

Procesado *on the edge* de señales EEG para reconocimiento de tareas de imaginación motora

Daniel Enériz, Ana Caren Hernández-Ruiz, Nicolas Medrano, Belen Calvo

Grupo de Diseño Electrónico (GDE)
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: eneriz@unizar.es

Resumen

El desarrollo de las interfaces cerebro-computador busca habilitar la comunicación entre humanos y máquinas, principalmente mediante dispositivos capaces de adquirir e interpretar las señales eléctricas emitidas por el cerebro del usuario al realizar acciones concretas e intencionadas. Este trabajo presenta la implementación sobre una FPGA de una red neuronal convolucional compacta capaz de clasificar correctamente este tipo de señales.

Introducción

Las interfaces cerebro-computador, BCI por sus siglas en inglés, son una tecnología que busca habilitar la comunicación entre los humanos y las máquinas a través del procesamiento y reconocimiento de las señales eléctricas emitidas por el cerebro, y cuya aplicación no solo está restringida a la medicina [1], sino que también se explora su uso en sectores como el entretenimiento, control de prótesis o la potenciación de capacidades [2, 3]. Uno de los paradigmas más destacados en el desarrollo de BCIs basados en el uso de señales de electroencefalograma (EEG) [4] es el de reconocimiento de actividad cerebral relacionada con imaginación motora (MI), es decir, señales eléctricas provocadas por la simulación o ensayo de movimientos. Una metodología común para resolver esta tarea es el uso de algoritmos de extracción de características, como el *Common Spatial Pattern* (CSP) o el *Independent Component Analysis* (ICA), tras el cual se realiza una clasificación, utilizando por ejemplo *k-Nearest Neighbours* (k-NN) o *Support Vector Machines* (SVM). Por otro lado, dadas ciertas similitudes entre el procesamiento de señales de EEG y las técnicas de Visión Artificial, algunos trabajos [5-6] han propuesto el uso de Redes Neuronales Convolucionales (CNN) cuya arquitectura incorpora ambas etapas, la extracción de características y su procesamiento para la detección de clases. La EEGNet es una red convolucional diseñada específicamente para el procesamiento de señales de EEG [7]. Cuenta con una arquitectura compacta y sus resultados son

comparables con los obtenidos por otras técnicas en el estado del arte.

Otro aspecto importante en el desarrollo de BCIs es el soporte físico en el cual se va a ejecutar todo el tratamiento de las lecturas de los sensores y su interpretación. Una posibilidad es hacer el procesamiento de datos en la nube, lo que restringe su utilidad a zonas con acceso adecuado a Internet, pudiendo incrementar la latencia del sistema y poner en riesgo la privacidad del usuario. Otra alternativa es el procesamiento *on the edge*, empleando sistemas totalmente locales, de forma que la carga computacional de la BCI se realice en el dispositivo del usuario. Esta posibilidad precisa de un estudio de las prestaciones computacionales básicas requeridas, de modo que pueda obtenerse un modelo lo más simple posible y compatible con procesadores de reducidas prestaciones.

En este trabajo se presenta una adaptación de la EEGNet que mejora los resultados del estado del arte obtenidos en la base de datos *Physionet EEG Motor Movement/Imagery* [8], y su implementación en una *Field-Programmable Gate Array* (FPGA) haciendo uso de representación numérica en punto fijo y funciones de activación sencillas que favorecen la optimización del uso de recursos hardware.

Metodología

Base de datos

Physionet EEG Motor Movement/Imagery está compuesta por las lecturas de 64 canales de EEG de 109 sujetos durante la realización de cuatro tareas de MI: imaginar abrir y cerrar el puño izquierdo (L), el puño derecho (R), descanso (0) o ambos pies (F). De esta forma, se consideran tres escenarios diferentes, de 2, 3 y 4 clases, usando las clases L/R, L/R/0 y L/R/0/F, respectivamente. Existen 21 muestras de cada clase para cada sujeto, que se obtienen durante un estímulo de 4 s de duración a una frecuencia de muestreo de 160 Hz. Para limitar el número de

elementos de entrada a la red, se incorporaron las técnicas de reducción a los datos presentados en [9]: (1) submuestreo de las señales con diferentes factores $ds = \{1, 2, 3\}$; (2) reducción del número de canales EEG $N_{\text{chan}} = \{64, 38, 19, 8\}$; y (3) reducción del tamaño de la ventana temporal de entrada $T = \{3, 2, 1\}$ s. Cabe recalcar que tras el submuestreo las medidas descartadas se han empleado para aumentar el tamaño del dataset, como se muestra en la Figura 1.

Arquitectura, entrenamiento y resultados

Para reducir los requisitos de la arquitectura EEGNet seleccionada se sustituyó la función de activación logística del modelo original por una LeakyReLU con factor alfa escalonado, cuya implementación en FPGA consume menos recursos. La arquitectura fue adaptada de acuerdo a las técnicas de reducción del tamaño de los datos previamente mencionada dando lugar a la arquitectura descrita en la Tabla 1. El entrenamiento se realizó en la API Keras de Tensorflow siguiendo el mismo método presentado en [6] y [9]: entrenamiento inicial de un modelo global con *cross-validation* (CV) de *5-folds*, y posterior *transfer-learning*, entrenando los modelos específicos para cada sujeto (SS-TL) validándolos con CV de *4-folds*. El modelo global se entrenó durante 100 épocas usando un *batch size* de 16 con el método *Adam*. Por su parte, cada modelo de sujeto es entrenado durante 5 épocas con los mismos hiperparámetros que en el caso global.

Los resultados de esta arquitectura tanto para el modelo global como para los SS-TL en los 3 escenarios de clases usando $T = 3$, $N_{\text{chan}} = 64$ y $ds = \{1, 2\}$ comparados con los de los trabajos [6] y [9] se muestran en la Tabla 2. Puede observarse que con $ds = 1$ los modelos globales mejoran ligeramente el estado del arte, mientras los modelos SS-TL con $ds = 2$ consiguen mejoras importantes, comparables con métodos mucho más sofisticados de procesamiento de EEG, gracias a la *data augmentation*.

Implementación en FPGA

La implementación de la red en FPGA se ha desarrollado para demostrar la viabilidad de ejecutar el modelo en este tipo de soporte físico. Se seleccionó la FPGA incluida en el módulo Xilinx XC7Z010 *System on-Chip* (SoC), de bajo coste. Además, este SoC cuenta con un procesador ARM Cortex-A9 de dos núcleos que funcionan a una frecuencia máxima de 667 MHz. Para facilitar el trabajo con el SoC, se eligió un kit Red Pitaya STEMLab 125-10, del

tamaño de una tarjeta de crédito y con un peso entorno a los 60 g.

Para desarrollar la implementación de la adaptación de la EEGNet se utilizó la herramienta Vivado *High-Level Synthesis* (HLS), que permite traducir a *Hardware Description Language*, una descripción algorítmica en un lenguaje de alto nivel (C/C++) del sistema. Para ello, se desarrolló a partir del modelo obtenido en Keras una versión en C/C++ sobre la que se realizaron diferentes estudios orientados a reducir la complejidad computacional del modelo, reduciendo así el consumo de recursos lógicos y permitiendo su programación dentro de la FPGA. Debido al impacto del tipo de representación numérica en la exactitud de los resultados obtenidos, se realizó un análisis para determinar la representación numérica que proporcionase el mejor compromiso entre ocupación de recursos/exactitud de resultados. En primer lugar, se seleccionó una representación en punto fijo para los valores de la red neuronal: entradas, parámetros, valores internos y salidas. Tras evaluar el impacto del número de bits empleado para la representación de números en la exactitud de la red implementada, se determinó la adecuación de una codificación numérica en 16 bits con 8 bits para la parte entera.

Con esta representación se implementó sobre la FPGA el modelo neuronal con $ds = 2$ y $T = 3$, dando un valor de 65.45 % del parámetro *accuracy* en su versión global tras la simulación en HLS, un 0.11 % por debajo del resultado que se conseguía tras el entrenamiento en Keras, 65.56%, pero todavía por encima de los resultados del estado del arte implementados en computadores tradicionales usando punto flotante.

Conclusiones

En este trabajo se ha desarrollado un modelo basado en EEGNet que mejoran los resultados en el estado del arte en el dataset *Physionet EEG Motor Movement/Imagery*. Aprovechando la naturaleza compacta de esta arquitectura se ha programado sobre una FPGA de bajo coste usando la herramienta HLS, demostrando la viabilidad de ejecución de este tipo de modelos *on the edge* usando FPGAs como soporte físico.

REFERENCIAS

- [1]. KINNEY-LANG, E., KELLY, D., FLOREANI, E.D., JADAVJI, Z., ROWLEY, D., ZEWDIE, E.T., ANARAKI, J.R., BAHARI, H., BECKERS, K., CASTELANE, K., CRAWFORD, L., HOUSE, S., RAUH, C.A., MICHAUD, A., MUSSI, M., SILVER,

- J., TUCK, C., ADAMS, K., ANDERSEN, J., CHAU, T. y KIRTON, A., 2020. Advancing Brain-Computer Interface Applications for Severely Disabled Children Through a Multidisciplinary National Network: Summary of the Inaugural Pediatric BCI Canada Meeting. *Frontiers in Human Neuroscience*, vol. 14, pp. 593883. ISSN 1662-5161. DOI 10.3389/fnhum.2020.593883
- [2]. LIAO, L.-D., CHEN, C.-Y., WANG, I.-J., CHEN, S.-F., LI, S.-Y., CHEN, B.-W., CHANG, J.-Y. y LIN, C.-T., 2012. Gaming control using a wearable and wireless EEG-based brain-computer interface device with novel dry foam-based sensors. *Journal of NeuroEngineering and Rehabilitation*, vol. 9, no. 1, pp. 5. ISSN 1743-0003. DOI 10.1186/1743-0003-9-5.
- [3]. SZOCIK, K. y WÓJTOWICZ, T., 2019. Human enhancement in space missions: From moral controversy to technological duty. *Technology in Society*, vol. 59, pp. 101156. ISSN 0160791X. DOI 10.1016/j.techsoc.2019.101156.
- [4]. ABIRI, R., BORHANI, S., SELLERS, E.W., JIANG, Y. y ZHAO, X., 2019. A comprehensive review of EEG-based brain-computer interface paradigms. *Journal of Neural Engineering*, vol. 16, no. 1, pp. 011001. ISSN 1741-2560, 1741-2552. DOI 10.1088/1741-2552/aaf12e.
- [5]. AN, X., KUANG, D., GUO, X., ZHAO, Y. y HE, L., 2014b. A Deep Learning Method for Classification of EEG Data Based on Motor Imagery. *ICIC*. S.l.: s.n., DOI 10.1007/978-3-319-09330-7_25.
- [6]. DOSE, H., MØLLER, J.S., IVERSEN, H.K. y PUTHUSSERYPADY, S., 2018. An end-to-end deep learning approach to MI-EEG signal classification for BCIs. *Expert Systems with Applications*, vol. 114, pp. 532-542. ISSN 09574174. DOI 10.1016/j.eswa.2018.08.031.
- [7]. LAWHERN, V.J., SOLON, A.J., WAYTOWICH, N.R., GORDON, S.M., HUNG, C.P. y LANCE, B.J., 2018. EEGNet: a compact convolutional neural network for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, vol. 15, no. 5, pp. 056013. ISSN 1741-2560, 1741-2552. DOI 10.1088/1741-2552/aace8c
- [8]. GOLDBERGER, A.L., AMARAL, L.A.N., GLASS, L., HAUSDORFF, J.M., IVANOV, P.Ch., MARK, R.G., MIETUS, J.E., MOODY, G.B., PENG, C.-K. y STANLEY, H.E., 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, vol. 101, no. 23. ISSN 0009-7322, 1524-4539. DOI 10.1161/01.CIR.101.23.e215.
- [9]. WANG, X., HERSCHE, M., TOMEKCE, B., KAYA, B., MAGNO, M. y BENINI, L., 2020. An Accurate EEGNet-based Motor-Imagery Brain-Computer Interface for Low-Power Edge Computing. *2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. Bari, Italy: IEEE, pp. 1-6. ISBN 9781728153865. DOI 10.1109/MeMeA49120.2020.9137134.

Tabla 1. Adaptación de la EEGNet usada en el trabajo

Capa	# Filtros	Padding	Kernel	# Parámetros	Activación	Forma salida
Input	-	-	-	-	-	$(N_{\text{chan}}, fs \cdot T/ds, 1)$
Conv2D	4	same	$(1, fs/2ds)$	$2fs/ds$	LReLU ($\alpha=0.6$)	$(N_{\text{chan}}, fs \cdot T/ds, 4)$
DepthConv2D	2-4	valid	$(N_{\text{chan}}, 1)$	$8N_{\text{chan}}$	LReLU ($\alpha=0.5$)	$(1, fs \cdot T/ds, 8)$
AvgPool2D	-	valid	$(1, 6/ds)$	-	-	$(1, fs \cdot T/6, 8)$
SepConv2D	8	same	$(1, 16)$	192	LReLU ($\alpha=0.4$)	$(1, fs \cdot T/6, 8)$
AvgPool2D	-	valid	$(1, 8)$	-	-	$(1, fs \cdot T/48, 8)$
Flatten	-	-	-	-	-	$fs \cdot T/6$
Dense	N_{classes}	-	-	$N_{\text{classes}} \cdot fs \cdot T/6$	Softmax	N_{classes}

Tabla 2. Accuracy en clasificación (%)

N_{classes}	Dose et al. [6]		Wang et al. [9]		Este trabajo			
	Global	SS-TL	Global	SS-TL	$ds = 1$		$ds = 2$	
					Global	SS-TL	Global	SS-TL
2	80.38	86.49	82.43	84.32	83.15	87.46	82.52	93.10
3	69.82	76.25	75.07	80.07	75.74	83.26	75.34	93.21
4	58.59	68.51	65.07	70.83	65.75	74.31	65.56	89.23

Para los resultados en [6] y [9] y los presentados aquí con $ds = 1$ no se ha usado *data augmentation*, mientras que los resultados presentados en este trabajo con $ds = 2$ cuentan con el doble de muestras.

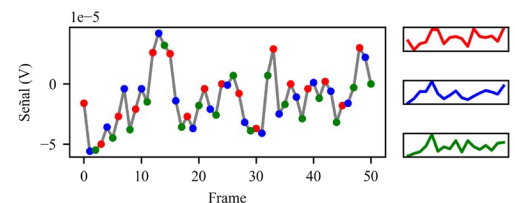


Fig. 1. Ejemplo de uso del *downsampling* como *data augmentation*. En este caso se muestra $ds = 3$, de forma que cada tercio se trata como una muestra diferente.