



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Diseño de sistema BMI no invasivo con procesamiento de  
señales EEG mediante Machine Learning

Tesis presentada a la  
**Facultad de Ciencias Físico-Matemáticas**  
y al  
**Cuerpo Académico de Física Médica**  
como requisito parcial para la obtención del grado de  
**MAESTRO EN CIENCIAS**

por

Ing. Ricardo Thaddeus Páez Amaro

Asesorado por

Dr. Benito de Celis Alonso  
Dr. Javier Miguel Hernández López

Puebla, Pue.

Agosto 2021



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Diseño de sistema BMI no invasivo con procesamiento de  
señales EEG mediante Machine Learning

Tesis presentada a la

**Facultad de Ciencias Físico-Matemáticas**

y al

**Cuerpo Académico de Física Médica**

como requisito parcial para la obtención del grado de

**MAESTRO EN CIENCIAS (FÍSICA APLICADA)**

por

Ing. Ricardo Thaddeus Páez Amaro

Asesorado por

Dr. Benito de Celis Alonso

Dr. Javier Miguel Hernández López

Puebla, Pue.

Agosto 2021



**Título:** Diseño de sistema BMI no invasivo con procesamiento de señales EEG mediante Machine Learning

**Estudiante:** RICARDO THADDEUS PÁEZ AMARO

## COMITÉ

---

Dr. Eduardo Moreno Barbosa  
Presidente

---

Dr. Mario Iván Martínez Hernández  
Secretario

---

Dr. Rodolfo Palomino Merino  
Vocal

---

Dr. Benito de Celis Alonso  
Asesor

---

Dr. Javier Miguel Hernández López  
Asesor



# Agradecimientos

Este trabajo no habría sido posible sin el apoyo de las personas e instituciones que me han apoyado.

Estoy particularmente agradecido con mis asesores, quienes me brindaron enseñanzas para la investigación y para la vida. Dr. Benito de Celis, muchas gracias por su seguimiento, ayuda constante, confianza, crítica, orientación, por creer en mí, y por no soltarme en manos de la burocracia. Dr. Javier Hernández, gracias por su guía, tanto académica como administrativa. Y a ambos, gracias por ser tan comprensivos siempre.

Por supuesto, debo agradecer a mi familia por su cobijo y soporte, indispensables también para la conclusión de este proyecto. Especialmente a mi madre por su apoyo inagotable, a Ricardo y Kaori que son mi motivación, y a Itzel, con quien comparto la dicha y el reto de la paternidad.

Quiero hacer una mención especial a mis amigos Ángel y Jeny, por las noches y domingos que me acompañaron estudiando, y por permitirle este ingeniero vislumbrar cómo son las matemáticas de verdad.

Gracias a la Vicerrectoría de Investigación y Estudios de Posgrado (VIEP) y, nuevamente, al Dr. Benito de Celis por el apoyo económico para presentar el avance de este trabajo en el LXII Congreso Nacional de Física.

Y gracias al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca para realizar mis estudios de maestría.

En verdad, lo reitero, literalmente, sin el apoyo de cada uno de ustedes esto no sería posible. ¡Gracias!

# Índice General

<b>INTRODUCCIÓN</b> .....	<b>2</b>
RESUMEN.....	2
DEFINICIÓN DEL PROBLEMA .....	2
IMPORTANCIA DEL PROBLEMA .....	3
HIPÓTESIS .....	4
OBJETIVO GENERAL.....	4
OBJETIVOS ESPECÍFICOS .....	4
CONTENIDO .....	5
<b>LISTA DE ABREVIATURAS</b> .....	<b>6</b>
<b>1. MARCO TEÓRICO</b> .....	<b>8</b>
1.1 NEUROFISIOLOGÍA.....	8
1.2 NEUROANATOMÍA DEL MOVIMIENTO.....	13
1.3 EEG .....	19
1.4 APRENDIZAJE AUTOMÁTICO .....	25
1.5 INTERFAZ CEREBRO MÁQUINA.....	54
<b>2. METODOLOGÍA</b> .....	<b>56</b>
2.1 DATOS PÚBLICOS .....	56
2.2 DATOS DE LABORATORIO .....	59
2.3 ARQUITECTURAS DE BMI.....	62
<b>3. DISCUSIÓN Y RESULTADOS</b> .....	<b>66</b>
3.1 DISCUSIÓN.....	69
3.2 TIEMPO DE CÓMPUTO .....	70
3.3 EVALUACIÓN DE LOS CLASIFICADORES .....	70
3.4 COMPARACIÓN DE RESULTADOS CON EL ESTADO DEL ARTE .....	72
3.5 LIMITACIONES.....	75
3.6 TRABAJO FUTURO .....	76
3.7 CONCLUSIONES.....	76
<b>ÍNDICE DE FIGURAS</b> .....	<b>78</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>79</b>
<b>BIBLIOGRAFÍA</b> .....	<b>80</b>

# Introducción

## Resumen

Una interfaz cerebro-máquina o brain machine-interface (BMI) es una configuración que permite comunicar señales cerebrales directamente con una máquina o dispositivo. Detecta, clasifica y usa las señales cerebrales como órdenes de control. Sin embargo, los equipos y métodos que usan las BMI's aún son costosos, por lo que se necesita una opción asequible y portable. En este estudio, señales cerebrales de imaginación motora (MI por sus siglas en inglés) obtenidas por EEG, de la base de datos pública PhysioNet y de datos propios obtenidos en nuestro laboratorio usando un equipo Emotiv, fueron clasificadas con cuatro algoritmos de aprendizaje automático: CSP en conjunto con LDA, redes neuronales profundas, redes neuronales convolucionales, y MDM con geometría riemanniana. La exactitud media para cada algoritmo fue de 78%, 66%, 60% y 80% respectivamente. Los mejores resultados se obtuvieron en el caso bicategorico de Línea Base vs MI. En el caso de entrenamiento global con datos públicos, se consiguieron exactitudes de entre 86.4% y 99.9%. Y en el caso de entrenamiento global, pero datos propios, la exactitud fue mayor de 99% solo usando CSP y RMDM. El tiempo de cómputo por evento para la clasificación/predicción de datos propios fue de 8.3 ms, 18.1 ms, 62 ms, y 9.9 ms con CSP, DNN, CNN y RMDM, respectivamente, mostrando la viabilidad para usar cualquiera de tales clasificadores en una BMI de tiempo real. En conclusión, los resultados de CSP y RMDM fueron excelentes, y demuestran la plausibilidad de su implementación en una BMI con total fiabilidad con un grado de libertad y buen desempeño con dos grados de libertad.

## Definición del Problema

Identificar movimientos imaginarios en señales EEG mediante algoritmos de clasificación de aprendizaje automático.

## **Importancia del Problema**

El 15% de la población mundial presenta discapacidad médica de algún tipo, es decir, alrededor de mil millones de personas en el mundo (OMS, 2011). Como ejemplo, una de cada 20,000 personas en el mundo es diagnosticada con esclerosis lateral amiotrófica (ALS, por sus siglas en inglés). Esta es una discapacidad neuro-motriz que degenera las neuronas motoras y reduce su capacidad para comunicarse con los músculos (CONADIS, 2018). En México hay 8 millones de personas con discapacidad (INEGI, 2020). De estas, 2.6 millones presentan discapacidad motriz, de las cuales un 38.5% son debidas a alguna comorbilidad (BPOGR, 2018). La discapacidad motriz es uno de los subtipos de discapacidad que más limitan la calidad de vida de los afectados.

Una interfaz cerebro-máquina (BMI por sus siglas en inglés) es un sistema de comunicación que permite una interacción directa entre un dispositivo electrónico y la actividad cerebral. Se puede dar a través de diferentes tecnologías como EEG, MEG, ECoG, EMG, entre otras. (Clerc, Bougrain, & Lotte, 2016; Graitmann, Allison, & Pfurtscheller, 2013)

Personas con derrame o parálisis cerebral, distrofia muscular, daños en la médula espinal, con miembros amputados, y en general con cualquier discapacidad motriz, podrían rehabilitarse física o socialmente a través del uso de alguna BMI. Esto les permitiría controlar una prótesis robótica, silla de ruedas eléctrica, cursor de computadora, o sencillamente comunicarse a través de un sistema binario interpretable como respuestas de “sí” o “no”. En casos menos graves como hemiparesia, por ejemplo, incluso la correcta detección de los movimientos imaginados puede ayudarlos a mejorar su relación con el mundo. Esto a través de un sistema de refuerzo que les indique qué tan bien están imaginando la tarea. Está demostrado que la mera imaginación de la realización de sus movimientos durante entrenamientos produce una mejora significativa en el desempeño de los deportistas de alto rendimiento. (Mizuguchi, Nakata, Uchida, & Kanosue, 2012)

Actualmente, las BMI's clínicas son costosas, intrincadas y deficientes aún al discriminar correctamente las señales cerebrales, limitadas a 2 o 3 clases. Ha habido diferentes estudios que buscan mejorar la precisión en la clasificación de las señales para casos multiclase, o reducir el tiempo de cómputo para proveer aplicaciones en tiempo real. (Lotte, et al.,2018) Sin embargo, todavía hace falta trabajo para reducir el costo, incrementar la portabilidad, y la facilidad de uso para un usuario final.

Por lo tanto, es importante diseñar opciones no invasivas, de bajo costo, portátiles, y certeras al discriminar varias clases, con el fin de generar una BMI eficaz que responda con precisión y en tiempo real, ante órdenes dadas mediante la imaginación de movimientos.

### **Hipótesis**

Es posible detectar en tiempo real la imaginación del movimiento de un brazo a través de EEG del equipo Emotiv EPOC+ con una exactitud mayor al 85%, usando ML.

### **Objetivo General**

Diseñar una BMI asequible, de fácil uso, y portable, basada en ML a partir de señales EEG con una exactitud de 85%, usando algoritmos de ML.

### **Objetivos Específicos**

- Establecer la interfaz que usará la diadema Emotiv para enviar los datos a la computadora.
- Diseñar el protocolo experimental para la toma de datos.
- Tomar los datos de EEG de 30 voluntarios y descargar los datos de PhysioNet correspondientes a MI.
- Definir las clases, limpiar los datos, separar y/o etiquetar los eventos.
- Determinar los algoritmos de ML a utilizar y su arquitectura.

- Programar los algoritmos y entrenarlos con ambos conjuntos de datos, tanto en forma global como individual.
- Evaluar su exactitud obtenida y su tiempo de cómputo, tanto de entrenamiento como de clasificación.

## **Contenido**

En el primer capítulo se sientan las bases para comprender y desarrollar una interfaz cerebro-máquina. Comenzando con la comprensión básica de los componentes cerebrales y su relación con el movimiento, o su intención. Se explican los fundamentos fisiológicos y físicos del EEG, y cómo su interpretación permite pensar en una BMI. Y se da un recorrido breve por algunos de los algoritmos más utilizados en aprendizaje automático, detallando principalmente aquellos implementados en este trabajo.

En el segundo capítulo se detalla el estudio realizado, comenzando con los conjuntos de datos utilizados y cómo fueron generados. Así como los modelos de aprendizaje automático y aprendizaje profundo implementados, junto con el detalle de sus arquitecturas e hiperparámetros.

En el último capítulo se muestran y se discuten los resultados, evaluando el desempeño de los clasificadores, sus tiempos tanto de entrenamiento como de predicción, y se comparan con los resultados hallados en el estado del arte. También se reflexiona sobre las limitaciones que presenta cada clasificador y cómo podrían optimizarse en un trabajo futuro.

# Lista de abreviaturas

AI - Artificial Intelligence

ANN - Artificial Neural Network

BCI - Brain-Computer Interface

BL - Baseline

BMI - Brain Machine Interface

BP - Backpropagation

CAR - Common Average Reference

CNN - Convolutional Neural Network

CSP - Common Spatial Pattern

DL - Deep Learning

DNN - Deep Neural Network

ECoG - Electrocorticografía

EEG - Electroencefalograma/Electroencefalografía/Electroencefalógrafo

EMG - Electromiografía

EPSP - Excitatory Postsynaptic Potential

ERD - Event-Related Desynchronization

ERP - Event Related Potential

ERS - Event-Related Synchronization

fMRI - functional Magnetic Resonance Imaging

GMDH - Group Method of Data Handling

GPU - Graphics Processing Unit

kNN - k Nearest Neighbors

LDA - Linear Discriminant Analysis

LMI - Left Motor Imagery

M1 - Corteza Motora Primaria

MDM - Minimum Distance to Mean

MEG - Magnetoencefalografía

MI - Motor Imagery

ML - Machine Learning

MLP - Multilayer Perceptron

MNIST - Modified National Institute of Standards and Technology

PMA - Premotor Area

RMDM - Riemannian Minimum Distance to Mean

RMI - Right Motor Imagery

S1 - Corteza Somatosensorial

SMA - Supplementary Motor Area

SNC - Sistema Nervioso Central

SNP - Sistema Nervioso Periférico

SVM - Support Vector Machine

VEP - Visually Evoked Potential

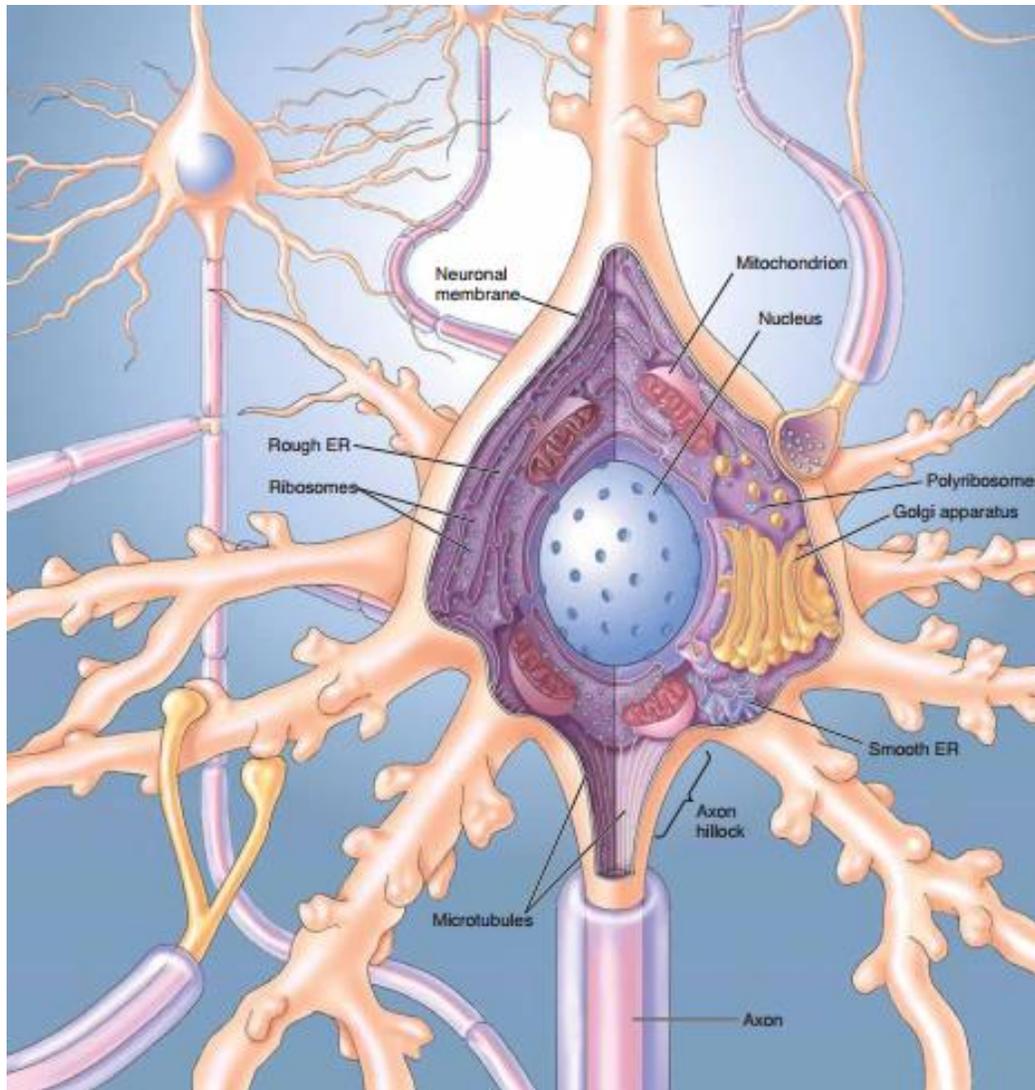
# 1. Marco Teórico

## 1.1 Neurofisiología

Para abordar el problema propuesto se requiere conocer el funcionamiento y comportamiento del cerebro. En particular, su constituyente principal: la *neurona*. Aunque además de estas, hay otro tipo de células como las gliales que dan estructura mecánica a las complicadas redes de neuronas, este trabajo se enfocará únicamente en describir a estas últimas. Y decir “únicamente” es aligerar la carga ya que en un cerebro humano hay alrededor de 100 mil millones de neuronas, esto es la cantidad de estrellas en una galaxia.

Como toda célula, la neurona posee un núcleo posicionado en el llamado cuerpo de la célula o *soma*, además de organelos, citosol, una pared celular y canales iónicos incrustados en ella que regulan el paso de elementos esenciales para su supervivencia y función. En la Figura 1.1 se representa la anatomía neuronal. Dentro de la neurona hay principalmente iones de sodio, potasio y calcio en concentraciones diferentes a las que se hallan más allá de su pared celular. Específicamente, la diferencia de potencial eléctrico que generan estos iones entre el interior y el exterior de la célula (potencial de reposo) es de aproximadamente -65 mV.

En su mayoría, los canales iónicos son dependientes del voltaje, es decir, requieren percibir cierta diferencia de potencial para que se abran o se cierren. Una vez abiertos no requieren energía para permitir el paso de los iones, pero los transportadores iónicos sí consumen ATP para transportar los iones, por lo que de no ser necesario no se activarán. De esta manera se mantiene el potencial de reposo prácticamente constante con breves fluctuaciones.



**Figura 1.1. Anatomía de una neurona.** Como toda célula, la neurona cuenta con un núcleo, presente en el soma, y organelos que le permiten llevar a cabo sus funciones. Modificada de *Neuroscience: Exploring the brain* (p. 30), por Bear et al., 2016, Wolters Kluwer.

La estructura neuronal se caracteriza por un largo tubo ramificado que se extiende desde el soma, llamado *axón*, rodeado de pequeñas protuberancias llamadas dendritas. A través de estas estructuras es que las neuronas se comunican mediante ondas electroquímicas denominadas *potenciales de acción*.

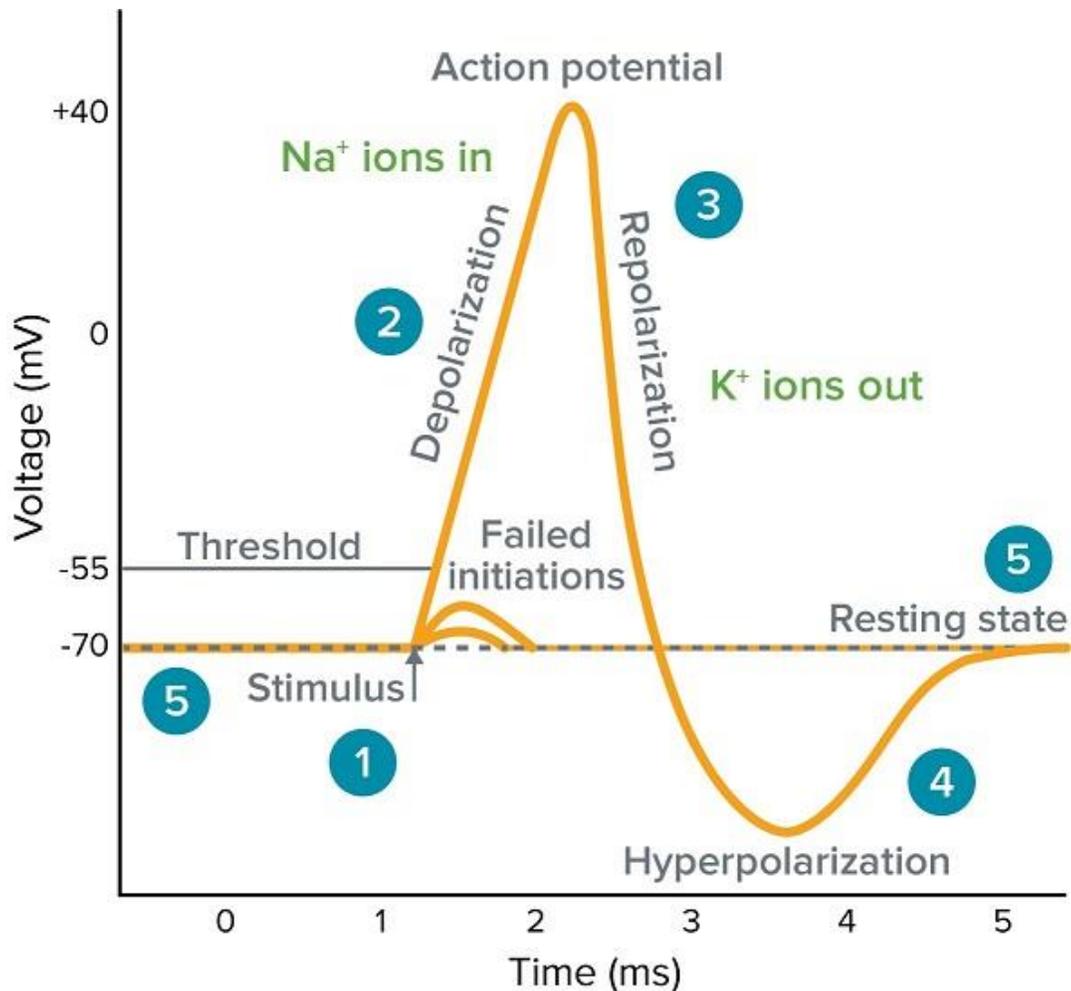
Para un detalle más escrupuloso sobre las bases neurofisiológicas y neuroanatómicas, se sugiere consultar los libros “*Neuroscience: Exploring the brain*” (2016) de Bear, Connors y

Paradiso; “Textbook of Medical Physiology” (2006) de Guyton y Hall; y “Neurociencia” (2007) de Purves et al., que fungieron de fundamento para el desarrollo de las secciones 1.1 y 1.2.

### **1.1.1 Potencial de Acción**

Un potencial de acción se da cuando el potencial membranal llega a estar por encima de -50 mV, los canales de sodio se abren violentamente y entra a la célula, incrementando aún más el voltaje hasta los +40 mV donde se abren los canales de potasio, que comienza a salir de la célula, polarizándola nuevamente más allá del potencial membranal, es decir, por debajo de los -65 mV. Al cerrarse los canales de sodio y potasio, y estar la célula hiperpolarizada, las bombas entran en acción y regulan el voltaje al potencial de descanso. Esto se muestra gráficamente en la Figura 1.2.

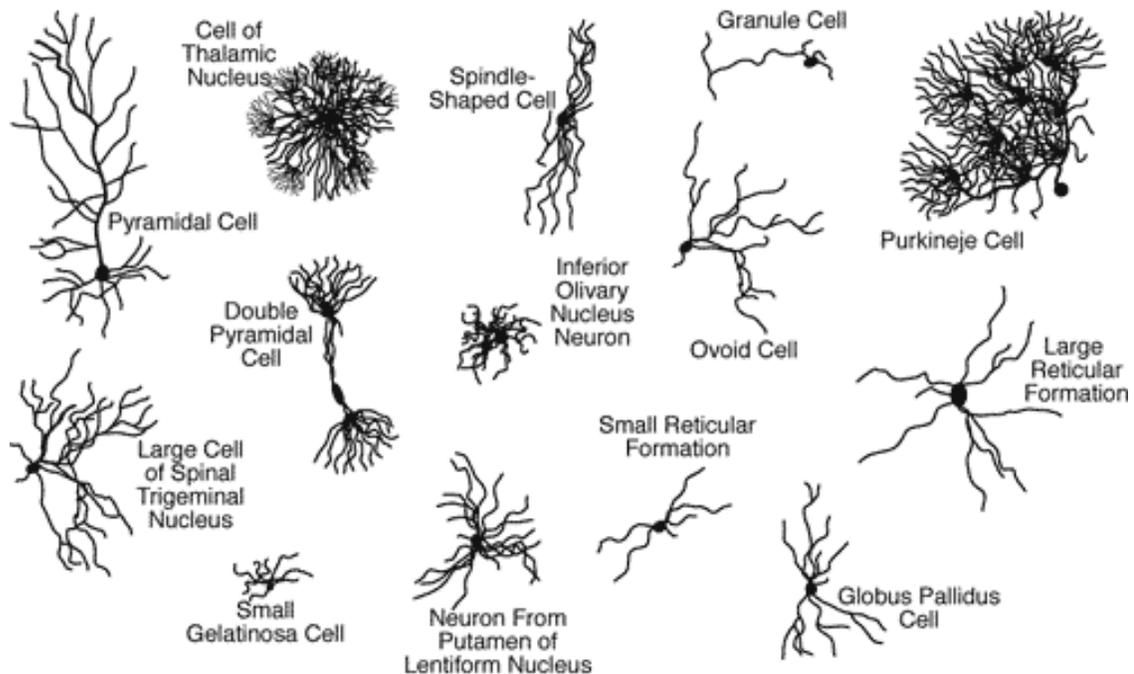
Los axones parten del soma desde una unión llamada *cono axonal* y concluyen en la *terminal axonal* donde entran en contacto con otras células, a esta región se le llama *sinapsis*, ahí transmiten información derivada principalmente del potencial de acción. A la neurona que emite el mensaje se le llama *neurona presináptica* y a la que lo recibe *neurona postsináptica*. En la sinapsis se liberan neurotransmisores que activarán o inhibirán a la neurona postsináptica. Los axones pueden medir desde unos milímetros hasta un metro. Están cubiertos con vainas de mielina que funcionan como aislantes. Estas vainas están separadas a su vez por pequeños espacios llamados *nodos de Ranvier*. Esta configuración acelera el flujo eléctrico del potencial de acción.



**Figura 1.2. Polarización durante el potencial de acción.** (1) Cuando una neurona recibe uno o varios estímulos que en suma superan su umbral de disparo, se genera un potencial de acción, (2) donde se despolariza al dejar entrar iones de sodio y (3) se repolariza dejando salir iones de potasio. (5) Por la inercia del flujo electroquímico se hiperpolariza por un instante, y (5) eventualmente alcanza su estado de reposo, lista para recibir nuevos estímulos. Reproducido de *What is an action potential*, por Molecular Devices, s.f. (<https://www.moleculardevices.com/applications/patch-clamp-electrophysiology/what-action-potential>).

### 1.1.2 Morfología Neuronal

Hay diferentes tipos de clasificación neuronal, una de ellas es por su morfología. Hay *neuronas piramidales* cuyo árbol axonal se ramifica en forma piramidal; las bipolares, que constan de 2 axones principales que se extienden hacia lados opuestos del soma como las del nervio óptico; las multipolares, que cuentan con axones radiales, como las neuronas motoras; entre otras. Algunos ejemplos se muestran en la Figura 1.3.



**Figura 1.3. Algunas morfologías neuronales.** Las neuronas toman diferentes formas y tamaños dependiendo de sus funciones. Aquí se muestran algunos ejemplos. Se pueden apreciar las neuronas piramidales y doble piramidales que se encuentran en de la corteza cerebral, entre otras regiones, y han servido de base para explicar el funcionamiento del EEG. Reproducido de *Neuron types*, por Stufflebeam, 2006, The Mindproject ([http://www.mind.ilstu.edu/curriculum/neurons\\_intro/imgs/neuron\\_types.gif](http://www.mind.ilstu.edu/curriculum/neurons_intro/imgs/neuron_types.gif)).

### 1.1.3 Neuroplasticidad

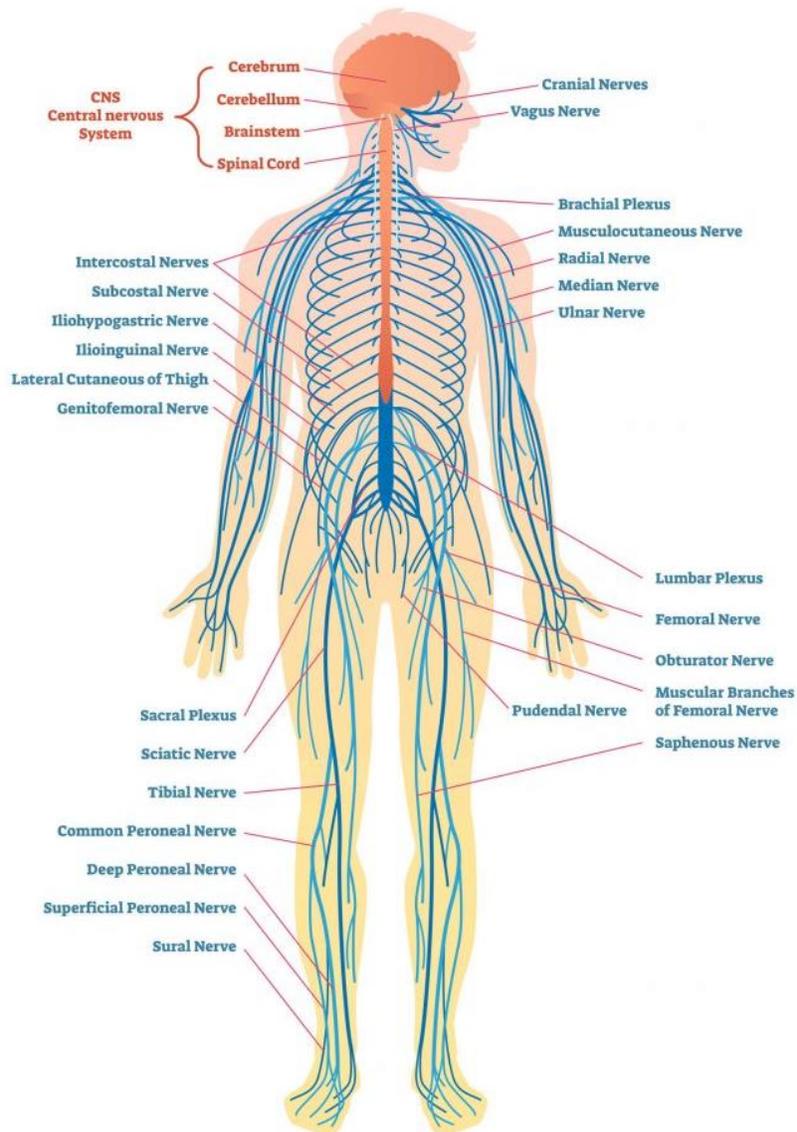
Los diferentes tipos de neuronas también distinguen la especialización de cada una. Esta diferenciación se da durante el desarrollo embrionario junto con el desarrollo axonal de cada neurona y sus conexiones sinápticas, que serán la base de las funciones cerebrales. Sin embargo, todas las redes neuronales que se generan en esta etapa se van a ir podando con la experiencia adquirida después de nacer, principalmente durante la infancia, para no desperdiciar energía y afinar la funcionalidad. La *neuroplasticidad* es la capacidad de reorganización neurosináptica, creando y eliminando axones y dendritas, a partir de estímulos internos o externos, con el fin de optimizar el funcionamiento de los circuitos neuronales especializados. (Herbert & Duffau, 2020; Stagg, 2014)

Hebbs supuso que si una neurona presináptica *A* dispara continuamente a una neurona postsináptica particular *B* de manera que *B* también genere un potencial de acción, la sinapsis entre ellas se estabiliza, haciéndose más fuerte. Por otro lado, si *B* no se activa tras la acción de *A*, pero sí tras la acción de otra de sus neuronas presinápticas *C*, entonces la sinapsis entre *A* y *B* se debilita al grado de poder llegar a perderse, manteniendo únicamente la conexión con *C*. A este tipo de sinapsis se les llama *sinapsis Hebbiana*. Y se resume en la idea de que neuronas disparando juntas se conectan juntas (Hebbs, 1949 citado por Gallistel & Matzel, 2013; Bear, Connors, & Paradiso, 2016)

## **1.2 Neuroanatomía del Movimiento**

### **1.2.1 Sistema Nervioso Central y Periférico**

Un conjunto de células compone un tejido con funciones específicas que van más allá de la suma de las funciones celulares. Las neuronas forman tejido que enerva todo el cuerpo humano en el llamado sistema nervioso. El sistema nervioso puede dividirse en 2: el sistema nervioso central (SNC) y el sistema nervioso periférico (SNP). El SNP está conformado por los nervios que reciben y envían señales a los órganos del cuerpo. El SNP somático controla los músculos que podemos manejar a voluntad, mientras que el SNP visceral se encarga del control de órganos internos, junto con los sistemas involuntarios. Por otro lado, el SNC está conformado por el cerebro, cerebelo, tallo cerebral y médula espinal. Una representación gráfica de esta división se muestra en la Figura 1.4.



**Figura 1.4. Sistema nervioso.** El sistema nervioso central, conformado por cerebro, cerebelo, tallo cerebral y médula espinal, se muestra en rojo. En azul se muestran varios elementos del sistema nervioso periférico. Adaptado de *Peripheral nervous system*, por CMT Research Foundation, s.f. (<https://cmtrf.org/cmt-101-video-cmt-the-peripheral-nervous-system/>)

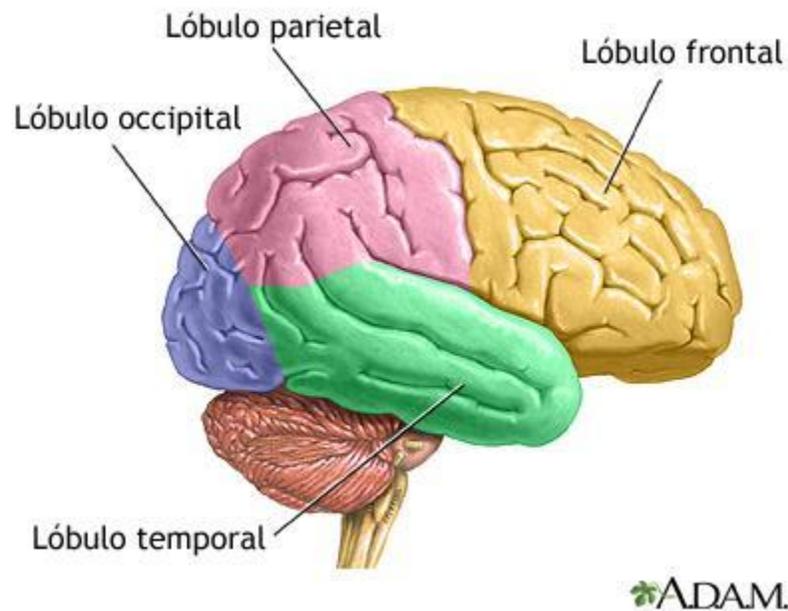
Genéricamente, se llama materia blanca a la parte central del SNC que tiene una coloración clara gracias a las cubiertas lipídicas de los axones; y se le llama materia gris a la capa más externa del SNC, una aglomeración de cuerpos neuronales.

### **1.2.2 Lóbulos Cerebrales**

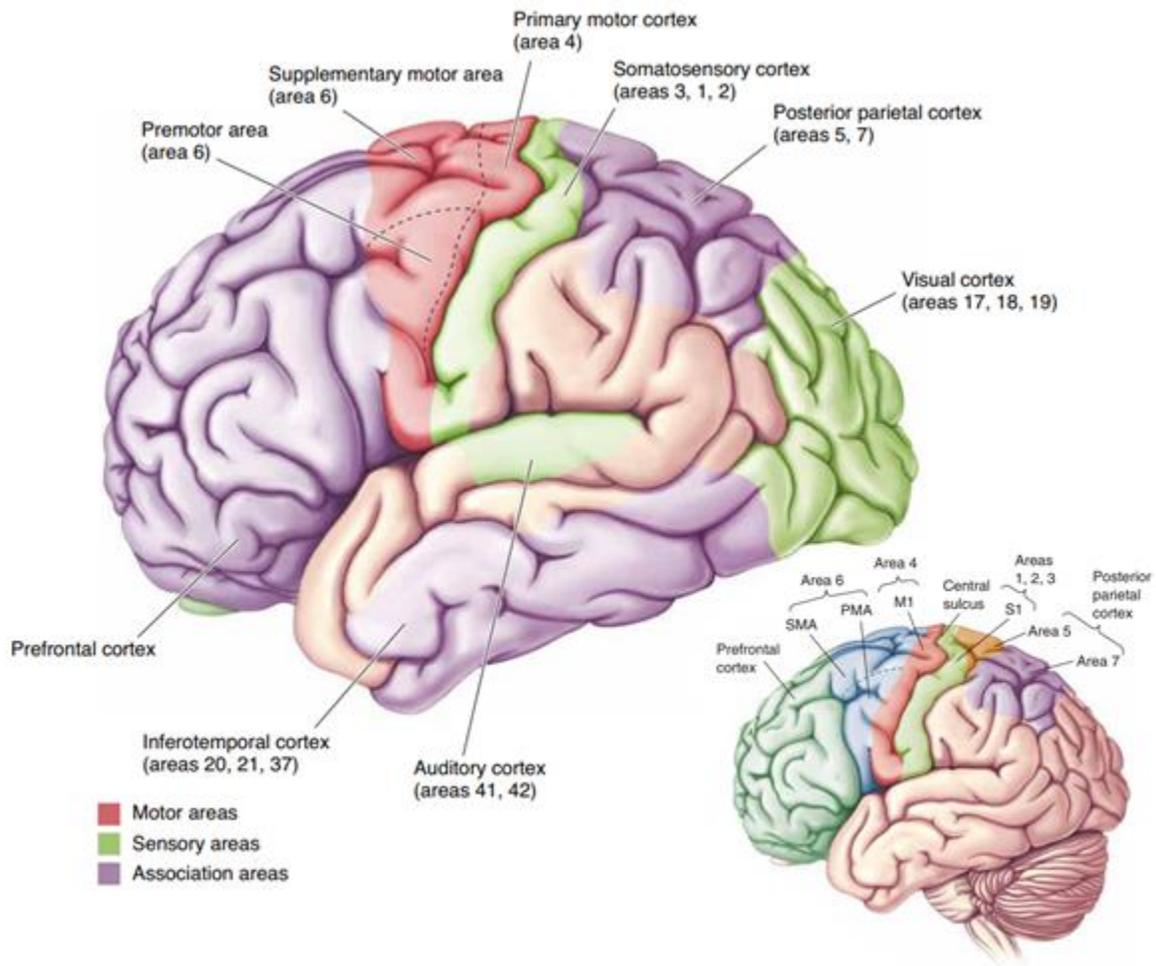
El cerebro se caracteriza por tener pliegues. Donde sobresale, se llama giro o circunvolución y en donde se hunde se denomina surco. Principalmente, el cerebro puede dividirse en sus 2 hemisferios, izquierdo y derecho, sin embargo, se usa una división basada en giros o surcos de referencia que lo separa en 4 lóbulos: frontal, parietal, temporal y occipital. Cada lóbulo tiene asociadas ciertas funciones. El lóbulo frontal se halla en la parte anterior del cerebro, delimitado por el surco central o cisura de Rolando, en él se encuentran las áreas asociadas a la voluntad, incluida la corteza motora. Inmediatamente después del surco central se encuentra el lóbulo parietal donde reside la región somatosensorial. El lóbulo temporal se encuentra caudal al surco lateral o cisura de Silvio, que lo delimita de los lóbulos frontal y parietal; aquí está la zona auditiva. Y el lóbulo occipital está en la parte posterior del cerebro, donde se encuentra la región asociada a la visión. En la Figura 1.5 se puede apreciar la división cerebral en lóbulos.

### **1.2.3 Cortezas Motora y Somatosensorial**

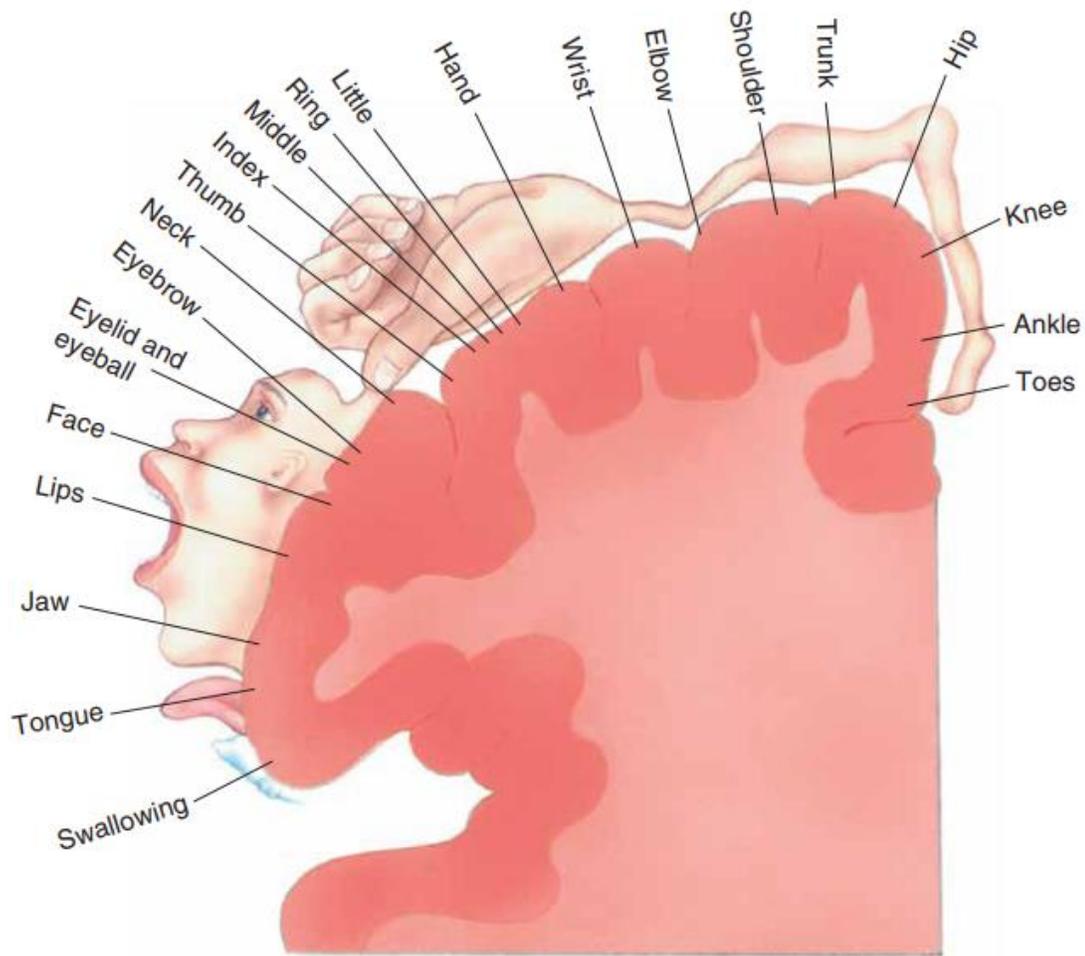
Penfield realizó electroestimulación directamente sobre las regiones anterior y posterior al surco central, y descubrió que los pacientes reportaban sensaciones o contracciones en partes específicas del cuerpo. (Penfield & Boldrey, 1937) Se concluyó que la región anterior al surco central era la encargada del movimiento, ahora conocida como la *corteza motora primaria* (M1); y la posterior era la encargada de las sensaciones, en general, relacionadas a las mismas partes del cuerpo, por lo que ahora se le denomina *corteza somatosensorial* (S1). Sus ubicaciones se pueden visualizar en la Figura 1.6. Y en la Figura 1.7 se puede ver las partes del cuerpo asociadas a cada zona de la corteza motora, como lo notó Penfield.



**Figura 1.5. Lóbulos cerebrales.** Representación gráfica de la división del cerebro en lóbulos. Reproducido de Lóbulos del cerebro, de Animated Dissection of Anatomy for Medicine (ADAM), 2018, Medline Plus ([https://medlineplus.gov/spanish/ency/esp\\_imagepages/9549.htm](https://medlineplus.gov/spanish/ency/esp_imagepages/9549.htm))



**Figura 1.6. Funciones asociadas a las regiones corticales.** Se muestran las regiones de la corteza cerebral asociadas al movimiento, en rojo y azul, y a la sensación, en verde claro. Adaptado de *Neuroscience: Exploring the brain* (pp. 224, 492), por Bear et al., 2016, Wolters Kluwer.



**Figura 1.7. Mapa somatotópico motor de M1.** Ilustración de un corte coronal del cerebro en la región precentral del hemisferio izquierdo. Se puede establecer una relación directa de control entre una zona de su superficie cortical y una parte del lado derecho del cuerpo. Modificado de *Neuroscience: Exploring the brain* (p. 493), por Bear et al., 2016, Wolters Kluwer.

De la corteza motora parte una larga lista de axones dirigidos a neuronas motoras en la médula espinal que continuarán hacia la mayor parte de los músculos del cuerpo. En el bulbo raquídeo se da la *decusación piramidal*, que es el cruce de casi todas las neuronas que parten de la corteza motora de un hemisferio hacia el lado contrario del cuerpo, por lo que el hemisferio derecho del cerebro controla el movimiento del lado izquierdo del cuerpo y el hemisferio izquierdo controla el lado derecho.

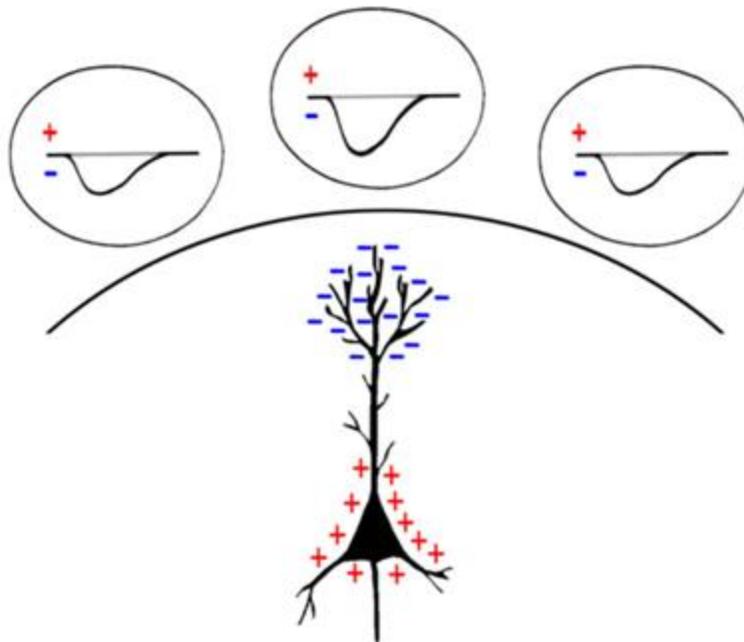
Anterior a la corteza motora, hay dos zonas más que son relevantes para el movimiento: el área premotora y el área motora suplementaria (PMA y SMA por sus siglas en inglés, respectivamente). PMA conecta neuronas reticuloespinales correspondientes a motoneuronas proximales, mientras que SMA conecta directamente con motoneuronas distales. En experimentos con monos se vio que las neuronas del SMA disparaban un segundo antes de que algún movimiento de mano o muñeca fuera realizado. También se vio que al dar una instrucción de movimiento se disparaban las neuronas en PMA, cesando después de que la acción misma hubiera iniciado (Bear, Connors, & Paradiso, 2016).

### 1.3 EEG

EEG son las siglas para referirse a *electroencefalografía*, que es una técnica no invasiva de detección de la actividad eléctrica del cerebro, que goza de alta resolución temporal, además de ser portable y de bajo costo comparado con otras técnicas de imagen, como fMRI, que ofrece una alta resolución espacial pero no temporal, no es portable y resulta bastante costosa.

También hace referencia a *electroencefalógrafo*, que es el aparato que permite realizar la técnica. Incluso puede significar *electroencefalograma*, que es el resultado gráfico que la técnica produce.

Una neurona piramidal cortical puede ser modelada como un dipolo, como se ve en la Figura 1.8, de manera que si recibe un potencial inhibitorio postsináptico (IPSP por sus siglas en inglés) cerca del soma, habrá una hiperpolarización intracelular, en consecuencia, la región extracelular que rodea al soma será más positiva, asimismo en la región intracelular apical, y por tanto, la región extracelular apical será más negativa.



**Figura 1.8. Origen del potencial eléctrico detectado por EEG.** Neurona cortical piramidal modelada como un dipolo produciendo un campo eléctrico. Las diferencias del potencial eléctrico son detectadas por electrodos en el cuero cabelludo. Adaptado de *The neurophysiological bases of EEG and EEG measurement: A review for the rest of us* (p. 1062), por Jackson & Bolger, 2014, *Psychophysiology*, 51.

En el caso de un potencial excitatorio postsináptico (EPSP) en el soma, ocurre una despolarización dentro de la membrana celular, por lo que en la zona extracelular habrá una concentración de iones negativos, en consecuencia, en las dendritas tendremos una concentración más positiva con respecto al otro extremo de la neurona.

Sin embargo, la actividad individual de una neurona no basta para ser medida en el cuero cabelludo, por lo que se requiere que haya neuronas alineadas y sincronizadas para que su actividad sea captable. (Jackson & Bolger, 2014) Hay consenso en afirmar que las señales medidas por EEG son producidas por la actividad sináptica sumada de poblaciones de  $\sim 10^8$  neuronas corticales y no tanto así por los potenciales de acción individuales. (Nunez & Srinivasan, 2005) Un grupo de neuronas piramidales alineadas y sincronizadas aún puede

modelarse como un dipolo con un extremo positivo y el otro negativo, correspondiente a la concentración iónica extracelular derivada de los potenciales de acción.

Debido a que se colocan electrodos en diferentes partes de la cabeza, es posible medir poblaciones de neuronas que en conjunto emulen tanto una configuración radial como una tangencial al cráneo, esto es, en los giros o surcos corticales respectivamente.

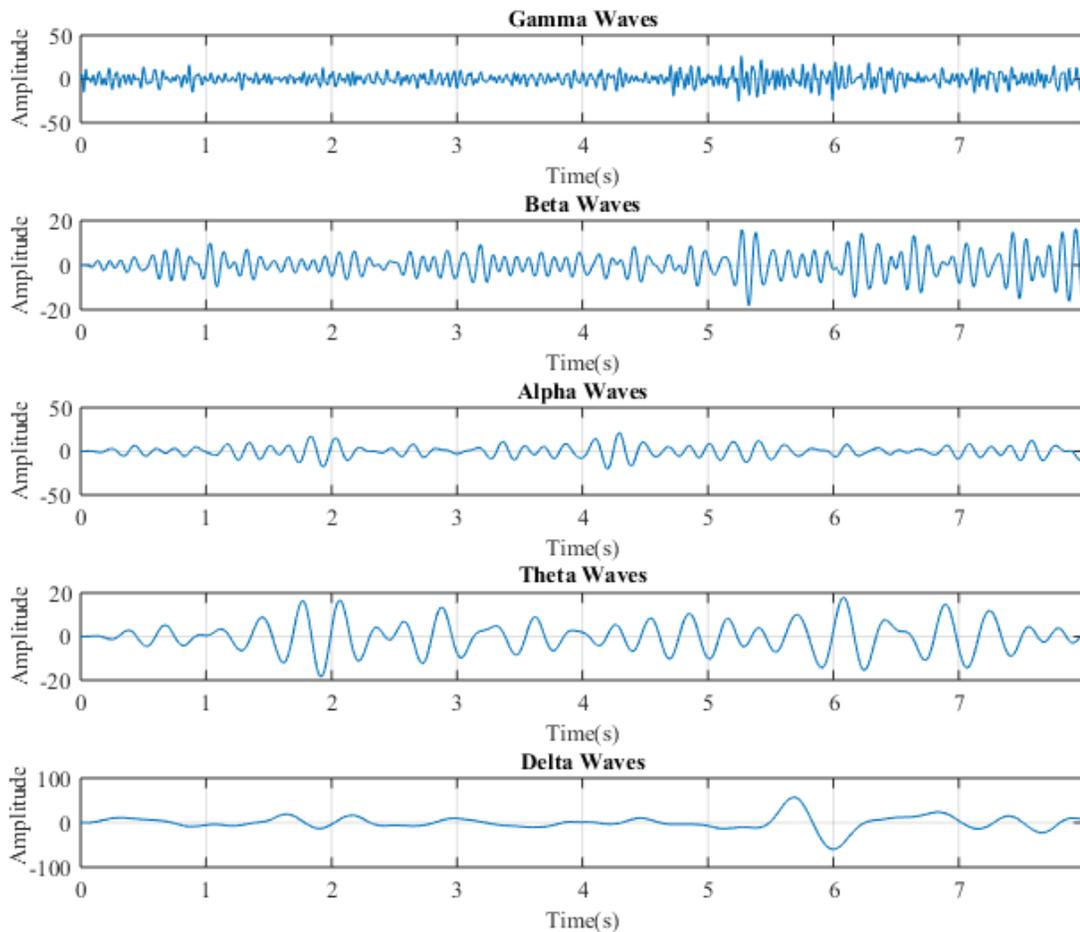
Los factores que determinan las propiedades de las oscilaciones en el EEG son: las propiedades intrínsecas de la membrana neuronal y la dinámica del proceso sináptico; la fuerza y el alcance de las interconexiones entre los elementos de la red; y las influencias moduladoras de los sistemas neurotransmisores. (Jackson & Bolger, 2014)

En el tejido biológico, al flujo de un campo eléctrico se le llama 'conducción de volumen', en el que los iones transitan por gradientes electroquímicos hasta llegar a las paredes que impiden su paso indiscriminado. Cada tipo de tejido posee una conductividad y capacitancia diferentes que definirán el comportamiento de la señal bioeléctrica. En el caso de las señales de EEG, una vez producida la corriente sincronizada de miles de neuronas vecinas, se genera un campo eléctrico que se transmite a través de la conducción de volumen hasta llegar al siguiente tejido; se forma una especie de pila de capacitores; simplificando: cerebro, fluido cerebroespinal, meninges, cráneo, cuero cabelludo, electrodo.

### **1.3.1 Ondas Cerebrales**

La principal clasificación de las ondas cerebrales se hace en función de la frecuencia. En el rango aproximado de 0.5 a 3 Hz están las ondas delta, halladas en el sueño profundo; de 4 a 7 Hz las ondas theta, que aparecen al estar durmiendo pero sin soñar; de 8 a 12 Hz las ondas alpha, se producen al parpadear o al estar en un estado de relajación, o de manera normal en niños; las ondas beta van de los 13 a los 30 Hz, se hallan normalmente en humanos adultos en

estado de alerta consciente; y finalmente las ondas gamma, de 30 Hz en adelante, no son comunes y derivan de un estado de suma actividad cerebral o incluso en personas con desórdenes mentales. En la Figura 1.9 se muestra cómo se ven las señales de EEG de acuerdo con cada banda de frecuencia mencionada.



**Figura 1.9. Ondas cerebrales normales en un adulto.** Reproducido de *Real Time EEG Based Automatic Brainwave Regulation by Music* (p. 781), por Hossan & Chowdhury, 2016, 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV).

Cierto tipo de eventos generan cambios en la actividad de las poblaciones neuronales en una determinada ventana de tiempo, y son llamados 'potenciales relacionados a eventos' o ERP (Event-Related Potential por sus siglas en inglés). Los ERP representan cambios específicos de

frecuencia, si incrementa la potencia en la banda de frecuencia dada puede considerarse que hubo un incremento en la sincronía de las poblaciones de neuronas subyacentes, y se denomina 'sincronización relacionada al evento' o ERS (Event-Related Synchronization por sus siglas en inglés); por otro lado, si disminuye la potencia, se asume que hubo un decremento en la sincronía, generando una 'desincronización relacionada al evento' o ERD (Event-Related Desynchronization por sus siglas en inglés). Además, se sabe que, en general, si la amplitud de una fluctuación disminuye, su frecuencia se incrementará. Se recomienda tener periodos de espera de 10 segundos entre cada acción para que los ritmos vuelvan a su estado neutral. Y se ha visto que los movimientos voluntarios provocan ERD en las bandas alpha superior y beta inferior, localizadas cerca de las áreas sensoriomotoras, que comienza aproximadamente 2 segundos antes del comienzo del movimiento, se vuelve bilateralmente simétrica inmediatamente antes de su ejecución, y muestra ERS un segundo después.

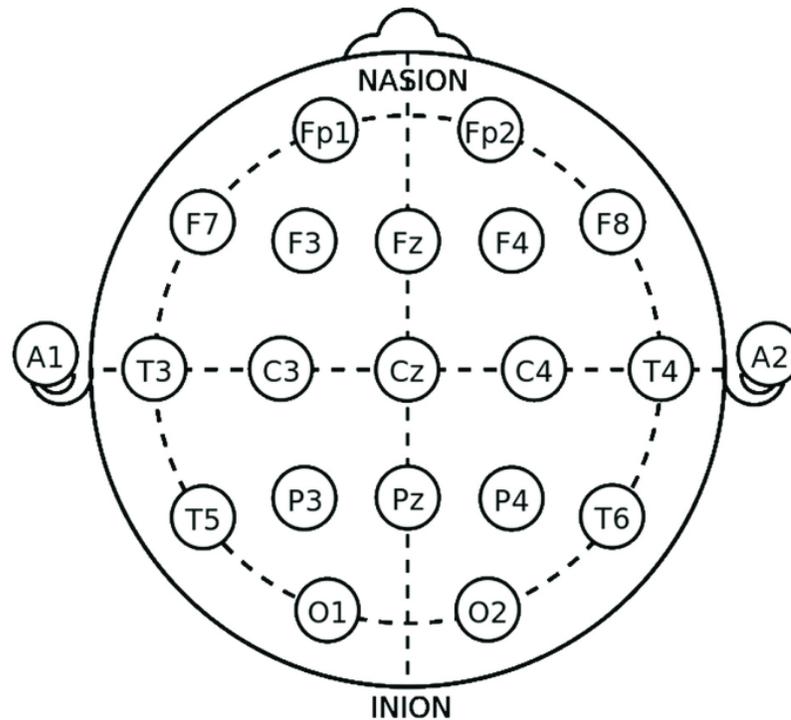
### **1.3.2 Artefactos**

Hay 2 tipos de ruido que pueden alterar la señal detectada por el EEG: internos y externos. El externo más importante es la red eléctrica local, con voltaje de 120V a 60Hz que, afortunadamente, su campo eléctrico decae con el cuadrado de la distancia. En cuanto a ruidos internos tenemos las señales eléctricas cardíaca y muscular, y los ERP derivados del movimiento o estímulos visuales, auditivos, olfativos, etc.

### **1.3.3 Sistema Internacional 10-20**

El sistema 10-20 o *protocolo de colocación de electrodos 10-20* es el estándar internacional para colocar los electrodos en la cabeza de un humano para EEG. Barre el cráneo desde el nasión hasta el inión, y de un punto preauricular al otro, colocando el primer y último electrodo a una distancia igual al 10% de la distancia total entre estos dos puntos, y los subsecuentes en espacios de 20% de la distancia total. La nomenclatura de cada electrodo

corresponde al lóbulo sobre el que se encuentra y su separación al plano sagital comenzando en el centro e incrementando en números pares hacia el lado derecho e impares hacia el izquierdo. Los electrodos posibles, su posición y nomenclatura se muestran en la Figura 1.10.



**Figura 1.10. Sistema internacional 10-20.** Reproducido de Study of resting-state functional connectivity networks using EEG electrodes position as seed (p. 2), por Rojas et al., 2018, 12 (235).

### 1.3.4 Imaginación Motora

Por ejemplo, un *Potencial Visual Evocado* o *VEP* (Visually Evoked Potential por sus siglas en inglés) podría ser percibido principalmente en los electrodos O1 y O2 en la región occipital donde está la corteza visual. Del mismo modo, un movimiento del brazo izquierdo sería observar en C4 donde está la corteza motora, particularmente el área que controla el lado izquierdo del cuerpo. De hecho, el pensamiento de un movimiento de las extremidades produce un tipo

especial de ondas, llamadas el ritmo mu, alrededor de 8-12 Hz en la región central. Este ERP se llama *imaginación motora* o *motor imagery* (MI).

## **1.4 Aprendizaje Automático**

### **1.4.1 Inteligencia Artificial**

Alan Turing propuso la pregunta “¿Pueden pensar las máquinas?” (Turing, 1950, pág. 433) Sin embargo, él mismo da un paso atrás al tener que definir los conceptos *pensar* y *máquina*. Y desarrolla lo que hoy se conoce como el Test de Turing, una prueba en la que un humano *A* debe determinar si está interactuando con otro humano o con una computadora. Esto bajo la premisa de que el pensamiento se deduce del lenguaje natural utilizado. Se infiere que, si el humano *A* es incapaz de diferenciar la naturaleza de su interactor, se dice que la computadora pasó el test, o bien, que la computadora piensa.

La perspectiva diametral, no antagónica, la ofrecen primero McCulloch y Pitts (McCulloch & Pitts, 1943) donde presentan a la red de neuronas como pequeños operadores lógicos interconectados que reciben una entrada, la operan y producen una respuesta. En ese momento se sabía relativamente poco sobre la complejidad de la fisiología cerebral pero ya se había descubierto y aceptado que la neurona es la unidad funcional fundamental del sistema nervioso, principalmente por el trabajo de Santiago Ramón y Cajal. Se basaron en la premisa de la acción “todo o nada” de cada neurona, heredada del concepto del potencial de acción. En su modelo cada sinapsis sería una entrada a un operador. De manera semejante a una neurona multi-postsináptica, un operador podría recibir varias entradas que evaluaría para determinar si genera o no una salida. Esto produce principalmente 2 ideas: matemáticamente, que cualquier proposición lógica puede traducirse a una red de operadores lógicos (equivalente a una máquina de Turing); y filosóficamente, que el cerebro funciona como una computadora.

Con lo anterior presente se desarrolló todo un nuevo campo de estudio denominado *Inteligencia Artificial* o *AI* por sus siglas en inglés. Término acuñado por John McCarthy en 1955 previo a la Conferencia Dartmouth, donde se sentaron las bases de tal campo, junto con Marvin Minsky, Nathaniel Rochester y Claude Shannon (McCorduck, 2004). Según él, “la meta de la AI es desarrollar máquinas que se comporten como si fueran inteligentes” (McCarthy, citado por Ertel, 2017). Pues buscaban generar algoritmos y sistemas que, de alguna manera, expusieran algún indicio de intelecto inherente a los humanos. Desde demostraciones matemáticas, procesamiento de imágenes, sistemas expertos, procesamiento de lenguaje natural, reconocimiento vocal, aprendizaje automático, o incluso juegos, entre otras (Russel & Norvig, 2004). Eventualmente, se presentaron otras definiciones para el concepto, sin embargo, debido a que cada una se inclinaba hacia alguna de las ramas mencionadas no satisfacían la generalidad. La definición aceptada aquí es la de Elaine Rich, que no presenta el problema de la especificidad y le permite prevalecer en la actualidad:

“La Inteligencia Artificial es el estudio de cómo hacer que las computadoras realicen cosas que, hasta el momento, las personas hacemos mejor” (Knight, Rich, & Nair, 2010, pág. 3).

Como la misma autora menciona, sacrifica la precisión a cambio de proveer perspectiva para discutir el tema. Pero es gracias a su holgura que permite abarcar la mayoría de los enfoques habidos y en ciernes, incluyendo las tareas que los humanos tampoco podemos realizar con efectividad.

#### **1.4.2 Machine Learning**

Consideremos el aprendizaje como actividad humana. Representa una tarea intrínseca para la población sana que, desde la gestación, el cerebro va ligando y cortando conexiones neuronales que nos permiten retener información no solo de estímulos externos sino también internos, por ejemplo, al mejorar la coordinación motora o desarrollar el control de esfínteres.

Tras los trabajos de McCulloch y Pitts se continuó la exploración de arquitecturas computacionales que emularan la actividad cerebral, entre ellas el aprendizaje. Cabe mencionar que el objetivo de tales empresas no era, en sí, hacer una máquina pensante, sino reproducir las bases fisiológicas que permitieran la total comprensión de una mente generada por estructuras físicas.

El *aprendizaje máquina*, *aprendizaje automático* o *machine learning* (ML) es una técnica de modelado computacional, derivada como rama de la inteligencia artificial (AI) que hereda principios estadísticos, en la que un algoritmo recibe datos externos y genera un modelo generalizado para esos datos, ya sea de predicción, clasificación o construcción. Los datos que recibe pueden ser imágenes, audio, archivos de texto, o virtualmente cualquier tipo de dato. Cada ejemplo corresponde a un *evento*. Generalmente, son vectorizados de manera que la máquina pueda entrenarse con operaciones computacionales iterativas basadas principalmente en álgebra lineal.

¿Pero qué significa que una máquina aprenda? La definición más aceptada y usada para esto es la de Tom Mitchell: “Se dice que un programa computacional aprende de la experiencia  $E$  con respecto a alguna clase de tareas  $T$  y medida de desempeño  $P$ , si su desempeño en las tareas  $T$ , medido por  $P$ , mejora con la experiencia  $E$ .” (Mitchell, 1997, pág. 2) Tal definición resulta suficientemente general para abarcar cualquier tarea e incluir cualquier experiencia, ya sea  $T$  reconocimiento facial, procesamiento de lenguaje natural o jugar ajedrez; donde  $E$  podría ser recibir ejemplos de rostros en imágenes, fonemas en audios o archivos de texto de partidas de ajedrez, respectivamente.

**1.4.2.1 Predicción.** En el caso de la predicción se busca ajustar una curva a los datos para que el modelo sea capaz de decir en qué lugar del espacio estará el siguiente punto o evento. La tarea es predecir, la experiencia es el conjunto de eventos de entrenamiento, y la

medida de desempeño común es la distancia a la que se encuentra el punto predicho del punto correcto.

**1.4.2.2 Clasificación.** Con respecto a la tarea de clasificación, se tiene un conjunto de datos que se desea ordenar o separar de acuerdo con alguna regla semántica determinada. Para ejemplificar, considérese nuestro problema de clasificación de MI a partir de señales de EEG. Sea  $U = \{D_1, \dots, D_S\}$  el conjunto de datos de EEG de todos los voluntarios  $S$ , y  $D_i = \{(X_1, y_1), \dots, (X_{N_i}, y_{N_i})\}$  el conjunto de pares de eventos de MI  $X_j$  y su etiqueta correspondiente  $y_j$ , donde  $N_i$  es el número total de eventos del sujeto  $i$ , y  $\forall X_j \in \mathbb{R}^{ExT} \exists y_j \in \mathcal{L} | \mathcal{L} = \{\ell_1 = \text{"MI izquierda"}, \ell_2 = \text{"MI derecha"}, \ell_3 = \text{"Descanso"}\}$ , donde  $E$  es el número de electrodos del EEG,  $T$  es el número de muestras por evento, y  $\mathcal{L}$  es el conjunto de etiquetas, correspondiente a las posibles clases a las que puede pertenecer un evento dado.

Entonces, el clasificador  $g$  es entrenado a través de los eventos de ejemplo etiquetados, para producir  $y_j$  al recibir  $X_j$ . Puede verse como una función  $g(X_j, \theta): \mathbb{R}^{ExT} \rightarrow \mathcal{L}$ . El entrenamiento corresponde a afinar el parámetro  $\theta$ , el cómo dependerá de cada algoritmo, y se detallará más adelante.

Puede darse un preprocesamiento de los datos y/o una extracción de características que transforme los eventos conservando aquellas características físicas o estadísticas que sean más relevantes para la clasificación, como la densidad del espectro de potencia (PSD por sus siglas en inglés) o la transformada de Fourier, entre otras, en materia de señales de EEG. En tal caso se tendría además una función  $f(X_j, \varphi): \mathbb{R}^{ExT} \rightarrow \mathbb{R}^F$ , donde  $F$  es la dimensión del espacio de características, es decir, el número de características conservadas. Y finalmente, el clasificador lo recibiría como entrada:  $g(f(X_j, \varphi), \theta)$ .

**1.4.2.3 El Perceptrón.** Marvin Minsky desarrolló “la primera gran implementación una red neuronal” (Skansi, 2018, pág. 6) artificial física con 3000 tubos de vacío que tenía 40 neuronas.

Buscaba hallar una teoría sobre la mente a partir de la simulación del cerebro, pero no tuvo éxito alguno. (Ertel, 2017; McCorduck, 2004). Posteriormente, Frank Rosenblatt presentó la idea del *perceptrón*, basado en la hipótesis de las sinapsis Hebbianas. Criticando la propuesta determinista de McColloug y Pitts, Rosenblatt buscaba integrar cualidades probabilísticas a su representación considerando que la plasticidad cerebral produce patrones axodendríticos distintos en cada persona y que hay tanto neuronas inhibitorias como excitatorias. Incluyó también la idea de la sinapsis hebbiana. Estableció 3 secciones para el perceptrón: un sistema receptivo, uno asociativo y uno responsivo. Y lo dotó de una regla de aprendizaje básica pero fundamental para el desarrollo ulterior del aprendizaje automático. (Rosenblatt, The Perceptron. A Perceiving and Recognizing Automaton, 1957; Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain., 1958)

En términos modernos, un perceptrón se ve así:

$$P(\mathbf{x}) = \begin{cases} 1, & \boldsymbol{\omega}^T \cdot \mathbf{x} - u > 0 \\ 0, & \text{en otro caso} \end{cases}$$

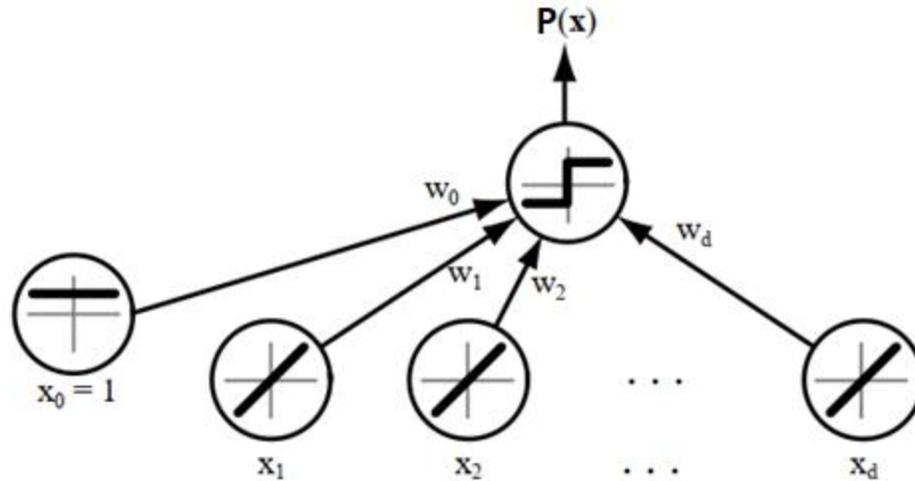
donde

$\boldsymbol{\omega} \in \mathbf{R}^n$  es un vector de parámetros llamados *pesos* que fungen como el sistema asociativo;

$\mathbf{x} \in \mathbf{R}^n$  es el vector de datos de entrada, es el sistema receptivo;

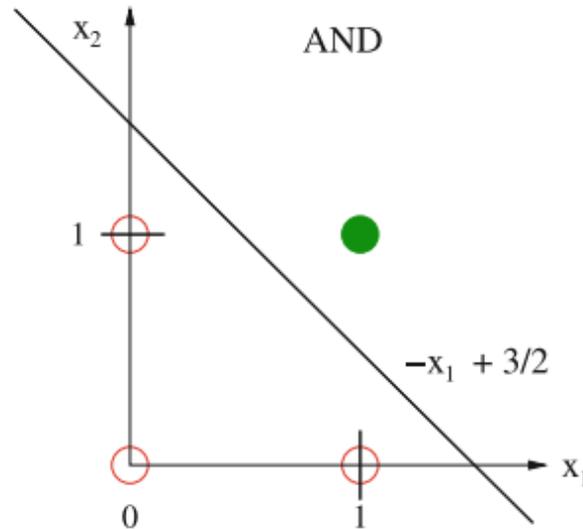
$u \in \mathbf{R}^n$  es un umbral que dota de sensibilidad a la neurona;

y la función  $P(\mathbf{x})$ , llamada *función de activación*, Heaviside en este caso, es el sistema responsivo. La Figura 1.11 ilustra cómo opera un perceptrón.



**Figura 1.11. Perceptrón.** El producto de las entradas  $x$  y los pesos  $w$  se hacen pasar por una función de activación que clasificará la entrada como positiva o negativa. Modificado de *Pattern Classification* (p. 216), por Duda et al., 2001, John Wiley & Sons.

**1.4.2.4 Máquina Lineal.** En general, hasta este punto, el perceptrón es un *clasificador o discriminante lineal*, donde  $\omega^T \cdot \mathbf{x} - u = 0$  ó  $g(\mathbf{x}) = \omega^T \cdot \mathbf{x} = \omega_1 x_1 + \dots + \omega_n x_n = u$  es la ecuación del hiperplano que separa el *espacio de características* que distinguen a uno de otro tipo de datos. Al tipo de dato se le llama *clase* y a los datos que alimentan el clasificador se les llama *características*. Algunos autores añaden un sesgo  $b = \omega_0$  -del inglés *bias*- al vector de pesos para elevar la tendencia hacia alguna clase, aunque es común tomar  $u = -b$  de modo que  $g(\mathbf{x}) = 0$ . Lo anterior se extiende fácilmente al caso de más de 2 clases. Simplemente se calcula  $g_i(\mathbf{x})$   $c$  veces, donde  $c$  es el número de clases, y se comparan. Si  $g_i(\mathbf{x}) > g_j(\mathbf{x}) \forall j \neq i$  entonces se elige la clase  $i$ . A tal clasificador se le llama *máquina lineal* (Duda, Stork, & Hart, 2001; Kubat, 2017). En la Figura 1.12 se muestra cómo una máquina lineal basta para resolver una compuerta lógica AND.



**Figura 1.12. La linealidad de la compuerta lógica AND.** Una línea es capaz de separar el espacio de características por lo que una máquina lineal sería capaz de clasificar las salidas de la compuerta AND. El punto verde representa Verdadero y los círculos rojos Falso. Modificado de *Introduction to Artificial Intelligence* (p. 184), por Ertel, 2017, Springer.

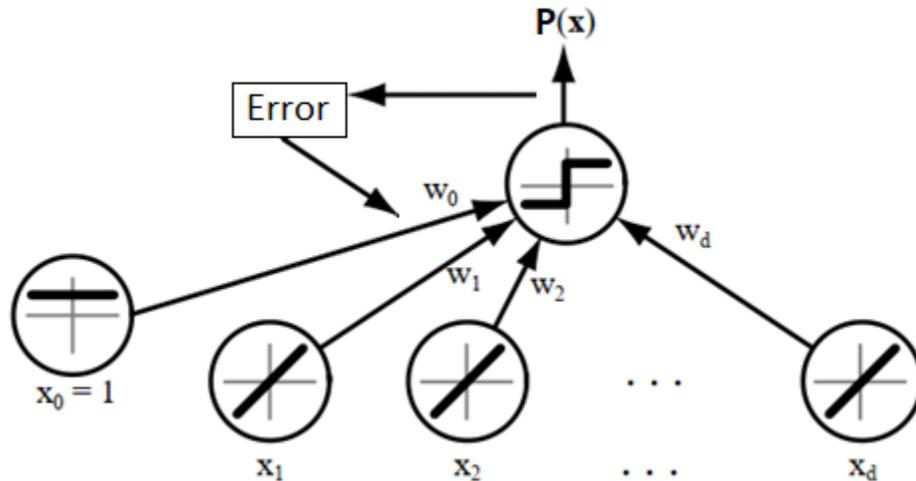
**1.4.2.5 Regla de Aprendizaje.** Una máquina lineal simplemente clasifica de acuerdo con los pesos determinados por  $w$ , pero el perceptrón puede aprender modificando los pesos mediante una regla de aprendizaje. Esta regla se basa en la premisa de que se conoce a priori la clase a la que pertenece cada uno de los datos de entrenamiento. Se dice que los datos están *etiquetados* si se tiene una tupla de la forma  $(x, y) = (\text{características}, \text{etiqueta})$  donde la *etiqueta* representa la clase a la que pertenece  $x$ , y al aprendizaje llevado a cabo con datos etiquetados se le llama *aprendizaje supervisado*. Entonces, si la etiqueta  $y$  y la predicción  $\hat{y}$  no coinciden se modificará  $\omega$  de la siguiente manera:

$$\text{Si } g(x) > 0 \rightarrow P(x) = \hat{y} = 1 \text{ y } y = 0, \text{ entonces } \omega = \omega + x.$$

$$\text{Si } g(x) \leq 0 \rightarrow P(x) = \hat{y} = 0 \text{ y } y = 1, \text{ entonces } \omega = \omega - x.$$

Y se repite tantas veces como sea necesario para que el perceptrón aprenda, es decir, produzca la salida esperada. A cada repetición se le llama *época*. La Figura 1.13 ofrece una

representación gráfica de la retroalimentación en un perceptrón. Es claro que converge pues en el primer caso la nueva  $g(x) = (\omega + x)^T \cdot x = \omega^T \cdot x + x^2$  será cada vez más positiva; y en el segundo caso  $g(x) = (\omega - x)^T \cdot x = \omega^T \cdot x - x^2$  será cada vez más negativa (Ertel, 2017).



**Figura 1.13. Regla de aprendizaje del perceptrón.** El resultado de  $P(x)$  se compara con la etiqueta del evento evaluado y se modifica el vector de pesos de acuerdo con lo requerido, repitiendo el ciclo hasta que los  $w$  tenga los valores necesarios para que la predicción sea igual al valor de la etiqueta. Modificado de *Pattern Classification* (p. 216), por Duda et al., 2001, John Wiley & Sons.

**1.4.2.6 El Problema de los Sistemas No Lineales.** Resultó interesante percatarse de que el perceptrón era capaz de identificar imágenes diferenciando hombres de mujeres, pero no de resolver correctamente algo relativamente básico como la compuerta lógica XOR cuya tabla de verdad se corresponde con la Tabla 1.1. En la Figura 1.14 se puede ver la no linealidad de XOR.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 1.1 Tabla de verdad XOR.

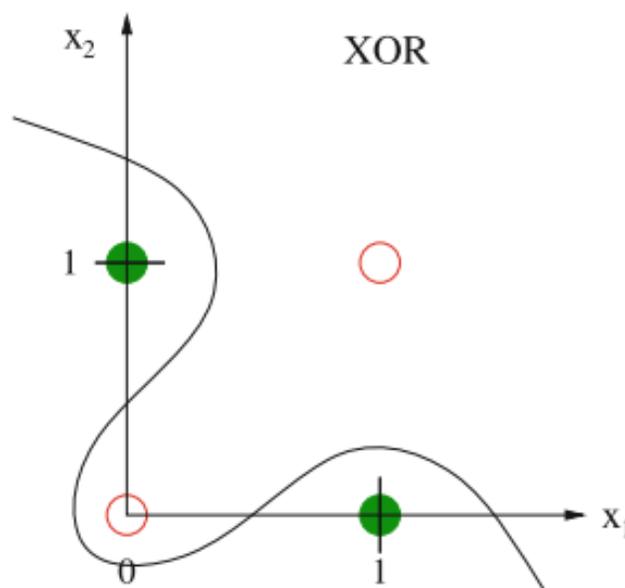


Figura 1.14. **La no linealidad de la compuerta lógica XOR.** El perceptrón no es capaz de clasificar las salidas de la compuerta XOR. Los puntos verdes representan *Verdadero* y los círculos rojos *Falso*. Modificado de *Introduction to Artificial Intelligence* (p. 184), por Ertel, 2017, Springer.

Rosenblatt mismo exploró la idea de acoplar perceptrones en varias capas para expandir sus capacidades, pero no halló la manera de aplicar la regla de aprendizaje en tal arquitectura. (Rosenblatt, 1961, citado por Skansi, 2018). Eventualmente, Minsky y Papert demostraron que el perceptrón solo funciona para clasificar clases linealmente separables. (Minsky & Papert, 1969)

**1.4.2.7 Tipos de aprendizaje.** Ya se mencionó lo que es el aprendizaje supervisado, donde se proveen las etiquetas para indicarle al modelo cuál es el comportamiento deseado. Hay ocasiones en que solamente se cuenta con el conjunto de datos sin etiquetas, ¿es posible generar un modelo a partir de estos? Sí es posible, mediante algoritmos de *agrupamiento* o *clustering*, que buscan relaciones entre los datos y son capaces de clasificarlos automáticamente. Como se esperaría, este tipo de aprendizaje se denomina *aprendizaje no supervisado*. Y hay una tercer categoría llamada *aprendizaje por refuerzo*, en la que se premia al modelo cuando se comporta como se desea y se le penaliza cuando no. No se comparan literalmente las etiquetas, sino que se ofrece una evaluación ponderada ante la respuesta generada, y tras cada iteración el algoritmo busca maximizar su puntaje. Por otro lado, el ML no sirve únicamente para clasificar, también puede utilizarse para predecir un comportamiento a través de algoritmos de regresión.

A continuación, se menciona brevemente algunos de los principales algoritmos de ML.

**1.4.2.8 Regresión Lineal.** Es un tipo de aprendizaje supervisado en que el modelo es una función lineal cuya recta se ajusta lo más posible a los datos de entrenamiento. Esto se logra minimizando una *función de pérdida*  $J$ , comúnmente se utiliza la función de error cuadrático medio (MSE por sus siglas en inglés):

$$J = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2, \quad f(\mathbf{x}) = \omega_0 + \sum_{j=1}^d \omega_j x_j$$

donde  $N$  es el número de datos de entrenamiento,

$y_i$  es la etiqueta del dato  $x_i$ ,

$d$  es la dimensión del vector de características  $\mathbf{x}$ ,

y  $\boldsymbol{\omega}$  es el vector de pesos. (Skiena, 2017)

**1.4.2.8.1 Gradiente Descendente.** Para minimizar la función de pérdida se busca el mínimo global de la función. En el caso de MSE, al ser cuadrática positiva, geométricamente representa una parábola, o hiperparaboloide en el caso general, con un único punto crítico que corresponde al mínimo global. Pero podría usarse cualquier otra función de pérdida que no necesariamente tuviera un único punto crítico, podría tener varios mínimos locales o puntos de silla. El algoritmo de gradiente descendente (GD por sus siglas en inglés), propuesto por primera vez por Cauchy (1867 citado por Tschöpe, 2019), aborda este problema. Su regla de actualización está dada por

$$\omega(t + 1) = \omega(t) - \eta \nabla_{\omega(t)} J$$

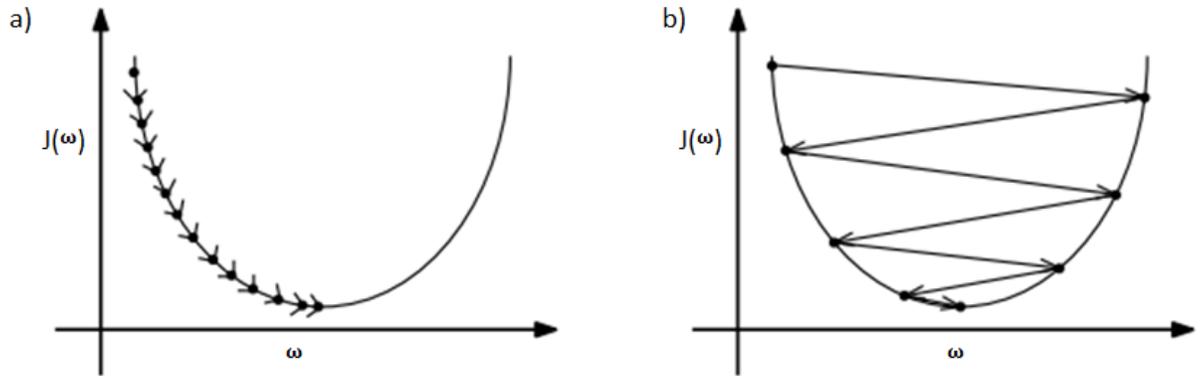
donde  $t$  es la época,

$\eta \in (0,1]$  es un hiperparámetro denominado *tasa de aprendizaje*, y

$\nabla_{\omega(t)} J$  es el gradiente de  $J$  con respecto a  $\omega(t)$ .

De modo que, tras cada época, la función  $J$  se irá acercando más a su mínimo global y al mismo tiempo los pesos  $\omega$  se irán optimizando. Los *hiperparámetros* son cantidades inmersas en distintas etapas de los modelos de ML que modifican su comportamiento.

**1.4.2.8.2 Tasa de Aprendizaje.** Vale la pena discutir el papel de la tasa de aprendizaje.  $\eta$  es un valor entre 0 y 1 que determina qué tanto se va a desplazar  $\omega(t + 1)$  en dirección contraria al gradiente. Si  $\eta$  es cercano a 1 dará pasos largos por lo que es posible que alcance más rápido el mínimo o que termine muy lejos para cierta época. Por otro lado, si es muy pequeño dará pasos muy cortos con la posibilidad de que tarde demasiado o no acabe por llegar al mínimo. Esto se ilustra en la Figura 1.15.



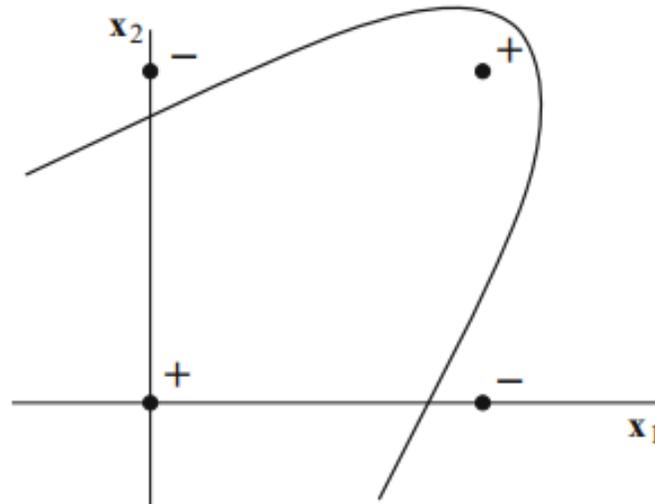
**Figura 1.15. Efecto de la tasa de aprendizaje.** Comparación del efecto en la velocidad con una a) tasa de aprendizaje pequeña contra una b) tasa de aprendizaje grande. Modificado *The Data Science Design Manual* (p. 284), por Skiena, 2017, Springer.

Una vez que se tienen los pesos óptimos, el modelo está listo para predecir, es entonces capaz de recibir datos totalmente ajenos a los datos de entrenamiento y emitir una respuesta coherente. A tal capacidad se le llama *generalización*. En términos de Mitchell, el algoritmo ha aprendido a seguir la tendencia (T) de los datos de entrenamiento (E) a través de la minimización del error cuadrático medio (P).

**1.4.2.9 Regresión Polinomial.** En la regresión lineal se busca ajustar una recta para predecir valores. Tal recta puede fungir como un clasificador lineal también. A pesar de que la regresión polinomial permite asimismo predecir valores de una distribución dada que se apegue a la función polinomial generada, comúnmente se usa para tareas de clasificación. Con la ventaja, precisamente, de que pueden clasificar conjuntos no linealmente separables también. La diferencia con la regresión lineal es la forma de la función a ajustar, que será del grado  $p$  que se desee,

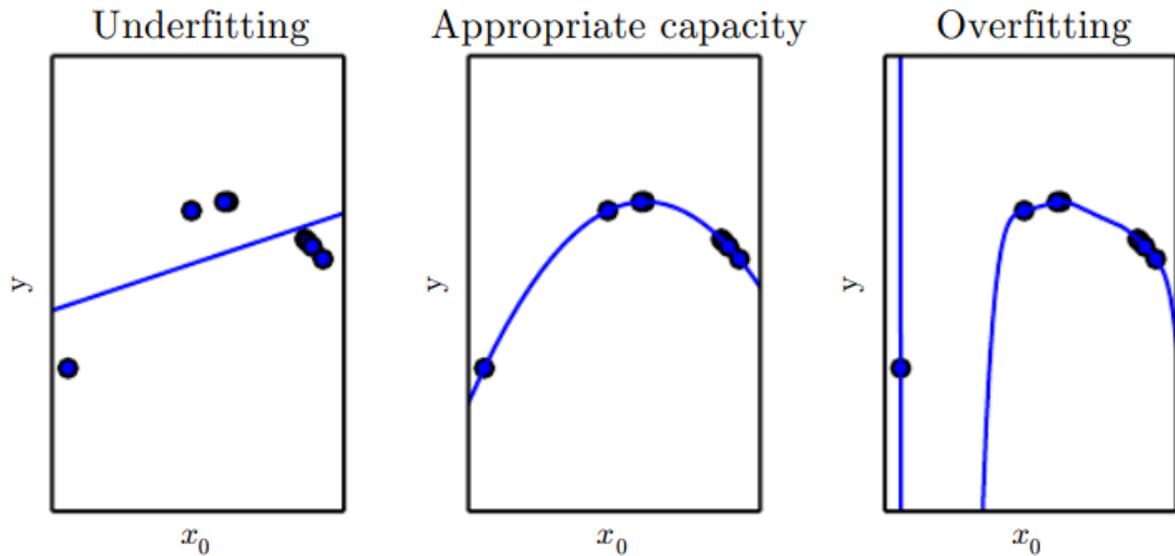
$$f(x) = \omega_0 + \sum_{i=1}^p \omega_i x^i,$$

pero desde la perspectiva de los pesos  $\omega$ , sigue siendo una función lineal que se puede optimizar de la misma manera que en el caso con  $p = 1$ , pues todos los valores de entrenamiento  $x$  son conocidos. En la Figura 1.16 se muestra cómo la regresión polinomial es capaz de resolver el problema XOR. (Kubat, 2017)



**Figura 1.16. Clasificación con regresión polinomial.** Un polinomio de segundo grado es suficiente para proponer una solución al problema XOR. '+' representa Verdadero y '-' representa Falso. Reproducido de An Introduction to Machine Learning (p.79), por Kubat, 2017, Springer.

**1.4.2.9.1 Sobreajuste y Subajuste.** Aunque la regresión polinomial trajo una respuesta para las dificultades en casos no linealmente separables del perceptrón, vino con un problema nuevo: el *sobreajuste*. Si un modelo presenta un error de entrenamiento bajo, es decir, predice correctamente la mayor parte de las etiquetas de los datos de entrenamiento, pero tiene un error de prueba alto, o bien, predice incorrectamente una cantidad significativa de datos de prueba, ajenos a los de entrenamiento, cuyas etiquetas se conocen a priori, se dice que el modelo está *sobreajustado*. En contraste un modelo *subajustado* es aquel que no es capaz de obtener un error pequeño ni para los datos de entrenamiento (Goodfellow, Bengio, & Courville, 2016). En la Figura 1.17 se ilustra un ejemplo de sobreajuste y subajuste en un modelo simple de regresión.

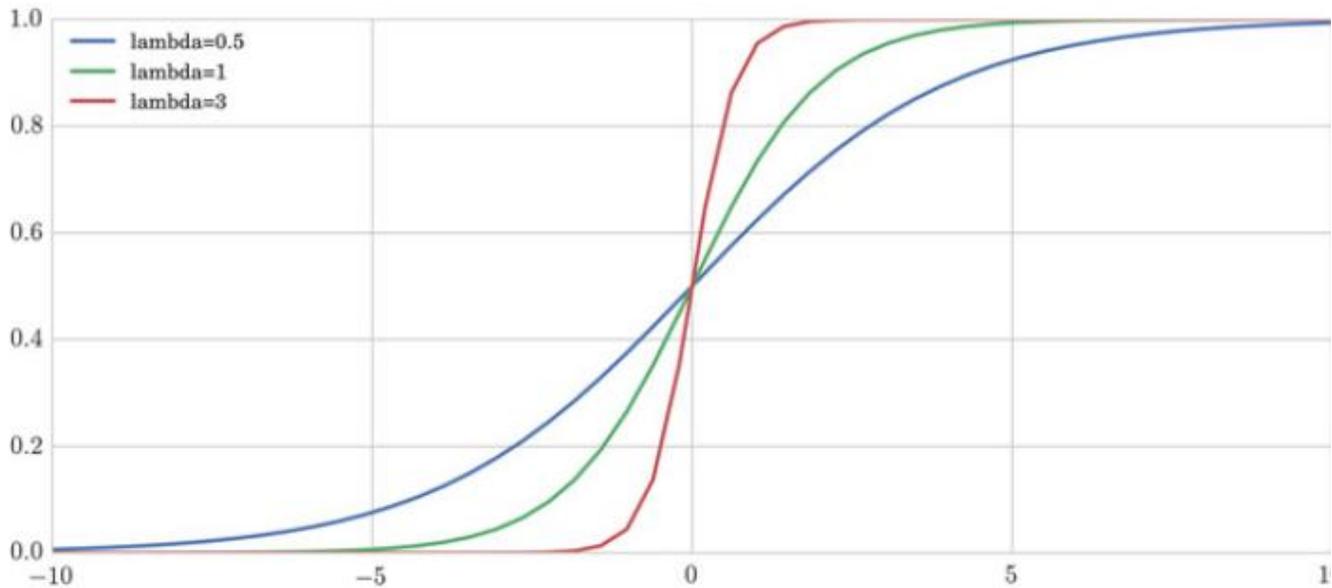


**Figura 1.17. Sobreajuste y subajuste.** El subajuste (izquierda) representa un modelo holgado que no separa con resolución el espacio, por lo que pierde precisión en su clasificación. Por otro lado, el sobreajuste (derecha) representa un modelo excesivamente preciso, que clasifica perfectamente todos los eventos de entrenamiento, sin embargo, pierde capacidad para generalizar y clasificar nuevos datos. Un ajuste correcto (centro) sacrifica la total exactitud a cambio de la generalización. Reproducido de *Deep Learning* (p. 111), por Goodfellow et al., 2016, The MIT Press.

**1.4.2.10 Regresión Logística.** Ya se ha mencionado que el aprendizaje automático heredó mucho de la estadística y la regresión logística es un ejemplo directo. Se usa para clasificación probabilística de eventos binarios. Debe su nombre a la *función logística*, también llamada *sigmoide* por la forma de su gráfica:

$$\sigma(x) = \frac{1}{1 + e^{-\lambda x}}$$

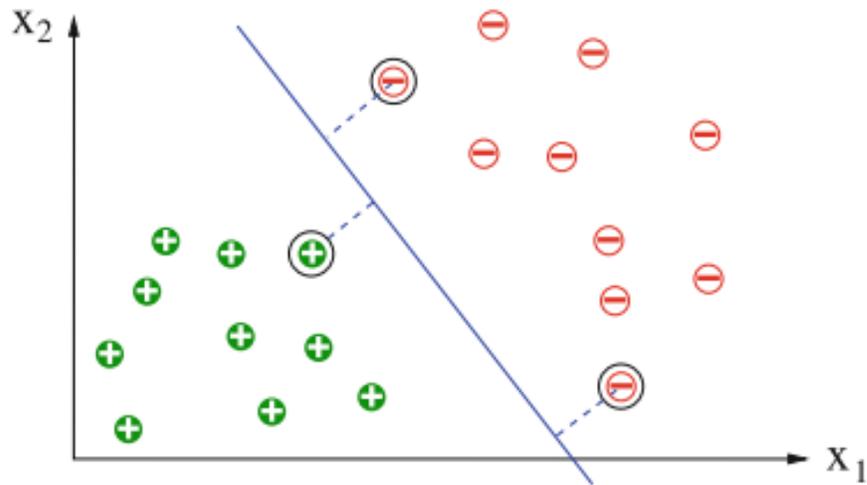
cuyo dominio está en los reales, pero su imagen está en el intervalo (0,1) (Igual & Seguí, 2017). En la Figura 1.18 se muestra la gráfica de la función sigmoide.



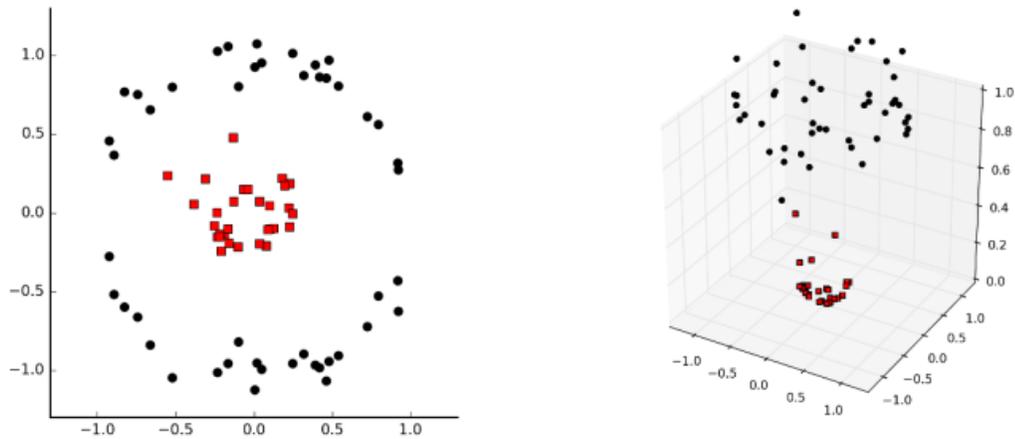
**Figura 1.18. Función sigmoide.** Gráficas de la función sigmoide para diferentes valores de  $\lambda$ . Reproducido de Introduction to Data Science (p. 111), por Igual & Seguí, 2017, Springer.

**1.4.2.11 Máquina de Soporte Vectorial (SVM).** Las máquinas de soporte vectorial son clasificadores lineales que producen hiperplanos que dividen el espacio de características acorde a las clases presentadas, por lo que es un algoritmo de aprendizaje supervisado. Se busca hallar el hiperplano que esté lo más alejado posible de ambas clases. Para ello se utilizan unos cuantos puntos de la zona fronteriza llamados vectores de soporte, Skiena sugiere no más del doble del número de dimensiones del espacio de características (Skiena, 2017). En la Figura 1.19 se ilustra el funcionamiento de una SVM.

Otra ventaja de SVM es que se ha desarrollado una manera de clasificar datos no linealmente separables. Esto a través del uso de funciones kernel que mapean los datos a una dimensión superior donde sean linealmente separables, como se muestra en la Figura 1.20.



**Figura 1.19. Clasificación con SVM.** La línea azul es la generada por el clasificador SVM y los puntos circunscritos son los vectores de soporte. Reproducido de *Introduction to Artificial Intelligence* (p. 276), por Ertel, 2017, Springer.



**Figura 1.20. Función kernel para hacer datos linealmente separables.** A la izquierda están los datos originales, en un espacio de 2 dimensiones. Es claro que las clases roja y negra no son linealmente separables. Al aplicar la función kernel de transformación  $(x,y) \rightarrow (x,y,x^2+y^2)$  se obtiene la gráfica de la derecha. Ahora los datos son linealmente separables en el nuevo espacio de 3 dimensiones. Fuente: "The Data Science Design Manual" (Skiena, 2017)

**1.4.2.12 Patrón Espacial Común (CSP).** Common Spatial Pattern es un algoritmo para obtener filtros espaciales de las matrices de covarianza de un conjunto de datos dado. Es

considerado un algoritmo de extracción de características debido a que necesita un decisor que haga la clasificación. (Olías, 2016)

La idea de CSP es simple, dadas 2 clases, hallar vectores que maximicen la varianza de una clase mientras minimizan la varianza de la otra, simultáneamente. Esto se logra resolviendo el problema de optimización

$$\max_w \left( \frac{w^T C_1 w}{w^T C_2 w} \right)$$

equivalente a resolver el problema de autovalores

$$C_1 w = \lambda C_2 w$$

Donde  $C_i$  es la matriz de covarianza de la clase  $i$  de la matriz de  $N \times T$  que contiene los datos de la señal de EEG, con  $N$  igual al número de electrodos y  $T$  al número de muestras (o de pasos temporales); los autovectores  $W$  son los filtros espaciales. Es común tomar 6 filtros, los 3 primeros y los 3 últimos. A  $W$  también se le conoce como la matriz demezcladora bajo la idea de que los patrones eléctricos corticales originales  $Z$  pasan por un filtro (meninges, cráneo, cuero cabelludo) de modo que la salida es una señal filtrada  $E = (W^{-1})^T Z$ , por lo que  $WE = W(W^{-1})^T Z = Z$ .

Para el caso multiclase se han propuesto el uso de una supermatriz de covarianza tomando la suma de  $c-1$  matrices de covarianza, donde  $c$  es el número de clases, y aplicar el algoritmo bajo el paradigma 'uno contra el resto'. (Blankertz, Tomioka, Lemm, Kawanabe, & Muller, 2008)

$$C_i' = \sum_j C_j, j \neq i$$

Ya que el dominio de la frecuencia de la señal no es tomado en cuenta por este algoritmo, se sugiere primero filtrar la señal conservando solo la banda de frecuencia requerida. En el caso de MI, se relaciona principalmente con los ritmos mu, entre 8 y 12 Hz, sin embargo, hay variaciones perceptibles también en las bandas alfa y beta, alrededor de los 8 a los 30 Hz, (Stock & Balbinot, 2016; Fakhruzzaman, Riksakomara, & Suryotrisongko, 2015; Elstob & Secco, 2016; Mohammadpour, Ghorbanian, & Mozaffari, 2016) así que generalmente también son consideradas, por ejemplo en (Blankertz, Tomioka, Lemm, Kawanabe, & Muller, 2008; Mahmood, Zainab, Ahmad, Saeed, & Kamboh, 2017; Acuña, 2017).

**1.4.2.13 Análisis de Discriminante Lineal (LDA).** Proyecta los datos en el hiperplano que maximice la distancia entre las medias de cada clase y minimice la varianza. En el caso de la biblioteca scikit learn, lo hace maximizando la distancia de Mahalanobis, partiendo de la regla de Bayes,

$$P(y = k|x) = \frac{P(x|y=k)P(y=k)}{P(x)},$$

$P(x|y)$  se modela como una distribución normal,

$$P(x|y = k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma_k^{-1}(x - \mu_k)\right).$$

Tomando el logaritmo de  $P(y = k|x)$ , se obtiene

$$\log P(y = k|x) = -\frac{1}{2}(x - \mu_k)^T \Sigma^{-1}(x - \mu_k) + \log P(y = k) + C,$$

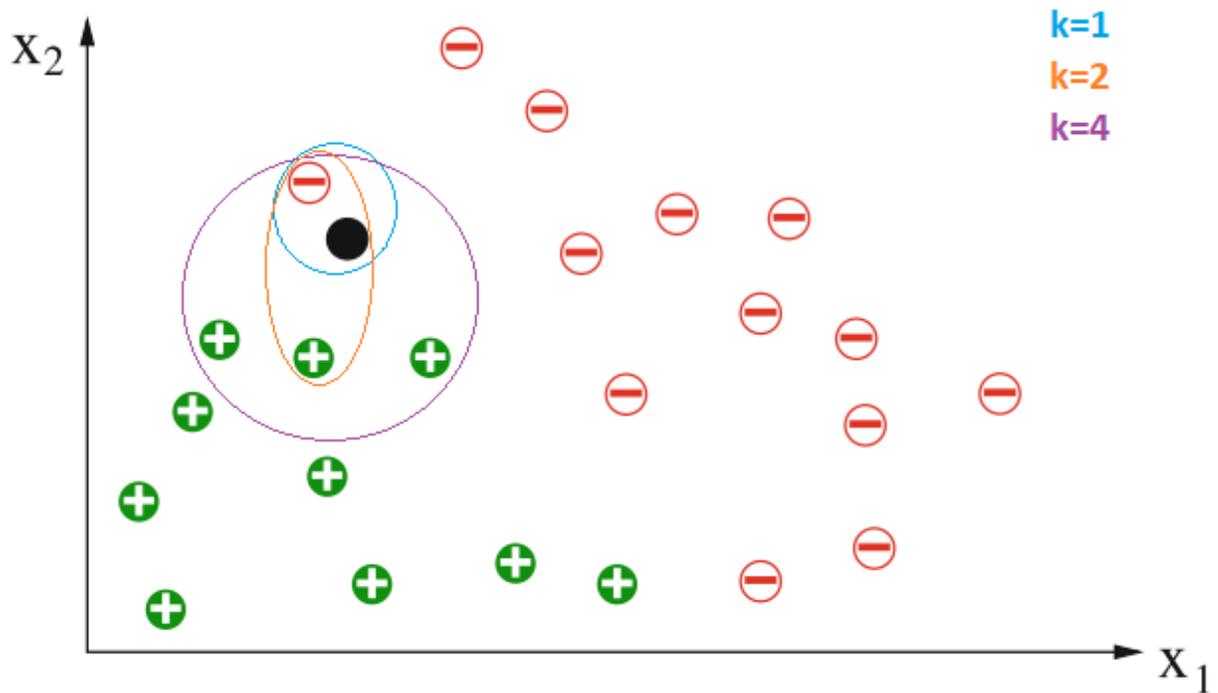
donde  $C$  es una constante.

Puede reescribirse como

$$\log P(y = k|x) = \omega_k^T x + \omega_{k0} + C,$$

que representa la ecuación del hiperplano que separa las clases.

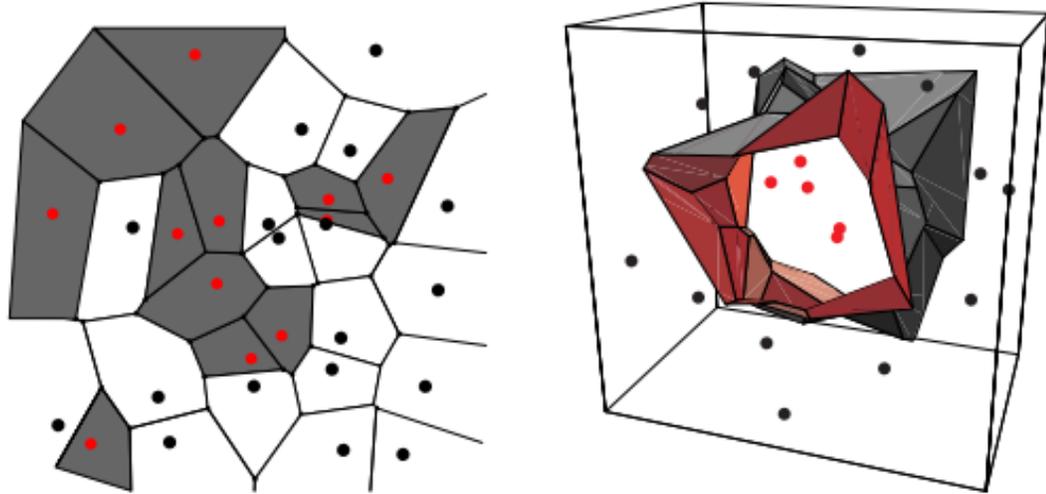
**1.4.2.14 K Vecinos Más Cercanos (kNN).** Método de aprendizaje supervisado basado en métricas. Separa el espacio en grupos o *clusters* de  $k$  datos, donde  $k \in \mathbb{N}$  es un hiperparámetro que determina el tamaño de los grupos. Requiere datos etiquetados pues asigna a cada dato nuevo la clase correspondiente a la de sus  $k$  vecinos más cercanos, como se ilustra en la Figura 1.21. Es un algoritmo sencillo de ejecutar, ya que no requiere entrenamiento, pero costoso, pues calcula la distancia entre el dato a clasificar y cada uno del resto de los datos para hallar los  $k$  más cercanos. (Ertel, 2017)



**Figura 1.21. Clasificación kNN para distintas  $k$ .** La correcta clasificación depende del valor de  $k$ , de la dimensión del espacio de características y del número de datos de entrenamiento. En este ejemplo se tienen 2 clases, la positiva en verde y la negativa en roja. Si  $k=1$ , el nuevo punto negro se cataloga negativo. Si  $k=2$ , el algoritmo no es capaz de decidir. Si  $k=4$ , se clasifica positivo. Modificado de *Introduction to Artificial Intelligence* (p. 190), por Ertel, 2017, Springer.

**1.4.2.14.1 Teselaciones de Voronoi.** La clasificación por kNN puede ser optimizada si en el espacio de características se generan celdas que rodeen los vecindarios de tamaño  $k$ . Luego solo es necesario ver en qué celda cayó el dato a clasificar y asignarle la clase

correspondiente a esa celda, como se muestra en la Figura 1.22. (Duda, Stork, & Hart, 2001; Ertel, 2017; Kubat, 2017).



**Figura 1.22. Teselaciones de Voronoi.** Se muestra el diagrama de Voronoi para un conjunto de datos bimodal, rojo y negro, con espacio de características en 2 (izquierda) y 3 (derecha) dimensiones. Modificado de *Pattern Classification* (p. 178), por Duda et al., 2001, John Wiley & Sons.

**1.4.2.15 K Medias.** Es un tipo de aprendizaje no supervisado que separa el espacio de características en  $k$  regiones de clasificación. Primero, se eligen arbitrariamente  $k$  puntos en el espacio. Se comparan las distancias entre cierto dato y los puntos elegidos, luego ese dato es clasificado en el grupo del punto hacia donde presente una menor distancia. Una vez que todos los datos fueron clasificados, se calcula el centroide de cada grupo y se establecen como los nuevos puntos de clasificación. Esto se realiza de manera iterativa hasta que las clases asignadas a cada punto no cambien (Igual & Seguí, 2017).

**1.4.2.16 Mínima Distancia a la Media (MDM).** Algoritmo de aprendizaje supervisado que compara eventos en un espacio euclidiano mediante la distancia entre ellos. La distancia euclidiana es definida como  $\delta_E(a, b) = |a - b|$ , donde  $a$  y  $b$  son puntos en el espacio. Considérese un caso simple con 2 clases: la clase ‘descanso’ con etiqueta 0 y la clase C con la

etiqueta  $C$ . Sea  $\sigma_0$  la varianza media de los eventos etiquetados 0,  $\sigma_C$  la varianza media de los eventos etiquetados  $C$ ,  $\sigma_k$  la varianza del evento  $k$ , y  $\delta(\cdot, \cdot)$  la función distancia.

El clasificador MDM simplemente comparará si  $\delta(\sigma_C, \sigma_k) > \delta(\sigma_0, \sigma_k) \Rightarrow k$  pertenece a la clase  $C$ .

Si se diera el caso contrario, entonces  $k$  es asignada a la clase 0.

**1.4.2.17 Mínima Distancia a la Media Riemanniana (RMDM).** Desafortunadamente, la distancia euclidiana no funciona bien cuando se trata de EEG como puntos en el espacio. Se ha tratado de implementar en su lugar la distancia log-euclidiana, también llamada hiperbólica o geométrica:

$$\delta_G(a, b) = |\log a - \log b|,$$

que goza de invarianza ante escalas ya que

$$\delta_G(xa, xb) = \delta_G(a, b), \quad \forall a, b, x > 0$$

e invarianza ante la inversión,

$$\delta_G(a^{-1}, b^{-1}) = \delta_G(a, b), \quad \forall a, b > 0$$

Pero también es necesario tener una media definida para usar MDM. Si  $\{c_1, \dots, c_k\}$  es el conjunto de  $k$  puntos, su media geométrica estará dada por

$$g = \sqrt[k]{c_1 \cdot c_2 \cdot \dots \cdot c_k}$$

Ahora, si se considera el caso de señales EEG con  $N=2$ , donde  $N$  es el número de electrodos o canales en el EEG, con  $x_{1k}$  y  $x_{2k}$  como sus respectivas series de tiempo en una ventana de tiempo  $k$ , su matriz de covarianza  $C_k$  puede representarse como un punto en un espacio tridimensional cuyos ejes sean  $\text{Var}(x_1)$ ,  $\text{Var}(x_2)$  y  $\text{Cov}(x_1, x_2) = \text{Cov}(x_2, x_1)$ . Dado que  $C_k$  es una matriz SPD, y considerando la desigualdad de Cauchy-Schwarz, los puntos están

contenidos en un cono alrededor del plano  $\text{Var}(x_1)\text{Var}(x_2)$ . Recuérdese que la varianza de una señal de EEG es equivalente a su energía, y la covarianza entre 2 electrodos puede traducirse como la sincronización entre las señales que miden, así que resulta una buena caracterización.

El conjunto  $S_{++}$  de matrices positivas de  $N \times N$  es una variedad diferenciable en la cual cada pequeña vecindad alrededor de algún punto  $P$  se ve como el espacio euclidiano de matrices simétricas. El espacio tangente es el de todas las matrices simétricas en cada punto base de la variedad. Y cada espacio tangente en la geometría de Riemann cuenta con un producto interno en el punto  $P$ ,  $\langle A, B \rangle_P = \text{tr}(P^{-1/2}AP^{-1/2}B)$ .

A la curva que representa la distancia más corta entre 2 puntos  $C_1$  y  $C_2$  en el espacio  $S_{++}(N)$  se le llama geodésica, o también, distancia riemannianna, dada por

$$\delta_G(C_1, C_2) = \|\text{Log}(C_1^{-1/2}C_2C_1^{-1/2})\|_F$$

donde  $\|\cdot\|_F$  es la norma de Frobenius. Goza de invarianza ante conjugaciones,

$$\delta_G(XC_1X^T, XC_2X^T) = \delta_G(C_1, C_2),$$

e invarianza ante inversiones,

$$\delta(C_1^{-1}, C_2^{-1}) = \delta(C_1, C_2).$$

Y, nuevamente, se requiere una media o centro de masa, que en este caso está dada por la media de Cartan, sin embargo, para  $k > 2$  no se conoce una forma cerrada definida, pero puede estimarse hallando la solución a la ecuación

$$\frac{1}{k} \sum_k \text{Log}(G^{-1/2}C_kG^{-1/2}) = 0$$

Finalmente, se compara la distancia riemanniana entre un evento de prueba y la media de Cartan de cada clase, y entonces el evento es asignado a la case más cercana. (Congedo,

Barachant, & Bhatia, 2017; Barachant, Bonnet, Congedo, & Jutten, Riemannian geometry applied to BCI classification, 2010; Barachant, Bonnet, Congedo, & Jutten, Multiclass Brain-Computer Interface Classification by Riemannian Geometry, 2012)

Recuérdese que los filtros espaciales parten de la idea de que la serie de tiempo de una señal de EEG esta dada por

$$x(t) = As(t),$$

Donde  $s(t)$  es el vector que contiene la serie de tiempo de la fuente original, esto es, las regiones corticales cerebrales, y  $A$  es la matriz mezcladora de la cual, suponiendo que sea invertible, se obtiene la matriz  $B$  de  $P \times N$ .  $B$  funge como filtro espacial, de manera que

$$y(t) = Bx(t) = s(t)$$

Si  $S_i$  y  $S_j$  son las matrices de covarianza de la fuente original de 2 eventos  $i$  y  $j$ , las matrices de covarianza de la señal obtenida por un electrodo son  $C_i = AS_iA^T$  y  $C_j = AS_jA^T$ , por lo que

$$\delta_G(S_i, S_j) = \delta_G(C_i, C_j).$$

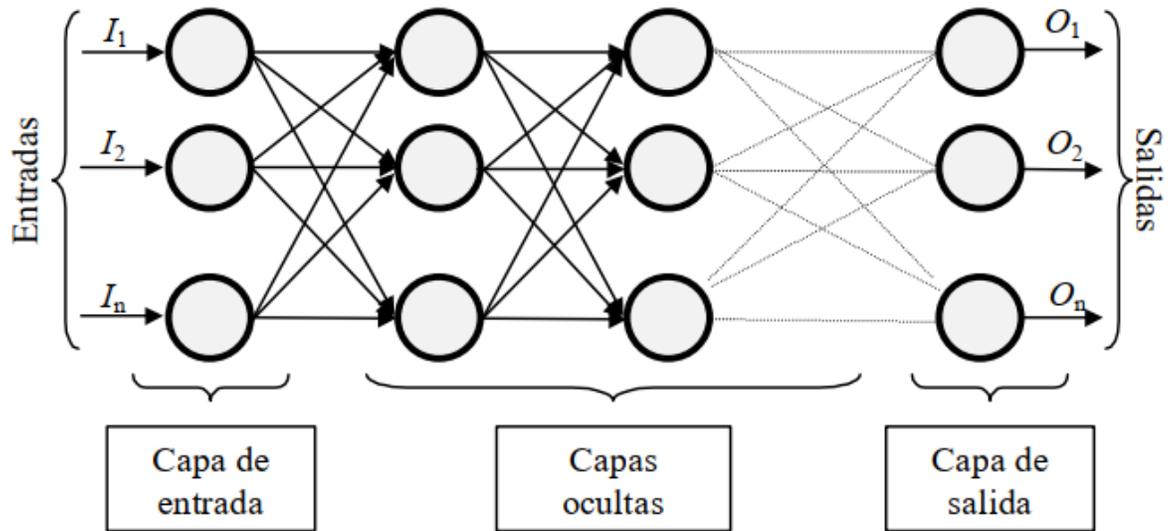
Entonces, no importa cuál es la matriz mezcladora, es equivalente a obtener los datos directamente de la Fuente original. Esto hace también al algoritmo de RMDM más robusto ante las variaciones que pudieran darse entre usuarios o entre sesiones con el mismo usuario. (Congedo, Barachant, & Bhatia, 2017)

### **1.4.3 Aprendizaje Profundo**

**1.4.3.1 Redes Neuronales Artificiales (ANN).** Como se mencionó previamente, Rosenblatt no pudo hallar una regla de aprendizaje para perceptrones de varias capas y Minsky

y Papert demostraron que el perceptrón era inútil ante datos no linealmente separables. (Minsky & Papert, 1969) Sin embargo, Ivakhnenko y Lapa introducen el método de agrupamiento para manejo de datos (GMDH, por sus siglas en inglés), consistente en funciones polinómicas anidadas entrenadas por regresión una capa tras otra. Con la ventaja de poder presentar comportamientos no lineales, se consideran las primeras redes profundas. En 1971 se registra un GMDH con 8 capas (Ivakhnenko y Lapa, 1965; Ivakhnenko, 1971; como se citó en Schmidhuber, 2014).

Actualmente, se considera perceptrón a la arquitectura con una neurona de entrada, una neurona intermedia que efectúa la activación y una capa de salida. Una red neuronal es una combinación de perceptrones, tanto en anchura como en profundidad. El *ancho* es la cantidad de neuronas que hay en una capa, y la *profundidad* se refiere a la cantidad de capas. A una red con más de una capa se le llama también *perceptrón multicapa* (MLP). Y a sus capas intermedias se les llama *capas ocultas*. Si una red posee solamente una capa oculta se conoce como *red neuronal vacía*. No hay consenso en la cantidad mínima de capas necesarias para llamar a red profunda pero comúnmente se considera que si tiene 4 o más capas puede llamarse *red neuronal profunda* (DNN), que se ilustra en la Figura 1.23.



**Figura 1.23. Red neuronal artificial.** Cada círculo es una neurona o nodo. Cada flecha corresponde a una conexión, las ANN pueden estar parcial o totalmente conectadas, esto último significa que cada neurona está conectada a todas las neuronas de la capa anterior y a todas las neuronas de la capa posterior. En cada nodo se realiza la sumatoria ponderada de las entradas que recibe y se le pasa a la función de activación que emite la salida a la siguiente capa. Reproducido de *Redes Neuronales: Conceptos Básicos y Aplicaciones* (p. 12), por Matich, 2001, Universidad Tecnológica Nacional.

**1.4.3.1.1 Retropropagación (BP).** Pero no fue hasta que se propuso el uso de la retropropagación de errores que las ANN retomaron su curso. Werbos mostró la viabilidad y conveniencia de usar la regla de la cadena como medio para identificar el peso del error en cada capa de una red (Werbos, 1982). Rumelhart et al. lo popularizaron mostrando su aplicabilidad en diferentes redes neuronales con distintos objetivos. (Rumelhart, Hinton, & Williams, 1986)

Sea  $\omega_{ij}^l$  el peso entre la neurona  $j$  de la capa  $l - 1$  y la neurona  $i$  de la capa  $l$ ,

$z_i^l = \sum_j \omega_{ij}^l a_j^{l-1}$  la suma ponderada en la neurona  $i$  de la capa  $l$ ,

$a_i^l = \sigma(z_i^l)$  la salida de la función de activación  $\sigma$  en la neurona  $i$  de la capa  $l$ ,

$L$  la capa de salida y  $J$  la función de pérdida. Definimos al error  $\delta$ , la tasa de cambio de la pérdida con respecto a la suma ponderada  $z$  que recibe de cada neurona de la capa de salida, como

$$\delta_i^L \equiv \frac{\partial J}{\partial z_i^L}$$

aplicado la regla de la cadena, se tiene que

$$\delta_i^L = \sum_j \frac{\partial J}{\partial a_j^L} \frac{\partial a_j^L}{\partial z_i^L} = \frac{\partial J}{\partial a_i^L} \frac{\partial a_i^L}{\partial z_i^L} = \frac{\partial J}{\partial a_i^L} \sigma'(z_i^L)$$

Más aún, es posible establecer un error para cada capa considerando

$$\delta_i^l \equiv \frac{\partial J}{\partial z_i^l}$$

así,

$$\delta_j^{l+1} = \frac{\partial J}{\partial z_j^{l+1}},$$

Y nuevamente, aplicando la regla de la cadena a la delta general

$$\delta_i^l = \sum_j \frac{\partial J}{\partial z_j^{l+1}} \frac{\partial z_j^{l+1}}{\partial z_i^l} = \sum_j \delta_j^{l+1} \frac{\partial z_j^{l+1}}{\partial z_i^l} = \sum_j \delta_j^{l+1} \omega_{ji}^{l+1} \sigma'(z_i^l).$$

Por otro lado,

$$\frac{\partial J}{\partial \omega_{ij}^l} = \frac{\partial J}{\partial z_i^l} \frac{\partial z_i^l}{\partial \omega_{ij}^l} = \delta_i^l a_j^{l-1}.$$

Basta con tales relaciones para pasar al algoritmo de retropropagación (Nielsen, 2019).

**Algoritmo 1.1** (Retropropagación)

- 1 **Inicio**
- 2  $\mathbf{z}^1 = \boldsymbol{\omega}^1 \mathbf{x}$
- 3  $\mathbf{a}^1 = \sigma(\mathbf{z}^1)$
- 4 **For**  $l = 2:L$
- 5  $\mathbf{z}^l = \boldsymbol{\omega}^l \mathbf{a}^{l-1}$

```

6       $a^l = \sigma(z^l)$ 
7       $\delta^L = \nabla_a J \odot \sigma'(z^L)$ 
8      For  $l = L - 1: -1: 2$ 
9           $\delta^l = (\omega^{l+1^T} \delta^{l+1}) \odot \sigma'(z^l)$ 
10      $\nabla_{\omega^l} J = a^{l-1} \delta^l$ 
11      $\omega^l = \omega^l - \eta \nabla_{\omega^l} J$ 
12 Fin

```

donde  $\odot$  es el producto de Hadamard.

De esta manera, la red aprende tras la actualización de los pesos de cada neurona en proporción al cambio que provoca en la función de pérdida la modificación de un peso dado.

**1.4.3.2 Redes Neuronales Convolucionales (CNN).** Una Red Neuronal Convolutiva o Convolutional Neural Network (CNN) toma su nombre de las convoluciones realizadas entre regiones acotadas de una capa neuronal y su kernel, que es una matriz de pesos. El resultado de tales convoluciones se pasa como entrada a la función de activación. Y la operación se va realizando a lo largo de toda la capa, tomando pequeños pasos hasta que todas las neuronas son tomadas en cuenta.

En 1989, LeCun et al. (citado por Goodfellow, Bengio, & Courville, 2016) implementó por primera vez una CNN “moderna”, probablemente inspirado en el Neocognitron de Fukushima y/o en los trabajos de Hubel y Wiesel en neurofisiología de la visión, consiguiendo un 95% de exactitud en el reconocimiento de los dígitos escritos a mano de la base de datos MNIST. (Goodfellow, Bengio, & Courville, 2016)

**1.4.3.2.1 Convolución.** Una convolución es una operación matemática, muy utilizada en análisis de señales, que puede interpretarse como el resultado de la superposición de una señal  $f$  con una señal  $g$  invertida y trasladada una distancia  $x$ , que se desplaza a lo largo de todo el dominio. Se define como

$$h(x) = f(x) * g(x) = \int_{-\infty}^{\infty} f(z)g(x - z)dz$$

Aunque la convolución es conmutativa, se le suele llamar filtro o *kernel* a la función  $g$  y se dice que es aplicado sobre  $f$ , y a la función resultante se le conoce como *mapa de características*. Su uso en computación requiere una discretización, dada por

$$h(x) = f(x) * g(x) = \sum_{z=-\infty}^{\infty} f(z)g(x - z)$$

Y para el tratamiento de imágenes o señales bidimensionales puede extenderse como

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

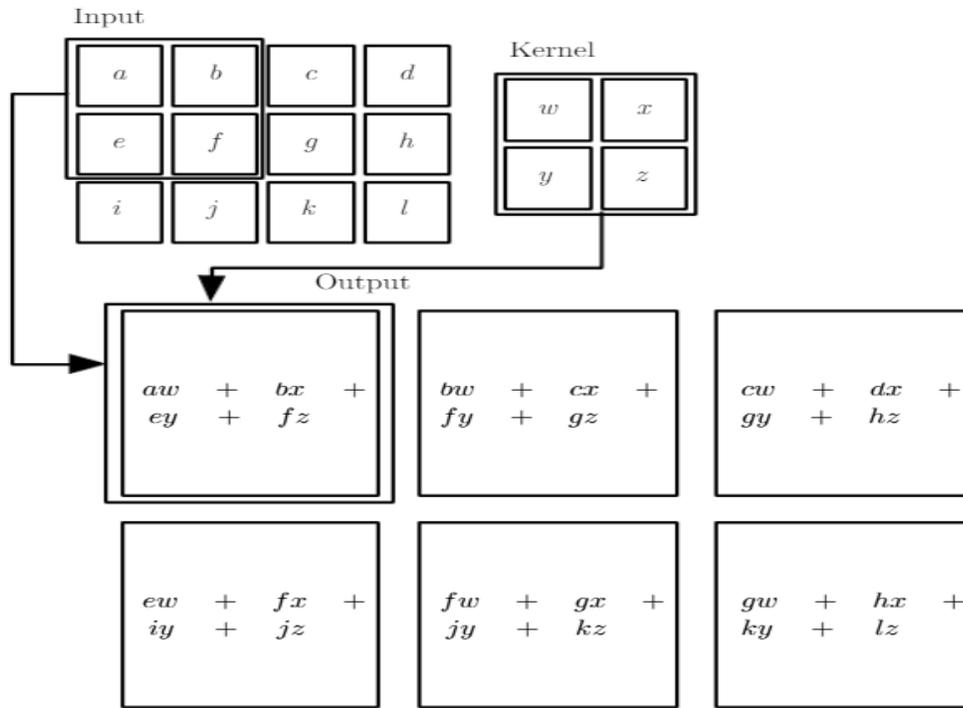
Así, una CNN toma la entrada de la capa neuronal y la convoluciona con el kernel, que funge como una matriz de pesos en las ANN, en el sentido que  $S$  se pasará a la función de activación, y serán los valores del kernel los parámetros que se actualicen mediante el algoritmo de retropropagación.

Sin embargo, nótese que mientras  $m$  incrementa el índice de  $I$  incrementa, pero el de  $K$  disminuye. Esto se debe a que la convolución invierte a  $g$ , como se mencionó arriba. Pero si la inicialización del kernel será aleatoria y sus entradas se modificarán tras cada iteración, resultaría más sencillo si no se tuviera que invertir el kernel. Actualmente, la mayoría de las implementaciones de CNN ya no aplican la convolución, sino que libran la complejidad innecesaria de invertir el kernel usando la correlación cruzada, que en 2D se ve así:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Pero en la práctica se le sigue llamando convolución y las Convolutional Neural Networks no han cambiado su nombre a *Correlational Neural Networks*. En fin, su ventaja es que simplifica

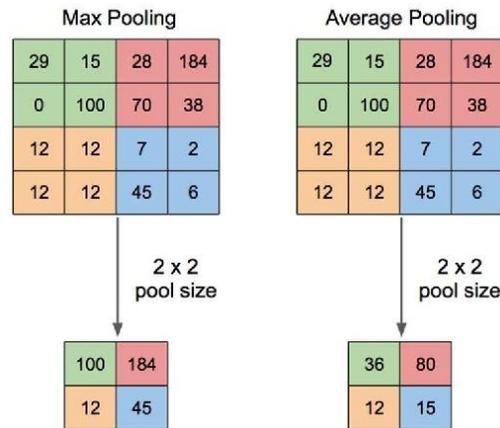
el cómputo y la comprensión de su implementación. (Goodfellow, Bengio, & Courville, 2016) En la Figura 1.24 se ve gráficamente cómo opera la correlación cruzada.



**Figura 1.24. Correlación cruzada.** Se tiene una matriz de 3x4 como entrada y un kernel de 2x2 que se le aplica como filtro mediante la correlación cruzada. El resultado se muestra debajo, el mapa de características es una matriz de 2x3. Reproducido de *Deep Learning* (p. 330), por Goodfellow et al., 2016, The MIT Press.

**1.4.3.2.2 Agrupación.** Un detalle que resulta limitante de las CNN es que aprenden a identificar pequeñas regiones del tamaño de su kernel, pero en ubicaciones muy específicas. Una solución sencilla y eficaz es la *agrupación* o *pooling*, que resume la información de una región en un pequeño bloque que dotará de holgura en la detección de características. Al mismo tiempo, reduce la dimensionalidad, reduciendo el tiempo de cómputo y promoviendo evitar el sobreajuste. Se coloca una capa adicional entre después de la capa convolucional para realizar la agrupación. El filtro de agrupación generalmente se elige de 2x2, lo que reduce a la mitad cada dimensión, y a su vez, reduce el mapa de características a un cuarto de su original.

Los algoritmos de agrupación más utilizados son: *agrupación promedio* o *average pooling*, que toma el promedio de los valores del mapa de características; y la *agrupación máxima* o *max pooling*, que conserva el solo el valor máximo en cada filtro. En la Figura 1.25 se ejemplifican ambos con la misma entrada para ver la diferencia en sus efectos.



**Figura 1.25. Diferencia entre max pooling y average pooling.** Los cuadros superiores representan el mapa de características donde se aplicará una agrupación con un filtro de 2x2 y un paso de 2, representado con los grupos de distintos colores. A la izquierda, se le aplica Max Pooling, que toma los valores máximos de cada grupo. A la derecha, se le aplica Average Pooling, que conserva el promedio de cada grupo. Abajo se muestran los nuevos mapas de características. Modificado de *Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail* (p. 3), por Yani et al., 2019, Journal of Physics Conference Series.

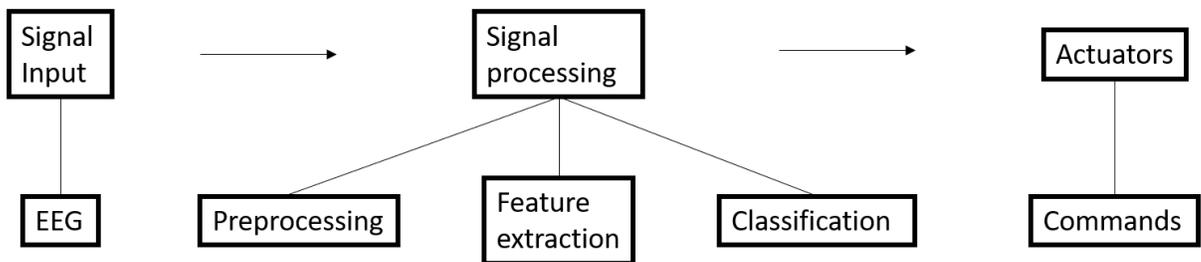
## 1.5 Interfaz Cerebro Máquina

Una interfaz cerebro máquina (BMI por sus siglas en inglés) es un sistema de comunicación que permite interactuar con un dispositivo electrónico a través de la mera actividad cerebral. Para lograr esto se requiere del uso de algoritmos de clasificación y reconocimiento de patrones.

El paradigma clásico de la estructura de un BMI, esquematizado en la Figura 1.26, incluye 3 etapas: recepción de datos, procesamiento de datos, y actuadores. En la primera etapa se adquiere la señal bioeléctrica, y pueden usarse diferentes tecnologías, como EEG, MEG, ECoG, EMG, entre otras. (Clerc, Bougrain, & Lotte, Brain-Computer Interfaces 1: Foundations and

Methods, 2016; Graimann, Allison, & Pfurtscheller, 2013) La más común es EEG debido a su bajo costo, portabilidad, resolución temporal, bajo requerimiento computacional, y no invasividad, en comparación con las otras técnicas. La segunda etapa es donde todo el análisis se lleva a cabo. Esto incluye el preprocesamiento que los algoritmos de clasificación pudieran conllevar, la extracción de características y la clasificación por sí misma. En la etapa final, la salida del clasificador es transformada en órdenes para producir una acción.

Un ejemplo de esta configuración es una silla de ruedas eléctrica cuyo movimiento hacia adelante o hacia atrás es controlado por un detector de parpadeos mediante EEG, usando LDA como clasificador. En la etapa de recepción, el EEG capta la señal mediante electrodos en el cuero cabelludo. Para el preprocesamiento, se conservan solo las señales de los electrodos más cercanos a cada ojo y el resto son descartados. Si el parpadeo incrementó la energía de la señal en el electrodo más cercano al ojo guiñado, y se redujo en el electrodo opuesto, tendríamos la característica a usar como discriminante para el clasificador. El cálculo de la energía en ventanas de tiempo es la extracción de características. El clasificador recibe entonces 2 cantidades, la energía de cada electrodo en una ventana de tiempo dada, permitiendo decir cuál fue el ojo guiñado. Finalmente, se asigna una acción a cada clase. Por ejemplo, si se guiña el ojo derecho la silla avanzará hacia adelante, y hacia atrás si es el izquierdo.



**Figura 1.26. Arquitectura clásica de una BMI.** Tiene 3 etapas: Entrada, procesamiento de los datos, y actuadores.

# 2. Metodología

Se utilizaron dos bases de datos de EEG, una pública provista por PhysioNet y una propia generada en nuestro laboratorio. Esto con el fin de comparar los resultados tras entrenar los clasificadores por separado. Por un lado, el entrenamiento con los datos públicos, y por otro, el entrenamiento con los datos obtenidos en el laboratorio. A continuación, se detallan los protocolos de obtención de cada uno y su tratamiento.

## 2.1 Datos Públicos

PhysioNet ofrece una base de datos pública que incluye series de tiempo de EEG bajo tareas de imaginación motora y movimiento real. Está disponible para su descarga en <https://physionet.org/content/eegmmidb/1.0.0/>. (Goldberger, y otros, 2000)

### 2.1.1 Voluntarios

Hay disponibles datos de 109 sujetos, pero solo se usaron los de 92 de ellos. Los voluntarios 14, 34, 37, 41, 51, 64, 69, 72, 73, 74, 76, 88, 92, 100, 102, 104, y 109 fueron excluidos debido a que presentaban un menor número de eventos o eventos incompletos. Esto con el fin de mantener el número de eventos homogéneo entre todos los sujetos. El experimento se realizó y grabó en el Wadsworth Center BCI Research and Development Program. (Wolpaw, Mcfarland, Vaughan, & Schalk, 2003) En el sitio web de PhysioNet no se brinda más información sobre los voluntarios.

### 2.1.2 EEG

Se usó un equipo EEG de 64 canales, con una frecuencia de muestreo de 160 Hz. Los electrodos fueron colocados de acuerdo con el sistema internacional 10-10, excluyendo: Nz, F9, F10, FT9, FT10, A1, A2, TP9, TP10, P9, y P10. Los datos fueron grabados usando el sistema

BCI2000 (<http://www.bci2000.org>), en una PC corriendo Windows 2000, con procesador Athlon de 1.4 GHz, 256 Mb de RAM, subsistema I/O IDE, y una tarjeta de adquisición de datos Data Translation DT3003 (Schalk, McFarland, Hinterberger, Birbaumer, & Wolpaw, 2004) y las secuencias de tiempo están en archivos formato edf.

### **2.1.3 Adquisición de Datos**

A cada voluntario se le solicitó realizar 14 acciones:

1. Línea base con ojos abiertos;
2. Línea base con ojos cerrados;
3. Tarea 1 (abrir y cerrar puño izquierdo o derecho);
4. Tarea 2 (imaginar abrir y cerrar puño izquierdo o derecho);
5. Tarea 3 (abrir y cerrar ambos puños o ambos pies);
6. Tarea 4 (imaginar abrir y cerrar ambos puños o ambos pies);
7. Tarea 1;
8. Tarea 2;
9. Tarea 3;
10. Tarea 4;
11. Tarea 1;
12. Tarea 2;
13. Tarea 3;
14. Tarea 4.

Se le llama línea base a la grabación de EEG mientras el sujeto no realiza alguna actividad en particular, física o mental. Simplemente se le solicita permanecer relajado, normalmente sentado. Por simplicidad, en este estudio solamente se usa la información de la línea base con ojos abiertos.

La grabación de las líneas base duró 1 minuto cada una, y 2 minutos la de cada tarea. En cada tarea se le pide realizar 5 o 6 veces cada uno de los 2 movimientos posibles, o imaginación de los movimientos, según sea el caso. Por ejemplo, en la tarea 1, se le pide en 5 ocasiones que abra y cierre el puño derecho durante 6 segundos; y en 6 ocasiones que abra y cierre el puño durante 6 segundos. Cada uno de estos intervalos de 6 segundos en que el voluntario imagina o ejecuta un movimiento es considerado un evento. Para cada uno de los 92 sujetos hay 45 eventos de cada tarea. Nuevamente, por simplicidad, aquí solo se consideraron los datos de la tarea 2 (acciones número 4, 8 y 12), que corresponde a imaginar abrir y cerrar el puño izquierdo o derecho. Y se recortaron los eventos, tomando solo los primeros 3 segundos de cada uno. Esto último para empatarlo con la longitud de los eventos en los datos de nuestro laboratorio (ver los detalles de la adquisición de los datos de laboratorio en las siguientes secciones). Asimismo, la totalidad de la grabación de línea base fue dividida en secciones de 3 segundos. De manera que se obtuvieron 20 eventos de línea base por cada sujeto.

Idealmente, en el entrenamiento de un modelo de aprendizaje automático, el número de eventos por clase debe ser similar. Ya que los datos de la línea base y la tarea no están balanceados, se implementó un procedimiento de aumentación de datos. Se generaron 'nuevos' eventos de línea base uniendo la mitad final de un evento con la mitad inicial del evento contiguo. Así, en vez de tener solo 20 eventos por sujeto, se contó con 39 eventos de línea base por sujeto. Esto produjo un total de 3588 eventos de línea base (considerando los 92 sujetos), y 4140 eventos de la tarea 2, de los cuales, 2086 corresponden al puño izquierdo y 2054 al puño derecho.

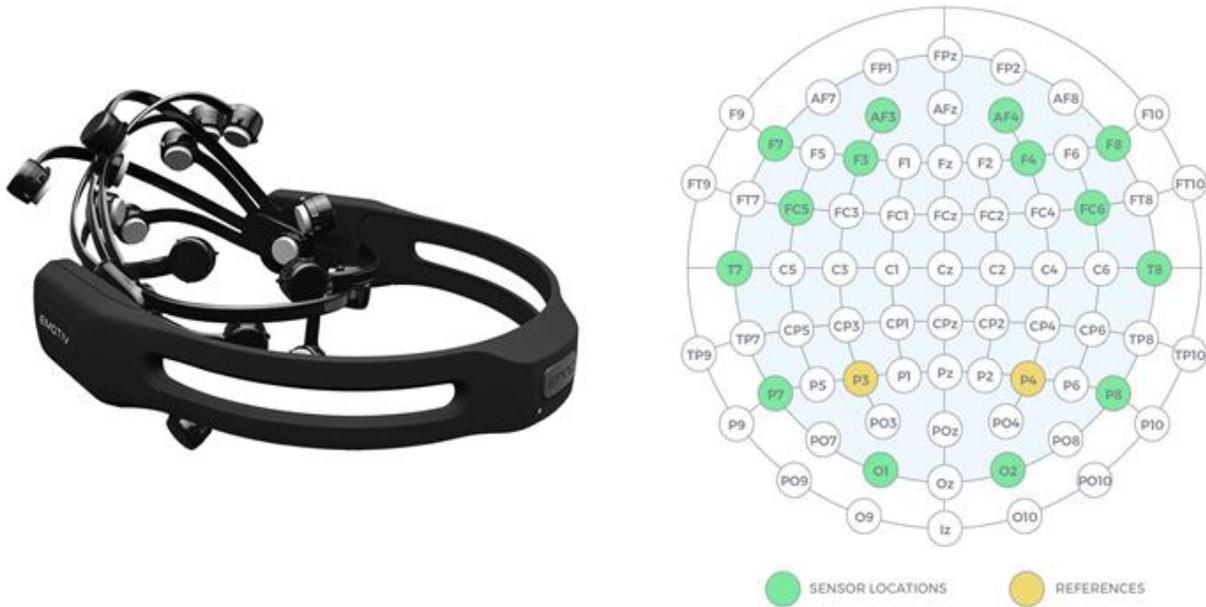
## **2.2 Datos de Laboratorio**

### **2.2.1 Voluntarios**

Los datos de EEG fueron generados de una muestra de 30 voluntarios masculinos, diestros, carentes de diagnóstico de patologías neurológicas y condiciones psiquiátricas o psicológicas. Sus edades se hallaron dentro del rango de 18 a 31 años. Todos aseguraron haber dormido más de 6 horas la noche antes del experimento, no estar bajo el efecto de psicoactivos, y bajo ningún tratamiento médico.

### **2.2.2 EEG**

Los datos se obtuvieron usando la diadema Emotiv EPOC+ (<https://www.emotiv.com/epoc/>). Su frecuencia de muestreo es de 128 Hz, cuenta con 14 electrodos dispuestos acordes al sistema internacional 10-20 (AF3 F7 F3 FC5 T7 P7 O1 O2 P8 T8 FC6 F4 F8 AF4), como se muestra en la Figura 2.1. Posee un filtro notch integrado para frecuencias de 50 y 60 Hz. Como interfaz PC-Emotiv se utilizó el software comercial del equipo: EmotivPRO v1.8.1. Conectada a través de su Bluetooth incluido.



**Figura 2.1. Emotiv EPOC+ y sus electrodos.** Se muestra la diadema Emotiv EPOC+ (izquierda) y los electrodos que posee según en sistema 10-10 (derecha). Adaptado de Emotiv, 2021, (<https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeeg/#tab-description>).

### 2.2.3 Adquisición de Datos

El protocolo implementado estuvo basado principalmente en los trabajos de (Brunner, Leeb, Müller-Putz, Schlögl, & Pfurtscheller, 2008), (Lee & Choi, 2018), y (Wu, Huang, Lin, Tsai, & Wang, 2018). Las grabaciones se realizaron en un laboratorio bien iluminado, con poco ruido, sin olores significativos, buscando reducir al máximo los estímulos externos que pudieran interferir con el experimento. Se les solicitó a los voluntarios permanecer sentados cómodamente. A un metro de distancia se dispuso una pantalla mostrando una secuencia de imágenes correspondientes con cada tarea, mediante una presentación en Microsoft PowerPoint (<http://www.office.com>). La pantalla usada fue la de una laptop con display LED, de 14 pulgadas, con resolución de 1920x1080, y una frecuencia de 60 Hz.

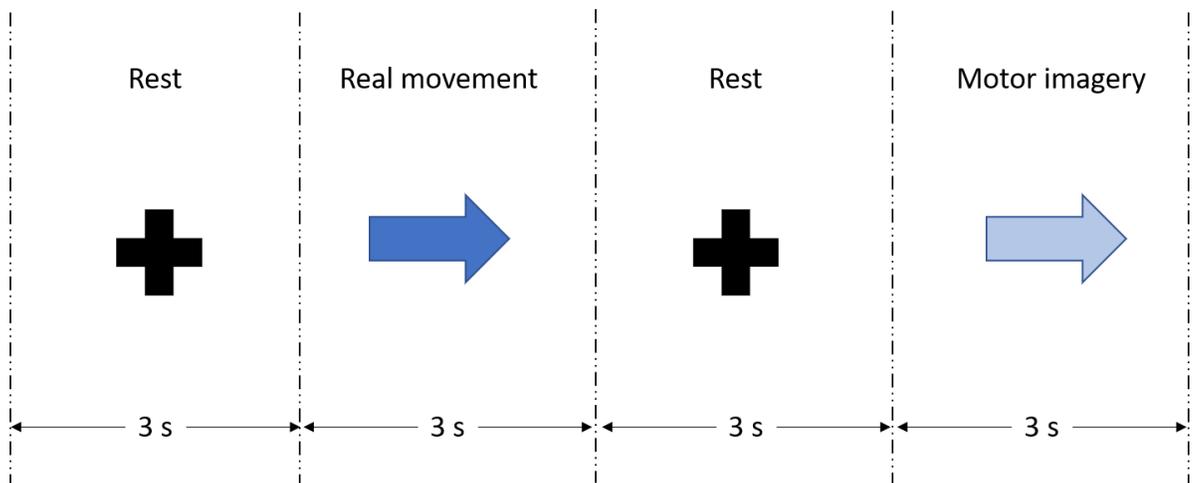
A cada voluntario se le pidió realizar las siguientes acciones:

- Línea base con ojos abiertos

- Levantar el brazo izquierdo
- Imaginar levantar el brazo izquierdo
- Levantar el brazo derecho
- Imaginar levantar el brazo derecho

La grabación de línea base duró 30 segundos. Posteriormente, hubo tres bloques de 16 ciclos de grabación. Cada ciclo consistió en un conjunto de descanso – movimiento – descanso – imaginación. A su vez, cada uno de los elementos del ciclo tuvo una duración de 3 segundos, por lo que cada ciclo tuvo una duración de 12 segundos.

Se mostró en pantalla una imagen diferente para cada tarea. Para la línea base y el descanso entre tareas, donde bajaban los brazos, se mostró una cruz negra; una flecha relleno sólido para el movimiento real y una flecha relleno suave para la imaginación, apuntando hacia el lado a mover o imaginar, azul para el lado derecho y verde para el izquierdo. En cada ciclo, el lado del movimiento y la imaginación fue el mismo. En la Figura 2.2 se muestran las imágenes de un ciclo derecho.



**Figura 2.2. Estructura de un ciclo derecho.** Cada ciclo consta de 4 elementos: descanso, movimiento real, descanso, imaginación de movimiento. Cada elemento dura 3 segundos. Los datos de EEG generados durante el elemento de imaginación corresponden a un evento En este caso el evento corresponde a la clase Imaginación Motora Derecha (RMI).

La lateralidad de los ciclos se fue alternando de acuerdo con una sucesión uniforme pseudoaleatoria generada con NumPy en Python, pero al final se mantuvo el balance, 24 ciclos por cada lado del cuerpo.

Los elementos de imaginación en cada ciclo corresponden a los eventos utilizados para entrenar los clasificadores. De manera que, considerando los 30 voluntarios, se obtuvieron 300 eventos de línea base y 720 eventos de imaginación motora por cada lado. De igual modo que para los datos de PhysioNet, se llevó a cabo el mismo procedimiento de aumentación de datos para lidiar con la falta de balance entre el número de eventos. Se tomó la mitad final de un evento de línea base y se unió con la mitad inicial de su evento contiguo para generar un 'nuevo' evento. Así, se contó con 570 eventos de línea base, una cantidad más cercana a los 720 eventos de cada una de las otras clases.

### **2.3 Arquitecturas de BMI**

Los eventos fueron etiquetados para el entrenamiento de los clasificadores en aprendizaje supervisado. Se consideraron 3 clases: línea base (BL), imaginación motora izquierda (LMI), e imaginación motora derecha (RMI). Cada algoritmo clasificador fue entrenado en 4 grupos de clasificación: LMI vs RMI, LMI vs BL, RMI vs BL, y el caso multiclase LMI vs RMI vs BL.

Se realizó el entrenamiento en dos formas diferentes. La global, alimentando el algoritmo con los datos de todos los sujetos. Y la forma por sujeto, entrenando y validando el modelo con los datos particulares de cada individuo. Esto para comparar la eficacia de cada clasificador con pocos o muchos datos, y sus capacidades de generalización.

### **2.3.1 Equipo**

Los modelos de aprendizaje automático fueron programados en Python (<https://www.python.org/>) usando el framework Keras (<https://keras.io/>), y las librerías sklearn (<https://scikit-learn.org/>) y pyRiemann (<https://pyriemann.readthedocs.io>). Y fueron ejecutados en una PC comercial corriendo Windows 10, con 32 GB de RAM, procesador Intel Core i7 – 6700 a 3.4 GHz, y una tarjeta gráfica NVIDIA GeForce GTX 1070.

### **2.3.2 CSP + LDA**

Los datos crudos fueron ordenados en una matriz de  $E \times N \times T$ , donde  $E$  es el número de eventos,  $N$  el número de electrodos, y  $T$  el número de muestras en la serie de tiempo. Posteriormente, fueron re-referenciados a la referencia de media común (CAR por sus siglas en inglés), balanceados y normalizados. Se hicieron pasar por un filtro pasa bandas Butterworth de sexto orden con una ventana de 8 a 30 Hz. Los CSP fueron calculados usando el paquete MNE (<https://mne.tools/>), mediante el método Ledoit-Wolf para estimación de covarianzas. Finalmente, se usaron 6 CSP como vector de características y entrada del clasificador de LDA.

### **2.3.3 DNN**

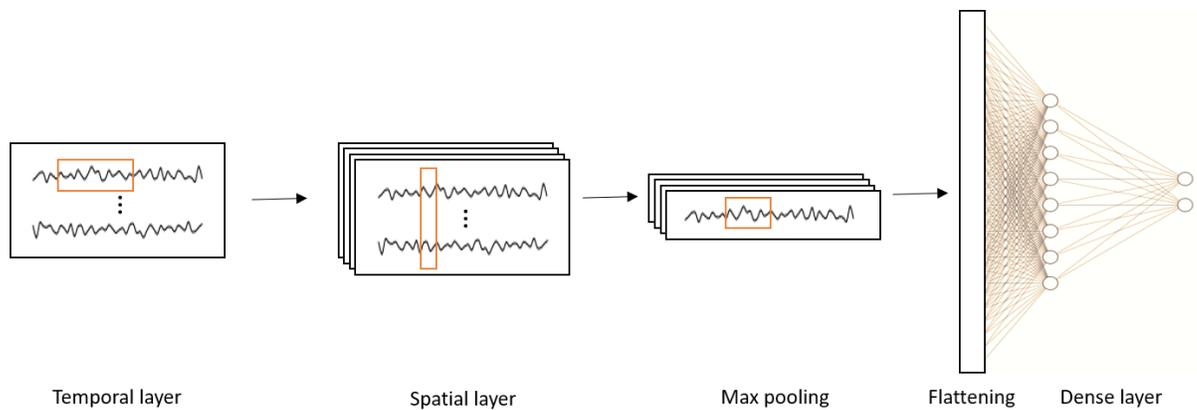
Primero los datos se centraron dentro del rango  $[-1,1]$  con la función `maxAbsScaler` de sklearn. Se utilizó una red neuronal profunda con 9 capas ocultas. El inicializador de kernel fue el mismo para toda la red: `Random Uniform` entre  $[-0.05, 0.05]$ , con 42 como semilla. Se eligió `Leaky ReLU` con  $\alpha=0.3$  como función de activación en las capas internas, a fin de evitar la muerte de las neuronas que emiten valores negativos, conservando un pequeño gradiente; y `Softmax` en la capa de salida debido a que funciona para cualquier cantidad de clases. El optimizador aplicado fue `Nadam` con tasa de aprendizaje entre  $1e^{-6}$  y  $1e^{-9}$ , y entropía cruzada como función de pérdida. Se implementó un 30% de dropout en cada capa. El número de épocas

se determinó empíricamente, cuidando evitar tanto el subajuste como el sobreajuste, entre 30 y 100.

#### **2.3.4 CNN**

El modelo de CNN implementado estuvo basado en la arquitectura propuesta en Dose et al. Consistió en 2 capas convolucionales con 40 neuronas cada una. En la primera capa no se aplicó padding y el tamaño de su kernel fue de  $30 \times 1$ . En la segunda se aplicó padding cero y su kernel fue de  $1 \times 64$ . En ambos casos, se usó el paso por defecto de  $(1,1)$ . Se hizo pasar por un max pooling de  $15 \times 1$  con padding cero. Después, se aplanaron los datos para pasar a una capa completamente conectada con 80 neuronas. Finalmente, la capa de salida consistió en 2 neuronas. Sin embargo, los tamaños de kernel de las capas se adaptaron al entrenar los datos de laboratorio de la siguiente manera:  $24 \times 1$ ,  $1 \times 14$ , y  $4 \times 1$  respectivamente. El inicializador del kernel y las funciones de activación conservaron las mismas condiciones que con DNN.

La idea detrás de este arreglo estuvo basada en permitir que la CNN extraiga las características de los datos crudos. Se esperaba que la primera capa convolucional fungiera como un filtro espacial entre canales. Mientras que la segunda servía de filtro temporal entre muestras. Entonces, la CNN recibió como entrada los datos crudos del EEG en forma de una matriz de  $T \times N$ , donde  $N$  fue el número de electrodos y  $T$  el número de muestras por evento. Luego, los datos 'filtrados' fueron aplanados y se hicieron pasar por una capa adicional para realizar la clasificación como tal. Tal estructura es mostrada en la Figura 2.3.



**Figura 2.3. Arquitectura de la CNN biclase.** La CNN consistió en una capa temporal en la cual un kernel horizontal extrajo las características en el dominio temporal de la serie de tiempo de cada canal. Luego, una capa espacial con un kernel vertical extrajo las características relacionadas a la posición de los electrodos en una cierta ventana de tiempo. Posteriormente, vino el Max pooling para resumir el aprendizaje previo. Finalmente, los datos fueron aplanados para pasar a una capa completamente conectada previa a la salida binaria para el caso biclase.

### 2.3.5 RMDM

Este clasificador estuvo basado en el trabajo de (Congedo, Barachant, & Bhatia, 2017) Los eventos fueron filtrados con un filtro pasabanda Butterworth de sexto orden con una ventana de 8 a 30 Hz. Los datos filtrados dieron lugar a una matriz de  $E \times N \times T$ , donde E fue el número de eventos, N el número de canales, y T el número de muestras por evento. Esta matriz fue la entrada al clasificador. La media riemanniana de cada clase y la distancia desde cada evento a tales medias fueron calculadas con la ayuda de la librería pyRiemann. Finalmente, se ejecutó una validación cruzada de 5 iteraciones sobre los valores resultantes de la clasificación.

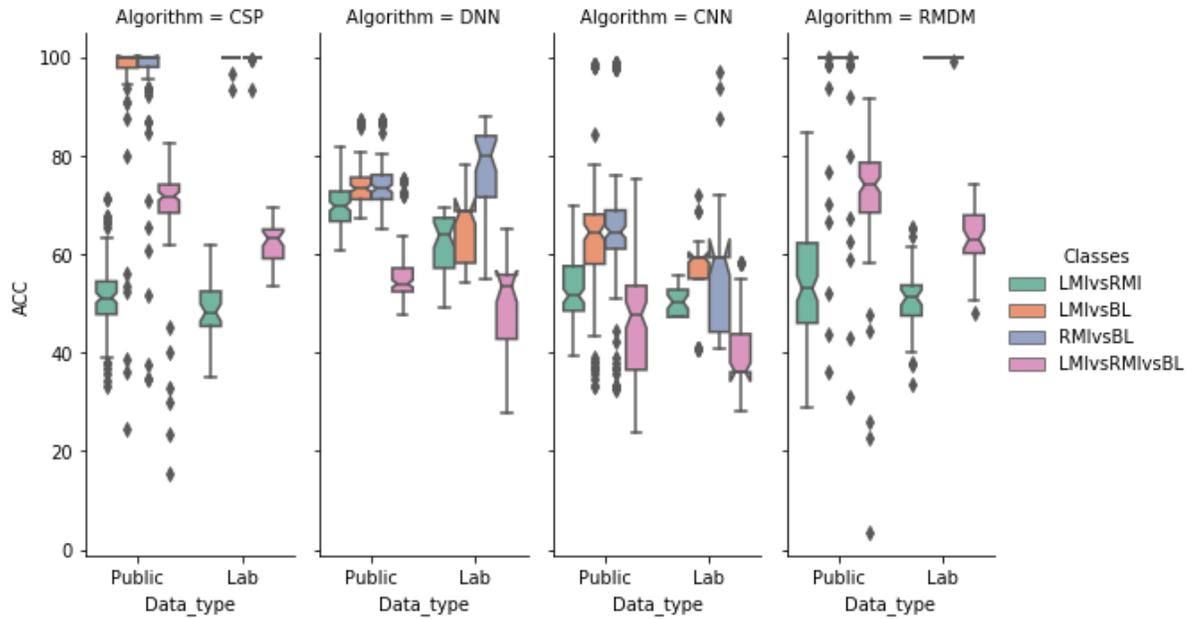
# 3. Discusión y Resultados

Un resumen de la exactitud media obtenida con los diferentes clasificadores se presenta en la Tabla 3.1. Muestra los resultados separados por algoritmo, tipo de datos (laboratorio o PhysioNet), y tipo de entrenamiento (por sujeto, global). Cada fila representa un grupo de clasificación (LMIvsRMI, LMIvsBL, RMIvsBL, or LMIvsRMIvsBL) usando un clasificador en específico.

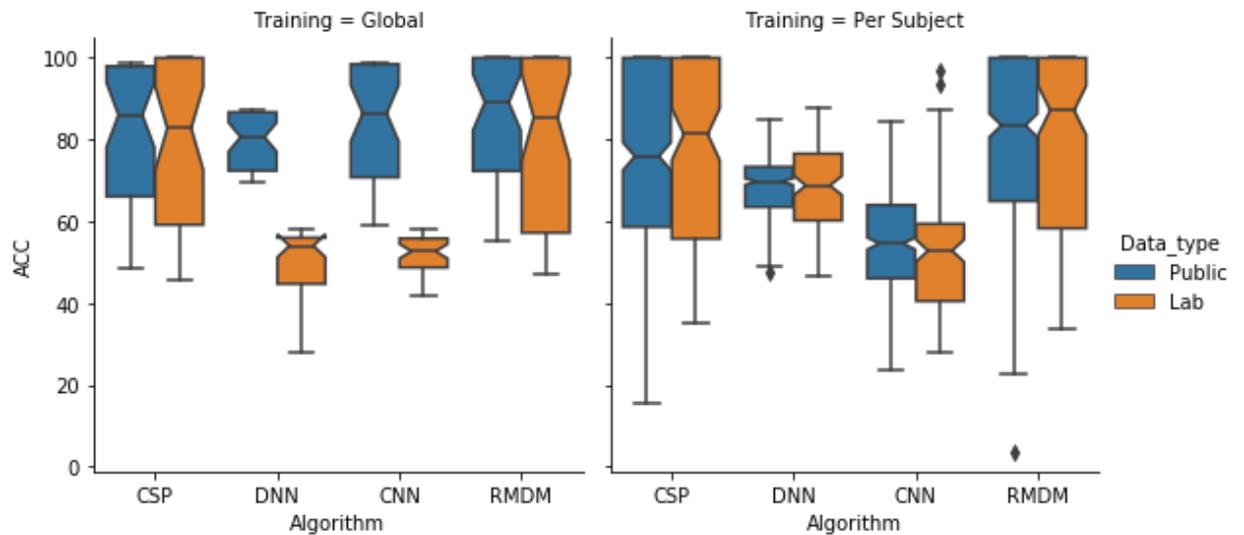
La Figura 3.1 contiene una representación gráfica de la exactitud en función de las clases. Esto con el fin de evaluar la relación entre ambos conjuntos de datos: los generados en nuestro laboratorio con Emotiv y los de PhysioNet. Siguiendo el formato de la Tabla 3.1, el gráfico cuenta con subdivisiones para cada algoritmo, que permiten ver cuál ha sido el mejor clasificador en cada caso. La comparación entre los resultados por tipo de entrenamiento se muestra en la Figura 3.2. De igual manera se ofrece con los datos separados por clasificador.

Algoritmo	Grupo	Datos de Laboratorio		Datos de PhysioNet		
		Por Sujeto	Exactitud Media (%)		Por Sujeto	Global
			Global	Por Sujeto		
<b>CSP+LDA</b>	<i>LMI/RMI</i>	48.1 ± 6.7	49.7 ± 3	51.4 ± 8.2	49.8 ± 1.7	
	<i>LMI/BL</i>	99.7 ± 1.3	100	95.1 ± 14.3	97.9 ± 0.3	
	<i>RMI/BL</i>	99.8 ± 1.2	99.9 ± 0.2	95.1 ± 13.7	98.1 ± 0.3	
	<i>LMI/RMI/BL</i>	61.6 ± 4.5	64.1 ± 1.6	69.1 ± 11.1	73.4 ± 1.3	
<b>DNN</b>	<i>LMI/RMI</i>	65.6 ± 3.5	50.2 ± 1.3	70.2 ± 4.2	71.4 ± 1.2	
	<i>LMI/BL</i>	68.8 ± 4.5	56.5 ± 1.8	72.5 ± 3.7	86.4 ± 0.4	
	<i>RMI/BL</i>	81.5 ± 2.7	56.6 ± 1.7	73 ± 12.1	86.9 ± 0.4	
	<i>LMI/RMI/BL</i>	54.5 ± 4.1	29.3 ± 1.6	54.2 ± 3.1	73.6 ± 1.3	
<b>CNN</b>	<i>LMI/RMI</i>	50.6 ± 2.9	50.1 ± 0.2	52.3 ± 6.5	59.5 ± 0.4	
	<i>LMI/BL</i>	56.8 ± 8.8	56.6 ± 1.7	59.5 ± 11.7	98.4 ± 0.2	
	<i>RMI/BL</i>	60.1 ± 13.5	43.4 ± 1.7	61 ± 10.7	98.3 ± 0.5	
	<i>LMI/RMI/BL</i>	33.7 ± 3.9	56.6 ± 1.7	44.2 ± 9.7	74.7 ± 0.3	
<b>RMDM</b>	<i>LMI/RMI</i>	51.1 ± 8.1	50.7 ± 2.4	53.8 ± 13.4	57.9 ± 1.9	
	<i>LMI/BL</i>	100	100	97.1 ± 11.2	99.9 ± 0.1	
	<i>RMI/BL</i>	100	99.9 ± 0.2	97 ± 11.4	99.9 ± 0.1	
	<i>LMI/RMI/BL</i>	63.3 ± 6.2	63.6 ± 4	72 ± 13.1	77.5 ± 0.9	

Tabla 3.1. Resumen de resultados.



**Figura 3.1. Exactitud vs. clases por clasificador y por tipo de datos.** El eje vertical representa la exactitud (%). Hay 4 bloques, separados por líneas verticales, correspondientes a los 4 algoritmos utilizados, y 2 sub-bloques correspondientes a los conjuntos de datos, a la izquierda PhysioNet y a la derecha los de nuestro laboratorio. Cada color representa un grupo de clasificación. LMI es MI izquierda, RMI es MI derecha, y BL es la línea base.



**Figura 3.2 Resultados por tipo de entrenamiento.** El bloque de la derecha muestra los resultados del entrenamiento por sujeto. Mientras que el izquierdo muestra los del entrenamiento global. El color azul corresponde a los datos de PhysioNet y el naranja a los datos de nuestro laboratorio.

### 3.1 Discusión

Los principales hallazgos de esta investigación fueron exactitudes medias de 78%, 66%, 60% y 80% para los clasificadores CSP, DNN, CNN y RMDM, respectivamente. Los mejores resultados fueron obtenidos con los casos de línea base contra MI. Con los datos públicos y entrenamiento global se obtuvieron exactitudes entre 86.4% y 99.9%. Con datos de nuestro laboratorio y entrenamiento global, la exactitud fue superior al 99% solo en los casos de CSP y RMDM. Asimismo, para datos propios, los tiempos de cómputo en la clasificación/predicción por evento fueron de 8.3 ms, 18.1 ms, 62 ms y 9.9 ms, para CSP, DNN, CNN y RMDM, respectivamente, lo que muestra la viabilidad del uso de estos algoritmos en una BMI.

Los cuatro algoritmos tuvieron algún problema al discriminar entre imaginación motora izquierda y derecha, pero casi todos tuvieron buen desempeño clasificando la imaginación motora contra la línea base, independientemente de la lateralidad, tanto con datos públicos como con los de nuestro laboratorio. Es importante notar que, en los cuatro algoritmos, el error medio en el entrenamiento por sujeto fue alrededor de 10% mayor que en el entrenamiento global. Esto es debido a la amplia varianza en los resultados entre los individuos; para un solo sujeto la exactitud fue del 20% mientras que la del resto se mantuvo cerca del 99%. Esta amplitud se ve difuminada en el entrenamiento global, donde internamente el clasificador considera estas diferencias.

Por otro lado, si se toma en cuenta solo que los datos no están perfectamente balanceados, se esperaría que en los datos de laboratorio se tuviera por lo menos una exactitud del 50% para el caso de MI izquierda contra MI derecha, 44.2% en MI contra línea base, y un 28.4% en el caso triclase. Mientras que para los datos públicos se esperarían exactitudes de 49.6%, 36.8%, 36.4%, y 26.6%, respectivamente. Claro que, para fines prácticos, la expectativa real está por encima del 50%, por lo menos, para pensar en una BMI útil.

### 3.2 Tiempo De Cómputo

La Tabla 3.2 muestra los tiempos de cómputo promedio por evento. El tiempo de entrenamiento dependió fuertemente en el hardware utilizado, y la GPU usada para este estudio fue un elemento comercial promedio. Considerando alrededor de 5000 eventos en un problema biclase, tomó cerca de 45 minutos para que la CNN estuviera lista, que fue la más tardada. El más rápido fue CSP + LDA, estando entrenado en menos de 5 minutos. Sin embargo, la parte más importante fue el tiempo de predicción para valorar su viabilidad como una verdadera BMI en tiempo real.

Algoritmo	Tiempo promedio de entrenamiento por evento (ms)		Tiempo promedio de clasificación por evento (ms)	
	Laboratorio	PhysioNet	Laboratorio	PhysioNet
CSP+LDA	14.2 ± 0.3	55.3 ± 1.2	8.3 ± 0.2	34.4 ± 0.8
DNN	20.7 ± 0.7	212.5 ± 2.5	18.1 ± 1.2	117.8 ± 7.3
CNN	85.1 ± 1.3	524.1 ± 3.2	62 ± 3.6	373.2 ± 11.7
RMDM	17.5 ± 0.3	98.5 ± 0.7	9.9 ± 0.4	46.2 ± 1.2

Tabla 3.2. *Tiempo de cómputo.*

Con los datos de laboratorio, la frecuencia de muestreo fue de 128 Hz (cada ~8 ms se toma una muestra). Si cada evento tuvo duración de 3000 ms, incluso el algoritmo más lento, CNN, habría podido hacer 48 predicciones por evento. Como máximo, pudo haber perdido 7 de las 384 muestras que conforman cada evento. El clasificador más rápido, CSP, habría podido hacer 361 predicciones; DNN 165, y RMDM 303 predicciones. Esto ilustra sencillamente que cualquiera de los clasificadores presentados aquí es viable como una opción de BMI, con la predicción, prácticamente, en tiempo real.

### 3.3 Evaluación de los Clasificadores

La mejor clasificación se obtuvo con RMDM, tanto con datos públicos como privados. Particularmente, en el caso de línea base contra MI, la clasificación alcanzó valores superiores al 97%. Para el caso de RMI contra BL con datos de laboratorio en entrenamiento por sujeto, se alcanzó un 100%. Esto significa que fue capaz de detectar la MI en cada sujeto y en cada evento.

CSP dio resultados similares con 99.8% en la misma categoría, y con más del 95% en MI contra BL. Se debe considerar que recibieron datos solo de 30 personas, por lo que posiblemente sean modelos sobreajustados, y se requiere trabajo futuro para probarlos con nuevos datos. Desafortunadamente, en la clasificación de MI izquierdo contra MI derecho, no fueron capaces de efectuar una predicción discriminante. Los valores fueron tan bajos que no fueron mejores que si solo adivinaran. Solo en el caso público global, RMDM dio un aceptable 61%. Esto sugiere la necesidad de una cantidad más grande de eventos para mejorar el resultado. Sin embargo, como se mencionó anteriormente, la similitud entre los resultados con los datos públicos y privados muestra la consistencia del algoritmo trabajando tanto con grandes cantidades de datos como con pocos. Incluso en el caso multiclase, tanto CSP como RMDM dieron buenos resultados, entre el 63% y el 76% después del entrenamiento global. DNN y CNN dieron resultados similarmente conservadores. Aun así, produjeron mejores resultados clasificando LMI contra RMI que los 2 clasificadores anteriores. Aunque con los datos de laboratorio obtuvieron una exactitud del 50%, con los datos públicos mostraron una mejora alcanzando hasta el 70%. Especialmente la DNN, consistente con la aptitud de del aprendizaje profundo para manejar grandes cantidades de datos, con la ventaja adicional de que no requiere prácticamente ningún preprocesamiento, puede trabajar con datos crudos directamente. Las Figuras 3.1 y 3.2 muestran que ambos tuvieron una deferencia significativa en los resultados entre los datos públicos y de laboratorio con el entrenamiento global, pero en el entrenamiento por sujeto no hubo diferencia sustancial. Esto pudo deberse a que la cantidad de datos de un solo usuario no fue lo suficientemente extensa para explotar los beneficios de las redes neuronales profundas. Asimismo, ambos obtuvieron resultados satisfactorios en la clasificación de MI contra línea base, y CNN alcanzó 98% de exactitud en el caso público global.

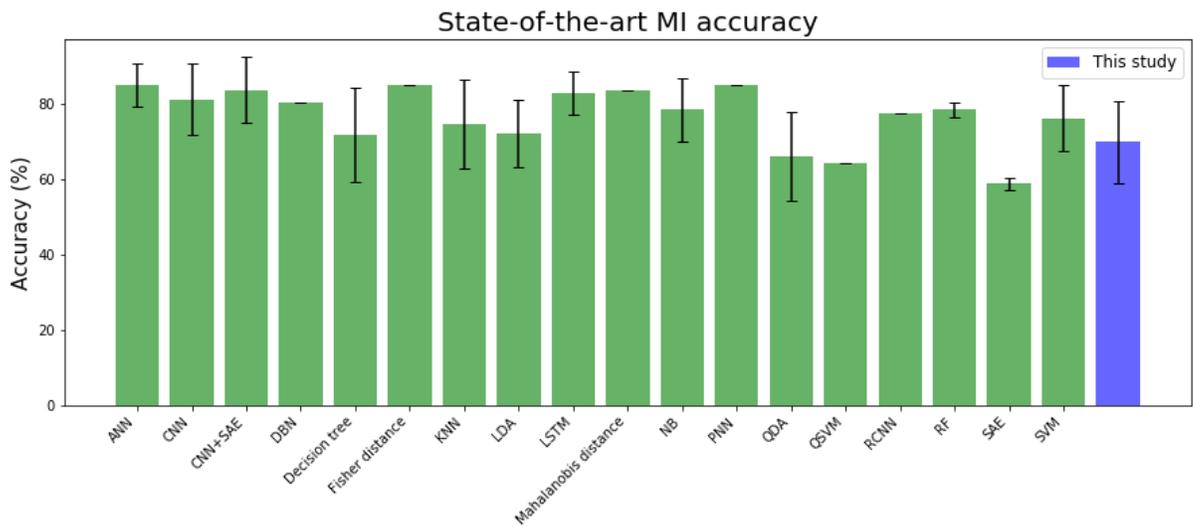
### 3.4 Comparación de Resultados con el Estado del Arte

En la literatura, los clasificadores de MI presentan exactitudes superiores al 60%. (Stock & Balbinot, 2016; Fakhruzzaman, Riksakomara, & Suryotrisongko, 2015; Amarasinghe, Wijayasekara, & Manic, 2014; Fatmawati, Wijaya, & Prawito, 2017; Wang, y otros, 2017; Kolodziej, Majkowski, Zapala, Rak, & Francuz, 2018; Xiao, Mu, & Hu, 2009; Elstob & Secco, 2016; Hamedi, Salleh, Noor, & Mohammad-Rezazadeh, 2014; Tabar & Halici, 2016; (Chatterjee & Bandyopadhyay, 2016; An, Kuang, Guo, Zhao, & He, 2014; Tyagi & Nehra, 2016; Bhattacharyya, Hossain, Konar, Tibarewala, & Ramadoss, 2014; Huang & Wu, 2010; Mohammadpour, Ghorbanian, & Mozaffari, 2016; Bentlemsan, Zemouri, Bouchaffra, Yahya-Zoubir, & Ferroudji, 2014; Bhaduri, Khasnobish, Bose, & Tibarewala, 2016; Shukla & Chaurasiya, 2018; Ko, Lin, Song, & Komarov, 2014; Tian & Liu, 2018; Lee & Choi, 2018; Dose, Müller, Iversen, & Puthusserypady, 2018; Chaudhary, Taran, Bajaj, & Sengur, 2019; Tayeb, y otros, 2019; Yang, Yao, & Wang, 2018)

La mayoría usan datos de 2 a 10 sujetos, con EEGs de 64 o hasta 128 canales, como se muestra en la Figura 5. Para estudios que usaron Emotiv como EEG, la exactitud resultó aun mayor (por encima del 80%), como en (Stock & Balbinot, 2016; Fakhruzzaman, Riksakomara, & Suryotrisongko, 2015; Amarasinghe, Wijayasekara, & Manic, 2014; Fatmawati, Wijaya, & Prawito, 2017; Elstob & Secco, 2016). Sin embargo, en nuestro caso, la exactitud promedio del estudio entero quedó alrededor de 70%. No obstante, si se considera solo los datos públicos, como en la mayoría de los estudios alegados, la exactitud estaría alrededor del 81%.

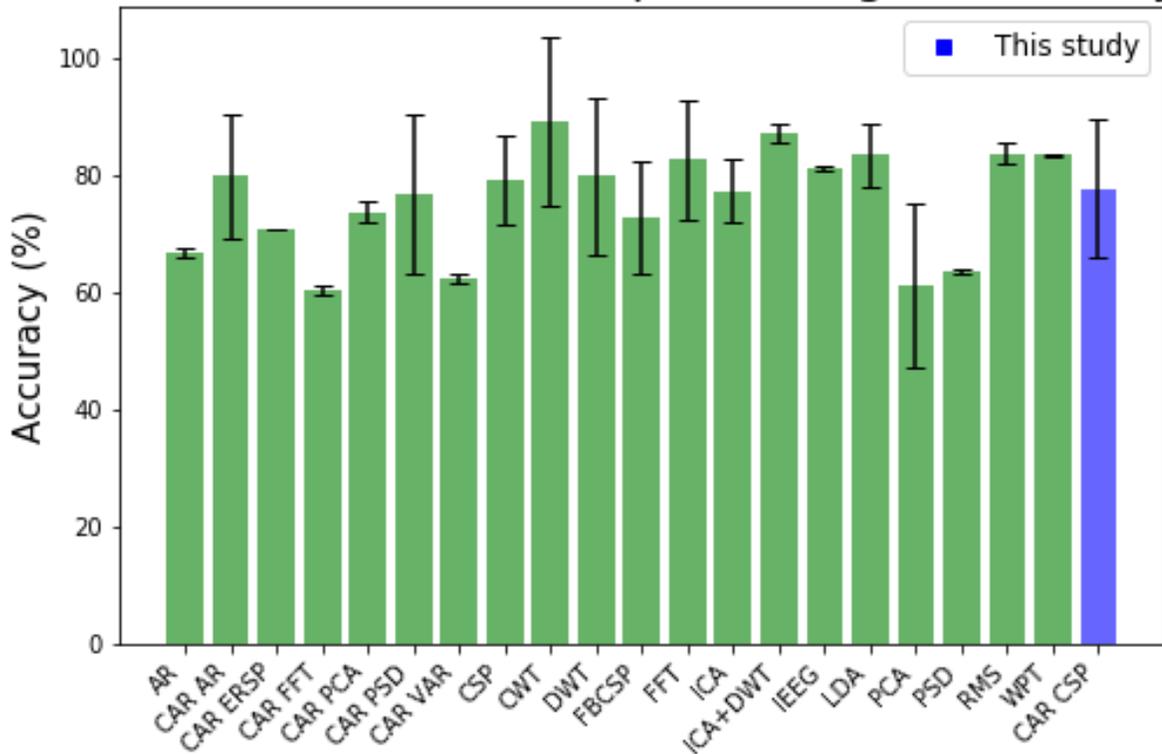
En cuanto al uso de CSP, la literatura coloca la barra de exactitud cerca del 80% (Stock & Balbinot, 2016; Fakhruzzaman, Riksakomara, & Suryotrisongko, 2015; Elstob & Secco, 2016; Mohammadpour, Ghorbanian, & Mozaffari, 2016; Bentlemsan, Zemouri, Bouchaffra, Yahya-Zoubir, & Ferroudji, 2014) y al de LDA alrededor del 72% (Aguiar, Yanez, & Benitez, 2016; Kolodziej, Majkowski, Zapala, Rak, & Francuz, 2018; Elstob & Secco, 2016; Ko, Lin, Song, &

Komarov, 2014; Yang, Yao, & Wang, 2018), como se muestra en la Figura 3.3 y en la Figura 3.4. En nuestro estudio, la exactitud mejoró en el caso de MI contra línea base, siendo mayor al 97%. Es interesante notar que el umbral de exactitud en los datos de laboratorio estuvo en 99.7%, 2% más que con los datos públicos. Pero la consistencia en la obtención de exactitudes altas en ambos casos confirmó la utilidad de este arreglo clasificador y su independencia de grandes cantidades de datos para funcionar bien.



**Figura 3.3 Clasificadores de MI vs. Exactitud.** Esta es una representación de la exactitud media obtenida por clasificadores de MI en la literatura. La barra azul de la derecha representa la exactitud promedio obtenida en este estudio considerando los 4 algoritmos, los 4 grupos de clasificación, con entrenamiento global. Fuente: Elaboración propia basado en datos de (Stock & Balbinot, 2016; Fakhruzzaman, Riksakomara, & Suryotrisongko, 2015; Amarasinghe, Wijayasekara, & Manic, 2014; Fatmawati, Wijaya, & Prawito, 2017; Wang, y otros, 2017; Kolodziej, Majkowski, Zapala, Rak, & Francuz, 2018; Xiao, Mu, & Hu, 2009; Elstob & Secco, 2016; Hamed, Salleh, Noor, & Mohammad-Rezazadeh, 2014; Tabar & Halici, 2016; (Chatterjee & Bandyopadhyay, 2016; An, Kuang, Guo, Zhao, & He, 2014; Tyagi & Nehra, 2016; Bhattacharyya, Hossain, Konar, Tibarewala, & Ramadoss, 2014; Huang & Wu, 2010; Mohammadpour, Ghorbanian, & Mozaffari, 2016; Bentlemsan, Zemouri, Bouchaffra, Yahya-Zoubir, & Ferroudji, 2014; Bhaduri, Khasnobish, Bose, & Tibarewala, 2016; Shukla & Chaurasiya, 2018; Ko, Lin, Song, & Komarov, 2014; Tian & Liu, 2018; Lee & Choi, 2018; Dose, Müller, Iversen, & Puthusserypady, 2018; Chaudhary, Taran, Bajaj, & Sengur, 2019; Tayeb, y otros, 2019; Yang, Yao, & Wang, 2018)

## State-of-the-art MI Preprocessing vs accuracy



**Figura 3.4. Algoritmos de procesamiento de MI vs. Exactitud.** Aquí, se comparan los resultados hallados en la literatura, de la clasificación de MI según el preprocesamiento aplicado a los datos, independientemente del clasificador que hayan implementado. La barra de la derecha muestra los valores obtenidos en este estudio cuando los datos fueron preprocesados con CAR y CSP. Fuente: Elaboración propia basada en (Stock & Balbinot, 2016; Fakhruzzaman, Riksakomara, & Suryotrisongko, 2015; Amarasinghe, Wijayasekara, & Manic, 2014; Fatmawati, Wijaya, & Prawito, 2017; Wang, y otros, 2017; Kolodziej, Majkowski, Zapala, Rak, & Francuz, 2018; Xiao, Mu, & Hu, 2009; Elstob & Secco, 2016; Hamed, Salleh, Noor, & Mohammad-Rezazadeh, 2014; Tabar & Halici, 2016; (Chatterjee & Bandyopadhyay, 2016; An, Kuang, Guo, Zhao, & He, 2014; Tyagi & Nehra, 2016; Bhattacharyya, Hossain, Konar, Tibarewala, & Ramadoss, 2014; Huang & Wu, 2010; Mohammadpour, Ghorbanian, & Mozaffari, 2016; Bentlemsan, Zemouri, Bouchaffra, Yahya-Zoubir, & Ferroudji, 2014; Bhaduri, Khasnobish, Bose, & Tibarewala, 2016; Shukla & Chaurasiya, 2018; Ko, Lin, Song, & Komarov, 2014; Tian & Liu, 2018; Lee & Choi, 2018; Dose, Müller, Iversen, & Puthusserypady, 2018; Chaudhary, Taran, Bajaj, & Sengur, 2019; Tayeb, y otros, 2019; Yang, Yao, & Wang, 2018)

Pasando al tema de redes neuronales, se puede apreciar su auge en la literatura por la gran cantidad de estudios implementando esta tecnología, y sorprendentemente, obteniendo buenos resultados, con exactitudes que rondan el 80% incluso en la clasificación de 2 tipos diferentes de MI, y cerca del 60% para el caso multiclase. (Amarasinghe, Wijayasekara, & Manic, 2014; Fatmawati, Wijaya, & Prawito, 2017; Hamed, Salleh, Noor, & Mohammad-Rezazadeh, 2014; Tabar & Halici, 2016; Chatterjee & Bandyopadhyay, 2016; An, Kuang, Guo, Zhao, & He,

2014; Tyagi & Nehra, 2016) Considerando solo el caso de datos públicos con entrenamiento global para hacer una comparación más justa, en este trabajo se obtuvo una exactitud media del 81%.

### 3.5 Limitaciones

Dado que CSP busca maximizar la varianza entre clases, se puede concluir que no hubo variabilidad suficientemente distintiva entre las señales de MI, o por lo menos no la suficiente distinción para ser separada linealmente por LDA. Esto puede atribuirse a la falta de canales de EEG sensitivos a MI en Emotiv, tales como C3 y C4. Pero sería inmediatamente refutado al contrastarlo con los resultados obtenidos con los datos de PhysioNet, en los que el EEG sí presentó los electrodos de la zona central, incluyendo C1, C2, C5 y C6.

La similitud en los resultados de CSP y RMDM pueden explicarse considerando que ambos métodos están basados en hallar las matrices de covarianza de cada clase. Para alcanzar alguna mejora, es necesario garantizar una alta variabilidad entre clases. Esto puede realizarse a través del método de obtención de las propias matrices de covarianza, o incrementando la resolución espacial el equipo, o usando una referenciación de EEG distinta acorde a la relación espacial entre las clases. Uno pudiera pensar en separar los electrodos, usando solo aquellos que se encuentran en el lado izquierdo de la cabeza durante las tareas de MI izquierda, y viceversa. Pero el reto sería entonces traducir los datos de manera que la interfaz sea capaz de incluir tal separación desde la implementación.

La comprensión actual sobre el funcionamiento interno de las redes neuronales no nos permite aún definir con precisión qué implica cada filtro y/o kernel. En cuanto a reconocimiento de imágenes, hay una analogía lineal a la que acercarse: pensando tales pesos como filtros que detectan ángulos y formas. Pero en cuanto a señales de EEG, no existe una analogía directa que se asemeje. En cierto sentido, se ve a las redes neuronales como cajas negras cuyo ajuste de hiperparámetros conlleva un proceso artesanal, afinándolos hasta lograr la arquitectura que

genere el modelo correcto. Para esperar un desempeño satisfactorio en EEG usando redes neuronales, con poco o nulo preprocesamiento, se requieren grandes cantidades de datos.

### **3.6 Trabajo Futuro**

En general, el entrenamiento global produjo exactitudes más altas. Esto muestra un camino viable para el desarrollo de BMI's basadas en DL. Esto es, realizar entrenamiento global para inicializar la red clasificadora, pero luego calibrarla mediante transferencia de conocimiento con datos del sujeto en cuestión para adaptar la interfaz. Puede hacerse un submuestreo de los datos públicos para que su forma coincida con la de los datos de laboratorio (14 canales y frecuencia de muestreo de 128 Hz), y entonces entrenar al algoritmo con la suma de ambos conjuntos de datos como entrada. Debe explorarse también el uso de equipos de EEG diseñados especialmente para BMI con pocos electrodos (alrededor de 4) posicionados en regiones estratégicas que garanticen la correcta resolución espacial para tareas de MI. Los resultados más prometedores fueron obtenidos con RMDM y CSP así que al ensamblarlos, tomando los CSP como las matrices de covarianzas de entrada para RMDM, sería posible incrementar la exactitud y tomaría lo mejor de cada algoritmo. Pero se necesitan más datos, idealmente tomados de diferentes voluntarios, para verificar la capacidad de los algoritmos y reducir el sobreajuste en estos clasificadores.

### **3.7 Conclusiones**

La exactitud media general fue mejor para el caso con datos públicos y entrenamiento global. Sin embargo, el entrenamiento por sujeto dio resultados prometedores en la mayoría de los casos, aunque la alta variabilidad entre los sujetos incrementó el error significativamente. Con respecto al caso de los datos de laboratorio, los resultados con CNN no fueron óptimos, pero con DNN fueron aceptables. En contraste, los resultados con CSP y RMDM fueron excelentes y muestran la viabilidad de su implementación en una BMI.

Es importante considerar el nivel mínimo de la señal que Emotiv permite es de 8400  $\mu\text{V}(\text{pp})$ , de modo que su ruido base estará cerca de 8 mV; por otro lado, el ruido base de una tarjeta de adquisición de datos típica es de alrededor de 1 mV. Entonces, la SNR de los datos de PhysioNet debe ser mayor que la de Emotiv. Esto tiene un impacto en la extracción de características y, en consecuencia, en la clasificación de las señales obtenidas.

# Índice de Figuras

Figura 1.1. Anatomía de una neurona.....	9
Figura 1.2. Polarización durante el potencial de acción.....	11
Figura 1.3. Algunas morfologías neuronales.....	12
Figura 1.4. Sistema nervioso.....	14
Figura 1.5. Lóbulos cerebrales. ....	16
Figura 1.6. Funciones asociadas a las regiones corticales.. ....	17
Figura 1.7. Mapa somatotópico motor de M1.....	18
Figura 1.8. Origen del potencial eléctrico detectado por EEG.....	20
Figura 1.9. Ondas cerebrales normales en un adulto.....	22
Figura 1.10. Sistema internacional 10-20.. ....	24
Figura 1.11. Perceptrón.. ....	30
Figura 1.12. La linealidad de la compuerta lógica AND.....	31
Figura 1.13. Regla de aprendizaje del perceptrón.. ....	32
Figura 1.14. La no linealidad de la compuerta lógica XOR.. ....	33
Figura 1.15. Efecto de la tasa de aprendizaje.....	36
Figura 1.16. Clasificación con regresión polinomial.. ....	37
Figura 1.17. Sobreajuste y subajuste.. ....	38
Figura 1.18. Función sigmoide.....	39
Figura 1.19. Clasificación con SVM.. ....	40
Figura 1.20. Función kernel para hacer datos linealmente separables.....	40
Figura 1.21. Clasificación kNN para distintas k.. ....	43
Figura 1.22. Teselaciones de Voronoi.. ....	44
Figura 1.23. Red neuronal artificial.....	49
Figura 1.24. Correlación cruzada.. ....	53
Figura 1.25. Diferencia entre max pooling y average pooling.....	54
Figura 1.26. Arquitectura clásica de una BMI.. ....	55
Figura 2.1. Emotiv EPOC+ y sus electrodos.....	60
Figura 2.2. Estructura de un ciclo derecho.....	61
Figura 2.3. Arquitectura de la CNN biclase.....	65
Figura 3.1. Exactitud vs. clases por clasificador y por tipo de datos.....	68
Figura 3.2. Resultados por tipo de entrenamiento.....	68
Figura 3.3. Clasificadores de MI vs. Exactitud.....	73
Figura 3.4. Algoritmos de preprocesamiento de MI vs. Exactitud.....	74

# Índice de Tablas

Tabla 1.1 <i>Tabla de verdad XOR</i> .....	33
Tabla 3.1. <i>Resumen de resultados</i> .....	67
Tabla 3.2. <i>Tiempo de cómputo</i> .....	70

# Bibliografía

- Acuña, K. (2017). Procesamiento de señales electroencefalográficas en un sistema embebido para una interfaz cerebro máquina. *Tesis*. Pontificia Universidad Católica del Perú.
- Aguiar, S., Yanez, W., & Benitez, D. (2016). Low complexity approach for controlling a robotic arm using the Emotiv EPOC headset. *2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. doi:10.1109/ropec.2016.7830526
- Amarasinghe, K., Wijayasekara, D., & Manic, M. (2014). EEG based brain activity monitoring using Artificial Neural Networks. *2014 7th International Conference on Human System Interactions (HSI)*, 61-66. doi:10.1109/hsi.2014.6860449
- An, X., Kuang, D., Guo, X., Zhao, Y., & He, L. (2014). A Deep Learning Method for Classification of EEG Data Based on Motor Imagery. *Lecture Notes in Computer Science*, 203-210. doi:10.1007/978-3-319-09330-7\_25
- Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2010). Riemannian geometry applied to BCI classification. *LVA/ICA 2010 - 9th International Conference on Latent Variable Analysis and Signal Separation*, 1-8. doi:10.1007/978-3-642-15995-4\_78
- Barachant, A., Bonnet, S., Congedo, M., & Jutten, C. (2012). Multiclass Brain-Computer Interface Classification by Riemannian Geometry. *IEEE Transactions on Biomedical Engineering, Institute of Electrical and Electronics Engineers*, 59(4), 920-928. doi:10.1109/TBME.2011.2172210
- Bear, M., Connors, B., & Paradiso, M. (2016). *Neuroscience: Exploring the brain*. Wolters Kluwer.

- Bentlemsan, M., Zemouri, E., Bouchaffra, D., Yahya-Zoubir, B., & Ferroudji, K. (2014). Random Forest and Filter Bank Common Spatial Patterns for EEG-Based Motor Imagery Classification. *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*. doi:10.1109/isms.2014.46
- Bhaduri, S., Khasnobish, A., Bose, R., & Tibarewala, D. N. (2016). Classification of lower limb motor imagery using K Nearest Neighbor and Naïve-Bayesian classifier. *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*. doi:10.1109/rait.2016.7507952
- Bhattacharyya, S., Hossain, M. A., Konar, A., Tibarewala, D. N., & Ramadoss, J. (2014). Detection of Fast and Slow Hand Movements from Motor Imagery EEG Signals. *Smart Innovation, Systems and Technologies Advanced Computing, Networking and Informatics*, 1, 645-652. doi:10.1007/978-3-319-07353-8\_74
- Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., & Muller, K. (2008). Optimizing Spatial filters for Robust EEG Single-Trial Analysis. *IEEE Signal Processing Magazine*, 25(1), 41-56. doi:10.1109/msp.2008.4408441
- BPOGR. (2018). *Biblioteca de Publicaciones Oficiales del Gobierno de la República*. Obtenido de Diagnóstico sobre la situación de las personas con discapacidad en México: <https://www.gob.mx/publicaciones/articulos/diagnostico-sobre-la-situacion-de-las-perso>
- Brunner, C., Leeb, R., Müller-Putz, G., Schlögl, A., & Pfurtscheller, G. (2008). BCI Competition 2008 – Graz data set A. *BCI Competition IV*.
- Brunner, C., Leeb, R., Müller-Putz, G., Schlögl, A., & Pfurtscheller, G. (2008). BCI Competition 2008 – Graz data set A. *BCI Competition IV*.

- Chatterjee, R., & Bandyopadhyay, T. (2016). EEG Based Motor Imagery Classification Using SVM and MLP. *2016 2nd International Conference on Computational Intelligence and Networks (CINE)*, 84-89. doi:10.1109/cine.2016.22
- Chaudhary, S., Taran, S., Bajaj, V., & Sengur, A. (2019). Convolutional Neural Network Based Approach Towards Motor Imagery Tasks EEG Signals Classification. *IEEE Sensors Journal*, 19(12), 1-7. doi:10.1109/jsen.2019.2899645
- Clerc, M., Bougrain, L., & Lotte, F. (2016). *Brain-Computer Interfaces 1: Foundations and Methods*. New York: Wiley.
- Clerc, M., Bougrain, L., & Lotte, F. (2016). *Brain-Computer Interfaces 1: Foundations and Methods*. New York: Wiley.
- CONADIS. (2018). *Consejo Nacional para el Desarrollo y la Inclusión de las Personas con Discapacidad*. Obtenido de La Esclerosis Lateral Amiotrófica ELA: <https://www.gob.mx/conadis/articulos/la-esclerosis-lateral-amiotrofica-ela>
- Congedo, M., Barachant, A., & Bhatia, R. (2017). Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces*, 4(3), 155-174. doi:doi: 10.1080/2326263X.2017.1297192
- Dose, H., Müller, J. S., Iversen, H. K., & Puthusserypady, S. (2018). An end-to-end deep learning approach to MI-EEG signal classification for BCIs. *Expert Systems with Applications*, 114, 532-542. doi:10.1016/j.eswa.2018.08.031
- Duda, R. O., Stork, D. G., & Hart, P. E. (2001). *Pattern classification* (Segunda ed.). John Wiley & Sons.

- Elstob, D., & Secco, E. (2016). A Low Cost Eeg Based Bci Prosthetic Using Motor Imagery. *International Journal of Information Technology Convergence and Services*, 6(1), 23-36. doi:10.5121/ijitcs.2016.6103
- Ertel, W. (2017). *Introduction to Artificial Intelligence*. Germany: Springer.
- Fakhruzzaman, M., Riksakomara, E., & Suryotrisongko, H. (2015). EEG Wave Identification in Human Brain with Emotiv EPOC for Motor Imagery. *Procedia Computer Science*, 72, 269-276. doi:10.1016/j.procs.2015.12.140
- Fatmawati, E., Wijaya, S. K., & Prawito. (2017). Development Prototype System of Arms Motor Imagery Utilizing Electroencephalography Signals (EEG) from Emotiv with Probabilistic Neural Network (PNN) as Signal Analysis. *2017 5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, 179-183. doi:10.1109/icici-bme.2017.8537727
- Gallistel, C., & Matzel, L. (2013). The neuroscience of learning: beyond the Hebbian synapse. *Annu Rev Psychol*, 64, 169-200. doi:10.1146/annurev-psych-113011-143807
- Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23), e215-e220. doi:doi:10.1161/01.cir.101.23.e215
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press. Obtenido de <https://www.deeplearningbook.org/>
- Graimann, B., Allison, B. Z., & Pfurtscheller, G. (2013). *Brain-Computer Interfaces Revolutionizing Human-Computer Interaction*. Berlin: Springer Berlin.
- Guyton, A., & Hall, J. (2006). *Textbook of medical physiology* (11 ed.). Elsevier.

- Hamed, M., Salleh, S.-H., Noor, A. M., & Mohammad-Rezazadeh, I. (2014). Neural network-based three-class motor imagery classification using time-domain features for BCI applications. *2014 IEEE REGION 10 SYMPOSIUM*, 204-207. doi:10.1109/tenconspring.2014.6863026
- Herbert, G., & Duffau, H. (2020). Revisiting the Functional Anatomy of the Human Brain: Toward a Meta-Networking Theory of Cerebral Functions. *Physiological Reviews*, *100*(3), 1181-1228. doi:10.1152/physrev.00033.2019
- Hossan, A., & Chowdhury, M. (2016). Real Time EEG Based Automatic Brainwave Regulation by Music. *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, 780-784. doi:10.1109/ICIEV.2016.7760107
- Huang, S., & Wu, X. (2010). Feature extraction and classification of EEG for imagery movement based on mu/beta rhythms. *2010 3rd International Conference on Biomedical Engineering and Informatics*, 891-894. doi:10.1109/bmei.2010.5639888
- Igual, L., & Seguí, S. (2017). *Introduction to Data Science: A python approach to concepts, techniques and applications*. Barcelona: Springer.
- INEGI. (2020). *Instituto Nacional de Estadística y Geografía*. Obtenido de Población con limitación o discapacidad por entidad federativa y tipo de actividad que realiza o condición mental según sexo, 2020: <https://www.inegi.org.mx/app/tabulados/interactivos>
- Jackson, A., & Bolger, D. (2014). The neurophysiological bases of EEG and EEG measurement: A review for the rest of us. *Psychophysiology*, *51*, 1061-1071. doi:10.1111/psyp.12283
- Knight, K., Rich, E., & Nair, S. (2010). *Artificial Intelligence*. Tata McGraw-Hill Education.

- Ko, L., Lin, S., Song, M., & Komarov, O. (2014). Developing a few-channel hybrid BCI system by using motor imagery with SSVEP assist. *2014 International Joint Conference on Neural Networks (IJCNN)*, 4114-4120. doi:10.1109/ijcnn.2014.6889901
- Kolodziej, M., Majkowski, A., Zapala, D., Rak, R. J., & Francuz, P. (2018). Methods of Power-Band Extraction Techniques for BCI Classification. *19th International Conference Computational Problems of Electrical Engineering*. doi:10.1109/cpee.2018.8506786
- Kubat, M. (2017). *An Introduction to Machine Learning*. Florida: Springer.
- Lee, H. K., & Choi, Y. (2018). A Convolution Neural Networks Scheme for Classification of Motor Imagery EEG based on Wavelet Time-Frequency Image. *2018 International Conference on Information Networking (ICOIN)*, 906-909. doi:10.1109/icoin.2018.8343254
- Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., & Yger, F. (2018). A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15, 1-28.
- Mahmood, A., Zainab, R., Ahmad, R. B., Saeed, M., & Kamboh, A. M. (2017). Classification of multi-class motor imagery EEG using four band common spatial pattern. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. doi:10.1109/embc.2017.8037003
- Matich, D. J. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Rosario: Universidad Tecnológica Nacional.
- McCorduck, P. (2004). *Machines Who Think*. A K Peters.
- McCulloch, W., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 115–133. doi:10.1007/BF02478259

Minsky, M., & Papert, S. (1969). *Perceptrons. An Introduction to Computational Geometry*. MIT Press.

Mitchell, T. (1997). *Machine Learning*. McGraw-Hill.

Mizuguchi, N., Nakata, H., Uchida, Y., & Kanosue, K. (2012). Motor imagery and sport performance. *The Journal of Physical Fitness and Sports Medicine*, 1(1), 103-111. doi:doi:10.7600/jpfsm.1.103

Mohammadpour, M., Ghorbanian, M., & Mozaffari, S. (2016). Comparison of EEG signal features and ensemble learning methods for motor imagery classification. *2016 Eighth International Conference on Information and Knowledge Technology (IKT)*. doi:10.1109/ikt.2016.7777767

*Molecular Devices*. (s.f.). Obtenido de What is an action potential?: <https://www.moleculardevices.com/applications/patch-clamp-electrophysiology/what-action-potential>

Nielsen, M. (2019). *Neural Networks and Deep Learning*. Obtenido de <http://neuralnetworksanddeeplearning.com/>

Nunez, P., & Srinivasan, R. (2005). *Electric Fields of the Brain*. Oxford University Press.

Olías, J. (2016). Estudio del método Common Spatial Patterns y sus variantes en interfaces cerebro-ordenador. *Tesis*. Escuela Técnica Superior de Ingeniería, Universidad de Sevilla.

OMS. (2011). *Organización Mundial de la Salud*. Obtenido de Informe Mundial Sobre la Discapacidad. Resumen: [http://apps.who.int/iris/bitstream/handle/10665/70672/WHO\\_NMH\\_VIP\\_11.03\\_spa.pdf?sequence=1](http://apps.who.int/iris/bitstream/handle/10665/70672/WHO_NMH_VIP_11.03_spa.pdf?sequence=1)

- Penfield, W., & Boldrey, E. (1937). Somatic Motor And Sensory Representation In The Cerebral Cortex Of Man As Studied By Electrical Stimulation. *Brain*, 389-443. doi:10.1093/brain/60.4.389
- Purves, D., Augustine, G., Fitzpatrick, D., Hall, W., Lamantia, A., Mcnamara, J., & Williams, S. (2007). *Neurociencia* (3 ed.). Panamericana Editorial.
- Rojas, G., Alvarez, C., Montoya, C., de la Iglesia-Vayá, M., Cisternas, J., & Gálvez, M. (2018). Study of Resting-State Functional Connectivity Networks Using EEG Electrodes Position As Seed. *Frontiers in neuroscience*, 12(235), 1-12. doi:10.3389/fnins.2018.00235
- Rosenblatt, F. (1957). The Perceptron. A Perceiving and Recognizing Automaton. *Cornell Aeronautical Laboratory*.
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408. doi:10.1037/h0042519
- Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning Representations by Back-Propagating Errors. *Nature*, 323(9), 533-536.
- Russel, S., & Norvig, P. (2004). *Inteligencia Artificial: Un Enfoque Moderno*. Pearson Educación.
- Schalk, G., McFarland, D., Hinterberger, T., Birbaumer, N., & Wolpaw, J. (2004). BCI2000: A General-Purpose Brain-Computer Interface (BCI) System. *IEEE Transactions on Biomedical Engineering*, 51(6), 1034-1043. doi:doi:10.1109/tbme.2004.827072
- Schmidhuber, J. (2014). Deep Learning in Neural Networks: An Overview. *arXiv*, 1-88.
- Shukla, P. K., & Chaurasiya, R. K. (2018). An Experimental Analysis of Motor Imagery EEG Signals Using Feature Extraction and Classification Methodologies. *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. doi:10.1109/gucon.2018.8675032

- Skansi, S. (2018). *Introduction to Deep Learning*. Springer.
- Skiena, S. (2017). *The data science design manual*. New York: Springer.
- Stagg, C. (2014). The physiological basis of brain stimulation. En R. C. (Editor), *The Stimulated Brain* (págs. 145-177). Elsevier.
- Stock, V., & Balbinot, A. (2016). Movement imagery classification in EMOTIV cap based system by Naïve Bayes. *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. doi:10.1109/embc.2016.7591711
- Stufflebeam, R. (2006). *The Mindproject*. Obtenido de Neurons, Synapses, Action Potentials, and Neurotransmission: [https://mind.ilstu.edu/curriculum/neurons\\_intro/neurons\\_intro.html](https://mind.ilstu.edu/curriculum/neurons_intro/neurons_intro.html)
- Tabar, Y. R., & Halici, U. (2016). A novel deep learning approach for classification of EEG motor imagery signals. *Journal of Neural Engineering*, 14(1), 1-11. doi:10.1088/1741-2560/14/1/016003
- Tayeb, Z., Fedjaev, J., Ghaboosi, N., Richter, C., Everding, L., Qu, X., & Conradt, J. (2019). Validating Deep Neural Networks for Online Decoding of Motor Imagery Movements from EEG Signals. *Sensors*, 19(210), 1-17. doi:10.3390/s19010210
- Tian, G., & Liu, Y. (2018). Study on Classification of Left-Right Hands Motor Imagery EEG Signals Based on CNN. *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI\*CC)*, 324-329. doi:10.1109/icci-cc.2018.8482042
- Tschöepe, M. (2019). Beyond SGD: Recent improvements of Gradient Descent Methods. doi:10.13140/RG.2.2.34359.88484
- Turing, A. (1950). Computing Machinery and Intelligence. *Mind*, 59(236), 433-460. doi:10.1093/mind/LIX.236.433

- Tyagi, A., & Nehra, V. (2016). Classification of motor imagery EEG signals using SVM, k-NN and ANN. *CSI Transactions on ICT*, 4(2-4), 135-139. doi:10.1007/s40012-016-0091-2
- Wang, K., Wang, Z., Guo, Y., He, F., Qi, H., Xu, M., & Ming, D. (2017). An EEG study on hand force imagery for brain-computer interfaces. *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*. doi:10.1109/ner.2017.8008439
- Werbos, P. (1982). *Applications of Advances in Nonlinear Sensitivity Analysis*. Department of Energy.
- Wolpaw, J., Mcfarland, D., Vaughan, T., & Schalk, G. (2003). The Wadsworth Center Brain-Computer Interface (BCI) Research and Development Program. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2), 204-207. doi:doi:10.1109/tnsre.2003.814442
- Wu, Y., Huang, T. H., Lin, C. Y., Tsai, S. J., & Wang, P. (2018). Classification of EEG Motor Imagery Using Support Vector Machine and Convolutional Neural Network. *2018 International Automatic Control Conference (CACCS)*. doi:doi:10.1109/cacs.2018.8606765
- Xiao, D., Mu, Z., & Hu, J. (2009). Classification of Motor Imagery EEG Signals Based on Energy Entropy. *2009 International Symposium on Intelligent Ubiquitous Computing and Education*. doi:10.1109/iuce.2009.57
- Yang, J., Yao, S., & Wang, J. (2018). Deep Fusion Feature Learning Network for MI-EEG Classification. *IEEE Access*, 6, 79050-79059. doi:10.1109/access.2018.2877452
- Yani, M., Irawan, B., & Setiningsih, C. (2019). Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail. *Journal of Physics Conference Series*, 1201-012052.

