

Salamanca, Gto., a 9 de diciembre del 2021.

M. en I. HERIBERTO GUTIÉRREZ MARTIN
COORDINADOR DE ASUNTOS ESCOLARES
P R E S E N T E.-

Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno(a) José Guadalupe Beltrán Hernández del **Programa de Doctorado en Ingeniería Eléctrica** y cuyo número de **NUA** es: 142490 del cual soy director. El título de la tesis es "Reconocimiento de Caracteres Manuscritos por Medio de Señales Mieléctricas"

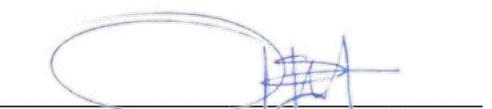
Hago constar que he revisado dicho trabajo y he tenido comunicación con los sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

A T E N T A M E N T E



DR. JOSÉ RUIZ PINALES
DIRECTOR DE TESIS
SECRETARIO

DIRECTOR DE TESIS



DR. OSCAR GERARDO IBARRA MANZANO
PRESIDENTE



DR. JOSÉ AMPARO ANDRADE LUCIO
VOCAL



DR. JOSÉ DE JESÚS RANGEL MAGDALENO
VOCAL



DRA. HAYDÉE PEREGRINA BARRETO
VOCAL



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO - SALAMANCA

DIVISIÓN DE INGENIERÍAS

*“Reconocimiento de Caracteres Manuscritos por Medio de
Señales Mioeléctricas”*

TESIS

QUE PARA OBTENER EL GRADO DE:

DOCTOR EN INGENIERÍA ELÉCTRICA

PRESENTA:

M. I. José Guadalupe Beltrán Hernández

DIRECTORES:

Dr. José Ruiz Pinales

Salamanca, Guanajuato

Junio 2021

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas.....	v
Agradecimientos	vi
Dedicatoria	ix
Capítulo 1 Introducción	1
1.1 Funciones de los músculos	5
1.2 Electromiografía	6
1.3 Objetivos	10
1.4 Objetivos específicos.....	10
1.5 Justificación.....	11
1.6 Organización de la tesis	11
Capítulo 2 Reconocimiento de caracteres manuscritos.....	13
Capítulo 3 Redes neuronales.....	21
3.1 Redes neuronales artificiales.....	22
3.2 Redes neuronales convolucionales	25
3.3 Redes neuronales recurrentes	27
3.4 Memoria a largo-corto plazo	28
3.5 Redes de unidad recurrente cerrada	30
3.6 Dropout	31
3.7 Normalización por lotes	33
3.8 Funciones de activación	34
3.9 Softmax	37
3.10 Entropía cruzada	38
3.11 Exactitud	40
3.12 MaxPooling	41
3.13 Algoritmo Backpropagation.....	42
3.14 Algoritmo Nadam.....	47
Capítulo 4 Enfoque propuesto	50
4.1 Construcción de la base de datos	51

4.2 Arquitectura de red neuronal	62
Capítulo 5 Pruebas y Resultados.....	66
Capítulo 6 Conclusiones	72
Bibliografía.....	73

Índice de figuras

Figura 1. Miofibrilla.	2
Figura 2. Miofilamentos.....	3
Figura 3. Tipos de músculos.	4
Figura 4 . Músculo esquelético.	4
Figura 5. Tejido cardíaco.	5
Figura 6. Señal Mioeléctrica.....	6
Figura 7. Medición de una señal EMG.	7
Figura 8. Señales EMG.....	8
Figura 9. Ejemplo de señales EMG capturadas por medio del brazalete Myo™.	13
Figura 10. Sistema de reconocimiento de caracteres manuscritos descrito en [6].	18
Figura 11. Neurona Biológica.....	21
Figura 12. Sinapsis.	22
Figura 13. Neurona Artificial.....	23
Figura 14. Capas de una red neuronal artificial.	23
Figura 15. Red Neuronal Convolucional.....	26
Figura 16. Una red neuronal recurrente y su desdoblamiento en el tiempo.....	28
Figura 17. Estructura de una unidad LSTM.....	29
Figura 18. Estructura de una unidad GRU.	30
Figura 19. Una red neuronal con y sin dropout.	33
Figura 20. Funciones de activación más comunes.....	35
Figura 21. Función de activación ReLU.	36
Figura 22. Una capa de salida softmax en una red neuronal.	38
Figura 23. MaxPooling (o agrupamiento máximo).....	41
Figura 24. Vista general del sistema de reconocimiento de caracteres manuscritos.	51
Figura 25. Marco general para la construcción de la base de datos utilizada.....	52
Figura 26. Un sujeto lleva un brazalete Myo en el antebrazo.	52
Figura 27. Brazalete Myo.	53
Figura 28. El software Myo Amband Manager.	54
Figura 29. Relajando el brazo para permitir la inicialización del software de calibración. .	54
Figura 30. Selección del perfil de calibración del brazalete Myo™.	55
Figura 31. Comienzo de la calibración del brazalete Myo™.	55
Figura 32. Movimiento de cerrar el puño para calibrar el brazalete Myo™.	56
Figura 33. Grabación del movimiento del puño.	56
Figura 34. Terminación de la calibración del brazalete Myo™.	57
Figura 35. Momento en que ya se ha guardado el perfil de calibración personal.	57
Figura 36. Estado del software de captura después de la calibración de la pulsera.	58
Figura 37. GUI para trazar un carácter manuscrito y capturar las señales sEMG.	58
Figura 38. Un carácter manuscrito trazado usando la GUI de captura de señales sEMG.	59
Figura 39. Ejemplo de tareas de escritura y captura.	60
Figura 40. Patrón representativo (promedio) de la base de datos sobre todas las clases.	61
Figura 41. Ejemplo de captura de señales sEMG guardada en un archivo.	61
Figura 42. Ejemplo de coordenadas x, y de una captura de trazo manuscrito.	62
Figura 43. Arquitectura CNN-LSTM.	64
Figura 44. Arquitectura CNN-GRU.....	65

Figura 45. Matriz de confusión para la arquitectura CNN_GRU con datos de un sujeto...	69
Figura 46. Matriz de confusión para la arquitectura CNN_GRU con todos los sujetos.	70
Figura 47. Curva ROC de la arquitectura CNN-GRU (una clase, un sujeto).....	71
Figura 48. Curva ROC de la arquitectura CNN-GRU (una clase, todos los sujetos).....	71

Índice de tablas

Tabla 1. Resultados obtenidos de cada arquitectura.....	67
Tabla 2. Resultados obtenidos para la arquitectura CNN-GRU por sujeto.	67
Tabla 3. Comparación del método propuesto con los mejores métodos.	68

Agradecimientos

Agradezco en especial a la **Universidad de Guanajuato (ABEJAS)** por permitirme realizar mis estudios de Maestría y Doctorado en el campus FIMEE ahora DICIS respectivamente a todos los Doctores que conforman la Universidad por haberme instruido en la captación de conocimientos para poder aplicarlos y desarrollarme profesionalmente dentro del campo de estudios que realizo.

Agradezco al Dr. Roberto Rojas Laguna por darme la oportunidad de realizar mis estudios en el campus DICIS cuando era director del campus gracias por todo el apoyo, al Dr. Horacio Rostro González por darme la oportunidad de realizar el doctorado en circunstancias especiales, al Dr. Juan Gabriel Aviña Cervantes por todo el apoyo de ir a dar clases hasta mi trabajo UABJO, al Dr. José Ruiz Pinales por ir a dar clases hasta mi trabajo UABJO, al Dr. Mario Alberto Ibarra Manzano por ir a dar clases hasta mi trabajo UABJO, y por el equipo de cómputo para realizar las pruebas de esta trabajo, al Dr. Juan Carlos Gómez Carranza por su paciencia y apoyo en los cursos de programación de Python y por prestarme un GPU para realizar mis pruebas de este trabajo, al Dr. Víctor Ayala Ramírez por todo el apoyo que me dio en el transcurso de mis estudios, al Dr. Arturo García Pérez por permitirme tener un lugar en su laboratorio, al Dr. Carlos García Capulín por la oportunidad de impartir clases en la división, al Dr. Juan Pablo Ramírez Paredes coordinador del doctorado por todo el apoyo recibido, a todos los Doctores que de alguna u otra manera me ayudaron en mi vida estudiantil Dra. Dora muchas gracias por las posadas a todo DSP.

Muy especialmente al **Dr. José Ruiz Pinales** por todo su apoyo incondicional por todas las enseñanzas que me da, por todo el trabajo realizado, por toda su paciencia para conmigo, por enseñarme a no darme por vencido, por todas las horas de clase que me dio y me da, por todo muchas gracias Dr. Pinales, que no tengo con que pagarle todo lo que hace por mí le agradezco de corazón.

A todos mis compañeros del grupo Pedro López Rodríguez por todo su apoyo muchas gracias, José Luis López Ramírez por todo el apoyo recibido muchas gracias por todo lo aprendido, a mis compañeros de cubo Guillermo Tatino, Roberto Huerta, Gerardo Valencia,

Dr. Jorge Cruz, Chucho Yáñez, Noe, a todos mis compañeros de DICIS Guillermina Vivar, Carlos Lastre, Tat'y, Alejandro Arrazola, David Matajira, Javier Antonio por las idas a las hamburguesas, Roberto Ayona, al Dr Julián Estudillo por esos momentos de descanso y armonía, y magnas celebraciones.

Al equipo de futbol FIMEE a todos mis compañeros de juego Eduardo Careta, Oscar Manzano, Osvaldo Villalon, Jacobo, Jerónimo Mireles, Roberto Rojas, Juan Montoya, Daniel Pérez, Marco Castro, Pedro Martínez, Sergio Cano, Ricardo Rodríguez, Guty Estrada, Adrián Pacheco, Jesús Antonio Herrera, Víctor Ayala, a todos por los grandes momentos de juego, en el RIAMA gracias por esos partidazos que nos aventábamos.

A la UBAJO por darme la oportunidad de realizar mis estudios de Doctorado fuera de la institución al Rector Dr. Eduardo Carlos Bautista Martínez, al secretario académico Javier Martínez Marín, por todas las facilidades otorgadas, a la secretaria Martha Elba Juárez Ruiz, por todo su apoyo y ayuda en lo administrativo, para poder concluir mis estudios para bien de la Universidad Autónoma "Benito Juárez" de Oaxaca UABJO.

A toda mi familia por todo el apoyo recibido que siempre están ahí para tenderme la mano en los momentos más difíciles de la vida, a mi papa José Guadalupe Beltrán Ortiz por todo el apoyo recibido de su parte para conmigo gracias papa, a mis hermanos Alejandro, Patricia, Angelica, Mercedes, gracias por todo los momentos que pasamos juntos en las buenas y en las malas, no tengo con que pagarles tanto amor para mí y tantas cosas que me comparten muchas gracias por todo, a mis sobrinos Ale, Beto, Brenda, Kevin, Dylan, Melanie Zharick, Kimberly, Elvis, a mis otros sobrinos Itzani Zuleyka, Alexa, Alisson Renata, Abril, Isaac, Romeo, Alexia Estefanía, Máximo, Daniela, Fernanda, Camila, Emilio, Emiliano gracias por todo lo que me dan y no les puedo pagar con nada, por enseñarme todas las cosas de la vida.

A mis tíos Teresa y Antonio por toda la paciencia que me tienen, a mis tíos Anselmo, Blas, Loreto, Julián, Oliva, Lorenza, Ofelia que siempre están ahí para lo que se me ofrezca, gracias por todo.

A mis primos que siempre están para ayudarme en cualquier cosa, Genaro, Heriberto, Erika, Beatriz, Daniel, Fernando, Lidia, Blas, Susana, Miriam, Ana, Julio, María, para Liliana y

Vero, por todas las atenciones que tienen para mí, a mi ahijado Aldebarán, un agradecimiento muy especial a Lorena, Vanessa y Jazmín por ayudarme en los momentos más difíciles de mi vida, porque siempre están ahí para mí cuando las requiero, muchísimas gracias por todo, no tengo con que pagarles todo lo que hicieron y hacen por mí, por todas las enseñanzas que me dan para saber más de esta vida y enseñarme a salir adelante, a Ma. del Rosario y Abel Ángel por todas las atenciones que tienen para conmigo, siempre que los necesito ahí están, muchas gracias por todo lo que me ofrecen siempre que ando por esas tierras tan bellas. Al grupo de oración mis maestros mayores, por enseñarme a crecer espiritualmente en todos los aspectos.

Y si alguien se me escapa por ahí muchas gracias por todo el apoyo recibido les agradezco de corazón muchas gracias por toda su ayuda y por todo el apoyo incondicional.

Dedicatoria

Al Padre Dios, por toda la abundancia que me da, por la vida, por la oportunidad que me da de poder servir a los demás por medio de él, muchas gracias de corazón por todas las cosas que me da, y también por todas las que no me da muchas gracias.

En memoria de mi hermana María Mercedes Beltrán Hernández (†)

Meche muchas gracias por todo lo que me diste, por todo lo que me enseñaste en esta vida, por todos los momentos alegres, tristes, bonitos, especiales, cumpleaños, fiestas, momentos especiales, por todas las preocupaciones, risas, pláticas, llantos, gritos, juegos, etc., por todas las cosas que pasamos juntos, muchas gracias por todo, no puedo pagarte todo lo que hiciste por mí, por todas las cosas que te sacrificaste por mí, por todos tus regaños sobre todo para ser mejor persona, por todos tus consejos, por todas las cosas que compartimos en esta vida, muchas gracias por todo, te amo hermana te quiero mucho sólo puedo decirte gracias, nunca me pasó por la mente que te fueras a ir tan pronto de nosotros, dejándonos tristes y un hueco muy difícil de llenar pero muy agradecido contigo, eres una guerrera de luz, por todas tus sonrisas, gritos y alegrías muchas gracias Meche, hasta el cielo donde estas con el Creador.

En memoria de mi mama Elvira Hernández Olvera (†)

Mi mama Elvira muchas gracias por darme la vida, sin ti mamá no estaría aquí, te agradezco de corazón todo lo que me protegiste, todo lo que pediste por mí al Creador para estar aquí en esta vida sano y salvo, estoy vivo gracias a Dios y a ti que nunca te rendiste por luchar por mí y porque nunca dejaste ni un sueño por cumplir, te agradezco todas tus enseñanzas y por mostrarme toda tu fortaleza y ganas de vivir, y porque nunca te diste por vencida, porque me mostraste que todo se puede a base de trabajo y dedicación muchas gracias, eres una guerrera que admiro y ejemplo a seguir, gracias por tu sonrisa, alegría por todo tu amor que me diste a mí, muchas gracias de corazón te lo agradezco, no tengo con que pagarte tanto amor y tanto cariño que me tienes sólo puedo decirte te amo mamá muchas gracias, hasta el cielo donde estás.

En memoria de mi tía Gracia Rangel Gallegos (†)

Gracia, tía, muchas gracias por todas tus enseñanzas que me diste, por todos los momentos que pasamos juntos en esta vida, por todas las cosas que compartimos juntos, por todas tus alegrías que me diste, por todos tus consejos que me hiciste, por todo lo que me diste de corazón, sólo puedo agradecerte todo lo que me diste muchas gracias, tía, te quiero mucho.

Capítulo 1

Introducción

En este trabajo se va a estudiar e implementar un nuevo sistema de procesamiento y reconocimiento de escritura manuscrita basado en electromiografía (EMG). El desarrollo de sistemas que pueden hacer interfaz con dispositivos externos por medio de señales bioeléctricas representa una gran promesa de la ciencia en el ámbito clínico. Por ejemplo, las prótesis neurales tratan de restaurar la movilidad de miembros y capacidades de comunicación a los discapacitados por medio de señales EEG, señales EMG con actuadores y dispositivos basados en estimulación eléctrica. Además de las aplicaciones clínicas se ha sugerido que las interfaces bioeléctricas podrían también utilizarse para mejorar ciertas funciones en sujetos normales.

Los sistemas de reconocimiento y reconstrucción de escrituras manuscrita están apenas en su etapa temprana de desarrollo, debido principalmente a dificultades técnicas percibidas por los diseñadores de interfaces bioeléctricas. Una interfaz que convierta actividad bioeléctrica en registros de texto tendría ´potenciales aplicaciones. Por ejemplo, el desarrollo de esta tecnología sustituiría a algunos periféricos de computadora o pantallas táctiles que se han utilizado para capturar y transmitir información textual.

Es por esto que en este proyecto se propone estudiar las señales EMG (Electromiograficas) para determinar las características que llevan la mayor parte de la información necesaria para el reconocimiento.

Así mismo se plantea la hipótesis para desarrollar un sistema practico de reconocimiento de escritura manuscrita basado en señales EMG de pequeño vocabulario y un multiescritor, con una buena tasa mínima de error.

Plantearemos algunos objetivos específicos para alcanzar esta hipótesis planteada, estudiaremos las señales EMG y su aplicación al reconocimiento de caracteres manuscritos, también estudiaremos a fondo las arquitecturas de redes neuronales y propondremos una que mejor se adapte para la tarea de reconocimiento, también se estudiaran los factores que afectan al reconocimiento, por ejemplo, efecto de la posición del sensor, numero de sensores, interferencia de señales ECG, etc., construir un sistema practico para la captura, reconstrucción y reconocimiento de palabras manuscritas en base a señales EMG, construiremos una base de datos grande para desarrollar este trabajo.

Con este proyecto se pretende tener los resultados esperados, como es, ya tener un sistema de reconocimiento y reconstrucción de escritura manuscrita basado en señales EMG de pequeño vocabulario y multiescritor, con una buena tasa de error, para esto entonces comenzamos a estudiar las señales EMG.

Originalmente, las señales mioeléctricas se estudiaron en los músculos esqueléticos, los cuales tienen la característica de contraerse, como respuesta a estímulos nerviosos provenientes del cerebro para causar movimiento.

Los músculos están compuestos de estructuras filiformes microscópicas llamadas miofibrillas, ver la Figura 1.

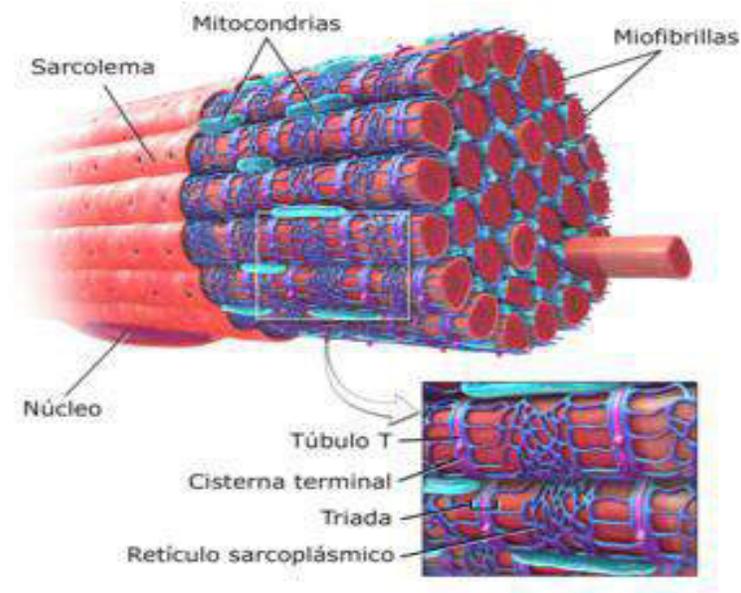


Figura 1. Miofibrilla.¹

Cada miofibrilla está compuesta de miles de miofilamentos: gruesos, hechos de miosina, y delgados, hechos de actina, ver la Figura 2.

¹ Tomada de https://commons.wikimedia.org/wiki/File:Blausen_0801_SkeletalMuscle_esp.png.

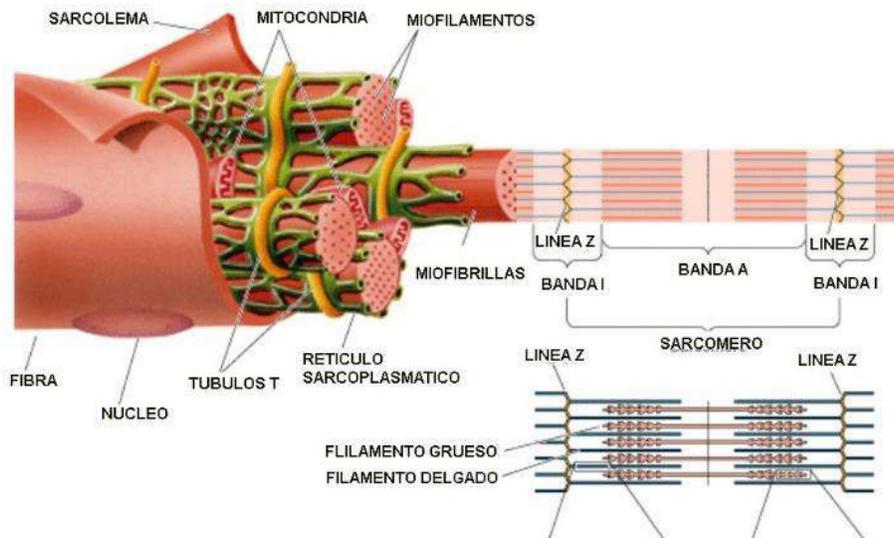


Figura 2. Miofilamentos.²

Los miofilamentos gruesos y delgados se traslapan en sus extremos y se deslizan unos sobre otros cuando ocurre una contracción muscular. Los músculos se clasifican en 3 grupos: lisos, esqueléticos y cardíacos, ver Figura 3.

² Tomada de <https://sites.google.com/site/atlashistologicodigital/tejido-muscular?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>

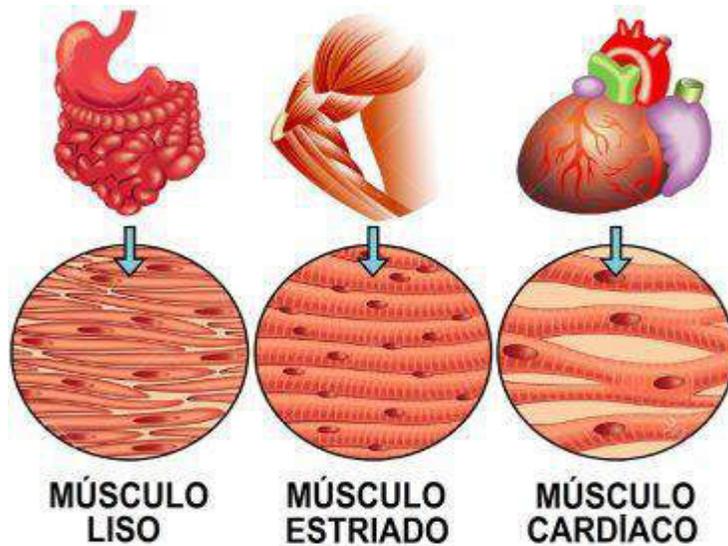


Figura 3. Tipos de músculos.³

Los músculos esqueléticos o estriados están formados por fibras largas (células alargadas), con un diámetro que va de 10 a 80 micras, dentro de una membrana celular llamada sarcolema, ver Figura 4.

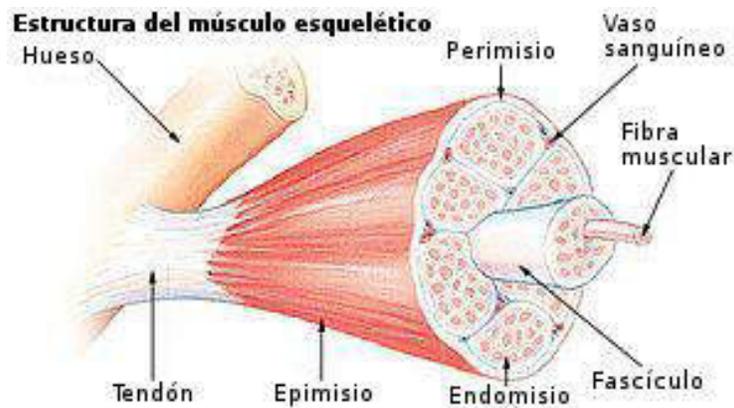


Figura 4 . Músculo esquelético.⁴

La mayoría de estos músculos están unidos a diversas zonas del esqueleto por medio de tejidos llamados tendones. De esta manera, las contracciones de los músculos hacen que

³ Tomada de <https://mundoentrenamiento.com/tipos-de-musculos-y-su-funcion/>

⁴ Tomada de https://es.wikipedia.org/wiki/Músculo_esquelético

los huesos se muevan. En los vertebrados, los músculos comprenden un gran porcentaje de su masa corporal.

Los músculos cardíacos son los que están localizados en las paredes del corazón [1]. Las fibras musculares cardíacas tienen un tamaño de 15 μm de diámetro y cerca de 100 μm de largo con una morfología similar a las de los músculos esqueléticos, pero son controladas de manera involuntaria, ver Figura 5.

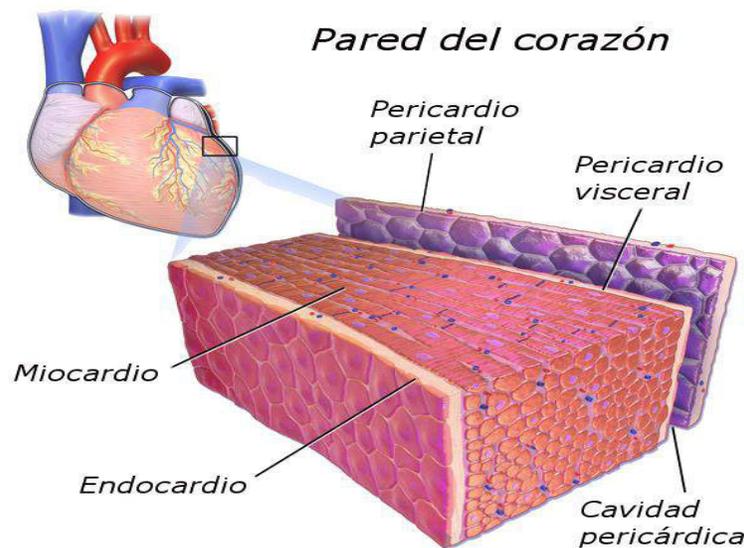


Figura 5. Tejido cardíaco.⁵

1.1 Funciones de los músculos

Los músculos reciben nombres que recuerdan su forma y funcionamiento, como el músculo trapecio del dorso que se parece a un trapecio. Los músculos esqueléticos están formados por fibras musculares que pueden ser de contracción lenta o de contracción rápida. Las de contracción rápida aportan potencia y las de contracción lenta aportan resistencia.

Las contracciones de las células musculares son el resultado de cambios eléctricos en su superficie, originados por liberaciones de calcio. La contracción tiene lugar por la

⁵ Tomada de <https://es.wikipedia.org/wiki/Miocardio>

estimulación de los impulsos nerviosos, y la relajación cuando los estímulos cesan, ver Figura 6.

Primero, un potencial de acción llega a las fibras musculares proveniente de los nervios. En respuesta, son liberados neurotransmisores (acetilcolina) en los extremos de los nervios. Estos neurotransmisores actúan sobre la membrana de la fibra muscular, haciendo que se abran canales de iones de calcio. Como resultado, la fibra muscular recibe grandes cantidades de iones de calcio, lo que hace que se produzca un potencial positivo (potencial de placa final, EPP) dentro de la fibra muscular que se propaga a lo largo de la membrana del músculo, causando esto la contracción del músculo.

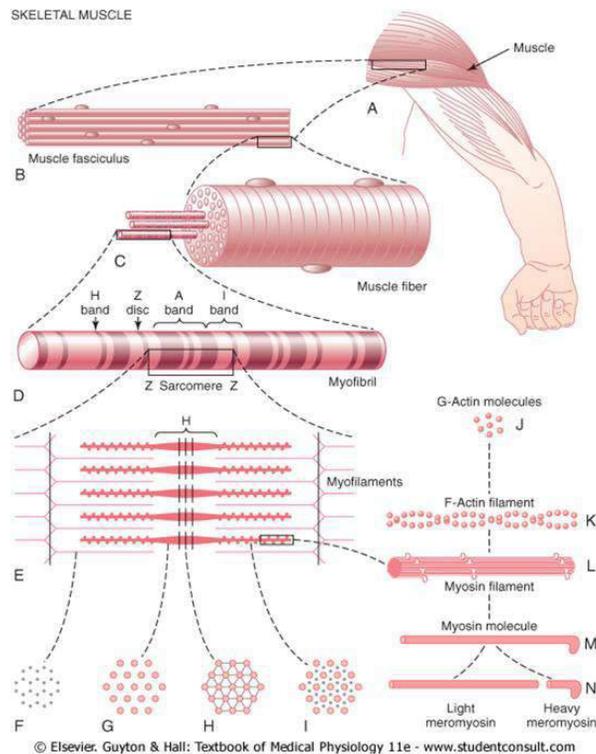


Figura 6. Señal Mioeléctrica.

1.2 Electromiografía

El electromiograma (EMG) es un registro de los potenciales eléctricos producidos por los músculos cuando son estimulados por las señales nerviosas. Estos potenciales eléctricos se pueden medir en la superficie del cuerpo (EMG de superficie) o directamente insertando

pequeñas agujas en los músculos. Las señales EMG dan una indicación de la cantidad de actividad de un músculo determinado, o un grupo de músculos. En realidad, estas señales EMG son una superposición de todos los potenciales de acción generados por cada fibra de los músculos.

Cuando se utilizan electrodos colocados sobre la piel, estos recogen potenciales de todos los músculos cercanos. Por lo tanto, las mediciones de señales EMG de superficie generadas por músculos pequeños pueden verse afectadas por las señales generadas por músculos grandes. En este caso de que se requiera medir las señales EMG de un solo músculo, se pueden utilizar electrodos de aguja, ver Figura 7. En cualquier caso, los potenciales generados en las fibras pueden tener tanto polaridades positivas como polaridades negativas, por lo que muchos de ellos pueden cancelarse entre sí. Por lo tanto, las señales EMG presentan una forma bastante compleja que las hace muy parecidas al ruido aleatorio, véase la Figura 8.

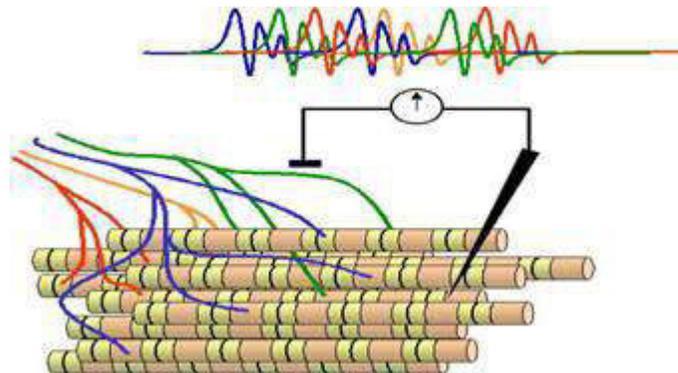


Figura 7. Medición de una señal EMG.⁶

Es necesario subrayar, que los potenciales bioeléctricos generados por los músculos son debidos a flujos de iones. Y para medir de manera correcta esos potenciales iónicos es necesario convertirlos en potenciales eléctricos para que así se puedan utilizar métodos de medición convencionales. Esto condujo a que se desarrollaran instrumentos de medición muy estables, con poco ruido, etc., [1].

⁶ Tomada de “Análisis de un Electromiograma”, Universidad Politécnica Salesiana, Ingeniería Eléctrica.

El procesamiento de señales es un área de la ingeniería electrónica que se concentra en la representación, transformación y manipulación de señales en tiempo discreto (Discrete Time Signal Processing), esto se refiere al procesamiento de señales discretas en el tiempo o en el espacio. Lo cual implica que sólo se conoce el valor de la señal en instantes o en puntos específicos. Sin embargo, la amplitud de la señal es continua, es decir, puede tomar un número infinito de valores diferentes [2].

La electromiografía tiene, hoy en día, múltiples aplicaciones, como el diagnóstico médico de enfermedades neurológicas [3], el diseño de interfaces BCI, la medicina del deporte, la evaluación del desempeño en deportes, el entretenimiento, y los videojuegos, entre otras.

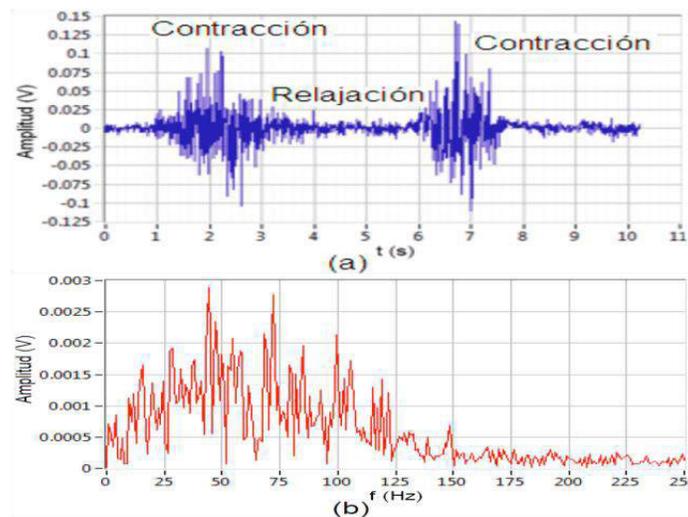


Figura 8. Señales EMG.⁷

No fue hasta la mitad del siglo XX que se desarrolló el primer equipo comercial de electromiografía para uso médico. Luego, la llegada del microprocesador hizo posible el desarrollo de sistemas cada vez mejores y más económicos para captar, representar, almacenar, analizar y clasificar las señales EMG.

⁷ Tomada de https://www.researchgate.net/figure/Figura-10-a-Senal-EMG-medida-en-los-biceps-del-brazo-derecho-con-los-electrodos-con_fig2_277931231

Actualmente, los avances en el campo de las tecnologías de la información y las comunicaciones han permitido el desarrollo de sistemas basados en inteligencia artificial, sistemas expertos de apoyo al diagnóstico electromiográfico [4].

En este trabajo se pretende estudiar, diseñar e implementar un nuevo sistema de procesamiento y reconocimiento de escritura manuscrita basado en electromiografía. El desarrollo de sistemas cerebro-computadora que puedan hacer interfaz con dispositivos externos por medio de señales bioeléctricas representan una gran promesa de la ciencia en el ámbito clínico. Por ejemplo, las prótesis neurales tratan de restaurar la movilidad de miembros y capacidades de comunicación a los discapacitados por medio de señales electroencefalográficas (EEG), señales EMG con actuadores y dispositivos basados en estimulación eléctrica.

Además de las aplicaciones clínicas, se ha sugerido que las interfaces BCI podrían también utilizarse para mejorar ciertas funciones en sujetos normales. Los sistemas de reconocimiento y reconstrucción de escritura manuscrita están apenas en su etapa temprana de desarrollo, debido principalmente a dificultades técnicas percibidas por los diseñadores de interfaces BCI. Una interfaz que convierta la actividad bioeléctrica en registros de texto tendría potenciales aplicaciones. Por ejemplo, el desarrollo de esta tecnología sustituiría a algunos periféricos de computadora o pantallas táctiles que se han utilizado para capturar y transmitir información textual.

En una primera etapa, las señales eléctricas de los músculos de la mano serán captadas por medio de sensores EMG de superficie al momento que una persona escribe. Luego, esas señales serán procesadas utilizando algoritmos de procesamiento digital de señales para poder eliminar en lo más mínimo el ruido y otras señales de interferencia (por ejemplo, las señales ECG). Enseguida, a partir de la señal procesada se utilizarán técnicas de reconocimiento de patrones para reconocer la palabra que la persona estaba escribiendo al momento de generarse la señal EMG. Se utilizarán sensores EMG de superficie de la marca Thalmic Labs ®.

La tarea de reconocimiento será multi-escritor y de pequeño vocabulario. Se emplearán técnicas de reconocimiento de patrones apoyadas en redes neuronales convolucionales y se elaborarán estrategias de reconocimiento analíticas y holísticas. De hecho, se pueden

utilizar muchas otras técnicas que ya se han desarrollado para el reconocimiento de la escritura. En este caso, la principal dificultad del desarrollo del presente sistema será principalmente de tipo técnico.

Se va a estudiar a fondo las señales EMG para determinar las características que llevan la mayor parte de la información necesaria para el reconocimiento. También se va a estudiar a fondo todas las arquitecturas de redes neuronales que mejor se adaptan para la tarea de reconocimiento de las señales EMG para conseguir los mejores resultados posibles de esta investigación. Se va a disponer de un sistema práctico para la captura, reconstrucción, y reconocimiento de letras manuscritas en base a señales EMG.

1.3 Objetivos

Desarrollar un sistema práctico de reconocimiento de escritura manuscrita basado en señales EMG de pequeño vocabulario y un multi-escritor, con una buena tasa de error.

1.4 Objetivos específicos

Estudiar a fondo las señales EMG para determinar las características que llevan la mayor parte de la información necesaria para el reconocimiento.

Estudiar a fondo las arquitecturas de redes neuronales y/o modelos de Markov ocultos que mejor se adaptan para la tarea de reconocimiento.

Estudiar a fondo factores que afectan al reconocimiento, por ejemplo, efecto de la posición del sensor, número de sensores, inclusión de señales EEG, interferencia de señales ECG, etc.

Disponer de un sistema práctico para la captura, reconstrucción, y reconocimiento de palabras manuscritas en base a señales de EMG.

1.5 Justificación

Las señales electromiográficas (EMG) tienen aplicaciones importantes en las interfaces cerebro-computadora (BCI), el ámbito clínico, la rehabilitación, el diagnóstico de padecimientos neurológicos, entre otros campos, principalmente porque son fáciles de adquirir de forma no invasiva y sin peligro para el individuo. Estas señales contienen información acerca de la actividad neuromuscular presente en el músculo activo correspondiente.

La escritura manuscrita es una habilidad comunicativa compleja y esencial para el ser humano que a menudo no se tiene en cuenta. Sin embargo, hay estudios que han mostrado que la escritura manuscrita es una habilidad motriz fundamental para el desarrollo de las facultades motoras, cognitivas y académicas de los niños [5].

El uso de las señales electromiográficas para el reconocimiento de escritura manuscrita es un tema novedoso por lo que hasta la fecha sólo se sabe de unos cuantos trabajos [6, 7].

En general, se puede afirmar que todos los enfoques existentes son muy restrictivos, ya que requieren de un ambiente bien controlado que permita obtener las señales EMG con pocas variaciones para facilitar su clasificación. Por un lado, muchos de ellos colocan los electrodos sobre los músculos para obtener señales EMG de músculos individuales. Por otro lado, todos los enfoques sólo se diseñaron para caracteres escritos de un solo trazo.

Por lo tanto, para aplicaciones prácticas es necesario un enfoque más general para el reconocimiento de caracteres manuscritos multiusuario, de estilo libre, de varios trazos y sin restricciones en la disposición de los electrodos.

1.6 Organización de la tesis

Este trabajo está dividido en siete capítulos los cuales se describirán a cada uno de ellos a continuación.

En el capítulo uno se verá una introducción a las señales Electromiográficas (EMG). De cómo se originaron y cómo han ido evolucionando hasta la actualidad.

En el capítulo dos se verá la teoría y el estado del arte de los sistemas de reconocimiento de escritura usando señales EMG, cómo han ido evolucionando los métodos, técnicas y procedimientos para el procesamiento de las señales electromiográficas en el mundo actual.

En el capítulo tres se hablará de las redes neuronales artificiales, las redes neuronales recurrentes (Recurrent Neural Network) RNN, las redes de memoria a corto y largo plazo (Long Short Term Memory, LSTM), y redes GRU.

En el capítulo cuatro se describirá el sistema o enfoque propuesto que se realizó y los métodos propuestos. También se describirá cómo se creó la base de datos de señales mioeléctricas.

En el capítulo cinco se verán las pruebas que se realizaron, los experimentos efectuados para obtener una arquitectura de red óptima que nos permitió alcanzar los mejores resultados.

Al final se presentan las conclusiones, así como los trabajos futuros.

Capítulo 2

Reconocimiento de caracteres manuscritos

Las señales electromiográficas (EMG) son muy importantes para un cierto número de aplicaciones las cuales incluyen el desarrollo de interfaces cerebro-computadora (BCI). Esto es debido a que esta señal contiene suficiente información acerca de la actividad muscular. Las señales EMG, como otras señales de origen biológico, son señales no estacionarias.

Thalmic Labs® desarrolló el dispositivo inalámbrico llamado Myo™ para capturar las señales EMG. La Figura 9 muestra el brazalete Myo™ y algunos ejemplos de las señales capturadas.

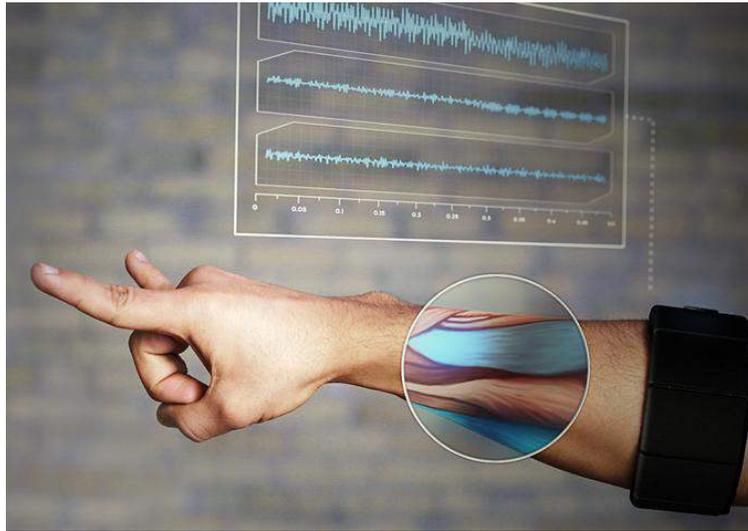


Figura 9. Ejemplo de señales EMG capturadas por medio del brazalete Myo™.⁸

⁸ Tomada de <https://mikejr1.es/index.php/noticias/167-hardware/22931-myo-el-brazalete-para-controlar-el-pc-por-gestos-que-sustituira-a-nuestros-ratones.html>.

Los primeros experimentos documentados relacionados con las señales EMG comenzaron cuando Francesco Redi en 1666 descubrió que un músculo altamente especializado de los peces raya eléctrica genera electricidad. Por ahí de 1773, Walsh demostró que el tejido muscular de la anguila es capaz de generar chispas luminosas de descargas eléctricas.

En 1792, una publicación titulada “De Viribus Electricitatis in Motu Musculari Commentarius”, escrita por Luigi Galvani, se demostró que la electricidad podría iniciar la contracción muscular. Posteriormente, Emil du Bois-Reymond demostró la existencia de actividad eléctrica durante contracciones musculares voluntarias en humanos. Fue hasta 1890 que Marey logró por primera vez grabar esa actividad eléctrica, y utilizó la palabra electromiografía (EMG).

En 1922, Gasser y Erlanger lograron observar en un osciloscopio las señales EMG. Debido a la naturaleza estocástica de la señal EMG, sólo podía obtenerse información aproximada a partir de su observación. La detección de señales EMG mejoró continuamente hasta que se desarrollaron mejores electrodos para el estudio de los músculos. Varios años después, se empezaron a aplicar las señales sEMG en el ambiente clínico para el diagnóstico de trastornos específicos. Hardyck y sus colegas fueron de hecho los primeros en utilizar las señales EMG. Cram y Steger desarrollaron un método clínico para la captura de señales EMG de una variedad de músculos.

No fue hasta la mitad de la década de 1980 que los avances en las técnicas de integración fueron suficientes para poder fabricar la circuitería de instrumentación con la pequeñez y ligereza requeridas. Herbert Jasper desarrolló la primera máquina de electromiografía, marcando así un amplio uso de las señales EMG en el ambiente clínico. En la actualidad, ya hay disponibles comercialmente amplificadores con características adecuadas.

Las investigaciones recientes han contribuido a mejorar la comprensión de las señales sEMG. Las señales sEMG se utilizan ampliamente para la registrar la actividad muscular en protocolos clínicos o kinesiología.

Existen muchas aplicaciones de las señales EMG, se utilizan en el diagnóstico del andar y en la evaluación de la biorretroalimentación o ergonomía. Las señales EMG también se

usan en muchos tipos investigación, incluyendo la biomecánica, control motor, fisiología neuromuscular, trastornos del movimiento, el control postural, y terapia física.

Es importante subrayar que la aplicación de las señales sEMG al reconocimiento de la escritura manuscrita es un tema novedoso por lo que hasta la fecha sólo se sabe de unos cuantos trabajos [6, 7].

La escritura manuscrita es una habilidad comunicativa compleja y esencial para el ser humano que a menudo no se tiene en cuenta. Sin embargo, se ha demostrado que la escritura manuscrita es una habilidad motriz fundamental para el desarrollo cognitivo y académico de los niños [5].

Considerando como ejemplo el caso de la composición musical, los expertos en esta materia muestran su preferencia por las ediciones manuscritas frente al uso de interfaces especializadas. Esto último debido al limitado uso del ordenador en aplicaciones artísticas. A pesar de todo, la detección automática de la notación musical manuscrita en soportes digitales es muy recomendable y rentable en una aplicación de este tipo, muy similar al reconocimiento de caracteres en el análisis de textos. Calvo-Zaragoza y Oncina propusieron un método de reconocimientos de notación musical basado en un bolígrafo electrónico y una máquina de estados finitos para detectar los trazos del usuario de forma competitiva [8].

Mendes investigó la interacción humano-robot basada en la detección de movimientos de las manos mediante el desarrollo de un sensor de EMG para capturar y discriminar movimientos válidos utilizando una técnica de Deep Learning [9].

De forma análoga, Slapekand y Paskiel propusieron un método basado en curvas de Bézier para extraer movimientos predefinidos de las manos sin utilizar un marcador al inicio o al final en la secuencia de puntos [10].

Este tipo de métodos trata de detectar el estilo de escritura manuscrita para desarrollar metodologías mejoradas en interfaces humano-ordenador inteligentes y el reconocimiento de la escritura manuscrita.

Por otro lado, el desarrollo de interfaces cerebro-ordenador ofrece una alternativa para aplicaciones clínicas y asistidas que implican bioseñales motoras.

Por ejemplo, la capacidad de comunicación de personas con o sin discapacidades puede mejorarse mediante dispositivos con interfaz de señales de electroencefalografía (EEG) y/o electromiografía.

Hay dos tipos de señales EMG [11]: señales EMG de superficie (sEMG) y señales EMG intramusculares. Las señales EMG intramusculares requieren la inserción de electrodos en un músculo esquelético. En cambio, las señales sEMG sólo requieren colocar electrodos en la superficie de la piel por encima de un músculo. En comparación con las señales EMG intramusculares, el uso de las señales sEMG tiene la gran ventaja de ser una técnica no invasiva que es clínicamente superior en el diagnóstico, la rehabilitación y el ensayo de nuevas terapias relacionadas con la actividad motora.

Sin embargo, las señales sEMG forman un intrincado patrón de interferencias de señales bioeléctricas generadas principalmente por la actividad dinámica de los músculos esqueléticos. Como consecuencia, las señales EMG de los músculos individuales no pueden distinguirse de forma fiable. El potencial de las membranas musculares puede sufrir cambios que pueden ir desde -70 mV (en reposo), hasta cerca de 30 mV. Por lo tanto, el rango de amplitudes de la señal sEMG puede ir desde unos cuantos microvolts hasta 20 a 30 mV.

Las señales sEMG han encontrado varias aplicaciones nuevas apoyadas por medio de Machine Learning y Deep Learning como el análisis del movimiento, la fatiga muscular, las terapias deportivas, los dispositivos variables, y particularmente en el reconocimiento de la escritura manuscrita [12].

La adquisición de señales sEMG y las formas de onda correspondientes se ven afectadas por muchos factores, como las propiedades fisiológicas de los músculos, el ruido añadido en la señal de los músculos, el ruido añadido a la señal por la instrumentación utilizada para la detección, así como la ubicación elegida de los electrodos en relación con las unidades musculares activas.

Por lo tanto, los métodos convencionales de extracción de características en señales sEMG generalmente requieren robustos pasos de preprocesamiento y filtrado. Además, la complejidad de la señal sEMG también puede explicarse por el hecho de que hay varias formas de realizar una tarea motora (redundancia motora) [13, 14].

Simao por ejemplo hizo una revisión de los sistemas basados en las señales sEMG utilizando técnicas de reconocimiento de patrones y describiendo procesos sensibles como las técnicas de adquisición y el filtrado de la señal. Su estudio se centró en las nuevas modalidades de interacción hombre-máquina [15].

Recientemente, los avances en biomedicina han hecho posible la intervención de prótesis mioeléctricas capaces de realizar movimientos reales de brazos o piernas [16]. Las prótesis neuronales pretenden restaurar la movilidad de las extremidades y la capacidad de comunicación de los sujetos con discapacidad, mediante la interconexión de potenciales cerebrales, actuadores o dispositivos artificiales en la estimulación eléctrica funcional. Una importante innovación fue la implantación de un dispositivo para predecir un parto prematuro con varias semanas de antelación.

En este contexto, las interfaces bioeléctricas, permiten extraer patrones de escritura con fines de diagnóstico en pacientes con la enfermedad de Parkinson y, más recientemente, con la discapacidad por discografía. Por lo tanto, las señales EMG generadas durante la escritura manuscrita pueden utilizarse para caracterizar trastornos neurológicos [17].

En la escritura manuscrita, las señales EMG se generan y capturan de los músculos de la mano y el antebrazo durante la actividad de la escritura, para ser decodificado por los sistemas de reconocimiento de fines específicos [7].

En los estudios, Linderman por ejemplo desarrolló un sistema fiable que utilizaba señales EMG para reconocer los caracteres de la escritura manuscrita para diagnosticar o medir la progresión de enfermedades en tratamientos médicos.

Linderman et al propusieron un sistema de reconocimiento de caracteres manuscritos basado en señales EMG [6]. En este caso, se utilizaron, como se muestra en la Figura 10, electrodos bipolares de superficie sobre 4 músculos del antebrazo y 4 músculos de la mano.

Al momento de la escritura, se observó que las señales EMG de cada músculo presentaron ráfagas. Entonces, las señales EMG fueron rectificadas y la reconstrucción del trazo de escritura fue efectuada por medio de filtros de Wiener. La trayectoria del lápiz fue obtenida como sumas ponderadas de las señales EMG rectificadas. La exactitud de la reconstrucción resultó ser comparable a otras interfaces bioeléctricas. Para el reconocimiento de caracteres numéricos (0 a 9) se utilizó análisis lineal discriminante (LDA). Muestras de señales EMG de un solo escritor fueron capturadas para un total de 500 muestras (50 muestras por clase). La mitad de las muestras fueron usadas para entrenamiento y la otra mitad para prueba. El porcentaje de reconocimiento promediado para 6 escritores fue de 90.4% con una desviación estándar de 7.0% [6].

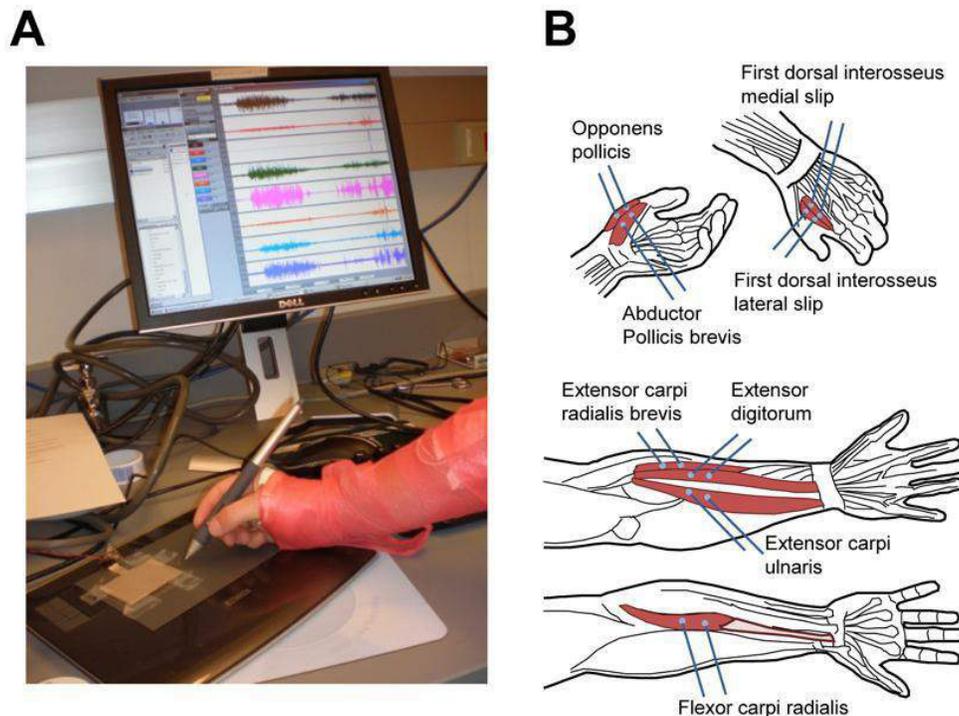


Figura 10. Sistema de reconocimiento de caracteres manuscritos descrito en [6]. A) Señales capturadas durante la escritura. B) Posiciones de los sensores sobre el brazo [18].

En general, el reconocimiento de la escritura manuscrita a partir de las señales EMG se ha abordado en la literatura mediante los siguientes enfoques: Template Matching [18], Dynamic Time Warping (DTW) [7, 19], Radial Basis Functions (RBF) [17].

Cabe destacar que las señales sEMG contienen bastante información sobre el movimiento efectuado por una extremidad. Siguiendo este razonamiento, se han propuesto varias aplicaciones en la literatura para el reconocimiento, modelado y reconstrucción de los movimientos de la escritura.

Además, la reconstrucción de la trayectoria del trazo de la escritura manuscrita a partir de señales EMG se ha abordado utilizando: filtros de Weiner [18], mínimos cuadrados recursivos [20], redes neuronales RBF [21], y filtros de Kalman [22].

Por lo tanto, los avances no sólo se limitan a los fines médicos, sino que también al entretenimiento y a los videojuegos que pueden ser controlados por la actividad eléctrica de los músculos. En un trabajo relacionado, Wei por ejemplo ideó un método de reconocimiento de movimientos mediante el uso de características multivista para movimientos unimodales y multimodales basados en señales sEMG por medio de una red neuronal convolucional profunda (DCNN) [23].

Dash por ejemplo [24], diseñó un enfoque llamado AirScript, para visualizar y reconocer caracteres escritos en el aire. En este caso, los datos de la IMU (Unidad de Medición Inercial) del brazalete Myo fueron utilizados para obtener una secuencia de coordenadas 2D y visualizar un dígito escrito a mano como una imagen. Para el reconocimiento de los dígitos escritos manuscritos, se utilizó una red neuronal convolucional (CNN) y dos redes neuronales recurrentes (GRU). Las imágenes se utilizaron como entradas para la CNN, la secuencia de coordenadas 2D generada se utilizó como entrada para la primera red neuronal GRU, y la señal cruda de la IMU como entrada a la segunda red neuronal GRU. Posteriormente, las salidas de esas tres redes neuronales se utilizaron para obtener la clase del dígito reconocido. El sistema alcanzó una tasa de acierto del 91.7% y 96.7% para las tareas independientes y dependientes, respectivamente.

En el método aplicado por Roy [25], los dígitos de un solo trazo se escribieron en el aire utilizando un marcador de color uniforme. Una cámara de video grabó la trayectoria de la escritura. Después de segmentar cada fotograma de video, la posición del marcador se utilizó para aproximar la trayectoria de la escritura y obtener una imagen del dígito escrito a mano. A continuación, se utilizó una red CNN para reconocer la imagen de cada dígito

manuscrito con una tasa de acierto del 97.7%, 95.4%, y 93.7% para los dígitos escritos en inglés, Bengali y Devanagari, respectivamente.

Como se pudo constatar, en algunos estudios, el reconocimiento de los caracteres manuscritos se ha limitado a utilizar pequeñas bases de datos, imponiendo varias restricciones en la colocación de los electrodos, en el número y la complejidad de los trazos de la escritura. La mayoría de los métodos de reconocimiento de la escritura manuscrita que utilizan señales sEMG se basan principalmente en el preprocesamiento y la extracción de características [6], mientras que otros utilizan el método DTW después del preprocesamiento [19].

Capítulo 3

Redes neuronales

En 1888, Ramón y Cajal descubrió que el cerebro está formado por unidades individuales llamadas neuronas que se interconectan entre sí formando redes. Una neurona es una célula con un tamaño de 10 a 80 micrones, que presenta numerosas ramificaciones (dendritas) y una prolongación (axón) de hasta 1 m de largo.

La neurona biológica se puede considerar como un procesador microscópico cuyas dendritas sirven como entrada de señales provenientes de otras neuronas (hasta 10,000), esas señales son acumuladas en el cuerpo o soma y transmitidas eléctricamente por el axón a otras neuronas por medio de las sinapsis (hasta cientos de miles) [26], ver Figura 11.

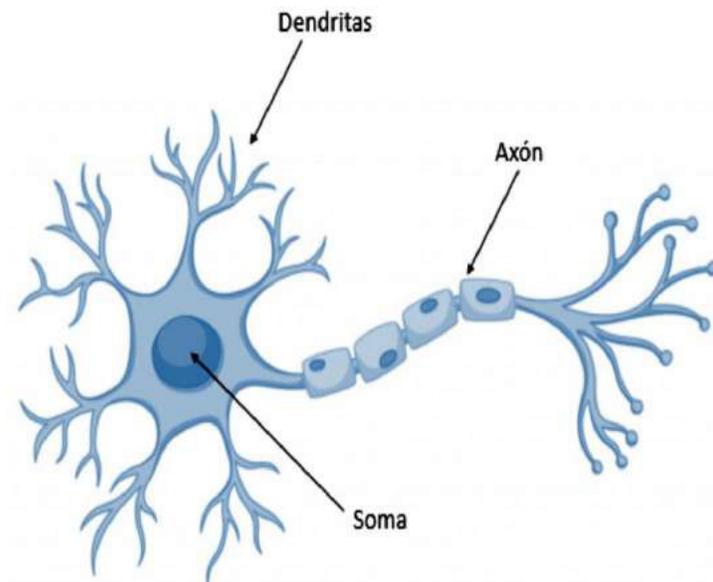


Figura 11. Neurona Biológica.⁹

⁹ Tomada de <https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>



Figura 12. Sinapsis.¹⁰

Las sinapsis permiten conectar una neurona con otra, aunque en realidad no hay contacto entre ellas. La sinapsis es un espacio muy estrecho (varios nanómetros) por el que pasan sustancias químicas llamadas neurotransmisores, ver Figura 12. Los neurotransmisores pueden ser excitatorios o inhibitorios.

El cerebro humano se puede considerar como la cosa más compleja e inteligente que existe, contiene alrededor de 10^{11} neuronas, cada neurona pueden tener hasta 10^4 conexiones, lo que equivale a un total de 10^{15} conexiones [27].

3.1 Redes neuronales artificiales

Las redes neuronales artificiales o también conocidas como métodos conexionistas, son una técnica de aprendizaje de máquina originalmente propuesta por Warren McCulloch y Walter Pitts en 1943, basándose en hallazgos sobre el funcionamiento del cerebro humano.

A nivel esquemático, una neurona artificial se puede representar como una simplificación de la neurona biológica de la siguiente manera. El modelo de la neurona artificial consiste

¹⁰ Tomada de <https://psicologiymente.com/neurociencias/tipos-neurotransmisores-funciones>

simplemente de: entradas, pesos de conexión de las entradas, un umbral (bias), un nodo suma y una función de activación para la salida, ver Figura 13.

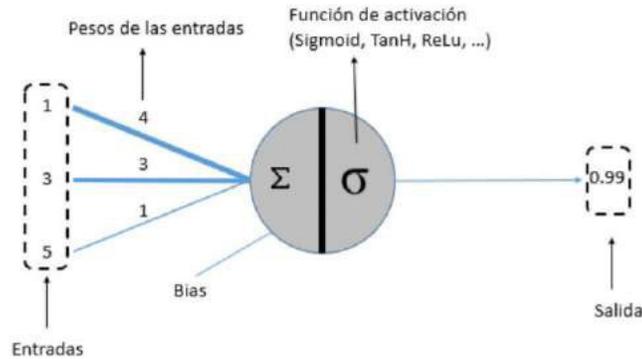


Figura 13. Neurona Artificial.¹¹

En analogía con las redes neuronales biológicas, una red neuronal artificial está compuesta de capas de neuronas artificiales conectadas entre sí. La Figura 14 muestra una red neuronal artificial de tres capas, incluyendo la entrada como una capa.

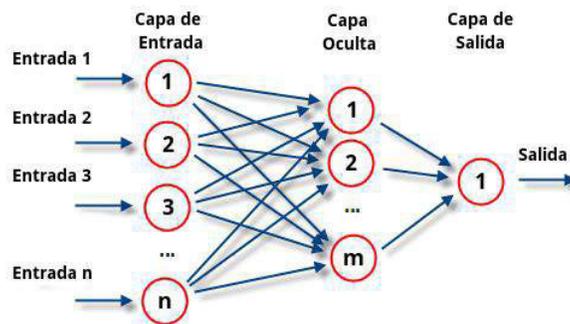


Figura 14. Capas de una red neuronal artificial.¹²

La primera capa es llamada capa de entrada, simplemente comprende los nodos de entrada. Las siguientes capas exceptuando la última son llamadas capas ocultas porque

¹¹ Tomada de <https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>

¹² Tomada de https://es.wikipedia.org/wiki/Perceptrón_multicapa

sus salidas no son normalmente visibles. Por último, la última capa es llamada capa de salida. Las redes neuronales artificiales pueden tener cualquier número de capas. Sin embargo, antiguamente se utilizaban redes con muy pocas capas ocultas. Actualmente, las redes neuronales profundas pueden contener muchísimas capas. El concepto de Deep Learning surgió cuando empezaron usarse redes neuronales con un gran número de capas [28].

Las redes neuronales artificiales son técnicas provenientes del campo de la inteligencia artificial. Obviamente, siendo el cerebro humano la red neuronal más poderosa, es lógico pensar que podría existir una red neuronal artificial que pueda simular la inteligencia humana [27].

Existen tres tipos de algoritmos para el aprendizaje de las redes neuronales artificiales: supervisado, no supervisado y por refuerzo.

El aprendizaje supervisado usa un conjunto de pares de entrenamiento que consisten cada uno de (entrada, salida correcta). Este método de aprendizaje es utilizado para resolver problemas de clasificación (reconocimiento de patrones) o regresión (aproximación de funciones).

El aprendizaje no supervisado usa un conjunto de datos de entrenamiento sin etiquetar. Este método de aprendizaje es utilizado para descubrir agrupamientos de patrones o clases.

El aprendizaje por refuerzo fue originalmente descubierto por Edward Thorndike quien observó que la conducta de un animal podía venir acompañada por respuestas del medio ambiente satisfactorias. Cuando las respuestas del medio ambiente se repetían, era muy probable que el animal asociara su conducta con esas respuestas. El aprendizaje por refuerzo consiste en que un agente debe explorar un espacio desconocido y determinar las acciones necesarias mediante ensayo y error. Esto es, el agente aprenderá por sí mismo la estrategia a seguir para maximizar las recompensas obtenidas de sus acciones.

Las redes neuronales se clasifican, según su topología de red, en:

- Red neuronal Monocapa.

- Red neuronal Multicapa.
- Red neuronal Convolutacional (CNN).
- Red neuronal recurrente (RNN).
- Red de base radial (RBF).

3.2 Redes neuronales convolucionales

Una red neuronal convolutacional (CNN), es una conocida arquitectura de Deep Learning originalmente inspirada en los descubrimientos sobre la percepción visual. De hecho, una red neuronal convolutacional es equivalente a una red con pesos compartidos, por lo que contiene menos parámetros que un perceptrón multicapa.

En 1959, Hubel y Wiesel descubrieron la existencia de células simples en la corteza visual de los gatos [29]. Estas células simples eran neuronas que respondían a barras o bordes con una orientación específica. Hubel y Wiesel propusieron el modelo jerárquico, en el cual una célula simple integraba las activaciones de las células ganglionares dentro de una región alargada.

En 1976, Marr [30] afirmó que los campos receptivos de las células simples tenían la forma de una barra o de borde y que calculaban la convolución lineal con una máscara (filtro, kernel) en forma de barra o de borde.

Inspirándose en los hallazgos anteriores, Fukushima propuso el Neocognitrón en 1980 [31], que quizás constituye una de las primeras arquitecturas de CNN. El Neocognitrón introdujo dos tipos básicos de capas: capa convolutacional y capa de submuestreo (average pooling).

En 1992, Weng propuso la arquitectura Cresceptrón que introdujo el Max-pooling [32] y una variante llamada blurring [33].

En 1990, LeCun creó la arquitectura LeNet, ver Figura 15, para resolver la tarea de la clasificación de imágenes de dígitos manuscritos [34]. La arquitectura LeNet y sus variantes fueron las primeras en ser entrenadas con el algoritmo de retropropagación [35]. Se encontró que esta arquitectura obtenía representaciones eficaces directamente de la

imagen, lo que permitía reconocer los patrones visuales a partir de píxeles sin procesar. Por ese tiempo, Zhang propuso una Red Neuronal Artificial Invariante (SIANN) para reconocer caracteres en una imagen [36].

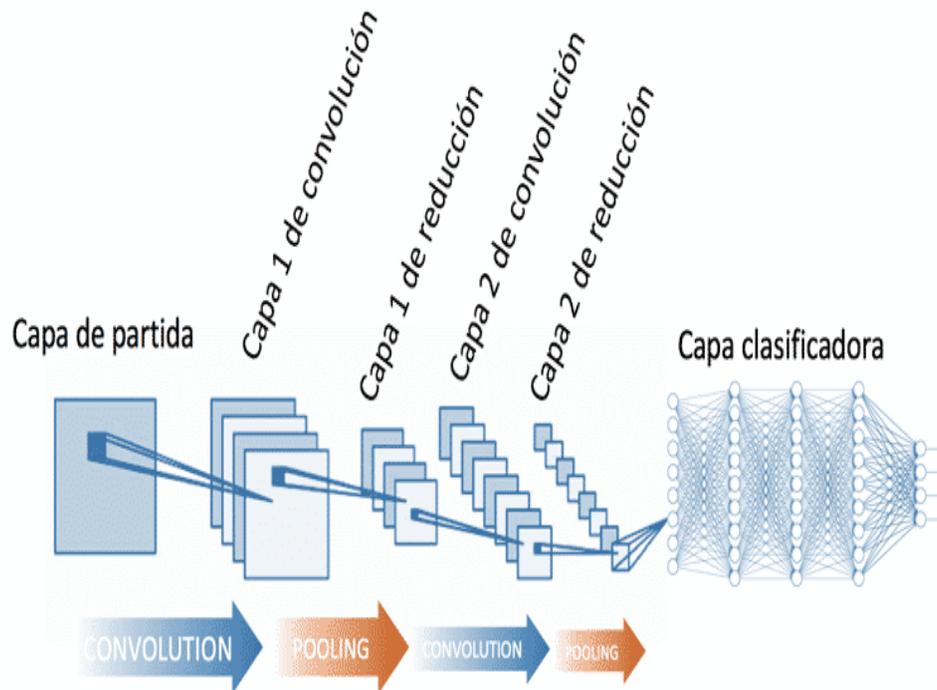


Figura 15. Red Neuronal Convolutiva.¹³

Al principio, a pesar del éxito obtenido por la LeNet-5 en el reconocimiento de dígitos, las CNNs no tuvieron mucho éxito en tareas más complejas. Las razones principalmente eran la escasa potencia de cálculo disponible, el pequeño número de muestras de entrenamiento, y diversos problemas que surgieron especialmente para las CNN profundas (DCNN) [37]. Tras varios esfuerzos por superar las dificultades de entrenamiento de las DCNN y gracias a la disponibilidad de nuevos y potentes recursos informáticos, Krizhevsky presentó la arquitectura AlexNet, con la cual ganó el reto de reconocimiento visual a gran escala de ImageNet (ILSVRC2012) [38].

¹³ Tomada de <https://www.diegocalvo.es/red-neuronal-convolutiva/>

La arquitectura general de ALexNet [39], es básicamente la misma que la arquitectura LeNet-5 pero considerablemente más profunda, con convoluciones en cascada y con más filtros por capa. Entre las innovaciones incorporadas en AlexNet, se encuentran el uso de la función de activación ReLU (Rectified Linear Unit) [40], las capas de agrupamiento máximo (Max-Pooling), Dropout [41], el aumento de datos [42], y algoritmos de inicialización ad-hoc. Jiuxiang Gu detalló recientes mejoras de las CNNs relacionadas con las innovaciones mencionadas [37].

Varios años después, He por ejemplo propuso la arquitectura ResNet, que ganó el concurso ILSVRC2015 [43]. Esta era varias veces más profunda que la AlexNet e incorporaba innovaciones como los bloques residuales y capas de normalización por lotes [44].

3.3 Redes neuronales recurrentes

Las redes neuronales recurrentes (RNNs) son una clase de redes neuronales concebidas originalmente como un esfuerzo por incorporar el tiempo en las redes neuronales.

En 1986 Jordan propuso una red neuronal con memoria dinámica que incorporaba conexiones recurrentes desde la capa de salida hasta la capa oculta. Varios años después, Elman [45] propuso otra RNN basada en la red de Jordan. Este modelo incorporaba conexiones recurrentes directamente desde la salida de la capa oculta permitiendo a la red mantener una especie de estado o memoria en la capa oculta. Esto permitió a la red realizar tareas difíciles como la predicción de secuencias temporales, el mapeo de secuencia a secuencia, etc.

Las RNNs se entrenan actualmente utilizando la técnica de retro-propagación en el tiempo (BPTT). En primer lugar, la red se desdobla en el tiempo. Como resultado, se obtiene una red neuronal estática (feedforward) de pesos compartidos muy profunda. A continuación, se realiza el entrenamiento mediante el algoritmo de retro-propagación.

En su tiempo, las RNNs de Jordan y Elman tuvieron un éxito muy limitado porque presentaban problemas de aprendizaje para aplicaciones con secuencias muy largas [46] [47]. Esto se denominó el “problema del desvanecimiento del gradiente” (vanishing gradient

problem) porque los gradientes retro-propagados utilizados para actualizar los pesos decaían rápidamente hasta llegar a cero. Como resultado, la red no era capaz de retener los patrones contenidos al principio de la secuencia. Esto significa que la red no era capaz de aprender dependencias a largo plazo.

Las RNNs no presentan una estructura de capas definidas, sino que presentan conexiones de retroalimentación, permitiendo que de esta manera la red tenga una especie de memoria.

La Figura 16 muestra el desdoblamiento en el tiempo de una RNN. Como se puede observar, la red primero procesa la entrada en instante 1, luego la misma procesa la entrada en el instante 2, y así sucesivamente [28].

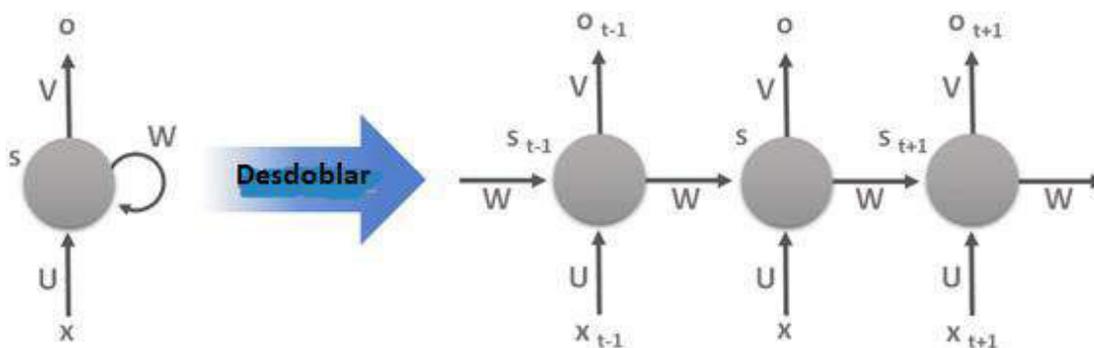


Figura 16. Una red neuronal recurrente y su desdoblamiento en el tiempo.¹⁴

3.4 Memoria a largo-corto plazo

En 1997 Hochreiter y Schmidhuber [48] propusieron las redes de memoria a largo-corto plazo (LSTM) como solución al problema del desvanecimiento del gradiente. La estructura básica de una red LSTM está compuesta por una cadena de unidades que pueden decidir cuándo recordar, olvidar y adquirir nueva información. En consecuencia, una unidad LSTM está compuesta por una célula de memoria y tres puertas multiplicativas, olvido, entrada y

¹⁴ Adaptada de <https://www.diegocalvo.es/red-neuronal-recurrente/>

salida. La Figura 17 ilustra la estructura básica de una unidad LSTM, que incluye en esencia unidades sigmoidales, tanh, sumadores y multiplicadores.

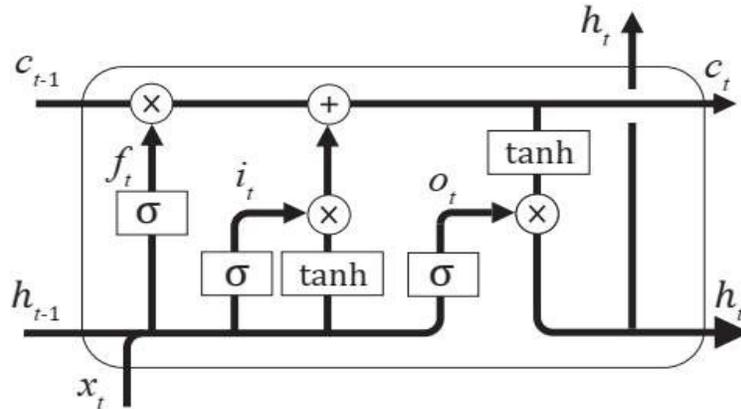


Figura 17. Estructura de una unidad LSTM.

Matemáticamente, las ecuaciones que rigen el funcionamiento de una unidad LSTM se describen a continuación.

La salida de la puerta de olvido viene dada por la Ecuación (3.1):

$$f_t = \sigma(W_{xt}x_t + W_{hf}h_{t-1} + b_f) \quad (3.1)$$

donde $\frac{1}{1+e^{-x}} \in [0,1]$ es la función de activación sigmoide, x_t es la entrada actual, h_t es la salida actual de la unidad LSTM, W y b son las matrices de pesos y de umbral, respectivamente.

La salida de la puerta de entrada viene dada por la ecuación (3.2);

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3.2)$$

donde h_{t-1} es la salida anterior, W y b son, respectivamente, la matriz de pesos y los parámetros de umbral que deben ser aprendidos. Además, la salida actual del estado de la célula c_t , se obtiene mediante la ecuación (3.3);

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc} x_t + W_{hc} h_{t-1} + b_c) \quad (3.3)$$

donde $\tanh(x)$ es la función de activación tangente hiperbólica. Por último, o_t es la salida de la puerta definida como la ecuación (2.4);

$$o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + b_o) \quad (3.4)$$

Las redes LSTM han encontrado innumerables aplicaciones desde su invención, que incluyen el reconocimiento del habla, el análisis de textos, el procesamiento del lenguaje, la traducción, automática y la generación de escritura manuscrita, entre otras.

3.5 Redes de unidad recurrente cerrada

Las redes de unidad recurrente cerrada (GRU, Gated Recurrent Unit) fueron introducidas por Cho [49] como una alternativa a las redes LSTM con menos parámetros. Una unidad GRU utiliza sólo dos puertas para decidir cuándo adquirir nueva información y su estado oculto es también su salida. Básicamente, una unidad GRU está compuesta por una célula de memoria y dos puertas multiplicativas: compuertas de actualización y compuerta de salida. La Figura 18 ilustra la estructura de una unidad GRU fundamental.

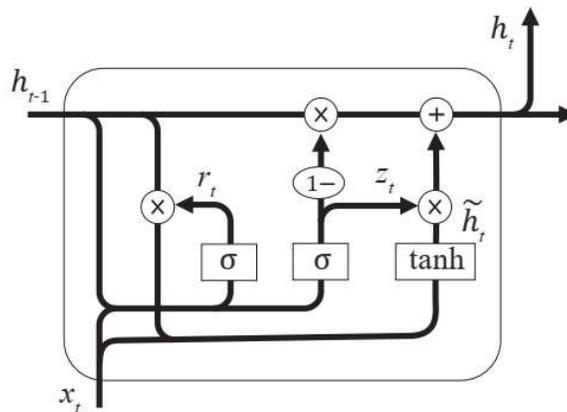


Figura 18. Estructura de una unidad GRU.

A continuación, se describen los cálculos realizados por una unidad GRU. La salida r_t , de la puerta de reinicio, es la que decide cuánta cantidad de información se debe olvidar, se define como la ecuación (3.5),

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (3.5)$$

Donde $\sigma(\cdot)$ es una función de activación sigmoideal, x_t es la entrada actual, h_{t-1} es el estado oculto del paso de tiempo anterior, W y b son la matriz de pesos y los parámetros de umbral, respectivamente. La salida de la puerta de actualización para el paso de tiempo t , z_t se describe mediante la ecuación (3.6).

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_i) \quad (3.6)$$

Por último, el estado oculto actual (o salida) h_t de la unidad GRU viene dado formalmente por la ecuación (3.7).

$$h_t = (1 - z_t)h_{t-1} + z_t\hat{h}_t \quad (3.7)$$

Donde \hat{h}_t se define simplemente en la ecuación (3.8),

$$\hat{h}_t = \tanh(W_{xh}x_t + r_tW_{hh}h_{t-1} + b_h) \quad (3.8)$$

que utiliza la salida de la puerta de reinicio para retroalimentar la información relevante pasada.

3.6 Dropout

El dropout es una técnica que desactiva de forma aleatoria las neuronas de una red neuronal. En cada capa, un cierto porcentaje de neuronas (tasa de dropout) son desactivadas de forma aleatoria, ver Figura 19. En cada iteración del algoritmo de entrenamiento, diferentes neuronas son desactivadas, lo que evita la coadaptación de las activaciones.

La tasa de dropout indica qué tan probable es que una neurona de una capa quede activa. El valor de la tasa de dropout va de 0.5 a 1.0, donde 0.5 y 1.0 indican que la mitad y todas las neuronas quedarán activas, respectivamente.

El dropout sólo se aplica durante el entrenamiento, por lo que, durante la prueba, las entradas netas de las neuronas son escaladas para compensar el efecto causado por la desactivación.

Puede aplicarse a cada capa de la red una tasa de dropout diferente, dependiendo de las características de cada capa.

El dropout es muy ampliamente usado como método de regularización para reducir los efectos del overfitting (o sobreajuste). De hecho, se demostró que el dropout es equivalente a un método de regularización. Esto es, ayuda a reducir el sobreajuste evitando que la red memorice alguna particularidad de los datos de entrenamiento.

El sobreajuste se produce cuando una red neuronal trata de aprender fielmente (memorizar) los datos de entrenamiento. Por lo tanto, su capacidad de generalización para reconocer o aproximar datos nunca vistos, que no tienen las particularidades de los datos de entrenamiento, es muy baja.

Actualmente, el dropout ha encontrado aplicación en la cuantificación de la incertidumbre de los modelos de aprendizaje profundo [50].

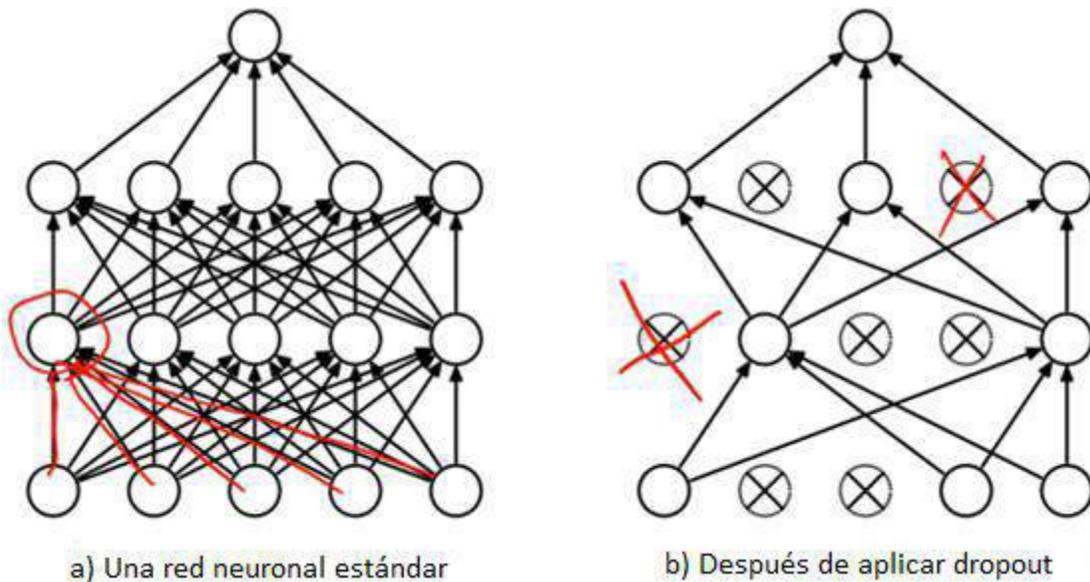


Figura 19. Una red neuronal con y sin dropout.¹⁵

3.7 Normalización por lotes

La normalización por lotes (Batch Normalization) es una técnica de normalización de las entradas de una capa de una red para cada lote de entrenamiento. Lo cual tiene el efecto de estabilizar el entrenamiento de la red y reduciendo grandemente el número de épocas de entrenamiento necesarias. La normalización es efectuada directamente sobre las entradas netas, antes de aplicar la función de activación.

La normalización por lotes también ayuda a reducir el fenómeno llamado desplazamiento de la covarianza (covariate shift). El desplazamiento de la covarianza se refiere al cambio de la distribución de las variables de entrada en el conjunto de entrenamiento y el conjunto de prueba. Además, la normalización por lotes tiene algún efecto de regularización, ayudando a reducir el error de regularización. Por lo tanto, es posible que no sea necesario aplicar dropout para regularizar la red. Sin embargo, se recomienda no usar dropout cuando se usa normalización por lotes. [51].

¹⁵ Adaptada de <https://medium.com/metadatos/técnicas-de-regularización-básicas-para-redes-neuronales-b48f396924d4>

3.8 Funciones de activación

Las funciones de activación en las redes neuronales artificiales son usualmente funciones no lineales. Esto ayuda a las redes neuronales profundas a extraer características más complejas. Cuando no se usa función de activación, varias capas de la red pueden ser reemplazadas por una sola capa. Las ANNs son capaces de separar espacios no linealmente separables, sólo por la no linealidad de sus funciones de activación.

Existen muchas funciones de activación (ver figura 20) cada una es usada para un propósito especial. Por ejemplo, cuando se quiere predecir la probabilidad, con un valor de salida en (0,1), entonces se puede utilizar la función de activación sigmoide [52].

La función de activación sigmoide está dada por:

$$\delta(z) = 1/(1 + e^{-z})$$

y tiene un rango de (0,1).

Cuando se requiere un valor de salida positivo y negativo, normalmente se usa $\tanh(z)$ como función de activación. La función de activación \tanh tiene un rango de (-1, 1).

En el caso de redes DCNNs, en las capas convolucionales, se usa regularmente ReLU. La función de activación ReLU se define como:

$$ReLU(x) = \max(0, x)$$

Las diversas funciones de activación se presentan en la Figura 20.

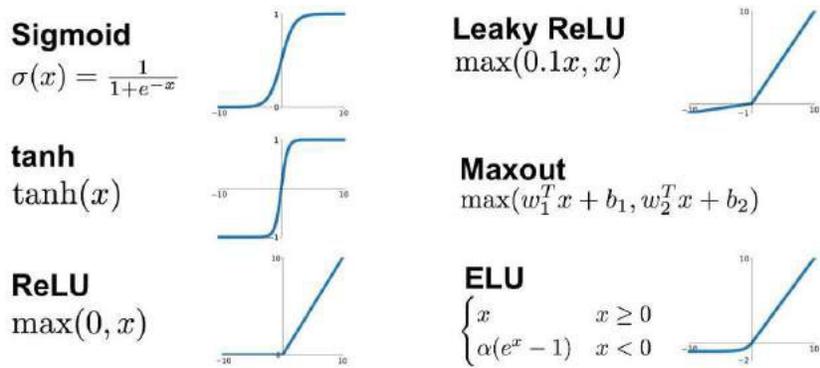


Figura 20. Funciones de activación más comunes.¹

Ya que las redes neuronales DCNNs utilizan la propagación hacia atrás, para el cálculo de los gradientes, la función de activación debe de ser continua. Sin embargo, la función de activación ReLU sólo es discontinua en 0, ver Figura 21.

Entonces, para la convergencia, es necesario que los pesos iniciales de la red no sean todos cero. Por consiguiente, los valores de los pesos usualmente se inicializan al azar utilizando algún método de inicialización.

Las funciones de activación, tales como ReLU, son un componente esencial de las redes neuronales cuyo propósito es incrementar su capacidad. De hecho, las imágenes usualmente contienen características complejas que son difíciles de extraer, usando simples filtros lineales (convoluciones lineales).

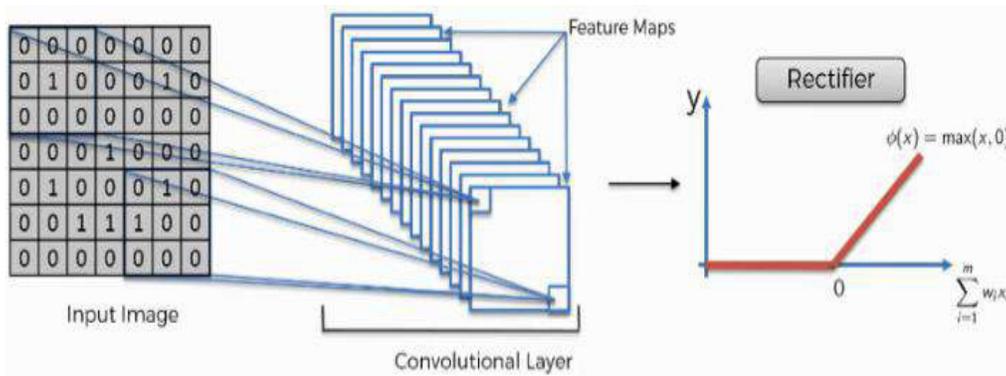


Figura 21. Función de activación ReLU.¹⁶

Las ventajas de la función de activación ReLU son: su cálculo es rápido y no presenta saturación. Sin embargo, para resolver los problemas que presenta, se han propuesto varias variantes de ésta (LeakyReLU, PReLU, ELU).

ReLU tiene las siguientes desventajas:

- No es continua en el origen.
- En la parte negativa, el gradiente es cero.
- Conduce al cambio del sesgo.

Conocemos las dos primeras desventajas, pero qué podemos decir del cambio del sesgo (bias shift). Cuando se utiliza la función de activación como *tanh*, las activaciones usualmente se distribuyen alrededor de cero. Sin embargo, en el caso de otras funciones de activación, tales como ReLU, las activaciones son positivas. Por lo tanto, en redes neuronales profundas, tales como las DCNNs, el sesgo (bias) se va desplazando ligeramente al pasar de capa a capa [53].

¹⁶ Tomada de <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1b-relu-layer/>

3.9 Softmax

La función de activación softmax, llamada también generalización de la función logística es utilizada principalmente en clasificadores basados en redes neuronales, cuando se requieren probabilidades de salida. Esta función de activación recibe como entrada un conjunto de valores (o logits) y da como salida un conjunto de valores (probabilidades) cuya suma es uno. También es llamada función softargmax o logística multiclase. La función softmax se puede usar en un clasificador sólo cuando las clases son mutuamente excluyentes.

Muchas redes neuronales multicapa terminan en una capa que genera puntajes de valor real 0 (scores) que no se escalan convenientemente y con los que puede ser difícil de trabajar. Aquí el softmax es muy útil porque convierte los puntajes en una distribución de probabilidad normalizada, que puede mostrarse a un usuario o usarse como entrada para otros sistemas. Por esta razón, es habitual agregar una función softmax a la capa final de la red neuronal [54], ver Figura 22.

La fórmula de softmax es la siguiente:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

donde:

- \vec{z} es el vector de entrada a la función softmax, compuesto por (z_0, \dots, z_k) .
- z_i es cada uno de los elementos del vector de entrada a la función softmax y pueden tomar cualquier valor real, positivo, cero o negativo.
- e^{z_j} es la función exponencial estándar que se aplica a cada elemento del vector de entrada. Esto da un valor positivo por encima de 0, que será muy pequeño si la entrada fue negativa y muy grande si la entrada fue grande. Sin embargo, todavía no está fijo en el rango $(0,1)$ que es lo que se requiere de una probabilidad.
- $\sum_{j=1}^K e^{z_j}$ es el término para efectuar la normalización. Asegura que todos los valores de salida de la función sumen 1 y que cada uno esté en el rango $(0,1)$, constituyendo así una distribución de probabilidad válida.

- K es el número de clases.

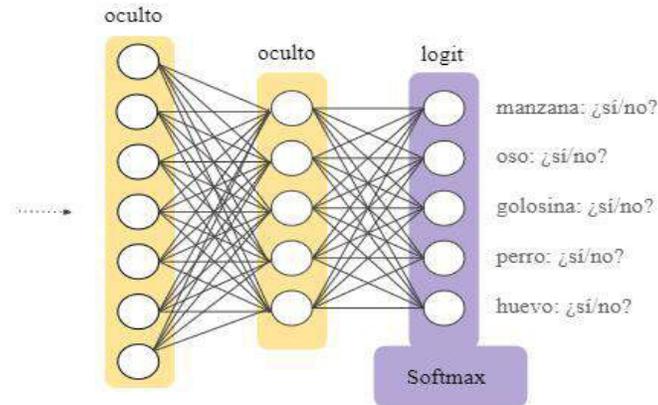


Figura 22. Una capa de salida softmax en una red neuronal.¹⁷

3.10 Entropía cruzada

La entropía cruzada es una función de pérdida basada en la teoría de la información de Shannon, en concreto la noción de entropía. Está estrechamente ligada, pero es completamente diferente a la divergencia de Kullback-Leibler (KL).

La entropía cruzada también se relaciona y a menudo se confunde con la pérdida logística, también denominada pérdida logarítmica. Aunque las dos medidas tienen un origen diferente, como funciones de pérdida en la clasificación, ambas medidas son equivalentes [55].

De acuerdo con el teorema de Shannon, la información cuantifica el número de bits necesarios para codificar y transmitir un evento. Los eventos de menor probabilidad tienen más información, los eventos de mayor probabilidad tienen menos información.

¹⁷ Tomada de <https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax?hl=es>

La cantidad de información para un evento x , dada su probabilidad de ocurrencia $P(x)$, es:

$$h(x) = \log(P(x))$$

La entropía de una variable aleatoria de X con función de probabilidad $P(x)$ es la siguiente:

$$H(x) = - \sum_{x \in \mathcal{X}} P(x) * \log(P(x))$$

La entropía cruzada entre las distribuciones Q y P , se puede representar formalmente como:

$$H(P, Q)$$

donde P puede ser la distribución objetivo y Q es una aproximación.

La entropía cruzada se puede calcular utilizando las probabilidades de los eventos de P y Q , de la siguiente manera:

$$H(P, Q) = - \sum_{x \in \mathcal{X}} P(x) * \log(Q(x))$$

donde $P(x)$ es la probabilidad del evento x , $Q(x)$ es la aproximación de la probabilidad del evento x y \log es el logaritmo en base 2, en este caso las unidades del resultado serán bits. Si se usa el logaritmo natural o base e , la entropía cruzada tendrá las unidades llamadas nats. Esta ecuación es para distribuciones de probabilidad discretas, pero se puede calcular la entropía para distribuciones de probabilidad continuas utilizando una integral en lugar de una sumatoria. Cuando las distribuciones P y Q son idénticas, la entropía cruzada es mínima (cero) [55].

Dado un clasificador que entregue como salida probabilidades de clase. La entropía cruzada se escribe como:

$$CE = - \sum_{i=1}^C d_i \log(y_i)$$

donde C es el número de clases, d_i es la salida deseada y y_i es la salida del clasificador para la i -ésima clase.

3.11 Exactitud

La exactitud es una métrica utilizada para evaluar el desempeño de modelos de clasificación. Esta consiste en el cociente entre las predicciones correctas que el modelo realizó y el número total de predicciones, por lo cual, la exactitud indica el porcentaje de datos correctamente clasificados.

Dentro del desarrollo de un modelo es de vital importancia obtener valores de salida cercanos a 1, es decir, que clasifique con la menor cantidad de errores posibles.

Utilizando los valores obtenidos de la matriz de confusión, la exactitud (o accuracy) se puede calcular de la siguiente forma [56].

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \times 100$$

La exactitud (o accuracy) mide el porcentaje de casos en los que el modelo ha acertado. Esta es una de las métricas más usadas y favoritas, que se recomienda evitar en ciertos casos.

Un problema con la exactitud es que nos puede llevar al engaño, es decir, nos puede indicar que un mal modelo es bueno, esto ocurre cuando las clases están desbalanceadas [56].

3.12 MaxPooling

El MaxPooling (o agrupamiento máximo) es una técnica de reducción de dimensionalidad, esta consiste en identificar el mayor valor en un grupo y sólo mantener dicho valor. En la Figura 23 se muestra más claramente el uso de esa técnica.

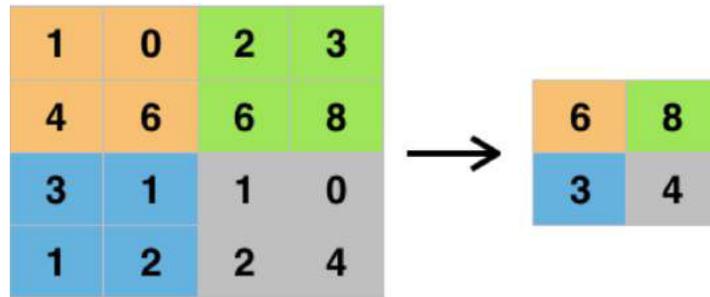


Figura 23. MaxPooling (o agrupamiento máximo).¹⁸

El MaxPooling es una operación comúnmente incorporada en las arquitecturas de redes neuronales convolucionales (CNN). La idea principal detrás del MaxPooling es preservar ciertas características de un mapa generado mediante la convolución de un filtro sobre una imagen.

Formalmente, su función es reducir progresivamente el tamaño espacial de la representación para reducir la cantidad de parámetros y el tamaño de la red. La forma más común de agrupamiento es el MaxPooling aunque existen otras técnicas de agrupamiento como el agrupamiento promedio (o AveragePooling) [57].

MaxPooling se utiliza en parte para ayudar a combatir los efectos del sobreajuste al proporcionar una representación más compacta. Además, reduce el costo computacional al reducir el número de parámetros a aprender y proporcionar cierta invarianza a la traslación en la representación interna. El MaxPooling se realiza aplicando un filtro máximo a subregiones normalmente no superpuestas de la representación inicial.

¹⁸ Tomada de <https://deepai.org/machine-learning-glossary-and-terms/max-pooling>

3.13 Algoritmo Backpropagation

El algoritmo de retropropagación del gradiente (backpropagation) es el método más ampliamente usado para entrenar redes estáticas (feed-forward). Este algoritmo forma parte de los algoritmos de aprendizaje supervisado. Se basa en el método de minimización llamado descenso del gradiente (gradient descent). En general, el algoritmo comprende dos fases: propagación hacia adelante, y propagación hacia atrás.

Durante la propagación hacia adelante, primero se presenta un patrón a la entrada de la red, y luego se propaga capa por capa hasta llegar a la salida. Durante la propagación hacia atrás, primero se calcula el error a la salida de la red, luego se actualizan los pesos de la capa de salida, se retropropaga el error hacia la capa anterior y se vuelven a actualizar los pesos, esto se repite hasta llegar a la capa de entrada (Fritsch, 1996).

Sea b_i el umbral, y w_{ij} los pesos de la i -ésima neurona de la capa de salida, la entrada neta está dada por:

$$\eta_i(t) = \sum_j w_{ji} a_j(t) + b_i \quad (1)$$

donde los $a_j(t)$ son las salidas (activaciones) de la capa anterior.

La salida $y_i(t)$ de la i -ésima neurona de la capa de salida está dada por:

$$y_i(t) = f_i(\eta_i(t)) \quad (2)$$

donde $f_i(\cdot)$ es la función de activación.

El error cuadrático (SE) de la red neuronal está dado por:

$$e_p^2 = \frac{1}{2} \sum_{k=1}^M (d_{pk} - y_{pk})^2 \quad (3)$$

donde d_{pk} es el valor deseado de la k -ésima salida y M es el número de salidas.

El error cuadrático medio (MSE) de la red neuronal está dado por:

$$E[W] = \frac{\sum_{p=1}^P e_p^2}{P} \quad (4)$$

donde P el número total de patrones.

La actualización de los pesos W en la t -ésima iteración se puede efectuar por medio de:

$$W(t+1) = W(t) + \Delta W(t) \quad (5)$$

donde $\Delta W(t)$ es el incremento del peso.

Utilizando el método del descenso del gradiente, tenemos que el incremento $\Delta W(t)$ debe ser proporcional y de dirección contraria al gradiente del error $E[W(t)]$ por lo que:

$$\Delta W(t) = -\alpha \nabla E[W(t)] \quad (6)$$

donde α es la tasa de aprendizaje, la cual debe tener un valor positivo pequeño, por ejemplo 0.0001.

De la ecuación (6), tenemos que:

$$\Delta w_{ji} = -\alpha \frac{\partial e_p^2}{\partial w_{ji}} \quad (7)$$

Aplicando la regla de la cadena tenemos que:

$$\frac{\partial e_p^2}{\partial w_{ji}} = \frac{\partial e_p^2}{\partial y_{pj}} \frac{\partial y_{pj}}{\partial \eta_j} \frac{\partial \eta_j}{\partial w_{ji}} \quad (9)$$

Usando la ecuación (7) tenemos que:

$$\frac{\partial \eta_j}{\partial w_{ji}} = \frac{\partial \sum_k w_{jk} h_k}{\partial w_{ji}} = h_i \quad (10)$$

Si escribimos el primer término de la ecuación (9) como:

$$\frac{\partial e_p^2}{\partial y_{pj}} = y_{pj} - d_{pj} \quad (11)$$

Tenemos que:

$$\frac{\partial y_{pj}}{\partial \eta_j} = f'(\eta_j(t)) \quad (12)$$

Para una función de activación sigmoideal tenemos:

$$\frac{\partial y_{pj}}{\partial \eta_j} = \frac{e^{-\eta_j}}{(1 + e^{-\eta_j})^2} = y_{pj}(1 - y_{pj}) \quad (12)$$

Y, por lo tanto, la ecuación (8) queda de la siguiente manera:

$$\Delta w_{ji} = -\alpha(y_{pj} - d_{pj})y_{pj}(1 - y_{pj})y_{pi} \quad (13)$$

Lo cual indica que el incremento de los pesos es proporcional al error.

La ecuación (13) puede ser simplificada como:

$$\Delta w_{ji} = -\alpha \delta_{pj} y_{pi} \quad (13)$$

donde

$$\delta_{pj} = \frac{\partial e_p^2}{\partial \eta_j} = \frac{\partial e_p^2}{\partial y_{pj}} \frac{\partial y_{pj}}{\partial \eta_j} = (y_{pj} - d_{pj})y_{pj}(1 - y_{pj}) \quad (14)$$

Las salidas de la red son función de las salidas de la capa anterior y éstas a su vez de la capa precedente. Por lo tanto, para para actualizar los pesos de las capas ocultas es necesario aplicar la regla de la cadena en la ecuación (9):

$$\frac{\partial e_p^2}{\partial \eta_j^{r-1}} = \sum_{k=1}^{k_r} \frac{\partial e_p^2}{\partial \eta_k^r} \frac{\partial \eta_k^r}{\partial \eta_j^{r-1}} = \sum_{k=1}^{k_r} \delta_{pk}^r \frac{\partial \eta_k^r}{\partial \eta_j^{r-1}} \quad (15)$$

donde k_r es el número de neuronas de la capa de salida (r -ésima capa).

La ecuación (15) se puede reescribir para obtener el error retropropagado:

$$\delta_{pj}^{r-1} = \sum_{k=1}^{k_r} \delta_{pk}^r \frac{\partial \eta_k^r}{\partial \eta_j^{r-1}} \quad (16)$$

donde

$$\frac{\partial \eta_k^r}{\partial \eta_j^{r-1}} = \frac{\partial \sum_{m=0}^{k_{r-1}} w_{km}^r a_m^{r-1}}{\partial \eta_j^{r-1}} = w_{kj}^r f'(\eta_j^{r-1}) \quad (17)$$

Sustituyendo la ecuación (17) en la ecuación (16) tenemos que el error retropropagado es:

$$\delta_{pj}^{r-1} = \sum_{k=1}^{k_r} \delta_{pk}^r w_{kj}^r f'(\eta_j^{r-1}) \quad (18)$$

Luego, tenemos que el incremento de los pesos de la r -ésima capa está dado por:

$$\Delta w_{ji}^r = -\alpha \delta_{pj}^r a_{pi}^{r-1} \quad (19)$$

Finalmente, la actualización de los pesos se puede escribir como:

$$w_{ji}^r(t+1) = w_{ji}^r(t) - \alpha \delta_{pj}^r a_{pi}^{r-1} \quad (20)$$

La retropropagación se efectúa calculando sucesivamente las ecuaciones (18), (19) y (20) para $r = L, L - 1, \dots, 1$ y utilizando la ecuación (14) para calcular δ_{pj}^L en la capa de salida.

Una de las desventajas del algoritmo de retropropagación del gradiente es que usualmente converge a un mínimo local. Este problema se puede compensar repitiendo el aprendizaje muchas veces, cada vez inicializando la red con valores seleccionados al azar. Otro problema es que su convergencia puede ser muy lenta debido a que la tasa de aprendizaje (α) debe ser pequeña. De hecho, la convergencia puede ser muy lenta, sobre todo cuando el algoritmo llega a puntos donde la magnitud del gradiente es casi cero. Obviamente, podemos incrementar la tasa de aprendizaje para hacer que el algoritmo sea más rápido. Sin embargo, cuando la tasa de aprendizaje es alta, el algoritmo puede no converger. Afortunadamente, existe una manera de acelerar el algoritmo utilizando el concepto de momento. Es decir, una partícula con masa tiende a moverse en la misma dirección cuando se le aplica una fuerza para cambiar su dirección. Por lo tanto, podemos describir la actualización de los pesos como sigue:

$$w_{ji}^r(t + 1) = w_{ji}^r(t) - \alpha \delta_{pj}^r a_{pi}^{r-1} + \beta \Delta w_{ji}^r(t) \quad (21)$$

donde β es el término de momento que debe ser un valor positivo menor que 1, por ejemplo 0.8. El momento tiene el efecto de acelerar el algoritmo de retropropagación del gradiente en regiones donde el gradiente es casi constante. En este caso, la tasa de aprendizaje equivalente está dada por:

$$\alpha_{eq} = \frac{\alpha}{1 - \beta} \quad (22)$$

Esto implica que para $\beta = 0.9$, el algoritmo se vuelve 10 veces más rápido.

Existen otros métodos para acelerar el aprendizaje que consisten en calcular una tasa de aprendizaje óptima. Sin embargo, estos métodos son muy pocas veces utilizados porque son computacionalmente muy pesados ya que implican calcular la inversa de una matriz Hessiana de gran tamaño.

3.14 Algoritmo Nadam

El algoritmo de optimización Adam (Adaptive Moment) puede considerarse como una combinación de RMSprop y momentum. El algoritmo RMSprop contribuye con un promedio exponencialmente decreciente de gradientes cuadrados pasados, mientras que el momento aporta un promedio exponencialmente decreciente de gradientes pasados.

El algoritmo Nadam (Momento adaptativo acelerado por Nesterov) [58] combina Adam y NAG (Gradiente acelerado por Nesterov). Para combinar NAG con Adam, sólo necesitamos modificar su término de momento. Enseguida adoptaremos la notación utilizada en [59].

Primero que nada, es conveniente reescribir la regla de actualización del momento como:

$$g_t = \nabla_{\theta_t} J(\theta_t) \quad (24)$$

$$m_t = \gamma m_{t-1} + \eta g_t \quad (25)$$

$$\theta_{t+1} = \theta_t - m_t \quad (26)$$

donde J es nuestra función objetivo, γ es el término de decaimiento del momento, y η es nuestra tasa de aprendizaje.

Expandiendo la ecuación (26) utilizando la ecuación (25) obtenemos:

$$\theta_{t+1} = \theta_t - (\gamma m_{t-1} + \eta g_t) \quad (27)$$

El segundo término de la derecha de la ecuación (27) indica que el momento implica dar un paso en la dirección del momento anterior y un paso en la dirección del gradiente actual.

Por su parte NAG nos permite realizar un paso más preciso en la dirección del gradiente actualizando los parámetros con el momento antes de calcular el gradiente. Por lo tanto, sólo necesitamos modificar el gradiente g_t como:

$$g_t = \nabla_{\theta_t} J(\theta_t - \gamma m_{t-1}) \quad (28)$$

$$m_t = \gamma m_{t-1} + \eta g_t \quad (29)$$

$$\theta_{t+1} = \theta_t - m_t \quad (30)$$

El algoritmo NAG puede modificarse de la siguiente manera: en lugar de aplicar el paso de momento dos veces, al actualizar el gradiente g_t y al actualizar los parámetros θ_{t+1} podemos aplicar el momento de búsqueda hacia adelante directamente para actualizar los parámetros actuales como sigue:

$$g_t = \nabla_{\theta_t} J(\theta_t) \quad (31)$$

$$m_t = \gamma m_{t-1} + \eta g_t \quad (32)$$

$$\theta_{t+1} = \theta_t - (\gamma m_t + \eta g_t) \quad (33)$$

Como podemos observar, en lugar de usar el momento anterior, ahora usamos el momento actual m_t para buscar hacia adelante.

Para agregar el momento de Nesterov a Adam, podemos reemplazar de manera similar el vector de momento anterior con el momento actual. Primero, recordemos que la regla de actualización de Adam es la siguiente:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (34)$$

$$\hat{m}_t = m_t / (1 - \beta_1^t) \quad (35)$$

$$\theta_{t+1} = \theta_t - [\eta / (\sqrt{\hat{v}_t} + \epsilon)] \hat{m}_t \quad (36)$$

sustituyendo la segunda ecuación las definiciones de \hat{m}_t y m_t obtenemos:

$$\theta_{t+1} = \theta_t - \eta / (\sqrt{\hat{v}_t} + \epsilon) [(\beta_1 m_{t-1}) / (1 - \beta_1^t) + (1 - \beta_1) g_t / (1 - \beta_1^t)] \quad (37)$$

Debemos observar que $(\beta_1 m_{t-1})/(1 - \beta_1^t)$ es sólo la estimación corregida de sesgo del momento del paso de tiempo anterior. Por lo tanto, podemos reemplazarlo en términos de \hat{m}_{t-1} :

$$\theta_{t+1} = \theta_t - \eta/(\sqrt{\hat{v}_t} + \epsilon) [\beta_1 \hat{m}_{t-1} + (1 - \beta_1) g_t / (1 - \beta_1^t)] \quad (38)$$

Nuevamente, debemos observar que, para simplificar, ignoramos que el denominador es $1 - \beta_1^t$ y no $1 - \beta_1^{t-1}$, ya que reemplazaremos el denominador en el siguiente paso de todos modos. Esta ecuación nuevamente se ve muy similar a la regla de actualización del momento expandido anterior.

Ahora podemos agregar el momento de Nesterov tal como lo hicimos anteriormente, simplemente reemplazando la estimación corregida de sesgo del momento del paso de tiempo anterior \hat{m}_{t-1} con la estimación corregida de sesgo del momento actual \hat{m}_t , que nos da la regla de actualización de Nadam [58]:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \left[\beta_1 \hat{m}_t + \frac{(1 - \beta_1) g_t}{1 - \beta_1^t} \right] \quad (39)$$

Capítulo 4

Enfoque propuesto

En este capítulo presentaremos nuestra propuesta para el reconocimiento de caracteres manuscritos basado en señales EMG, el desarrollo y prueba de diferentes arquitecturas de red, así como la construcción de una base de datos grande con señales EMG.

Después de revisar la literatura nos dimos cuenta de que la mayoría de los enfoques para el reconocimiento de caracteres de manuscritos que usan señales sEMG están basados en el preprocesamiento y la extracción de características, seguidos de un clasificador lineal como LDA (Linear Discriminant Analysis) [6]. Sin embargo, observamos que hay algunos otros enfoques que se basan en el preprocesamiento seguido de un método de alineamiento óptimo de secuencias tal que DTW (Dynamic Time Warping) [7].

De manera general, se puede afirmar que todos los enfoques son muy restrictivos, en el sentido de que requieren un ambiente bien controlado que permita obtener las señales EMG con pocas variaciones para facilitar su clasificación. Por ejemplo, usualmente los electrodos se colocan cuidadosamente sobre los músculos para obtener señales EMG de músculos individuales. Además, todos los enfoques sólo se diseñaron para caracteres escritos de un solo trazo.

En este trabajo se propone un enfoque más general para el reconocimiento de caracteres manuscritos multiusuario, de estilo libre, de varios trazos y sin restricciones en la colocación de los electrodos.

La Figura 24 muestra el diagrama general del sistema propuesto. Los pasos más importantes son la calibración del brazalete, la tarea de escritura manuscrita, la captura de señales sEMG (brazalete Myo™: Generación de señales sEMG en bruto, filtrado y muestreo), normalización de la magnitud y la longitud de la secuencia para finalmente enviar los datos de la señal sEMG de 8 canales a la arquitectura neuronal para su reconocimiento.

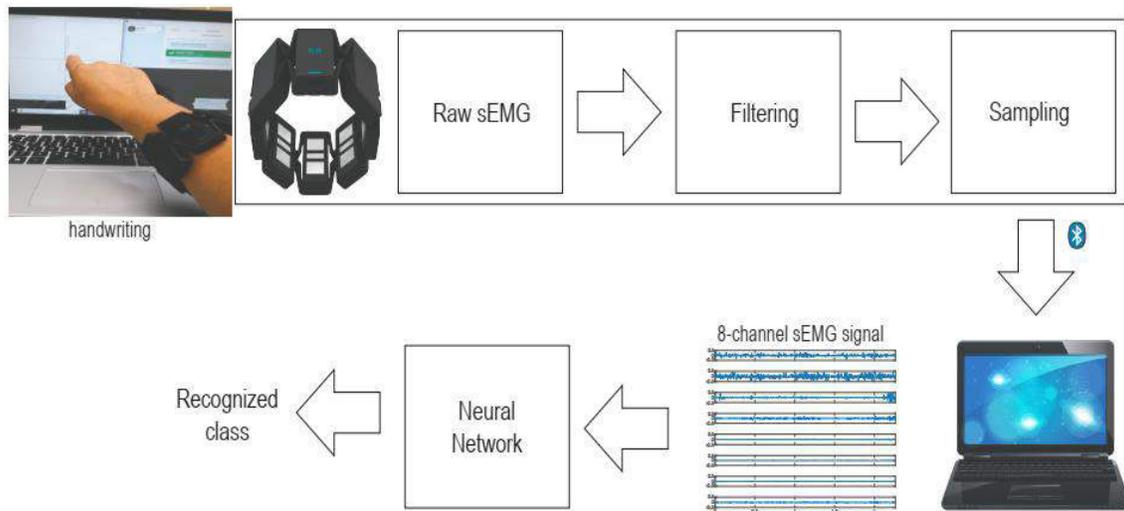


Figura 24. Vista general del sistema de reconocimiento de caracteres manuscritos.

4.1 Construcción de la base de datos

En esta parte, nuestro objetivo es construir una base de datos de señales EMG que sea suficientemente grande y representativa para entrenar redes DCNN. Cada sujeto de prueba llevará un brazalete Myo™ en el antebrazo, y no se impondrá ninguna restricción en cuanto a la posición predefinida de los electrodos. El individuo puede estar sentado, acostado o parado en la posición que prefiera para estar cómodo, para comenzar a escribir.

Cuando ya estamos cómodamente dispuestos, abrimos nuestra aplicación para efectuar el experimento de escritura y captura de señales EMG. Una vez abierta la aplicación, procedemos a la calibración del brazalete Myo™. Enseguida, ya calibrado el brazalete, comenzamos el experimento de adquisición de las señales EMG generadas durante el proceso de escritura manuscrita. Durante el proceso de creación de la base de datos y una vez terminado el trazo de un carácter, se procede a graficar el trazo manuscrito para verificar si se trazó correctamente el carácter manuscrito.

La Figura 25 nos muestra los componentes globales de la construcción de la base de datos de señales EMG de caracteres manuscritos con el enfoque de reconocimiento.

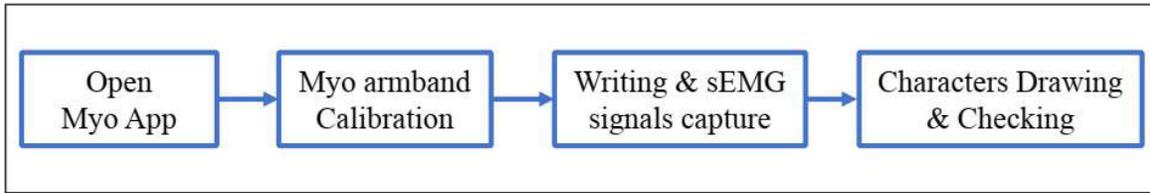


Figura 25. Marco general para la construcción de la base de datos utilizada.

Como se puede observar, hay cuatro bloques dedicados al sistema de adquisición de caracteres y especialmente a la construcción de la base de datos mediante un software programado para controlar los patrones capturados.

Los datos en bruto de las señales sEMG se capturaron por medio de un brazalete Myo™. El brazalete Myo™ es un dispositivo electrónico comercial desarrollado por Thalmic Labs®. Este sistema es un dispositivo de bajo coste fácil de usar e inalámbrico de ocho canales con electrodos secos que funciona a una frecuencia de muestreo de 200 Hz. En la Figura 26 se muestra un sujeto portando un brazalete Myo™.



Figura 26. Un sujeto lleva un brazalete Myo en el antebrazo.

La aplicación de captura se desarrolló en Python, bajo Windows 10. A grandes rasgos, el procedimiento de captura funciona de la siguiente manera. En primer lugar, la aplicación para capturar las señales sEMG se inicia desde la línea de un comando. La aplicación despliega una GUI con la opción de terminar la aplicación en cualquier momento cuando el usuario lo quiera hacer. A continuación, una persona se coloca un brazalete Myo™ en el

antebrazo izquierdo o derecho (véase la Figura 27). Posteriormente, el brazalete se enciende y se conecta a la PC a través de un adaptador vía Bluetooth que se conecta a un puerto USB.



Figura 27. Brazalete Myo.

Una vez que el dispositivo se ha detectado correctamente, hay que crear y seleccionar un perfil de calibración personal utilizando el software Myo™ Armband Manager, ver Figura 28.

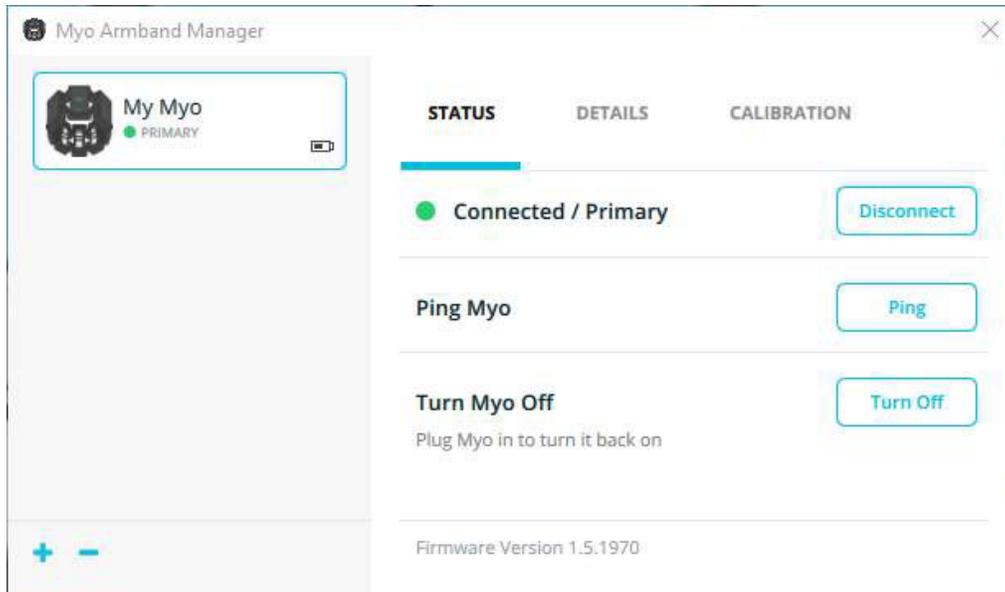


Figura 28. El software Myo Armband Manager.

En la Figura 29 se muestra un mensaje indicando que tenemos que relajar el brazo para permitir que el brazalete Myo™ se inicialice.

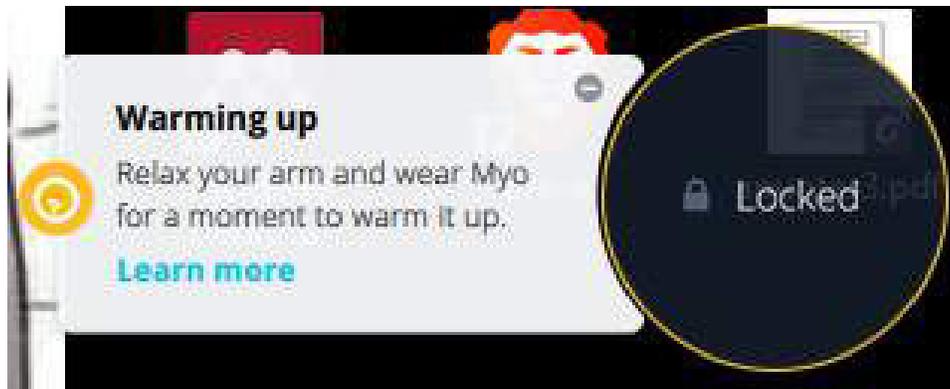


Figura 29. Relajando el brazo para permitir la inicialización del software de calibración.

La Figura 30 muestra el software para crear un perfil de calibración del brazalete Myo™ una vez que lo ponemos en nuestro brazo, para poder utilizarlo con exactitud y poder obtener nuestras señales sEMG.

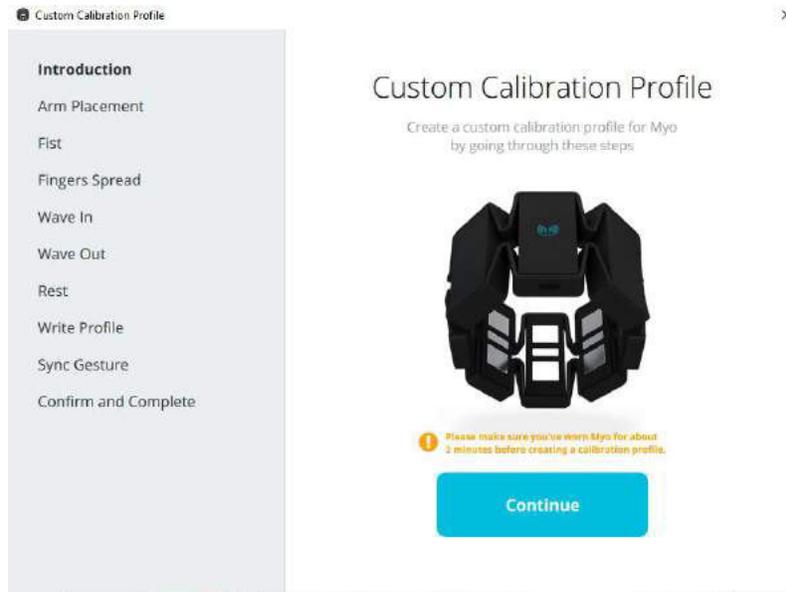


Figura 30. Selección del perfil de calibración del brazalete Myo™.

Una vez que indicamos continuar en la Figura 30, comenzamos la calibración del brazalete Myo™ como se ve en la Figura 31.

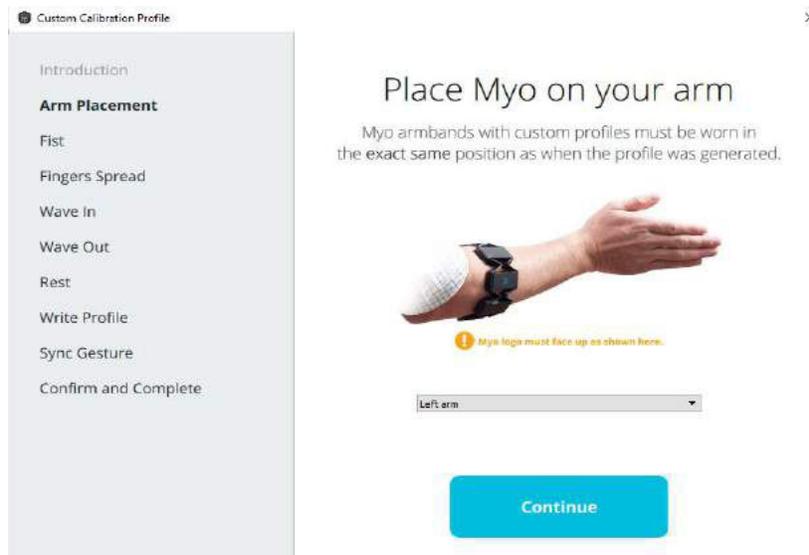


Figura 31. Comienzo de la calibración del brazalete Myo™.

Cerrar el puño es uno de los tantos movimientos que nos pide el software hacer para que el brazalete se calibre, ver la Figura 32.

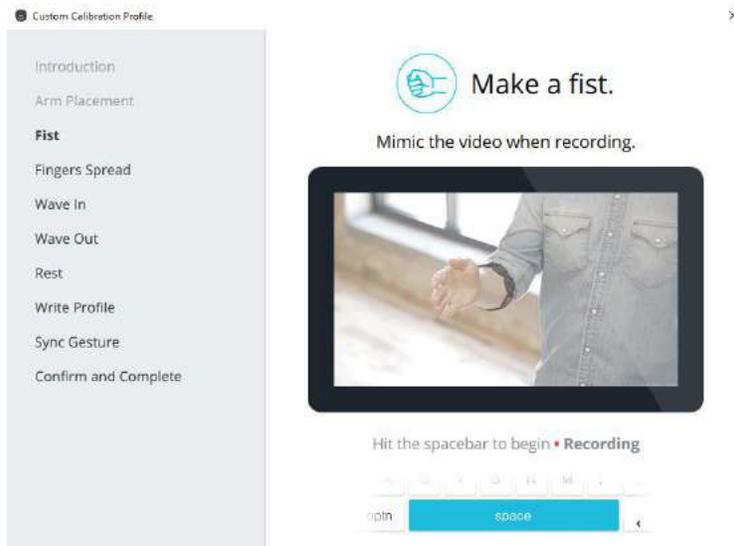


Figura 32. Movimiento de cerrar el puño para calibrar el brazalete Myo™.

Después de cerrar el puño, el software solicita efectuar otros movimientos para poder terminar de calibrar exitosamente el brazalete Myo™, ver la Figura 33.

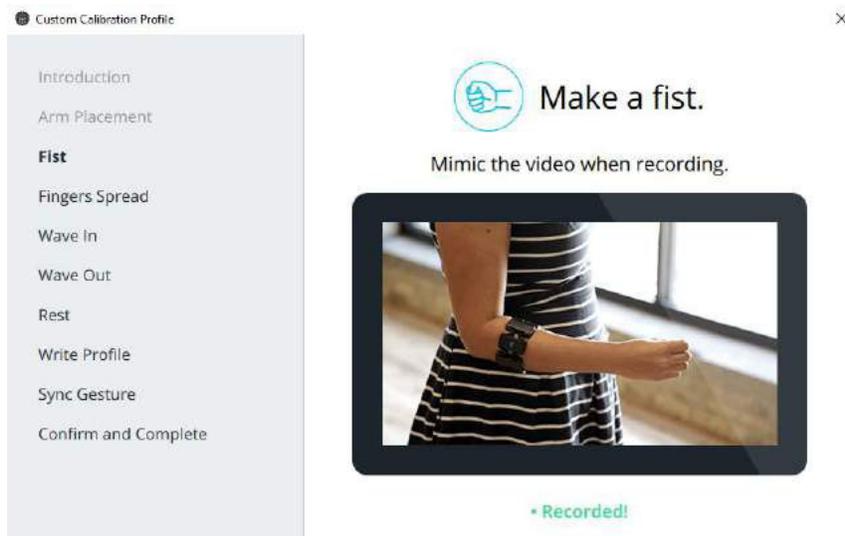


Figura 33. Grabación del movimiento del puño.

En la Figura 34 se muestra el momento cuando ya terminamos y quedó calibrado el brazalete Myo™, y a partir de este momento podemos utilizarlo para capturar datos.

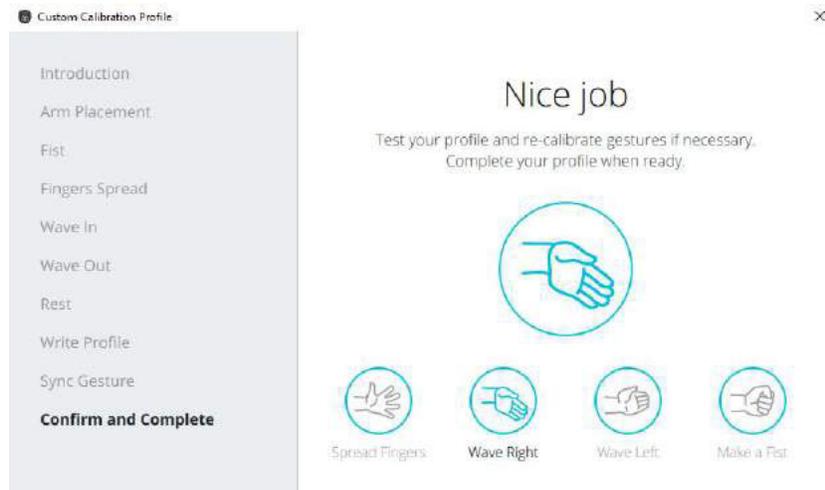


Figura 34. Terminación de la calibración del brazalete Myo™.

En la Figura 35 se muestra el momento en que ya quedó guardado nuestro perfil personalizado o nuestra calibración personal del brazalete Myo™.

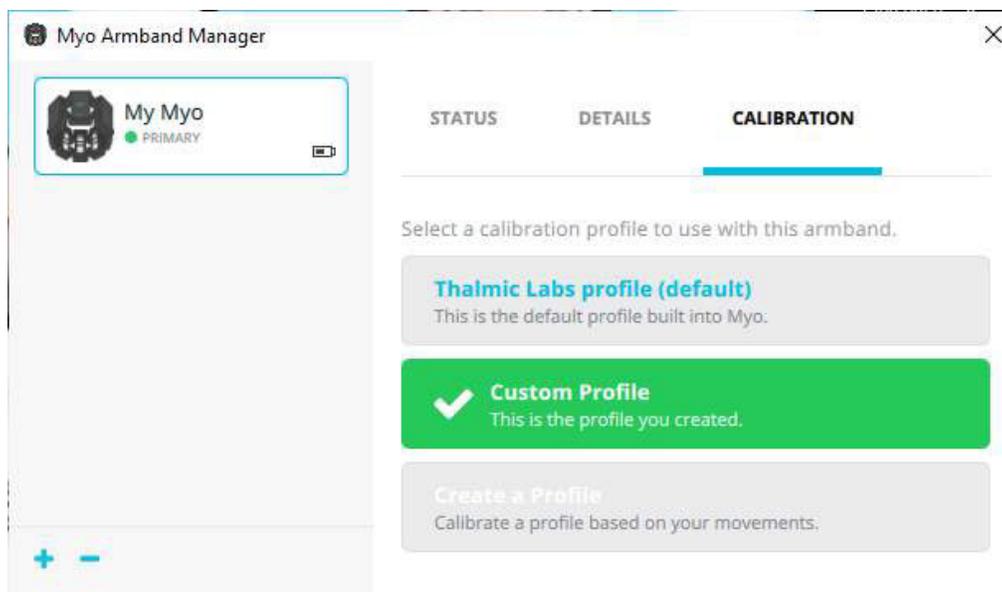


Figura 35. Momento en que ya se ha guardado el perfil de calibración personal.

La Figura 36 muestra el estado del software del brazalete ya calibrado para que sea utilizado en la adquisición de las señales EMG.

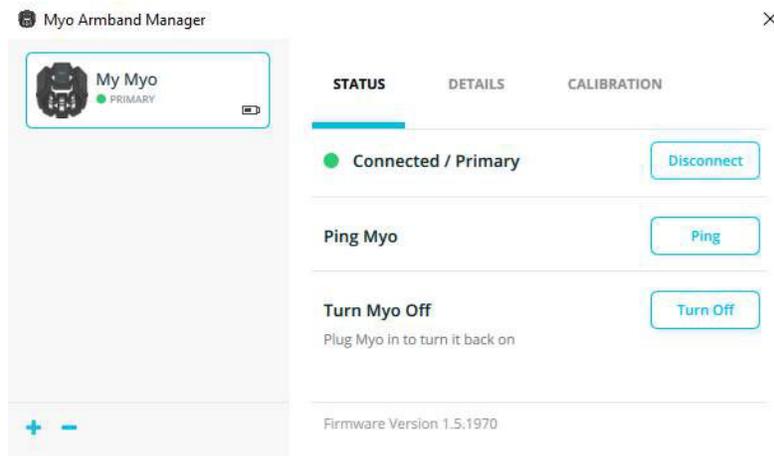


Figura 36. Estado del software de captura después de la calibración de la pulsera.

A continuación, el usuario debe escribir un carácter (letra o dígito) en un campo de texto en la interfaz gráfica del usuario. Si el carácter tecleado (o la clase) es errónea, el software sigue ofreciendo la posibilidad de cancelar la captura. Luego, el usuario debe hacer clic en el botón Iniciar para comenzar el proceso de captura. En este momento el sujeto debe comenzar a escribir con el dedo índice (o un lápiz óptico) en la pantalla táctil, como se ve en la Figura 37.

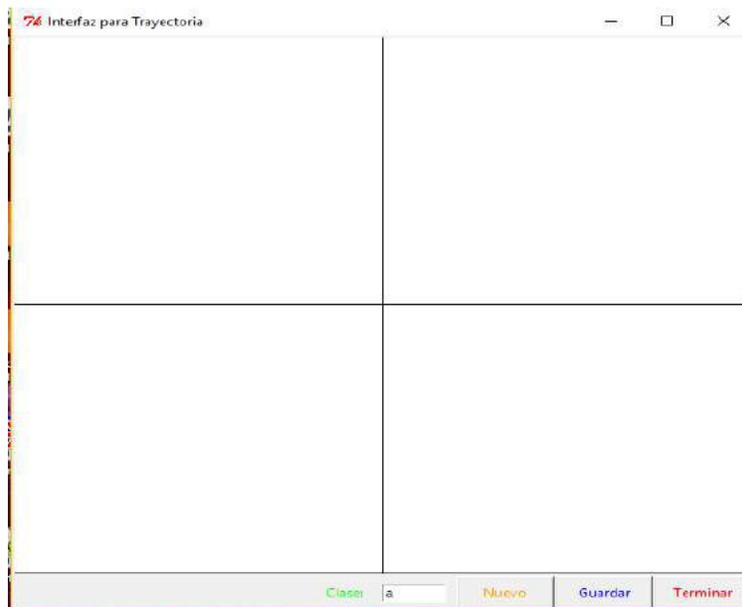


Figura 37. GUI para trazar un carácter manuscrito y capturar las señales sEMG.

Cuando el sujeto ha terminado de trazar el carácter manuscrito, debe hacer clic en el botón Guardar para escribir los datos de captura en archivo. Los datos capturados comprenden ocho señales correspondientes a los ocho sensores del brazalete Myo™. La aplicación termina cuando el sujeto hace clic en el botón Terminar, ver Figura 38.

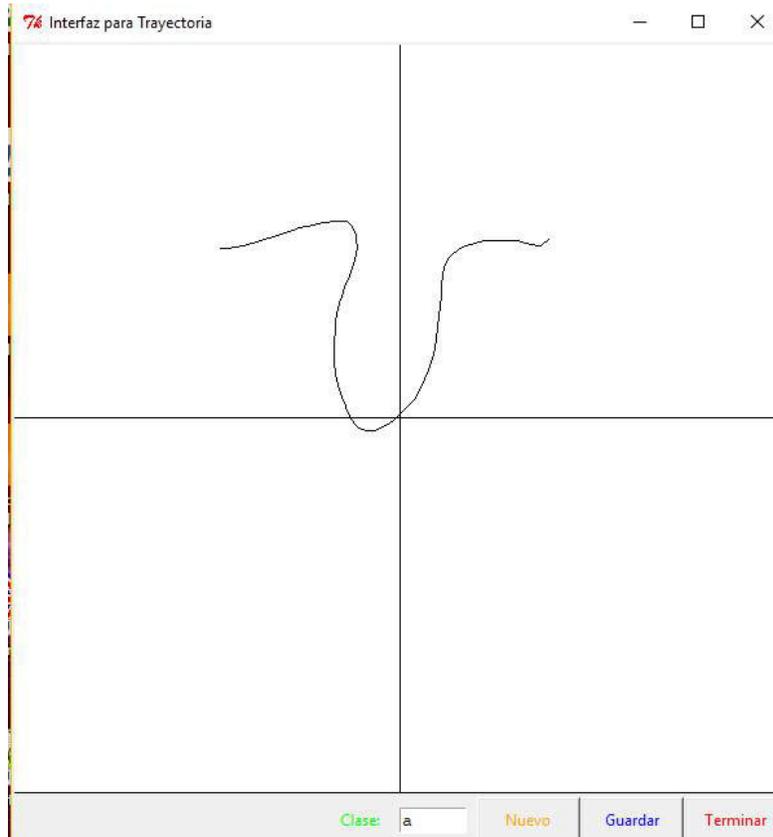


Figura 38. Un carácter manuscrito trazado usando la GUI de captura de señales sEMG.

Los caracteres se escriben a mano alzada con el dedo índice o con un lápiz óptico. Además, las coordenadas (x, y) de los movimientos de escritura se visualizarán en la pantalla. Para ello, recordemos que antes de cada tarea de escritura manuscrita, el brazalete Myo™ se calibra para que el dispositivo siempre comience con la misma configuración personal para los diferentes usuarios.

En total, tres sujetos sanos sentados en una posición cómoda participaron en la creación de la base de datos. Cada tarea consistía en dibujar con el dedo índice (o con el lápiz óptico), caracteres en minúsculas ("a"-“z”), y dígitos ("0"-“9”). Los sujetos eran libres de

utilizar su estilo particular de escritura con uno o más trazos. En la Figura 39 se muestran tres ejemplos de tareas de escritura y captura.

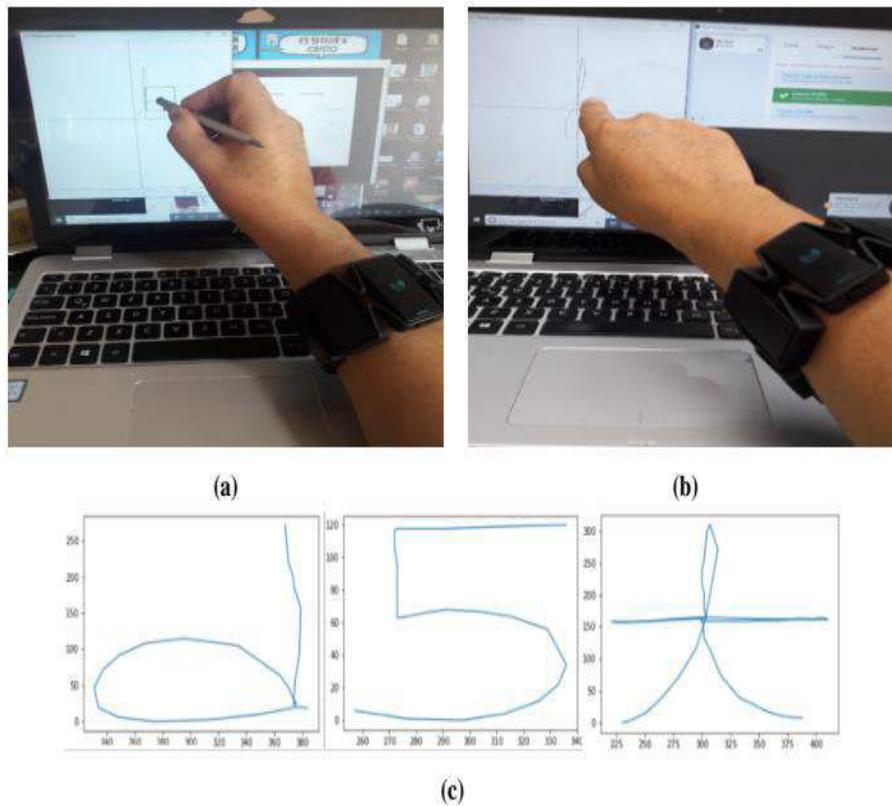


Figura 39. Ejemplo de tareas de escritura y captura. a) Usando un lápiz óptico. b) Usando el dedo índice. c) Algunos ejemplos de caracteres manuscritos.

Dado que un total de 3 sujetos participaron en la creación de la base de datos, el número de clases es de 36 (26 letras y 10 dígitos), y cada sujeto o persona escribió cada clase de carácter mil veces, de manera que el número total de capturas es de 108,000 (es decir, $3 \times 36 \times 1,000$). Por tanto, la base de datos contiene en total 108,000 capturas de señales sEMG multicanal, el número de clases es 36 y el número de capturas por clase es 3,000. Además de señales sEMG, la base de datos también proporciona la trayectoria de cada carácter manuscrito junto con el número de trazo. Esto puede ser útil para la visualización o para investigación (es decir, la reconstrucción de la trayectoria de la escritura).

La Figura 40 muestra el patrón representativo (promedio) de todas las clases. Las amplitudes son pequeñas en la parte derecha porque hay menos patrones con longitudes superiores a 2.5 segundos.

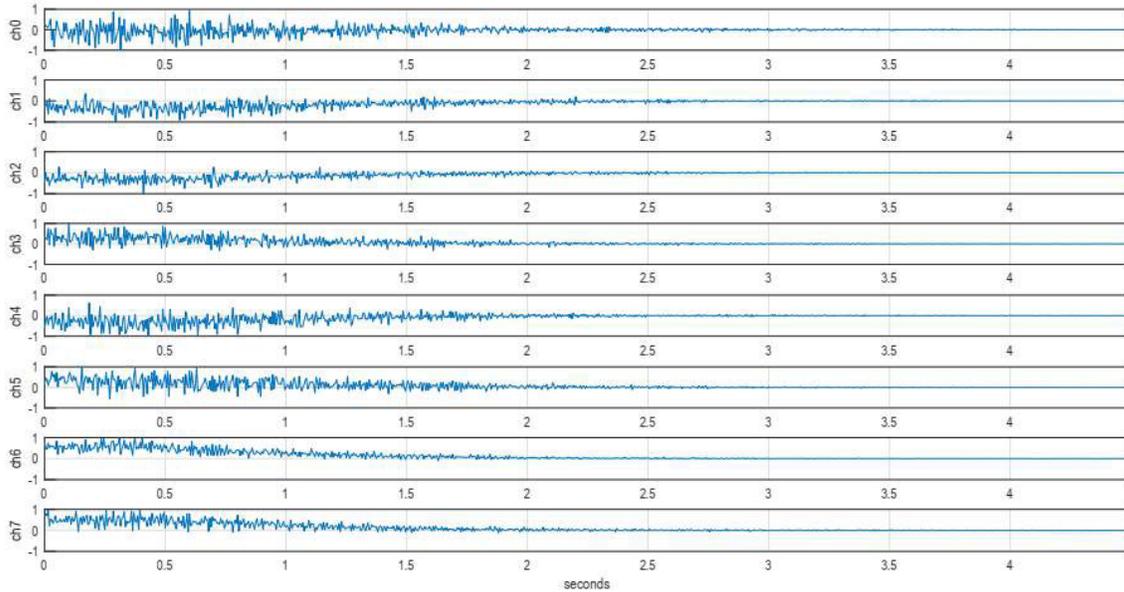


Figura 40. Patrón representativo (promedio) de la base de datos sobre todas las clases.

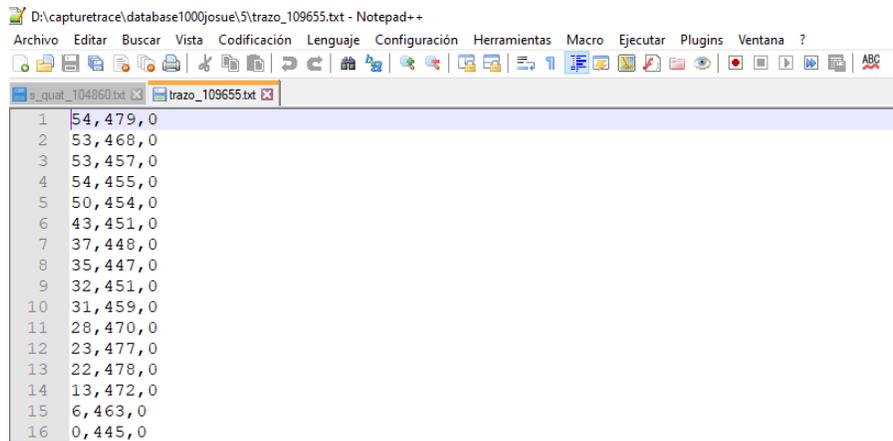
```

s_quat_104860.txt
1 -4, -6, -4, 0, 3, 0, 4, 11, 0
2 -31, -10, -6, -2, -2, -1, 3, -17, 0
3 17, -8, -6, -2, 2, 4, 9, 0, 0
4 32, 11, 4, -4, -5, -10, -18, 12, 0
5 29, 2, -6, -3, 0, -4, -6, -34, 0
6 12, -6, 3, 3, -1, 2, 0, 27, 0
7 4, -2, -1, 1, 3, 1, 5, 9, 0
8 -15, 4, -7, -4, -5, -6, -6, -12, 0
9 13, 2, 2, 2, 3, 5, 1, -6, 0
10 -13, -10, -3, -1, -3, -10, -11, -16, 0
11 -8, -12, 5, -4, -2, 5, 6, -4, 0
12 -3, 10, -16, -2, -1, -1, 1, 5, 0
13 4, -2, 1, -1, -1, -4, -8, 11, 0
14 -41, -13, -3, 1, 1, 4, -3, -35, 0
15 45, 9, 2, -2, -2, 5, 9, 40, 0
16 -14, -6, 13, 0, -8, -10, -6, 10, 0
17 13, -5, -14, -5, 0, -2, -1, 6, 0
18 -11, 9, 21, 4, 1, 1, -1, 5, 0
19 -18, -11, -17, 2, 3, 6, 3, -15, 0
20 -2, 9, 2, -1, -3, -7, -8, -18, 0
21 -6, -7, -5, 1, 3, 8, 8, 17, 0
22 4, 0, 12, 4, 1, -1, 2, -2, 0
23 -18, 1, -7, 2, 2, 1, -2, -9, 0
24 4, 5, 4, 4, 5, 5, 7, 0
  
```

Figura 41. Ejemplo de captura de señales sEMG guardada en un archivo.

La Figura 41 muestra un ejemplo de las señales capturadas y guardadas en un archivo. Los primeros 8 valores enteros corresponden a los 8 canales y el último valor corresponde al número de trazo.

La Figura 42 muestra un ejemplo de las coordenadas x, y de una captura del trazo de un carácter manuscrito guardada en archivo.



```
D:\capturetrace\database1000josue\5\trazo_109655.txt - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
s_qual_104860.txt trazo_109655.txt
1 54,479,0
2 53,468,0
3 53,457,0
4 54,455,0
5 50,454,0
6 43,451,0
7 37,448,0
8 35,447,0
9 32,451,0
10 31,459,0
11 28,470,0
12 23,477,0
13 22,478,0
14 13,472,0
15 6,463,0
16 0,445,0
```

Figura 42. Ejemplo de coordenadas x, y de una captura de trazo manuscrito.

La base de datos que se creó es una gran aportación a la comunidad científica para que en lo futuro se desarrollen más trabajos, ya que no existe un repositorio de tantos caracteres manuscritos convertidas en señales EMG, y hay muy pocos caracteres hechos por diferentes personas, en esta base de datos son 3 personas diferentes con mil caracteres hechos de cada letra del abecedario que se construyó para la realización de este trabajo.

4.2 Arquitectura de red neuronal

La base de datos construida contiene señales sEMG con todo tipo de variaciones debidas en gran parte al cambio de la posición de los sensores y al número de sujetos. Para hacer frente a la variabilidad de las señales sEMG, se sugiere el uso de DCNNs para el procesamiento de la señal y la extracción de características [60, 61].

Las arquitecturas DCNN se caracterizan por su eficacia y robustez para extraer las características discriminantes directamente de los datos brutos. Además, las DCNNs utilizan una representación jerárquica para aprender y decodificar los datos originales a partir de capas especializadas aplicadas a una tarea de reconocimiento.

Por otro lado, las Redes Neuronales Recurrentes (RNN), en particular, las redes de Memoria de Largo y Corto Plazo (LSTM) y GRU se han aplicado con éxito al reconocimiento de movimientos [62]. Por lo tanto, para el procesamiento y reconocimiento de secuencias,

optamos por las redes neuronales recurrentes (RNN). En particular, estamos interesados en utilizar dos de las mejores redes recurrentes: LSTM y GRU.

Por lo tanto, se propusieron cuatro arquitecturas de redes neuronales para la etapa de clasificación de secuencias, que combinan redes DCNN y redes RNN, denominadas DCNN-LSTM, DCNN-2LSTM, DCNN-GRU y DCNN-2GRU.

Cabe mencionar que la arquitectura DCNN-GRU es similar a la arquitectura DCNN-LSTM. La principal diferencia radica en el uso de una capa GRU en lugar de una capa LSTM. Por su parte, las arquitecturas DCNN-2LSTM y DCNN-2GRU usan dos capas LSTM y GRU apiladas, respectivamente.

La Figura 43 muestra la arquitectura de la red CNN-LSTM. Esta arquitectura comprende una red DCNN que consta de dos capas convolucionales y una capa de agrupamiento máximo (MaxPooling).

La salida de la red DCNN (es una secuencia) es aplicada a la entrada de una red LSTM que la procesa y luego es clasificada usando una capa totalmente conectada. Cada capa convolucional tiene una función de activación ReLU y filtros de tamaño 3×1 .

La capa totalmente conectada tiene 36 salidas y una función de activación SoftMax.

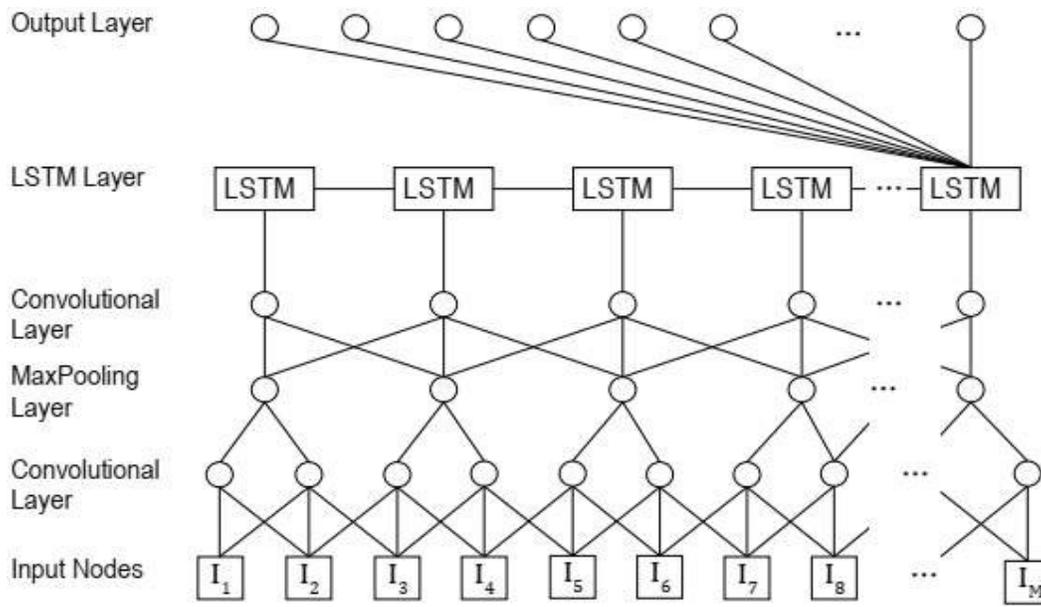


Figura 43. Arquitectura CNN-LSTM.

La Figura 44 muestra la arquitectura de la red CNN-GRU. Notamos que esta arquitectura también comprende una red DCNN que consta de dos capas convolucionales y una capa de agrupamiento máximo (MaxPooling). La salida de la red DCNN (es una secuencia) es aplicada a la entrada de una red GRU que la procesa y luego es clasificada usando una capa totalmente conectada. Cada capa convolucional tiene una función de activación ReLU y filtros de tamaño 3×1 . La capa totalmente conectada tiene 36 salidas y una función de activación SoftMax.

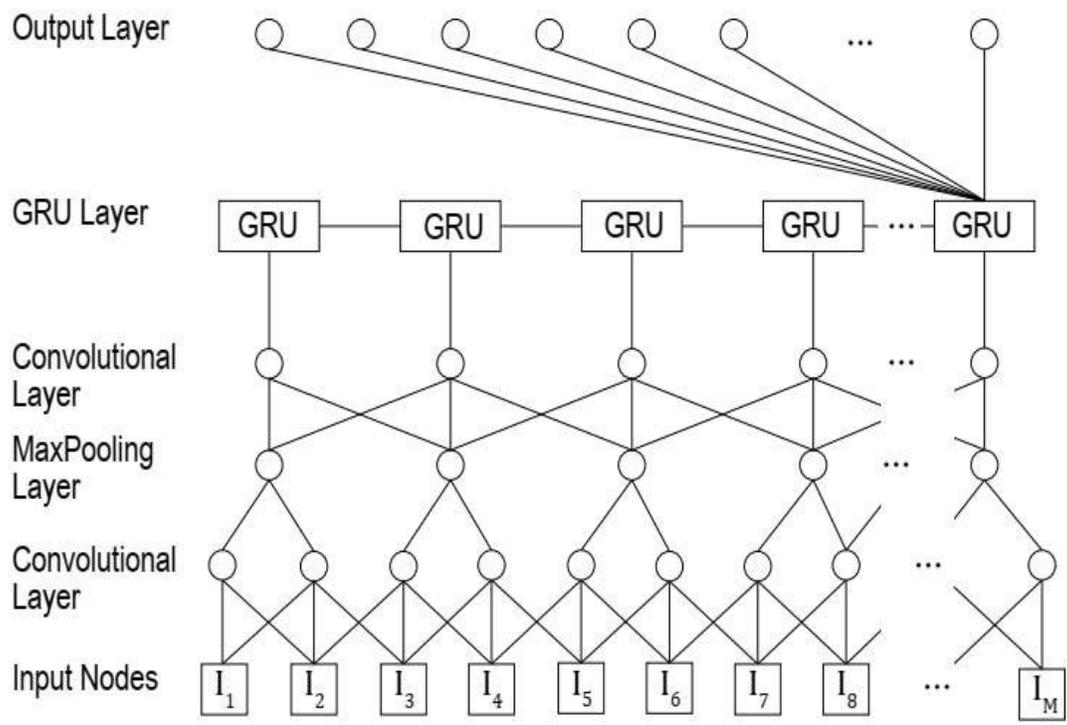


Figura 44. Arquitectura CNN-GRU.

Capítulo 5

Pruebas y Resultados

Las arquitecturas de las redes neuronales discutidas en el capítulo anterior fueron implementadas en Python usando Keras y Tensorflow con: la función de pérdidas basada en la entropía cruzada categórica, el optimizador conocido como Nadam, las métricas fundamentales definidas como exactitud. Por cierto, las clases de la base de datos están balanceadas.

En concreto la entrada a cada red neuronal es una matriz de tamaño (900, 8), con secuencias normalizadas de señales EMG de ocho canales, un patrón obtenido del brazalete en cada captura, el número de salidas es de 36. Cada salida representa una estimación de una probabilidad de clase.

La base de datos se dividió en conjunto de entrenamiento y de prueba para evaluar el rendimiento de las arquitecturas propuestas. El conjunto de prueba contiene el 10% de los datos, mientras que el resto de los datos formaron el conjunto de entrenamiento. Así, se utilizaron 97,200 capturas para el entrenamiento y 10,800 para las pruebas.

Todas las arquitecturas se entrenaron y probaron en una estación de trabajo con un procesador INTEL® Core i7, 16 GB de RAM, GPU Nvidia® GeForce RTX™ 2080 y Windows 10, utilizando Keras y Tensorflow™.

Para todos los casos, se utilizó el algoritmo Nadam [63] para el entrenamiento utilizando la función de pérdida de entropía cruzada y un tamaño de lote de 1024. En primer lugar, cada arquitectura se entrenó con parámetros similares de ajuste durante 8,000 épocas utilizando el algoritmo de optimización de tasa de aprendizaje cíclica [64]. Así, las tasas de aprendizaje mínima y máxima se fijaron en 0.000001 y 0.001, respectivamente. Además, el tamaño del paso se definió como cinco veces el número de iteraciones en cada época.

En segundo lugar, cada ensayo de entrenamiento se repitió varias veces con diferentes condiciones iniciales para ayudar a evitar los mínimos locales. Además, la tasa de Dropout

se varió de 0.8 a 0.6 para ayudar a reducir los efectos del sobreajuste (overfitting). Asimismo, el número de unidades en las capas de LSTM y GRU se varió de 80 a 200.

La tabla 1 muestra los resultados obtenidos para las cuatro arquitecturas utilizando sólo muestras de un sujeto o persona.

Arquitectura	Exactitud (%)
CNN-LSTM	96.37
CNN-2LSTM	96.48
CNN-GRU	97.03
CNN-2GRU	96.29

Tabla 1. Resultados obtenidos de cada arquitectura.

Como se puede observar, la mejor exactitud (97.03%) se consiguió para la arquitectura CNN-GRU, mientras que para la arquitectura CNN-2GRU se obtuvo la menor exactitud (96.29%).

La tabla 2 muestra los resultados para la arquitectura CNN-GRU por cada uno de los sujetos.

Sujeto	Exactitud (%)
Sujeto 1	97.04
Sujeto 2	95.05
Sujeto 3	94.89
Todos los sujetos	94.85

Tabla 2. Resultados obtenidos para la arquitectura CNN-GRU por sujeto.

Podemos observar que la mayor exactitud se obtuvo para el sujeto 1 y la menor para el sujeto 3. La exactitud promedio para todos los sujetos fue del 94.85%, la cual es más baja que la exactitud para un solo sujeto o persona. Este efecto se lo atribuimos a la mayor variabilidad de las señales sEMG para todos los sujetos.

La tabla 3 muestra una comparación cuantitativa con estudios representativos del estado del arte.

Método	Ejemplos	Sujetos	Clases	Exactitud (%)
Template matching [13]	350	6	10	97
DTW [12]	520	3	26	84.29
DTW [14]	780	4	26	78.24 a 92.42
Propuesto	108k	3	36	94.85

Tabla 3. Comparación del método propuesto con los mejores métodos.

A partir de los resultados numéricos, se puede observar y concluir que el enfoque propuesto es altamente competitivo y en general, se compara bien con los métodos ya existentes en el estado del arte.

La Figura 45 muestra la matriz de confusión obtenida para la arquitectura CNN-GRU para una persona.

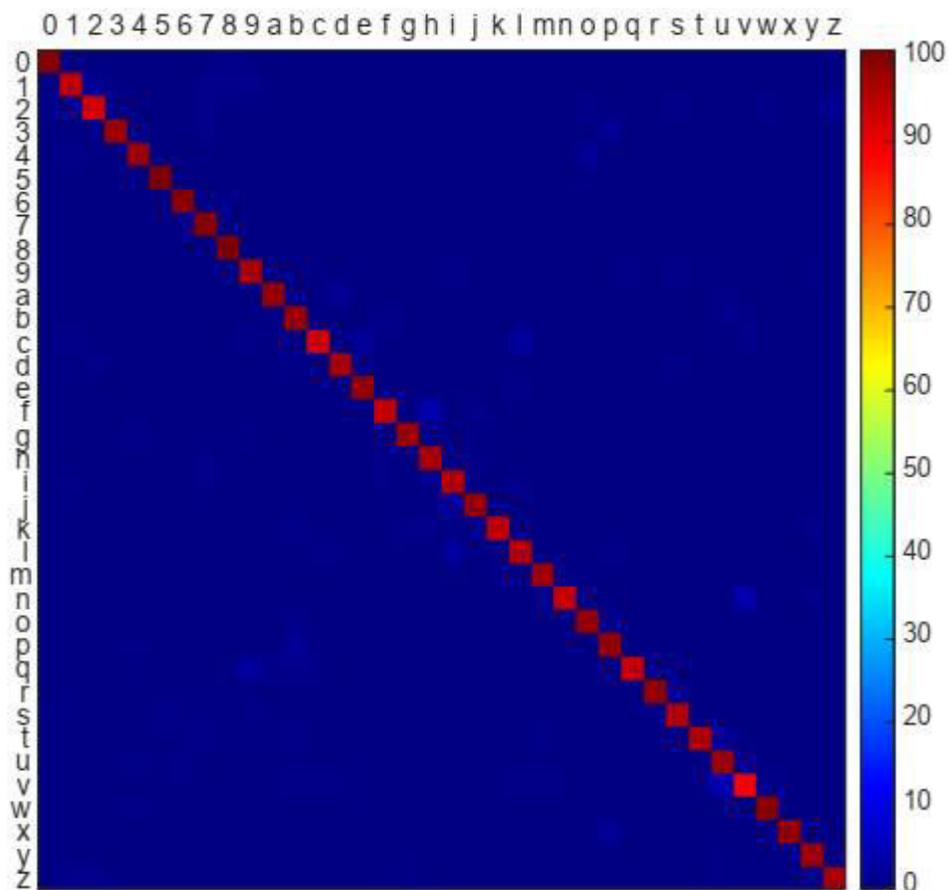


Figura 45. Matriz de confusión para la arquitectura CNN_GRU con datos de un sujeto.

La Figura 46 muestra la matriz de confusión obtenida para la arquitectura CNN-GRU para todos los sujetos.

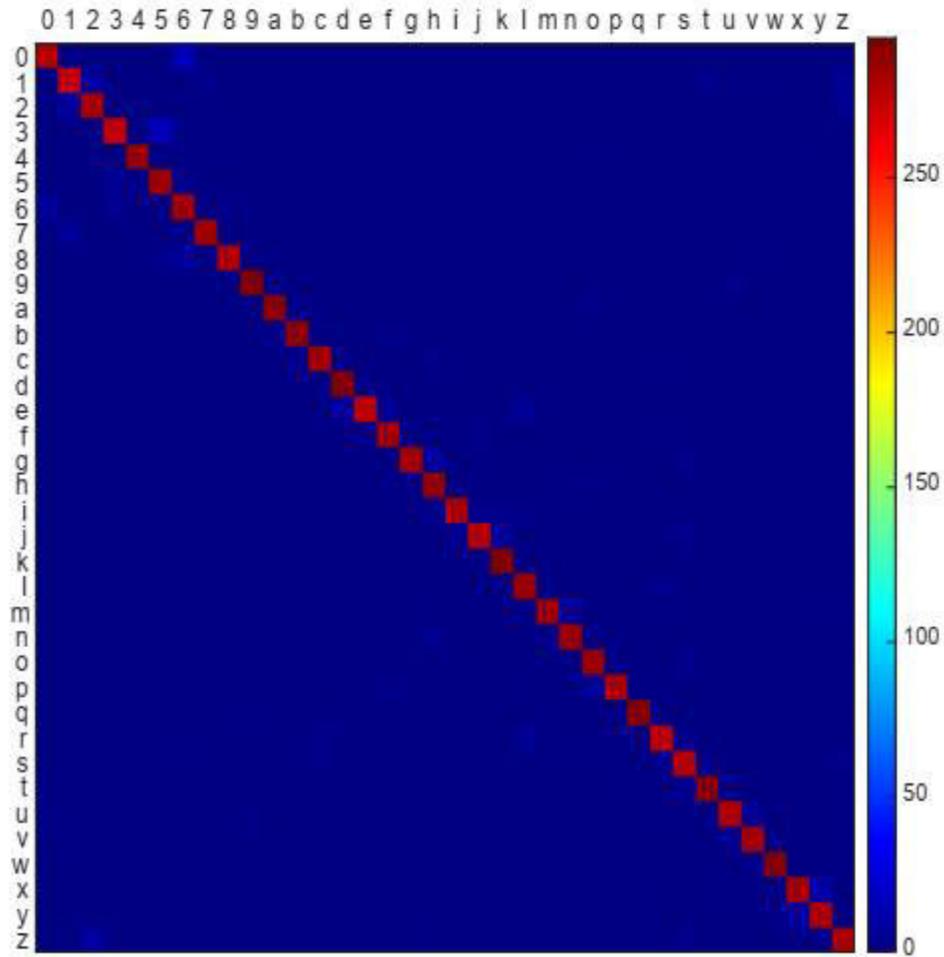


Figura 46. Matriz de confusión para la arquitectura CNN_GRU con todos los sujetos.

Cabe destacar que todas las clases se reconocen con alta exactitud para las muestras de todos los sujetos de prueba. Las figuras 47 y 48 muestran la curva ROC obtenida para la arquitectura CNN-GRU para una persona y para todos los sujetos.

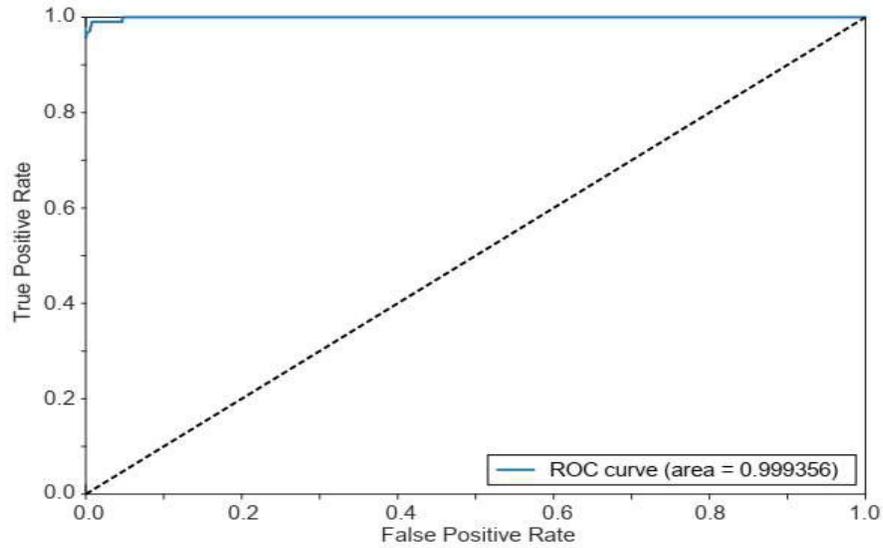


Figura 47. Curva ROC de la arquitectura CNN-GRU (una clase, un sujeto).

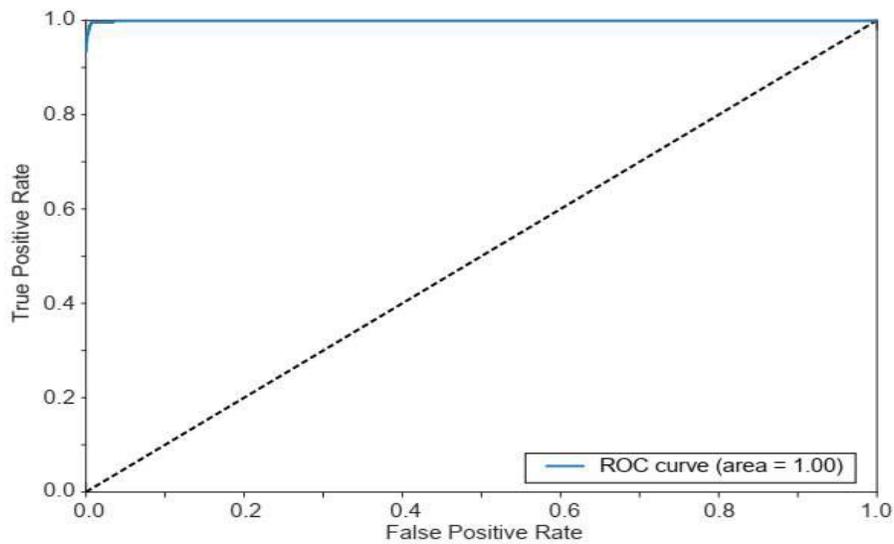


Figura 48. Curva ROC de la arquitectura CNN-GRU (una clase, todos los sujetos).

Como puede verse, el área ROC está muy cerca de uno, lo que indica una excelente capacidad de separación entre clases. Se obtuvo un resultado similar para las otras 35 clases.

Capítulo 6

Conclusiones

Se hicieron varios experimentos para lograr los resultados mostrados y encontrados, a continuación, se propuso una arquitectura de red y se empezaron a variar los hiperparámetros y se hicieron varias repeticiones de cada entrenamiento, las redes se probaron explorando un gran número de combinaciones de hiperparámetros hasta encontrar la arquitectura más adecuada para alcanzar el máximo porcentaje de reconocimiento.

Los resultados numéricos mostraron que las cuatro arquitecturas propuestas presentaban exactitudes comparables. Sin embargo, la arquitectura más rápida (CNN-GRU) también presentó la mayor exactitud y vale la pena mencionar que estos últimos resultados arrojaron una alta exactitud, aunque no se realizó ningún preprocesamiento.

Por estas razones, se eligió la arquitectura CNN-GRU para la tarea de reconocimiento de caracteres manuscritos de estilo libre y multiusuario. No obstante, siempre hay opciones para modificar los sistemas actuales. Por ejemplo, una oportunidad de mejora es la posibilidad de probar nuestro enfoque utilizando una base de datos, multiusuario más amplia de señales sEMG.

Se ha presentado una nueva metodología para el reconocimiento de caracteres de manuscritos de varios trazos a partir de señales sEMG utilizando un brazalete Myo™. El enfoque propuesto consiste en utilizar arquitecturas robustas de Deep Learning para la extracción de características (DCNN) y arquitecturas de reconocimiento de secuencias como las RNNs. Una de las ventajas del enfoque propuesto es que los caracteres también pueden escribirse en el aire. La arquitectura CNN-GRU logró una exactitud promedio del 94.85% para la tarea de reconocimiento de caracteres manuscritos de varios usuarios. En comparación con trabajos anteriores, los resultados obtenidos en este trabajo son prometedores a pesar de que las señales sEMG presentaron variaciones en sus formas debido a las diferencias en la colocación del brazalete.

Por lo tanto, este estudio es una contribución esencial al área de las interfaces cerebro-ordenador. Las interfaces cerebro-ordenador ofrecen una alternativa para restaurar o mejorar la capacidad de comunicación de las personas y sin discapacidad.

De acuerdo con los resultados obtenidos, se cumplieron los objetivos establecidos en la introducción, confirmando esto la hipótesis planteada. Se logró tener un sistema de reconocimiento de escritura manuscrita basada en señales EMG, con una muy buena tasa de error, se construyó una base de datos grande de señales EMG de diferentes personas para tener una muestra representativa de caracteres manuscritos, se propuso una arquitectura de la red neuronal que es adecuada para el reconocimiento de los caracteres manuscritos.

Bibliografía

- [1] L. Cromwell, Instrumentación y medidas biomédicas, Primera Edición ed., Marcombo, Ed., Barcelona: Boixareu Editores, 1980.
- [2] I. Somlai-Schweigner, Adquisición Superficial de Señales Mioeléctricas: Procesamiento y Clasificación de la Actividad Eléctrica Muscular del Antebrazo para el Control de una Prótesis de Mano, primera ed., Editorial Academia Española, 2011.
- [3] G. Michelle, Valoración de la electrografía de superficie multicanal para el análisis de la mialgia de esfuerzo en músculos del antebrazo, UPC, Ed., Barcelona, 2010.
- [4] L. Gila, A. Malandra y I. Rodríguez-Carreño, «Métodos de procesamiento y análisis de señales electrográficas.,» *Sist. Sanit Navar*, vol. 32, nº 3, 2009.
- [5] J. Maldarelli, B. Kahrs, S. Hunt y J. Lockman, «Development of early handwriting: Visual-motor control during letter copying,» *Dev. Psychol*, p. 879–888, 2015.
- [6] M. Linderman, M. A. Lebedev y J. S. Erlichman, «Recognition of Handwriting from Electromyography,» *PLoS ONE*, p. e679, 2009.
- [7] G. Huang, D. Zhang, X. Zheng y X. Zhu, «An EMG-based handwriting recognition through dynamic time warping,» de *Proc. of Annual International Conference of the IEEE Engineering in Medicine and Biology*, 2010.
- [8] J. Calvo-Zaragoza y J. Oncina, «Recognition of pen-based music notation with finite-state machines,» *Expert Systems with Applications*, p. 395–406, 2017.
- [9] N. Mendes, M. Simão y P. Neto, «Segmentation of electromyography signals for pattern recognition,» de *Proc. of 45th Annual Conference of the IEEE Industrial Electronics Society*, 2019.
- [10] M. Słapek y S. Paszkiel, «Detection of gestures without begin and end markers by fitting into Bezier curves with least squares method,» *Pattern Recognition Letters*, p. 83–88, 2017.
- [11] K. A. Lamkin-Kennard y M. B. Popovic, «Sensors: Natural and Synthetic Sensors,» *Biomechatronics*, p. 81–107, 2019.
- [12] J. Wu, X. Li, W. Liu y Z. J. Wang, «sEMG Signal Processing Methods: A Review,» *Journal of Physics: Conference Series*, vol. 1237, nº 3, p. 032008, 2019.
- [13] E. Guigon, P. Baraduc y M. Desmurget, «Computational Motor Control: Redundancy and Invariance,» *J. Neurophysiol.*, p. 331–347, 2007.

- [14] C. J. De Luca, «Physiology and Mathematics of Myoelectric Signals,» *IEEE Trans. Biomed. Eng.*, p. 313–325, 1979.
- [15] M. Simão, N. Mendes, O. Gibaru y P. Neto, «A Review on Electromyography Decoding and Pattern Recognition for Human-Machine Interaction,» *IEEE Access*, vol. 7, p. 39564–39582, 2019.
- [16] Y. Gloumakov, J. Bimbo y A. M. Dollar, «Trajectory Control For a Myoelectric Prosthetic Wrist,» de *Proc. of MEC Symposium Conference*, 2020.
- [17] A. Lansari, F. Bouslama, M. Khasawneh y A. Al-Rawi, «A novel electromyography (EMG) based classification approach for Arabic handwriting,» de *Proc. of the International Joint Conference on Neural Networks*, 2003.
- [18] M. Linderman, M. A. Lebedev y J. S. Erlichman, «Recognition of Handwriting from Electromyography,» *PLoS ONE*, 4, p. e6791, 2009.
- [19] C. Li, Z. Ma, L. Yao y D. Zhang, «Improvements on EMG-based handwriting recognition with DTW algorithm,» de *Proc. of 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2013.
- [20] I. Chihi, A. Afef y B. Mohamed, «Analysis of Handwriting Velocity to Identify Handwriting Process from Electromyographic Signals,» *Am. J. Appl. Sci.*, p. 1742–1756, 2012.
- [21] M. A. Slim, A. Abdelkrim y M. Benrejeb, «An efficient handwriting velocity modelling for electromyographic signals reconstruction using Radial Basis Function neural networks,» de *Proc. of 7th International Conference on Modelling, Identification and Control (ICMIC)*, 2015.
- [22] E. Okorokova, M. Lebedev, M. Linderman y A. Ossadtchi, «A dynamical model improves reconstruction of handwriting from multichannel electromyographic recordings,» *Front. Neurosci.*, p. 1–15, 2015.
- [23] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli y W. Geng, «Surface-Electromyography-Based Gesture Recognition by Multi-View Deep Learning,» *IEEE Trans. Biomed. Eng.*, p. 2964–2973, 2019.
- [24] A. Dash, A. Sahu, R. Shringi, J. Gamboa, M. Z. Afzal y M. I. Malik, «AirScript-Creating Documents in Air,» de *Proc. of 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2017.
- [25] P. Roy, S. Ghosh y U. Pal, «A CNN Based Framework for Unistroke Numeral Recognition in Air-Writing,» de *Proc. of 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2018.
- [26] D. Purves, «Neurobiologia,» de *Nuerociencia*, Panamericana, 2016, p. 760.

- [27] F. Berzal, «Redes Neuronales and Deep Learning,» de *Redes Neuronales and Deep Learning*, F. Berzal, Ed., 2018, p. 753.
- [28] World Tech, «DATA SCIENTIST,» de *Aprendizaje de Refuerzo con Python: Una Guía Sin Sentido para Dominar el Aprendizaje de Refuerzo con Diferentes Técnicas de Entrenamiento y Aplicaciones*, C. I. P. Platform, Ed., CreateSpace Independent Publishing Platform, 2018, p. 110.
- [29] D. H. Hubel y T. N. Wiesel, «Receptive fields and functional architecture of monkey striate cortex,» *J. Physiol.*, p. 215–243, 1968.
- [30] D. Marr, «Analyzing natural images: A computational theory of texture vision,» de *Cold Spring Harb. Symp. Quant. Biol.*, 1976.
- [31] K. Fukushima y S. Miyake, «Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,» de *Competition and Cooperation in Neural Nets*, Berlin, Heidelberg, Springer, 1980, p. 267–285.
- [32] C. Lee, P. W. Gallagher y Z. Tu, «Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree,» *Artificial intelligence and statistics*, pp. 464-472, 2016.
- [33] J. Weng, N. Ahuja y T. S. Huang, «Cresceptron: A self-organizing neural network which grows adaptively,» de *Proc. of International Joint Conference on Neural Networks (IJCNN)*, 1992.
- [34] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard y W. Hubbard, «Backpropagation Applied to Handwritten Zip Code Recognition,» *Neural Comput*, p. 541–551, 1989.
- [35] Y. LeCun, L. Bottou, Y. Bengio y P. Haffner, «Gradient-based learning applied to document recognition,» *Proc. IEEE*, p. 2278–2324, 1998.
- [36] W. Zhang, K. Itoh, J. Tanida y Y. Ichioka, «Parallel distributed processing model with local space-invariant interconnections and its optical architecture,» *Appl. Optics*, p. 4790–4797, 1990.
- [37] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy y B. Shuai, «Recent advances in convolutional neural networks,» *Pattern Recognit.*, p. 354 – 377, 2018.
- [38] A. Krizhevsky, I. Sutskever y G. E. Hinton, «Imagenet classification with deep convolutional neural networks,» *Advances in Neural Information Processing Systems*, 2015.
- [39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh y S. Ma, «ImageNet Large Scale Visual Recognition Challenge,» *J. Comput. Vis.*, vol. 115, p. 211–252, 2015.
- [40] V. Nair y G. E. Hinton, «Rectified linear units improve restricted boltzmann machines,» de *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010.

- [41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever y R. Salakhutdinov, «Dropout: A simple way to prevent neural networks from overfitting,» *J. Mach. Learn. Res.*, vol. 15, p. 1929–1958, 2014.
- [42] J. Wang y L. Perez, «The effectiveness of data augmentation in image classification using deep learning,» *Convolutional Neural Networks Vis. Recognit.*, 2017.
- [43] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» de *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [44] S. Ioffe y C. Szegedy, «Batch normalization: Accelerating deep network training by reducing internal covariate shift,» de *Proc. of the 32nd International Conference on Machine Learning ICM*, 2015.
- [45] J. L. Elman, «Finding Structure in Time,» *Cognit. Sci.*, p. 179–211, 1990.
- [46] S. Hochreiter, Untersuchungen zu dynamischen neuronalen Netzen, Master's thesis, Institut für Informatik, 1991.
- [47] Y. Bengio, P. Simard y P. Frasconi, «Learning Long-Term Dependencies with Gradient Descent is Difficult,» *IEEE Trans. Neural Networks*, p. 157–166, 1994.
- [48] S. Hochreiter y J. Schmidhuber, «Long Short-Term Memory,» *Neural computation*, p. 1735–1780, 1997.
- [49] K. Cho, B. V. Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares y H. Schwenk, «Learning phrase representations using RNN encoder-decoder for statistical machine translation,» *arXiv preprint arXiv:1406.1078*, p. 1724–1734, 2014.
- [50] Y. Gal y Z. Ghahramani, «Dropout as a bayesian approximation: Representing model uncertainty in deep learning,» de *Proc. of International conference on machine learning (ICML)*, 2016.
- [51] V. Rodriguez, Dropout y Batch Normalization, Rodriguez, 2018.
- [52] k. He, X. Zhang, S. Ren y J. Sun, «Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,» de *Proceedings of the IEEE international conference on computer vision*, 2015.
- [53] J. Duran-Suárez, «Redes neuronales convolucionales en R : Reconocimiento de caracteres escritos a mano,» *Depósito de Investigación Universidad de Sevilla*, 22 junio 2017.
- [54] L. M. Surhone, «Softmax Activation Function,» de *Softmax Activation Function*, Betascript Publishing, 2010, p. 126.
- [55] J. Brownlee, «Probability for Machine Learning,» de *Discover How To Harness Uncertainty With Python*, 2021.

- [56] E. Lopez, «Deep Neural Networks for fault diagnostics and prognostics,» *Facultad de ciencias físicas y matemáticas*, p. 149, 2018.
- [57] N. Jawad, F. Ducatelle, G. A. D. Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber y L. M. Gambardella, «Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition,» de *Proc. of 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011.
- [58] T. Dozat, «Incorporación de Nesterov Momentum en Adam,» *Taller de ICLR 2013-2016*, 2016.
- [59] S. Ruder, «An overview of gradient descent optimization algorithms,» 19 January 2016. [En línea]. Available: <https://ruder.io/optimizing-gradient-descent/>. [Último acceso: 21 June 2021].
- [60] N. Rusk, «Deep learning,» *Nat. Methods*, p. 35, 2016.
- [61] H. Chen, Y. Zhang, G. Li, Y. Fang y H. Liu, «Surface electromyography feature extraction via convolutional neural network,» *Int. J. Mach. Learn. Cybern.*, p. 185–196, 2019.
- [62] M. Simao, P. Neto y O. Gibaru, «EMG-based online classification of gestures with recurrent neural networks,» *Pattern Recognition Letters*, p. 45–51, 2019.
- [63] T. Dozat, «Incorporating nesterov momentum into adam,» *International Conference on Learning Representations, ICLR*, 2016.
- [64] L. N. Smith, «Cyclical Learning Rates for Training Neural Networks,» de *Proc. of 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.
- [65] L. F. Bertona, «Entrenamiento de redes neuronales basado en algoritmos evolutivos,» Buenos Aires, 2005.
- [66] A. BoschRúe, J. Casas-Roma y T. Lozano-Bagén, «Deep learning: principios y fundamentos,» de *Manual. Tecnología*, 2. UOC, Ed., Universitat Oberta de Catalunya, 2019, p. 260.
- [67] L. Cromwell, *Instrumentación y medidas biomédicas*, Barcelona: Boixareu Editores, 1980.