

Few-Shot Learning para Interfaz Cerebro-Máquina basada en imágenes motoras

Javier Pérez

Departamento de Ingeniería Biomédica
Universidad de los Andes
Bogotá, Colombia
je.perezl@uniandes.edu.co

Christian Picón

Departamento de Ingeniería Biomédica
Universidad de los Andes
Bogotá, Colombia
c.picon@uniandes.edu.co

I. INTRODUCCIÓN

Una Interfaz Cerebro-Máquina (Brain-Computer Interface, BCI) es un sistema que traduce los patrones de la actividad cerebral de un usuario en comandos que se pueden emplear para controlar dispositivos o aplicaciones [1]. Actualmente, para registrar la actividad cerebral de forma no invasiva se tienen técnicas como la electroencefalografía (EEG), la magnetoencefalografía (MEG) o la imagen por resonancia magnética funcional (fMRI). Una BCI basada en imágenes motoras utiliza la actividad inducida en la corteza motora que un usuario realiza mediante la imaginación del movimiento motor sin producir ningún movimiento de las extremidades ni recibir estímulo externo alguno [2]. Por otro lado, en un sistema BCI se requieren dos fases. La fase de entrenamiento, en la que se enfoca esta investigación buscando hacer esta fase más corta, corresponde a la calibración del sistema [1]. La segunda fase es la fase operacional donde el sistema puede reconocer los patrones de la actividad cerebral para traducirlos en comandos para la computadora.

En la fase de entrenamiento, se han aplicado varias técnicas de Machine Learning (ML) que han permitido obtener algoritmos de clasificación más robustos. Estos clasificadores se pueden dividir en cuatro categorías principales: clasificadores adaptativos, clasificadores de matrices y tensores, clasificadores basados en aprendizaje por transferencia y aprendizaje profundo, y otros clasificadores misceláneos [1]. Sin embargo, en la mayoría de estos clasificadores es necesario obtener una gran cantidad de datos para realizar un entrenamiento adecuado. En especial, para los clasificadores de aprendizaje profundo es necesario contar con un gran conjunto de datos que nos permita entrenar este modelo complejo.

Para solucionar esta problemática, proponemos recurrir a técnicas de ML conocidas como Few-Shot Learning (FSL). FSL consiste en el uso de la experiencia previa obtenida mediante el entrenamiento en un conjunto de datos grande para realizar tareas específicas en dominios de los cuales hay pocos datos disponibles [3]. Para ello, FSL aborda este objetivo desde tres diferentes metodologías. La primera consiste en generar nuevos datos a partir de los ya obtenidos. Este aspecto se conoce como *data augmentation*. Segundo, el enfoque por métodos de FSL, el cual busca reducir las posibles hipótesis

de parámetros para ajustar los datos. Tercero, también se busca optimizar la búsqueda de los parámetros de la hipótesis correcta; estos últimos dos se basan en el uso del conocimiento previo. Con ello se han propuesto varios modelos que permiten conseguir dicho objetivo [3].

Nosotros escogimos el modelo propuesto por Vinyals *et al.* [4] llamado Matching Networks. Dicho modelo consiste en el uso de un conjunto de datos grande y otro más pequeño llamado conjunto de soporte. El conjunto de soporte contiene etiquetas que no se encuentran en el conjunto de datos más grande. Así, su objetivo es utilizar el conjunto de datos grande como conocimiento previo para clasificar datos de entrada con etiquetas que solo están contenidos en el conjunto de soporte. En este modelo se aplican mecanismos de atención: estos comparan la similitud coseno entre el valor predicho por una red neuronal convolucional entrenada en el conjunto grande frente a otra red entrenada con el conjunto de soporte. Este proceso usa sistemas de aprendizaje incrustado para reducir la dimensionalidad de los datos de entrada y utiliza Long Short-Term Memory (LSTM) para clasificar nuevas clases contenidas en el conjunto de soporte teniendo en cuenta las contenidas en este conjunto [4].

II. MATERIALES Y MÉTODOS

Para el proyecto utilizaremos el conjunto de datos publicado por Kaya *et al.* [5] en 2018. Este conjunto de datos corresponde a imágenes motoras de EEG para BCI y cuenta con registros que suman 60 horas. Los experimentos fueron realizados con 13 individuos: 8 hombres y 5 mujeres entre 20 y 35 años. Se realizaron 75 sesiones, cada una tuvo una duración de 55 minutos divididas en 3 subsesiones de 15 minutos y tiempo de descanso. En total, se registraron alrededor de 60,000 ejemplos de imágenes motoras. Por estos motivos, el conjunto de datos corresponde a uno de los más grandes publicados hasta la fecha.

Además, los 60,000 ejemplos de imágenes motoras se clasifican según 4 paradigmas de interacción: CLA, HaLT, 5F y FreeForm [5]. CLA es el paradigma clásico que incluye los movimientos de la manos izquierda y derecha, y una imagen mental pasiva donde los participantes no realizan ninguna imagen motora. HaLT es una extensión de CLA en el que se incluyen las imágenes de los pies izquierdo y

derecho, y el movimiento de la lengua. 5F contiene imágenes motores de los cinco dedos de la mano. En FreeForm, los participantes realizaban movimientos voluntarios guiados por la información en pantalla. Tanto 5F como FreeForm contienen pocos ejemplos porque no fueron estudiados detalladamente por los desarrolladores del conjunto de datos. En este proyecto, trabajaremos con las imágenes motoras según los paradigmas CLA y HaLT.

Nosotros nos basaremos en los dos modelos con mayor precisión publicados por Schirrneister *et al.* [6] para compararlo con un modelo que utiliza técnicas de Few-Shot Learning. Estos modelos están basados en una red neuronal convolucional (ConvNet) y se demostró que tienen un desempeño ligeramente superior a Filter Bank Common Spatial Patterns (FBCSP), el método del estado del arte [6]. Las ventajas de utilizar redes convolucionales incluyen que estas pueden ser entrenadas *end-to-end* (a partir de los datos originales sin extraer características), escalan bien en conjuntos de datos grandes y pueden aprovechar la estructura jerárquica que poseen las señales naturales. Los modelos son llamados Shallow ConvNet y Deep ConvNet.

Shallow ConvNet es una red convolucional con menos parámetros que Deep ConvNet. Las primeras dos capas realizan una convolución temporal y un filtrado espacial análogos a un filtro pasabandas y a un filtro espacial CSP (Common Spatial Pattern) usados en FBCSP [6], como se muestra en la Figura 1. Le siguen una no-linealidad cuadrática, una capa de mean pooling y una función de activación logarítmica, estos pasos son análogos al *trial log-variance computation* en FBCSP. Shallow ConvNet busca decodificar las características de las bandas de potencia [6]. Por otro lado, Deep ConvNet tiene cuatro bloques convolucionales con max pooling. Luego de las capas convolucionales le sigue una capa de clasificación softmax densa, como se muestra en la Figura 2. Deep ConvNet fue diseñada para extraer un amplio rango de características y que no se restrinja a tipos específicos de estas [6].

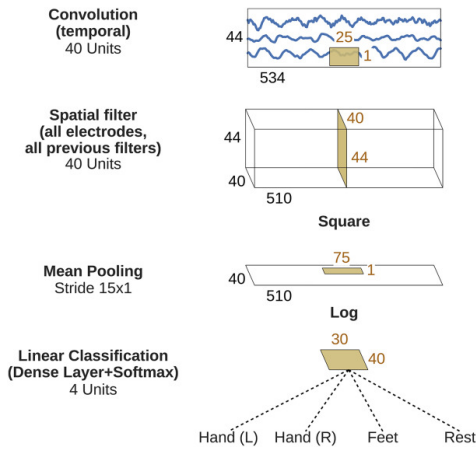


Fig. 1. Arquitectura de Shallow ConvNet para decodificación de las señales de EEG utilizadas en un sistema BCI basado en imágenes motoras. Imagen tomada de [6].

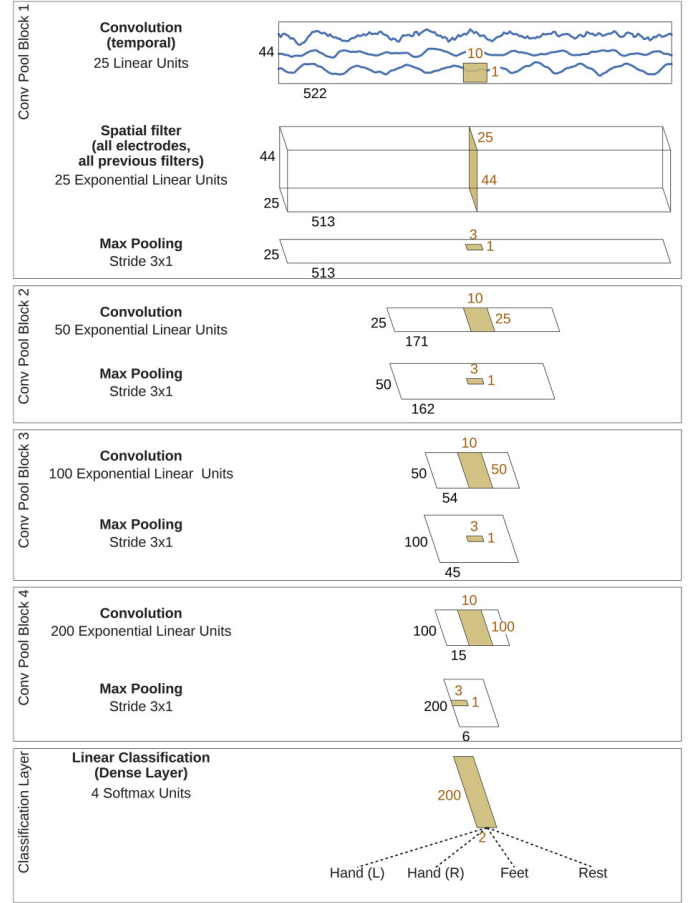


Fig. 2. Arquitectura de Deep ConvNet para decodificación de las señales de EEG utilizadas en un sistema BCI basado en imágenes motoras. Imagen tomada de [6].

En el diseño de las ConvNets, los autores utilizaron Exponential Linear Unit (ELU) como función de activación [7], max pooling, dropout [8], normalización del batch [9] y una división en las convoluciones temporales y espaciales (primero se realiza la convolución temporal) [6]. Para entrenar un modelo de ConvNet todos los parámetros son entrenados en conjunto, permitiendo realizar el entrenamiento con los datos EEG en bruto. Respecto al entrenamiento de las ConvNets, usaron *cropped training* que consiste en obtener más ejemplos de entrenamiento por cada ejemplo etiquetado, mediante una ventana deslizante con un tamaño de tiempo definido [6]. Con los conjuntos de datos que utilizaron, lograban incrementar el tamaño del conjunto de entrenamiento por un factor de 625. Además, utilizaron el optimizador de Adam y un criterio de *early stopping* [6]. Por otro lado, los autores propusieron nuevas metodologías para visualizar las características aprendidas por las ConvNets y mostraron que estas usan las modulaciones espectrales de potencia en las bandas de frecuencias alfa, beta y gamma [6]. El código está disponible en <https://github.com/TNTLFreiburg/braindecode>.

En este proyecto, se plantea la hipótesis de que un modelo de FSL puede alcanzar el desempeño de un algoritmo de

clasificación de imágenes motoras para BCI, pero con menos datos y utilizando el conocimiento previo que haya adquirido con otros datos. Para ello, se entrenarán las ConvNets para clasificar las imágenes motoras según el paradigma HaLT (que tiene 6 clases) con los datos de 8 pacientes y se reservarán los datos 5 pacientes para la prueba del modelo de FSL, en este caso, Matching Networks. Pocos datos (aproximadamente 5 por clase para cada paciente) de los 5 pacientes se utilizarán como el conjunto de soporte. Se evaluará el desempeño del modelo de Matching Networks con los datos restantes de los 5 pacientes. Para compararlo con un enfoque tradicional, los datos de entrenamiento de los 13 pacientes son usados para entrenar las ConvNets y se evalúa en datos de prueba de los 5 pacientes seleccionados en el modelo de FSL. Además, se realizará esto mediante la calibración (entrenamiento) por cada paciente y otra que sea independiente del paciente. Esto tendría implicaciones en la forma de entrenar algoritmos de clasificación para BCI, ya que no será necesario registrar muchos datos de imágenes motoras para nuevos usuarios y, por tanto, el tiempo de entrenamiento del usuario será reducido.

III. CRONOGRAMA

El cronograma propuesto, con sus actividades divididas en revisión bibliográfica, experimentos y elaboración del artículo y/o póster, se muestra en la Tabla I. La revisión bibliográfica se enfoca en cuatro temas: BCI, algoritmos de clasificación basados en ML, redes convolucionales y FSL. Los experimentos consisten en la implementación, entrenamiento y evaluación de los modelos de ML, ConvNets y FSL. Paralelamente a esto, se trabajará en las distintas secciones del artículo.

TABLE I
CRONOGRAMA DEL PROYECTO DE INVESTIGACIÓN DURANTE UN SEMESTRE ACADÉMICO

Semana	Revisión bibliográfica	Experimentos	Paper/Póster
Semana 1	Brain-Computer Interface	Exploración de datos	
Semana 2	Brain-Computer Interface	Exploración de datos	
Semana 3	Algoritmos de clasificación	Exploración de datos	Introducción
Semana 4	Algoritmos de clasificación	Algoritmos de ML	Introducción
Semana 5	ConvNets	Algoritmos de ML	Introducción
Semana 6	ConvNets	Implementación ConvNets	Introducción
Semana 7	ConvNets	Implementación ConvNets	Materiales y métodos
Semana 8	FSL	Implementación ConvNets	Materiales y métodos
Semana 9	FSL	Implementación ConvNets	Materiales y métodos
Semana 10	FSL	Implementación FSL	Resultados
Semana 11	FSL	Implementación FSL	Resultados
Semana 12	FSL	Implementación FSL	Resultados
Semana 13		Implementación FSL	Resultados
Semana 14		ConvNets y FSL	Discusión
Semana 15		ConvNets y FSL	Discusión
Semana 16			Conclusiones

REFERENCES

- [1] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update.," *Journal of neural engineering*, vol. 15, no. 3, p. 031005, 2018.
- [2] H. Cho, M. Ahn, M. Kwon, and S. C. Jun, "A step-by-step tutorial for a motor imagery-based BCI," in *Brain-Computer Interfaces Handbook*, pp. 445–460, CRC Press, Jan. 2018.
- [3] Y. Wang, Q. Yao, J. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," 2019.
- [4] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *NIPS*, 2016.
- [5] M. Kaya, M. K. Binli, E. Ozbay, H. Yanar, and Y. Mishchenko, "A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces," *Scientific Data*, vol. 5, Oct. 2018.
- [6] R. T. Schirrmeister, J. T. Springenberg, L. D. J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball, "Deep learning with convolutional neural networks for EEG decoding and visualization," *Human Brain Mapping*, vol. 38, pp. 5391–5420, Aug. 2017.
- [7] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015.
- [8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [9] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv*, vol. abs/1502.03167, 2015.