814224	-
1.1701921665899637226;4.4986069551408149891	0.049877041390386331332	-
3.9137598312531820532;0.77587209396072653877	-7.0252888571915379501	-
2.3826386446795431162;-3.0397154616908581559	-4.6679073143234317556	-
2.0861161732808084501;3.4519539959577496546	3.5403143586160541467	-
2.7919572942752224343;-3.8537912598808823361	-2.537597879184797718	-
3.4893792890772217241;-0.2965894933047452664	3.0770595059050220854	-
5.6319477818858771911;-3.3225010230092033581	0.94523151530870230808	-
8.7524447830376015389;1.0527590507974580181	8.2708912883686931394	-
1.8285083460165976188;1.5432116594711877688	-6.7993266706263275623	-
0.95202060848047898034;-5.461441069936800119	-1.2104631504713041501	-
1.1223070354119488723;0.030727379426830354892	-6.1200526195899502824	-
3.7994461307159652641;-1.0591315933645861769	5.9273636066512889187	-
0.56074008937071906367;2.361115491065091998	-6.7946337637640539242	-
0.71187454196672095286;3.6449018006357882804	1.8630984138901036662	-
4.3019544092434491844;2.3296717997471478867	4.7435634356868279227	-
4.3213827269295457256;-3.0106722628198965275	5.2190442837860802072	-
1.4152865412789428223;3.0514841212206866849	-6.3869520425590682322	-
1.4890798639787143909;1.4461404155343979827	-1.5146992214716152869	-
6.8852396989187658605;-5.0767996608656140722	-2.7767743017534680661	-
0.20063276307278182786;-4.9864090740009423541	3.7317954988762567226	-
2.2466781181160673242;0.18522616698645066879	-0.90369660179017385904	-
5.5260672748039647573;3.7662218177653468132	3.25032036643998401	-
2.5564504728515538012;-1.8910010520228923703	-6.0737822831985210925	-
1.8404431070534823522;2.4603430364958320808	-6.7275042154150295204	-
2.5909918811131964489;-2.4464431265994233122	-5.7788879515303399614	-
0.23452016889769186636;2.6793808363659366023	-4.8467600952993743491	-
4.7332106245887768381;-1.6361966468791939722	6.6768613873072988341	-
3.19104939658602893;-0.21875187835255699431	4.7438274175752512818	-
1.6507895144847006819;4.5293265504774788965	0.68228766249446004544	-
6.2526834571820550934;3.7126358962841585765	-2.9733917879824272035	-
3.6725984298350429214;4.2251265647225029198	1.4087081927472491216	-
5.9813213093508919016;0.54176867155173147328	2.8331443221921275288	-

```

5.3289649815401114452;-6.2073311825343164827      -4.9637431797272739686      -
4.2569547763767365822;-4.0289666475507885934      0.45391681524141253012      -
3.393492556831368745;-6.2674084939158349883      -1.7171048597427920335      -
2.0844695904410537324;3.3966682109938890477      -4.285997109524686266      -
2.1017591068859449166;-2.4067855741727135488      -2.5547529187816198082      -
4.6065843220911499856;3.6930749050981930282      -4.7195393162896914774      -
1.0621350013851291738;-4.3075693001304342644      -5.4746595739711869655      -
4.1191466365403401895;4.5523259511084397388      -3.3652381363187817698      -
3.5156535018750374455];

```

```
% Layer 2
```

```
b2 = -0.68039796613929959257;
```

```

LW2_1      =      [0.92857749615615881034      -0.31805455699930268443      -
0.57208750378988870722      -0.68244656579979512401      -0.18589859743308784212
0.34901231639149965513      -0.27461594448983128736      -0.20897026596722820257
0.39224435923923234304      -1.1401042213808918468      -0.56700318047115427778      -
0.47523354003334794537      -0.48138244169293820107      -0.27765586671852643397      -
0.40349572832522445642      -0.61441673609656877186      0.11200479239809481291
0.56659362422447578922      -0.58648240317565369928      -0.71442692057490275737
2.4420303685995268239      0.30332964719905841644      0.41534591122324326928      -
0.67493020307262385771      0.62203011200898028488      0.66310499540680378772      -
0.19385729516478072965      0.40400691794701337756      -0.4178073797214684193      -
0.012462154565277629853      0.94870085410689874994      0.50530672298523682251
0.90011698560597164231      -0.088039208926740691874      1.3111813911871221805
1.36939065430435436      2.6324524221485088304      0.21866828081900260838
0.54759619312936913538      0.33671841488957221511      -0.349200239584722405      -
0.70705514457535800776      1.248268082072681695      -1.1813542216214443226      -
0.0672574366477877994      -1.6535401990673574968      0.5144002726207176357      -
1.2612697435278374947      -0.82869817236859044662      0.86155089803813045268
0.18607728190709668659      2.0044979703147975947      -0.38034970903658404939
0.15225353393598733964      -0.54742098418373819069      -0.1725258600560506117
1.6572678530134827835      -0.99158808295342215633      1.0296122788805881498
0.9866785761405547639      -0.72503809439916122592      1.0199452683185685764
0.3331528644478150003      1.0425701922023031809      0.75133270055248746466      -
0.8279420954643359476      -0.41323341260182888668      3.176593713192873647      -
0.78486452407895435091      0.89492844743383836992      -0.55460716042756008637      -
0.4137339359582415943      0.35253228682270132399      -2.5686155725876949063      -
0.47145665538156589225];

```

```
% Output 1
```

```
y1_step1.ymin = -1;
```

```
y1_step1.gain = 1.39308203879941;
```

```
y1_step1.xoffset = -1.40893428082795;
```

```
% ===== SIMULATION =====
```

```
% Dimensions
```

```
Q = size(x1,2); % samples
```

```
% Input 1
```

```
xp1 = mapminmax_apply(x1,x1_step1);
```

```
% Layer 1
```

```
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);
```

```
% Layer 2
```

```
a2 = repmat(b2,1,Q) + LW2_1*a1;
```

```
% Output 1
```

```
y1 = mapminmax_reverse(a2,y1_step1);
```

```
end
```

```
% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x, settings)
y = bsxfun(@minus, x, settings.xoffset);
y = bsxfun(@times, y, settings.gain);
y = bsxfun(@plus, y, settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n, ~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y, settings)
x = bsxfun(@minus, y, settings.ymin);
x = bsxfun(@rdivide, x, settings.gain);
x = bsxfun(@plus, x, settings.xoffset);
end
```

## REFERENCIAS

- [1] A. Fernández, «Tecnologías de Inteligencia Artificial y sus categorías,» AuraQuantic, s.f. [En línea]. Available: <https://www.auraquantic.com/es/tecnologias-de-inteligencia-artificial-y-sus-categorias/>. [Último acceso: 2021 Junio 21].
- [2] Parlamento europeo, «¿Qué es la inteligencia artificial y cómo se usa?,» 8 Septiembre 2020. [En línea]. Available: <https://www.europarl.europa.eu/news/es/headlines/society/20200827STO85804/que-es-la-inteligencia-artificial-y-como-se-usa>. [Último acceso: 2021 Junio 21].
- [3] Iberdrola, «¿Qué es la inteligencia artificial?,» 2021. [En línea]. Available: <https://www.iberdrola.com/innovacion/que-es-inteligencia-artificial>. [Último acceso: 21 Junio 2021].
- [4] Agencia Universitaria DQ, «Inteligencia artificial ventajas y desventajas,» s.f. [En línea]. Available: <https://agenciauniversitariadq.online/inteligencia-artificial-ventajas-y-desventajas/>. [Último acceso: 21 Junio 2021].
- [5] Iberdrola, «Qué es el Machine Learning,» s.f. [En línea]. Available: <https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico>. [Último acceso: 21 Junio 2021].
- [6] F. Sancho Caparrini, «Aprendizaje Supervisado y No Supervisado,» Disqus, 14 Diciembre 2020. [En línea]. Available: <http://www.cs.us.es/~fsancho/?e=77>. [Último acceso: 21 Junio 2021].
- [7] P. Recuero, «¿Sabes en qué se diferencian las redes neuronales del Deep Learning?,» Blogthinkbig, 26 Abril 2019. [En línea]. Available: <https://blogthinkbig.com/redes-neuronales-deep-learning>. [Último acceso: 21 Junio 2021].
- [8] P. Larrañaga, I. Inza y . A. Moujahid, *Redes Neuronales*, Universidad del País Vasco–Euskal Herriko Unibertsitatea: Departamento de Ciencias de la Computación e Inteligencia Artificial.
- [9] D. Calvo, «Perceptrón – Red neuronal,» 2008 Diciembre 2018. [En línea]. Available: <https://www.diegocalvo.es/perceptron/>. [Último acceso: 21 Junio 2021].
- [10] L. Manjarrez, «Capas de una red neuronal,» ResearchGate, Junio 2014. [En línea]. Available: [https://www.researchgate.net/figure/Figura-III4-Capas-de-una-Red-Neuronal-Capa-de-entrada-neuronas-que-reciben-datos-o\\_fig3\\_315762548](https://www.researchgate.net/figure/Figura-III4-Capas-de-una-Red-Neuronal-Capa-de-entrada-neuronas-que-reciben-datos-o_fig3_315762548). [Último acceso: 21 Junio 2021].
- [11] Ramanujan, «Redes Neuronales y Aprendizaje profundo,» Blogspot, 7 Octubre 2017. [En línea]. Available: <http://ramanujan25449.blogspot.com/2017/10/redes-neuronales-y-aprendizaje-profundo.html>. [Último acceso: 21 Junio 2021].
- [12] Na8, «Breve Historia de las Redes Neuronales Artificiales,» AprendeMachineLearning, 12 Septiembre 2018. [En línea]. Available: <https://www.aprendemachinlearning.com/breve-historia-de-las-redes-neuronales-artificiales/>. [Último acceso: 21 Junio 2021].

- [13] D. Calvo, «Clasificación de redes neuronales artificiales,» 13 Julio 2017. [En línea]. Available: <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>. [Último acceso: 21 Junio 2021].
- [14] Mathworks, «Redes neuronales convolucionales,» Mathworks, s.f. [En línea]. Available: <https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html>. [Último acceso: 21 Junio 2021].
- [15] V. S. Buitrago, Simulador de las redes de bas radial, Universidad Carlos III de Madrid: Trabajo de Fin de Grado.
- [16] SmartPanel, «¿Qué es el Deep Learning?,» SmartPanel, 10 Abril 2021. [En línea]. Available: <https://www.smartpanel.com/que-es-deep-learning/>. [Último acceso: 21 Junio 2021].
- [17] Universidad de Alcalá, «El origen del Deep Learning,» s.f. [En línea]. Available: <https://master-deeplearning.com/origen-deep-learning/>. [Último acceso: 21 Junio 2021].
- [18] Enzyme Advising Group, «Redes Neuronales Artificiales y Deep Learning, explicado para dummies,» Enzyme, 14 Agosto 2019. [En línea]. Available: <https://blog.enzymeadvisinggroup.com/redes-neuronales-artificiales-y-deep-learning>. [Último acceso: 21 Junio 2021].
- [19] P. Recuero, «¿Sabes en qué se diferencian las redes neuronales del Deep Learning?,» BlogThinking, 26 Abril 2019. [En línea]. Available: <https://blogthinkbig.com/redes-neuronales-deep-learning>. [Último acceso: 21 Junio 2021].
- [20] L. Fuertes, «¿Qué es deep learning?: la evolución de la inteligencia artificial,» Crehana, 4 Febrero 2021. [En línea]. Available: <https://www.crehana.com/es/blog/tech/que-deep-learning/>. [Último acceso: 21 Junio 2021].
- [21] D. Knudson, Fundamentals of biomechanics, California State University at Chicago: Springer, 2007.
- [22] R. De Azevedo Guaura, «Planos anatómicos,» Lifeder, 12 Diciembre 2020. [En línea]. Available: <https://www.lifeder.com/planos-anatomicos-ejes/>. [Último acceso: 21 Junio 2021].
- [23] J. Saez Hurtado, «Ejes de movimiento de articulaciones,» Paradigma, s.f. [En línea]. Available: <https://paradigmia.com/curso/locomotor/modulos/generalidades-de-las-articulaciones/temas/ejes-de-movimiento-de-articulaciones/>. [Último acceso: 2021 Junio 21].
- [24] Curiosoando.com, «¿Cuál es la diferencia entre flexión y extensión?,» 26 Abril 2020. [En línea]. Available: <https://curiosoando.com/cual-es-la-diferencia-entre-flexion-y-extension>. [Último acceso: 21 Junio 2021] «Los diferentes tipos de pisada en el running,» Fisiohm, s.f. [En línea]. Available: <https://fisiohm.com/los-diferentes-tipos-pisada-running/>. [Último acceso: 2021 Junio 21].
- [25] Teckscan, «The Gait Cycle: Phases, Parameters to Evaluate & Technology,» Teckscan, [En línea]. Available: <https://www.teckscan.com/blog/medical/gait-cycle-phases-parameters-evaluate-technology>. [Último acceso: 21 Junio 2021].
- [26] UdeA, «Biomecánica de la marcha,» Aprendeonlinea, 16 Febrero 2016. [En línea]. Available: <http://aprendeonlinea.udea.edu.co/lms/moodle/login/index.php>. [Último acceso: 21 Junio 2021].
- [27] P. S. Arun, «Normal Gait Cycle and Abnormalities of Gait,» BoneAndSpine, 2017. [En línea]. Available: <https://boneandspine.com/normal-gait-cycle/>. [Último acceso: 2021 Junio 21].



- [28] Alonso Sánchez, F. J. (Octubre de 2005). *Los errores del análisis biomecánico en el cálculo de las fuerzas de reacción en las articulaciones*. Obtenido de Escuela de Ingenierías Industriales, Universidad de Extremadura: <https://www.efdeportes.com/efd89/biom.htm>
- [29] R. Johnson, W., Mian, A., A. Robinson, M., Verheul, J., G. Lloyd, D., & A. Alderson, J. (2020). *Multidimensional ground reaction forces and moments from wearable sensor accelerations via deep learning*. IEEE
- [30] Conte Alcaraz, J., Moghaddamnia, S., & Peissig, J. (2021). *Efficiency of deep neural networks for joint angle modeling in digital gait assessment*. SpringerOpen.
- [31] Sharma, D., Davidson, P., Müller, P., & Piché, R. (2021). *Indirect Estimation of Vertical Ground Reaction Force from a Body-Mounted INS/GPS Using Machine Learning*. Sensors.
- [32] J. Ojeda Granja, Aplicación de las técnicas MBS al sistema locomotor humano, Universidad de Sevilla: Tesis doctoral, 2012.
- [33] US, «Bibing,» [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/5315/fichero/05-Protocolo+de+reconstrucci%C3%B3n.pdf>. [Último acceso: 21 Junio 2021].
- [34] Mathworks, «Redes neuronales convolucionales,» Mathworks, s.f. [En línea]. Available: <https://es.mathworks.com/discovery/convolutional-neural-network-matlab.html>. [Último acceso: 21 Junio 2021].
- [35] J. Durán, «Técnicas de Regularización Básicas para Redes Neuronales,» MetaDatos, 8 Octubre 2019. [En línea]. Available: <https://medium.com/metadatos/t%C3%A9cnicas-de-regularizaci%C3%B3n-b%C3%A1sicas-para-redes-neuronales-b48f396924d4>. [Último acceso: 2021 Junio 21].
- [36] M. Bernico, «Controlling variance with dropout,» O'reilly Media, 2021. [En línea]. Available: <https://www.oreilly.com/library/view/deep-learning-quick/9781788837996/d3e5de85-0d34-4a20-a4ca-3c498b1f5f7a.xhtml>. [Último acceso: 21 Junio 2021].
- [37] A. B. Barrueta, Estudio de diferentes modelos de redes neuronales para el desarrollo de un clasificador de frases, Universidad del País Vasco: Trabajo de Fin de Grado, 2019.
- [38] J. Brownlee, «Difference Between a Batch and an Epoch in a Neural Network,» Machine Learning Mastery, 20 Julio 2018. [En línea]. Available: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>. [Último acceso: 21 Junio 2021].
- [39] J. Brownlee, «Understand the Impact of Learning Rate on Neural Network Performance,» Machine Learning Mastery, 25 Enero 2019. [En línea]. Available: <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>. [Último acceso: 2021 Junio 21].



# GLOSARIO

3D: tres dimensiones.....	11
PiG: Plug-in-Gait.....	11
UK: Reino Unido.....	11
STA: Error producido por el movimiento relativo de la piel respecto al hueso.....	11
DL: Deep Learning.....	16
IA: Inteligencia Artificial.....	16
ML: Machine Learning.....	17
RN: Red Neuronal.....	18
CNN: Red Neuronal Convolutacional.....	20
ReLU: Unidad lineal rectificada.....	20
RNN: Red Neuronal Recurrente.....	20
LSTM: Long-Short Term Memory.....	20
RBF: Red neuronal de base radial.....	21
GRF: Fuerza de reacción con el suelo.....	23
RMSE: Error cuadrático medio.....	25
LM: Algoritmo de entrenamiento de Levenberg-Marquardt.....	28
BR: Algoritmo de entrenamiento de Bayesian Regularization.....	28
SCG: Algoritmo de entrenamiento de Scaled Conjugated Gradient.....	28
GPS: Algoritmo de entrenamiento de Scaled Conjugated Gradient.....	28
IMU: Algoritmo de entrenamiento de Scaled Conjugated Gradient.....	28

