

# **Desarrollo de una interfaz cerebro-computadora multicanal para el reconocimiento de intenciones de movimiento mediante redes neuronales convolucionales**

**Rodrigo Ferrer, Geovanny Giorgana**

Universidad Anáhuac Mayab

Carr. Mérida-Progreso Km. 15.5, CP. 97310, Mérida, Yucatán, México

jrodrigo\_ferrer@hotmail.com; geovanny.giorgana@anahuac.mx

## **RESUMEN**

En este proyecto se implementó una red neuronal convolucional (CNN, por sus siglas en inglés) como etapa de clasificación de un sistema de interfaz cerebro computadora (BCI, por sus siglas en inglés) basado en un electroencefalograma (EEG) de 22 canales, capaz de distinguir entre cuatro intenciones de movimiento. Se utilizó la base de datos 2ª de la cuarta *BCI Competition* auspiciada por la Universidad de Graz, la cual se encuentra compuesta por los sets de señales obtenidas durante múltiples pruebas desempeñadas por nueve sujetos.

Con el fin de mejorar la compatibilidad de las señales con la CNN se utilizó el set de señales de entrenamiento para calcular la correlación de cada canal con el resto de canales, y en base a ello reordenar los canales para maximizar la correlación vertical.

Se estableció un modelo de CNN como referencia a partir del cual se realizaron múltiples pruebas modificando un único parámetro del modelo a la vez, esto con el fin de analizar sus efectos en el sistema y encontrar una mejor configuración. Los mejores parámetros de cada prueba se utilizaron para actualizar el modelo de referencia.

De la segunda prueba se obtuvieron resultados que llevaron a la conclusión de que había que separar las convoluciones sobre las señales de las de los canales, por lo que se realizaron pruebas posteriores para analizar los efectos de una capa de filtros sobre los canales (verticales) tras una serie de capas con filtros sobre las señales (horizontales).

Los resultados del proyecto muestran la capacidad de la CNN al clasificar las señales, alcanzando una exactitud de hasta un 54.7% y con un coeficiente kappa de 0.40.

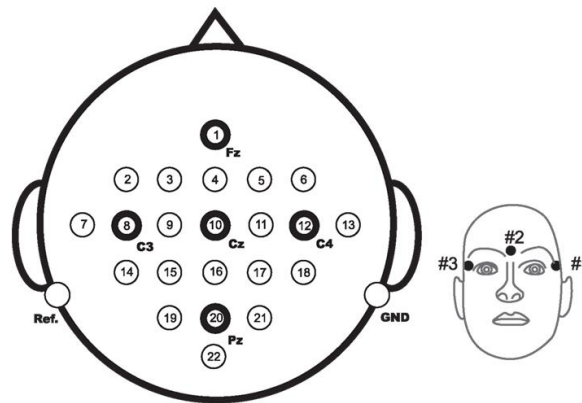
**PALABRAS CLAVE:** CNN, BCI, EEG, Intención de movimiento, Graz Competition

## **1 INTRODUCCIÓN**

Este proyecto desarrollo una interfaz cerebro-computadora capaz de procesar las señales eléctricas de ondas cerebrales provenientes de la base de datos 2ª de la Graz BCI Competition IV, y reconocer entre cuatro intenciones de movimiento (mano izquierda, derecha, ambos pies y lengua) mediante el uso de redes neuronales convolucionales. Se experimentó con distintos

parámetros como: tamaño y forma de los filtros, número de capas convolucionales, funciones de activación, etc.; y analizo los efectos de cada uno de estos sobre la exactitud del sistema, además de la viabilidad de la CNN en esta área de aplicación.

Una interfaz cerebro-computadora es un sistema que establece una comunicación indirecta entre un cerebro y un dispositivo externo. Al medir la actividad cerebral, e inferir la intención del usuario, es posible controlar un dispositivo sin necesidad de movimiento [1]. La base de datos 2ª contiene las señales obtenidas de nueve sujetos a través de una serie de pruebas utilizando un dispositivo EEG de 25 canales ordenados como se muestra en la Figura 1 [2].



**Figura 1.** Posición de los electrodos en el EEG junto con la numeración utilizada por Graz BCI para crear la base de datos.

Las redes neuronales artificiales son modelos que emulan la forma en la que el cerebro humano procesa la información a través de neuronas biológicas, para ello hace uso de la neurona artificial, la cual recibe valores numéricos ponderados independientemente a través de sus entradas, los suma junto con el *bias* y evalúa en una función de activación para comunicar el resultado con todas las neuronas de la capa siguiente.

Las redes neuronales convolucionales son un tipo de red neuronal artificial formada por una serie de capas de filtros y reducciones alternantes, seguidas de una red neuronal totalmente conectada [c].

La CNN se utiliza con el fin de: extraer de manera automática las características más relevantes tanto de las señales como de los canales a través de las convoluciones, y reducir el número muestras temporales para facilitar su clasificación entre las cuatro intenciones de movimiento mediante la red neuronal totalmente conectada.

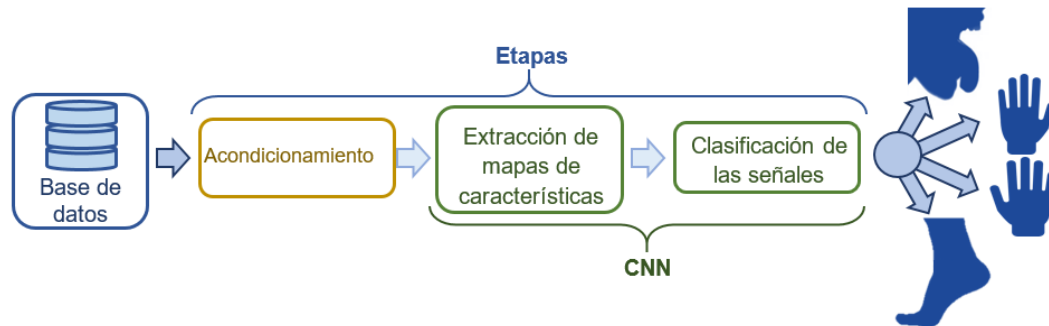
## 2 MATERIALES Y MÉTODOS

### 2.1 Visión general del método

El proyecto se realizó a través del lenguaje de programación Python en la plataforma Anaconda y utilizando la librería TensorFlow, siguiendo como referencia para la construcción de la CNN el algoritmo publicado por Admin [3].

La Figura 2 ilustra las etapas principales llevadas a cabo durante el proyecto para la correcta clasificación de las señales. Las señales de la base de datos 2ª se acondicionaron para mejorar su compatibilidad con la forma de extraer características de la CNN, posteriormente se realizaron

múltiples pruebas sobre un modelo de referencia para encontrar los parámetros que permitan extraer los mejores mapas de características y así poder clasificar las señales con una mayor exactitud.



**Figura 2.** Diagrama de la etapas que conforman el sistema desarrollado durante el proyecto.

## 2.2 Descripción de la base de datos

La base de datos 2ª contiene las señales de nueve sujetos, las cuales fueron capturadas durante seis sesiones de doce pruebas para cada una de las cuatro intenciones de movimiento, generando así una base con 2,592 sets de señales, algunos sets no fueron correctamente etiquetados por lo que se redujo a 2,281 sets. La base de datos fue dividida en un set de entrenamiento y uno de evaluación, con el 1823 y 458 de los sets respectivamente.

## 2.3 Acondicionamiento de las señales

Utilizando la base de entrenamiento se promedió todos los sets de señales dentro de ella, obteniendo una sola matriz de 22 canales y 1000 muestras temporales, la cual se utilizó junto con la función *np.corrcoef()* para calcular la correlación entre cada canal y el resto de los canales, estos coeficientes fueron guardados en una matriz de 22 filas y 22 columnas, la cual fue analizada con el objetivo de reordenar los canales, tanto en el set de entrenamiento como en el de evaluación, y así maximizar las correlación entre un canal y sus canales vecinos inmediatos, haciendo las señales más compatibles con la CNN.

## 2.4 Arquitectura del modelo de referencia

Los parámetros seleccionados para el modelo de CNN utilizados como referencia se encuentran en la Tabla 1, estos valores se seleccionaron a consideración personal tras haber hecho un par de pruebas. El modelo cuenta con cinco capas convolucionales con 64, 32, 16, 16 y 8 filtros de tamaño  $3 \times 3$ , junto con su respectiva capa de reducción por máximo usando una ventana de tamaño  $2 \times 1$ ; y una sola capa oculta de 1000 neuronas. Se utilizó la función de unidad lineal rectificada (ReLU) y *bias* tanto en capas convolucionales como ocultas.

**Tabla 1.** Parámetros de configuración para la CNN usada como modelo de referencia

Capas convolucionales					
Capa	Filtros			Reducción	
	Numero	Tamaño	Avance	Tamaño	Avance
1	64	[3,1]	[1,1]	[2,1]	[2,1]
2	32	[3,1]	[1,1]	[2,1]	[2,1]
3	16	[3,1]	[1,1]	[2,1]	[2,1]
4	16	[3,1]	[1,1]	[2,1]	[2,1]
5	8	[3,1]	[1,1]	[2,1]	[2,1]
Red completamente conectada					
Capas ocultas			Capas ocultas		
1			1000		

## 2.5 Pruebas realizadas

Se hicieron pruebas experimentando con distintos parámetros usando al modelo de referencia como punto de partida con el objetivo de analizar los efectos de cada parámetro en la CNN y crear un nuevo modelo con los mejores parámetros encontrados. Se probó modificando:

1. **El número de capas convolucionales:** de tres a siete capas, manteniendo 64 filtros en la primera capa, 32 en la segunda, 8 en la última y 16 en las capas intermedias.
2. **El tamaño de los filtros:** con tres y cinco muestras de las señales; y uno, tres y cinco canales.
3. **La función de activación:** con las funciones ReLU, Sigmoidal y Tanh
4. **El uso de *bias*:** en las capas convolucionales y en las capas ocultas.

De la segunda prueba se observó una oportunidad para seguir mejorando la exactitud del sistema por lo que se continuó experimentando con:

5. Filtros de 5, 10, 15, 20 y 25 muestras de las señales sobre un solo canal (horizontales).
6. Modificando la sexta capa a filtros de 3, 5, 7 y 9 canales sobre una sola muestra de las señales (verticales).
7. Agregando una séptima capa de filtros horizontales con 3, 5 y 10 muestras.

## 3 RESULTADOS

Se deberán enfatizar y describir claramente los resultados obtenidos durante la realización del trabajo. Trabajos por realizarse o sin resultados, al menos preliminares, no serán aceptados.

### 3.1 Metodología de los experimentos

Cada modelo de las pruebas fue entrenado al menos dos veces hasta alcanzar su punto de saturación, lo cual tomaba entre 20 a 40 epochs, en caso de que los resultado divergieran se realizaban dos pruebas más. En las tablas 3 a la 6 se presentan los resultados de las primeras pruebas, remarcando en negro el resultado del modelo de referencia presentado en la tabla 2, en verde el mejor resultado de la pruebas y en rojo cuando el mejor modelo de la prueba coincidiera con el de referencia.

En la tabla 7 se presentan los resultado del modelo generado a partir de los resultados obtenidos en las primera cuatro pruebas, este modelo se le denomino “modelo mejorado”.

En la tabla 8,9,y 10 se resalta en negro el resultado del mejor modelo de la prueba anterior a ella.

### 3.2 Resultados de los experimentos

**Tabla 2.** Promedio de mejores resultados del modelo de referencia descrito en la sección 2.4

Modelo base	
Coeficiente kappa	0.2248
Porcentaje de acierto (%)	42.5

**Tabla 3.** Mejores resultados obtenidos al modificar el número de capas convolucionales del modelo de referencia.

Número de capas convolucionales	3	4	5	6	7
Coeficiente kappa	0.1971	0.2255	<b>0.2248</b>	<b>0.2260</b>	0.2450
Porcentaje de acierto (%)	39.7	41.7	<b>42.5</b>	<b>41.5</b>	43.5
Epochs requeridas para alcanzar la saturación	17	18	<b>20</b>	<b>25</b>	40

**Tabla 4.** Mejores resultados obtenidos al modificar el tamaño de los filtros del modelo de referencia

Tamaños de los filtros	[3,1]	[5,1]	[3,3]	[5,3]	[3,5]	[5,5]
Coeficiente kappa	<b>0.2248</b>	<b>0.2320</b>	0.1335	0.1924	0.1924	0.2044
Porcentaje de acierto (%)	<b>42.5</b>	<b>42.6</b>	33.8	38.6	38.6	40.1

**Tabla 5.** Mejores resultados obtenidos al modificar la función de activación del modelo de referencia

Función	ReLu	Sigmoidal	Tanh
Coeficiente kappa	<b>0.2248</b>	0.0	0.1843
Porcentaje de acierto (%)	<b>42.5</b>	27.0	37

**Tabla 6.** Mejores resultados obtenidos al habilitar y deshabilitar las bias en el modelo de referencia

Usando <i>bias</i> en:	Capas convolucionales y ocultas	Solo en capas convolucionales	Solo en capas ocultas	En ninguna capa
Coeficiente kappa	<b>0.2248</b>	0.2248	0.2976	<b>0.2988</b>
Porcentaje de acierto (%)	<b>42.5</b>	43.1	47.0	<b>46.4</b>

**Tabla 7.** Promedio de los mejores resultados del modelo mejorado, en comparación con los del modelo de referencia.

Modelo	Modelo base	Modelo mejorado	Incremento (%)
Coeficiente kappa	0.2248	0.3077	36.9
Porcentaje de acierto (%)	42.5	48.1	13.2
Epochs requeridas para alcanzar la saturación	20	20	0.0

**Tabla 8.** Mejores resultados obtenidos al modificar el tamaño de los filtros en el modelo mejorado.

Tamaños de los filtros	[5,1]	[10,1]	[15,1]	[20,1]	[25,1]
Coeficiente kappa	<b>0.3077</b>	<b>0.3226</b>	0.2590	0.2700	0.2641
Porcentaje de acierto (%)	<b>48.1</b>	<b>48.8</b>	44.5	45.9	45.1

**Tabla 9.** Mejores resultados de la aplicación de una sexta capa convolucional, a lo largo de los canales.

Tamaños de los filtros	Sin capa	[1,3]	[1,5]	[1,7]	[1,9]
Coeficiente kappa	<b>0.3226</b>	0.3119	<b>0.3544</b>	0.3274	0.3307
Porcentaje de acierto (%)	<b>48.8</b>	48.6	<b>50.7</b>	49.5	50.1

**Tabla 10.** Mejores resultados de la aplicación de una sexta capa convolucional, a lo largo de los canales.

Tamaños de los filtros	Sin capa	[3,1]	[5,1]	[10,1]
Coeficiente kappa	<b>0.3544</b>	<b>0.3735</b>	0.3544	0.3656
Porcentaje de acierto (%)	<b>50.7</b>	<b>52.7</b>	51.5	52.5

**Tabla 11.** Tabla comparativa del mejor modelo obtenido, junto a los del estado del arte y los ganadores de la competencia.

Método usado	Coeficiente kappa	
Estado del arte		
Perez-Zapata et al. [5]	0.84	
Ko, W., Yoon, J. et al [6]	0.45	
Ganadores de la competencia		
Keng Ang, K.	0.57	
Guangquan, L.	0.52	
Nuestro modelo	Promedio	Máximo
	0.37	0.40
Song, W.	0.31	
Coyle, D.	0.30	
Wu, J.	0.29	

## 4 CONCLUSIONES

Los resultados obtenidos durante el proyecto revelan que existe una mejora en el desempeño al clasificar señales multicanal EEG utilizando métodos de aprendizaje profundo, específicamente CNN, la red fue capaz de superar al quinto, cuarto y tercer lugar de la BCI Graz Competition IV dataset 2<sup>a</sup>, quienes utilizaron múltiples etapas de limpieza y acondicionamiento de las señales, mientras que la CNN recibió únicamente las señales crudas correspondientes del segundo dos al cuatro de cada set de señales.

Cada prueba realizada mostró información relevante sobre los efectos de los hiperparámetros de la CNN y que pueden ser utilizados como referencia para futuros trabajos.

De estas se puede concluir que el uso de *bias* en las capas ocultas no produce efectos significativos al clasificar señales EEG, sin embargo el deshabilitar las *bias* en las capas convolucionales sí muestra una mejora importante en la precisión de la red.

Además, que el tratar de extraer información tanto de las señales como de los canales a la vez parece perjudicar el aprendizaje de la red, aunque si existen características relevantes entre los canales para la clasificación, por lo que deben realizarse por separado las convoluciones sobre las señales y los canales. Al tratar de extraer información de entre los canales como primera capa, se observó que el porcentaje de acierto disminuyó drásticamente, en comparación a cuando se realizó en las últimas capas, por lo que hay que priorizar la extracción de características en las señales.

De igual manera, que el aumentar el número de capas convolucionales causa, tras superar un determinado número de capas, un incremento significativo en el número de epochs requeridos para alcanzar la saturación en el aprendizaje del sistema, lo cual significa sesiones de entrenamiento más tardadas.

También se encontró que de las funciones de activación utilizadas sólo la función *ReLU* y la función *Tanh* permitían una correcta clasificación de las señales, aunque esta última lo hiciese con un desempeño significativamente menor. Estas dos funciones tienen la característica en común de estar centradas en el origen.

## REFERENCIAS

- [1] Graimann, B., Allison, B., & Pfurtscheller, G. (2009). Brain–computer interfaces: A gentle introduction. In *Brain-Computer Interfaces*. Springer Berlin Heidelberg. Pág. 1-27
- [2] Tangermann, T., Muller, K., et al (13 de Julio de 2012), Review of the BCI competition IV, *Frontiers Media S.A.*, 3 de Noviembre de 2018, recuperado de <https://www.frontiersin.org/articles/10.3389/fnins.2012.00055/full#h6>
- [3] Admin, (24 de Abril del 2017), Convolutional Neural Networks Tutorial in TensorFlow, *Adventures in Machine Learning*, recuperado de: <https://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-tensorflow/> el 25 de noviembre de 2018.