ITM
Institución Universitaria
Acreditada en Alta Calidad

Classification of Motor Imagery EEG Signals Using a CNN Architecture and a Meta-heuristic Optimization Algorithm for Selecting Training Parameters

Andrés Felipe Pérez Zapata

Instituto Tecnológico Metropolitano
Faculty of Engineering
Medellín, Colombia
2019

Classification of Motor Imagery EEG Signals Using a CNN Architecture and a Meta-heuristic Optimization Algorithm for Selecting Training Parameters

Andrés Felipe Pérez Zapata

Research work submitted as requirement to obtain the degree of:
Magister en Automatización y Control Industrial

Advisors:
Jorge A. Jaramillo, PhD. and
Gloria M. Díaz, PhD.

Research Area:
Máquinas Inteligentes y Reconocimiento de Patrones (MIRP)
Research Group:
Automática, Electrónica y Ciencias Computacionales (AEyCC)

Instituto Tecnológico Metropolitano
Faculty of Engineering
Medellín, Colombia
2019

I will not follow where the path may lead, but I will go where there is no path, and I will leave a trail.

-Muriel Strode

A mis padres y mi hermana por su apoyo durante estos años de estudio, por darme fuerzas para continuar cuando las necesité. A mi abuela que siempre estuvo pendiente de mí. A Yirley Román Herrera por su apoyo, consejos e invaluable compañía. Y a toda mi familia que creyó en mí, mil gracias.

Acknowledgements

Thanks to my thesis advisors, professors PhD. Gloria Mercedes Díaz Cabrera and PhD. Jorge Alberto Jaramillo Gárzón for their guidance and support during this work. Thanks to professor PhD. Carlos Andrés Madrigal González for his guidance in my early training as researcher, without him I would not have chosen this path.

I also thank to the members of my thesis commitee, Jorge Iván Padilla Buritica, Gustavo Adolfo Isaza Echeverry and Diego Fabián Collazos Huertas for their relevant advice that helped improve the quality of this work.

Finally, I want to thank my partner Andrés Felipe Cardona Escobar for his teachings and contributions to this work, to all my colleagues from the Máquinas Inteligentes y Reconocimiento de Patrones (MIRP) laboratory from the Instituto Tecnológico Metropolitano, Rubén Fonnegra, Julián Arango, Óscar Ossa, Gabriel Acevedo, Irving Solsol, Pablo Villegas, Carlos Duarte and my teachers and partners Andrés Giraldo and Hermes Fandiño.

Abstract

A Brain Computer Interface (BCI), is a system created for performing communication and control of computational devices by the analysis of brain signals. Particularly, Motor Imagery (MI) based BCI systems allow a real-time interaction by using the electrical signals that are generated by the brain when the user imagining certain movements or actions. Acquisition of these signals can be done by invasive and non-invasive devices, among which, electroencephalography is a non-invasive technique that is widely used. It is based on the superficial placement of the electrodes on the scalp, which avoid affecting the health and well-being of potential users, also is a portable technology and lower-cost than other alternatives. From the different stages required to develop a BCI, the signal characterization and the classification tasks continue to be the main research challenges, since the performance of the whole system depends on them. This thesis proposes the use of convolutional neural networks (CNN) for the classification of electroencephalographic signals, in order to identify the action imagined by a person. The proposed network architecture is trained with representations of the spectral power density (PSD) of the signals; and the hyperparameters of the network are defined by a metaheuristic optimization algorithm, which obtains the best accuracy in the signal classification in training stage. The proposed approach was evaluated using two public and well-known databases, i.e. BCI Competition IV 2a and BCI Competition IIIa. According to the results, this approach provides a reliable strategy to differentiate movement imagination, outperforming state of the art that used the same datasets. These results demonstrate that this is a valuable and promising strategy for the design of brain computer interfaces.

Keywords: Brain Computer Interfaces (BCI), Electroencephalography (EEG), Power Spectral Density (PSD), Deep Learning, Convolutional Neural Network (CNN), Metaheuristic Optimization Algorithm (MOA), Hyperparameter Tuning .

Resumen

Una interfaz cerebro-computadora (BCI, por sus siglas en inglés), es un sistema que permite la comunicación y control de dispositivos a partir del análisis de las señales cerebrales del usuario. Entre los diferentes esquemas BCI, se destacan los basados en Imaginación Motora (MI, por sus siglas en inglés); los cuales, permiten una interacción en tiempo real, utilizando sólo las señales eléctricas generadas por el cerebro cuando imagina ciertos movimientos o acciones. La adquisición de estas señales se puede realizar mediante diferentes metodologías invasivas y no invasivas, siendo la electroencefalografía una de las técnicas no invasivas más empleadas, debido a que, como se basa en la colocación superficial de los electrodos en el cuero cabelludo, no afecta la salud y el bienestar de los potenciales usuarios, además de ser una tecnología portable y de bajo costo, con respecto a otras alternativas. Entre las diferentes etapas que conforman un BCI, las tareas de caracterización y clasificación de las señales, continuan siendo los mayores retos de investigación, pues de estas depende el desempeño del sistema en general. Esta tesis propone la implementación de redes neuronales convolucionales (CNN, por sus siglas en inglés) para la clasificación de señales electroencefalográficas, con el fin de identificar la acción imaginada por una persona. La arquitectura propuesta es entrenada con representaciones de la densidad espectral de potencia de las señales, y los hiperparámetros de la red son definidos por un algoritmo de optimización metaheuristico, para obtener la mejor exactitud en la clasificación de las señales. El desempeño del método propuesto fue evaluado utilizando bases de datos públicas y bien conocidas, como son BCI Competition IV 2a y BCI Competition IIIa. De acuerdo con los resultados obtenidos, el método propuesto proporciona una estrategia confiable para diferenciar imaginación de movimiento, superando resultados del estado del arte para el mismo conjunto de datos, lo que demuestra que esta es una estrategia valiosa y prometedora para el diseño de interfaces cerebro computador. Además, se encontró que la arquitectura definida mediante el algoritmo de optimización metaheuristico mostró un desempeño muy superior al de la arquitectura sintonizada manualmente, lo cuál demuestra que el uso de algoritmos de optimización metaheuristicos para definir los hiperparámetros de una red neuronal profunda es una estrategia altamente recomendada.

Palabras clave: Interfaz cerebro computador (ICC), Electroencefalografía (EEG), Densidad espectral de potencia (DEP), Aprendizaje Profundo, Red Neuronal Convolucional (RNC), Algoritmo de Optimización Metaheurística (AOM), Sintonización de Hiperparametros.

Table of Contents

1. Introduction

A Brain-Computer Interface (BCI) is a communication and control system that measures and analyzes brain signals. Among the different BCI schemes, there are those based on motor imagination (MI), which consists in the use of signals based on the modulation of brain activity for allowing real-time interaction between the brain and computer devices, using only the electrical signals generated by the imagination of particular movements or actions [56]. Therefore, a BCI system replaces the nerves and muscles that cause the movement, with hardware and software that measures brain signals and translates it into actions.

The first attempts to develop BCI systems date back to 1973 [77]; however, it was in the last two decades when this research area showed the most relevant advances [43, 42]. Those advances have been motivated by the development of new acquisition and processing technologies but also by the emerging use of BCI into new applications besides of clinical applications to disabled people [13], including video-games, toys, advertising, safe-driving, among others [59, 23, 68, 20].

A BCI system is composed of four main stages: neurophysiological signal acquisition, signal feature extraction, signal classification, and device control or communication. According to this, neurophysiological signals captured by acquisition systems are translated into output orders that represent the user intention movements and do not depend on the standard output pathways of the peripheral nerves and muscles [51].

Neurophysiological signals acquisition can be performed by different invasive and non-invasive techniques. In invasive technologies, electrodes are embedded into the brain surgically, causing physical injury to the individual. Thus, non-invasive technologies, where the measurement sensors are placed outside the head, are preferred. Among them, electroencephalography (EEG) has been the most used due to its portability and low cost. Moreover, EEG preserves the health and well-being of patients, since it is based on the superficial placement of electrodes on the scalp. EEG technique measures the cerebral activity caused by the flow of electrical currents during synaptic excitations of neurons dendrites. However, this technique has several drawbacks, since the signals must pass through the skull and scalp, introducing noise, so that, when it reaches the electrodes, they have a very low quality and intensity [58]. Therefore, signal processing and classification are the most challenging processes of BCI.

On the other hand, deep learning methods based on Artificial Neural Networks (ANN), such as the Convolutional Neural Networks (CNN), have the potential to automatically detect seemingly unrelated features of a wide range of data thanks to the modeling of high level abstractions through nonlinear transformations that are not affected by noise, thus being unnecessary data filtering and preprocessing steps [78]. In addition, the use of new techniques that allow to avoid over-fitting effectively, such as Dropout and new activation functions such as Rectified Linear Units (ReLU) which have the advantage of being faster to calculate and do not suffer from the gradient leakage problem [57]. Among the techniques of deep learning based on ANN are Recurrent Neural Networks, Deep Belief Networks, Neural Networks with Time Delays [54], and Convolutional Neural Networks [84].

Although DCNN are naturally suitable for its use in images due to its characteristic of be able to successfully capture the spatial and temporal dependencies in an image, through the application of relevant convolutional filters; its suitability to be used in unidimensional signals, as EEG, opened a door to research in that kind of methodologies [72]. The great capacities of CNN over brain signals had been shown in the prediction of patient's epileptic seizures by classifying brain activity signals[53], [4] and [74]. In these cases, CNN exhibit a great ability to filter the input data with the convolutional layers, demonstrating the preprocessing required in a CNN is lower than other classification algorithms. Thus, in this thesis, the evaluation of CNN as a learning method for identifying the activity imagined by a person, based on the analysis of their EEG, is proposed.

Important issues in the development of CNN include the need for a large amount of data for training [40], the propensity to over-fitting [70], the gradient leakage problem [81], and the high time-consuming process of the training, by which, the exhaustive search of the network hyper-parameters becomes prohibitive[8]. To deal with these issues, some state of the art techniques were implemented. Initially, a data augmentation strategy was implemented, which allowed to have a large set of training and testing instances. Then, the technique of using Dropout layers [70] was implemented to avoid over-fitting, and the ReLU activation function to avoid the gradient leakage problem. Finally, to deal with the time-consuming process, the implementation was done over a GPU instead of the CPU's cores. Additionally, because the strong dependency on the correct selection of the network architecture hyper-parameters, i.e., minor changes in any hyper-parameter generates a different ANN architecture, and therefore, a different learning model; the use of an optimization algorithm was implemented to obtain the network architecture to reach the best accuracy in the classification task.

The proposed methodology was tested using two public and well-known databases, i.e. the multiclass BCI Competition IV-2a dataset, and the multiclass BCI Competition IIIa dataset.

Experimental design included, 1) the evaluation of the proposed CNN with network architecture hyper-parameters searched by a manual tuning strategy; 2) the evaluation of the proposed CNN using an automatic hyper-parameters tuning through the meta-heuristic optimization algorithm; and 3) evaluation of the optimized network architecture as learning model for classifying signals in a different database (multiclass BCI Competition III dataset). Both strategies, i.e. manual and automatic hyper-parameter tuning, reached an accuracy that outperforms previous works that reported results for the same datasets. However, the architecture defined by the meta-heuristic optimization algorithm exhibits the best classification accuracy, which shows that the use of optimization algorithm for searching network hyperparameters is a strategy highly recommended.

1.1. Hypothesis

Implementation of deep learning techniques, specifically Convolutional Neural Networks with ReLU activation functions between the intermediate layers, will allow developing a scheme for classification of motor imagery signals obtained by EEG. In addition, the use of a meta-heuristic optimization algorithm for selecting the free parameters of the network architecture will allow outperforming the classification accuracy of another state of the art machine learning techniques for BCI systems.

1.2. Objectives

1.2.1. General Objective

To develop a methodology for classifying motor imagery EEG signals by implementing a CNN architecture with ReLU activation functions, using a meta-heuristic optimization algorithm for selecting the network parameters.

1.2.2. Specific Objectives

- To design a classifier based on convolutional neural networks by defining convolutional filters in the space of the EEG electrodes.

- To establish a meta-heuristic optimization strategy for selecting free parameters of the CNN that allows easy parallelization of the learning process.

- To validate the proposed training and classification method in a multi-class recognition problem to a BCI system based on motor imagery.

1.3. Thesis Outline

To document the process followed to comply with the proposed objectives, this document is organized as follows: Chapter 2 introduces the technical background of BCIs, and the main theoretical foundations of deep neural networks and optimization algorithms. Section 2.1 is devoted to describing the main stages of an EEG based BCI system, i.e., signal acquisition, signal processing, classification, and control of the output devices. For each stage, the main previous works are summarized. In Section 2.2., the main aspects of deep neural networks are introduced, with a special emphasis into CNN architectures. At the end of that chapter, Section 2.3. presents the foundations of optimization algorithms, specifically, the Derivative Free Optimization based on Radial Bases Function algorithm (RBFOpt), which was the optimization technique used in this work for selecting the optimal hyper-parameters of the CNN. In chapter 3, the proposed method for classifying motor imagery EEG signals based on deep Convolutional Neural Network (CNN) is presented. This method uses the power spectral density (PSD) representation of EEG signals for training a CNN that is able to differentiate among four movement imaginations, i.e., left-hand movement, right-hand movement, feet movement, and tongue movement, which are they recorded into the BCI Competition IV-2a and BCI competition III datasets. Additionally, the use of RBFOpt algorithm for tuning the network hyper-parameter is also described. Chapter 4 presents the experimental results of the proposed approach. Results of the three main experiments are reported, i.e., the classification performance to the manual tuning of the hyper-parameters of the proposed CNN architecture to the BCI Competition IV-2a dataset, the classification performance to the automatic tuning of the hyper-parameters performed by using the RBBOpt algorithm to the same dataset, and the classification performance reported when the optimal architecture is trained using the instances in the BCI Competition IIIa dataset. Comparisson with state of the art approaches that used these databases is also included. Finally, in chapter 5 are presented the conclusions of this work, and the considerations and future works.

2. Theoretical Background and Previous Works

2.1. EEG based Brain Computer Interfaces

A BCI system based on EEG from MI signals consists usually of four fundamental stages, the first one concerns to the signal acquisition, in which a set of electrodes is placed into the scalp for capturing the neurophysiological electrical signals of the brain. In the second stage pre-processing and feature extraction techniques are used; in the pre-processing step, signals are filtered to the necessary frequencies and aims to eliminate signal artifacts since the EEG is very noise-prone, for this purpose different kind of filters has been implemented, such as high-pass, Low-pass, Band-stop, and Band-pass filters. Filter selection mainly depends on the frequencies that are expected to be related to the activities that are to be identified. The second step, feature extraction, corresponds to the computation of measurements or characteristics from signal segments, which are the most important for distinguishing between the different movements categories that must be recognized. Several feature extraction algorithms have been proposed [64]. However, the accurate representation of those features is still one of the most challenging tasks in BCI development. The third stage is the classification of the samples into one of the expected imagined movement or action by using the feature vector generated by the second stage. Finally, the last stage consists in executing the activity imagined by the subject in an electronic device. Figure 2.1 illustrates the four stages comprising a BCI. Additionally, the main methods reported in the literature to address each of these stages are described below, going in depth into those that were used in the approach proposed in this thesis. To delve into other methods, the reader must consult the references included in each item.

2.1.1. Signal Acquisition

EEG is a technique that allows to record signals from the surface of the skull generated by the brain activity through a set of electrodes located on the scalp, commonly positioned between 18 and 40 electrodes as indicated by the International Federation standard of EEG and neurophysiology. These electrodes are connected to a device that registers electric currents of an ionic nature. Later the ionic currents are converted into electric currents and taken to an instrumentation amplifier, depending on the frequency produced by these impulses we

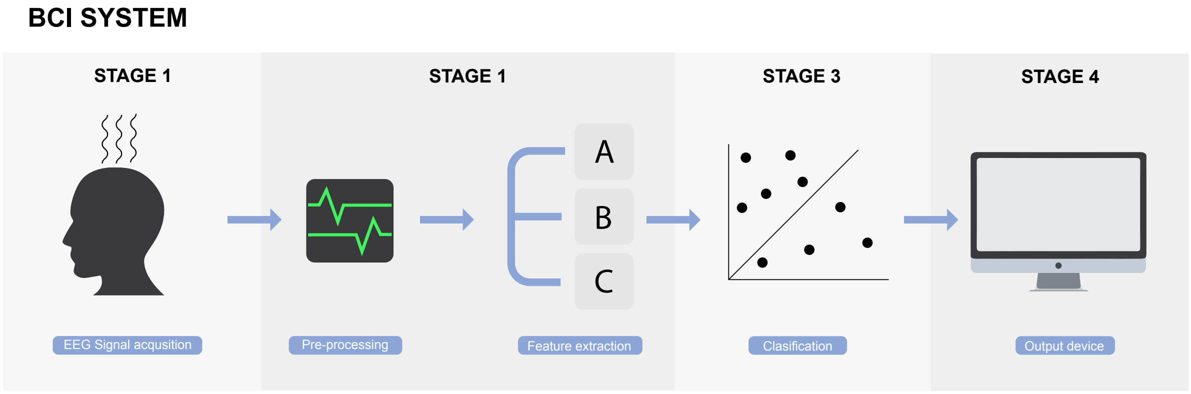

Figure 2.1.: Common BCI system stages, first the EEG Signal Acquisition, second the Pre-processing and Feature Extraction stage, third the Classification stage and finally the Final Device.

could identify several types of waves that can be measured in four bands: the delta bands 0.5 Hz - 4 Hz, theta 4 Hz - 8 Hz, alpha 8 Hz - 13 Hz and beta 13 Hz - 30 Hz, where each of these frequencies has the information of a specific task [3].

The functioning of the central nervous system is based on electrical impulses that travel through neurons. The information travel from neuron to neuron through the dendrites, which are responsible for reception and the axon is responsible for transmission. When a stimulus is generated, it is perceived thanks to the sensorial systems that we possess, be it vision, smell, listening, among others. The information perceived by these natural sensors is transported by the neurons of the nervous system to an integrating component that is responsible for analyzing this information, then a response is generated, which is conducted through the neurons to organs or muscles depending on the type of stimulus [3].

The acquisition of EEG signals of MI is done through the catchment of electromagnetic impulses generated by the simulation of the movements using the imagination. For this, acquisition protocols are created, in which the test subjects must imagine that they perform the movement of a limb or body part without actually moving it. For this purpose, it is usually indicated by means of auditory and/ or visual signals the type of action that should be performed. It has been proven that the practice of MI considerably improves the quality of the electrical signals generated, as well as putting rest phases between the different MI actions that allow to separate them clearly[10].

With the boom in the design of BCI applications by large laboratories, different hardware has been promoted for EEG low-cost samples acquisition, such as NeuroScan SynAmp, Neurosky MindSet, and Emotiv EPOC. These tools have similar technical specifications being portable and easy to use, these have between 8 and 64 channels for sampling, a great autonomy up to 12 hours and are widely used tools for the experimentation and development of brain-computer interfaces [56].

2.1.2. Pre-processing

The main objective of the signal pre-processing stage is to eliminate noise and artifacts generated by signals that are not inherent to the MI process, since these unwanted signals create unnecessary data that difficult the extraction of relevant characteristics to discriminate the carried out actions, since these signals can be overlapped on the time and frequency of the MI signals [14]. Additionally, pre-processing techniques are also useful to adapt the data to the requirements of the methodology that is to be applied, some pre-processing actions could include ordering, data augmentation, windows sliding, among other techniques.

There are many frequency filtering techniques that are commonly implemented in BCI systems to eliminate noise and artifacts. Its use must be careful about avoiding to eliminate the desirable signals with a wrong frequency selection. The most common type of filters used in BCI are[64]:

- **Low-Pass filtering**: this filter allows to pass only low frequency signals.
- **High-Pass filtering**: this filter allows to pass only high frequency signals.
- **Band-pass filtering**: this filter allows to pass a determined range of frequencies attenuating the rest of them.
- **Band-stop filtering**: this filter does not allow the pass of signals whose frequencies are included in a determined range.

Although frequency and spatial filters are commonly used to its utility to eliminate noise characteristic of EEG signals, in this work was not used due to CNN potential to automatically eliminate useful information and detect seemingly unrelated features of a wide range of data that may have been omitted previously thanks to the modeling of high level abstractions through nonlinear transformations. However, it is considered important to mention the frequency filters due to they are common in the most methodologies [64].

In this work the channel selection was not carried out due to three things, the first one is the data augmentation scheme to obtain a large amount of training and test samples used, which creates pseudo images using all the EEG channels available from the dataset. Second, a clustering algorithm to order the electrodes according to their location in the scalp to generate spatial relationship among them. This pre-processing scheme is detailed in section 3.3 Preprocessing. And finally, due to CNN's ability to discriminate between non relevant and relevant information that may be omitted when a channel selection is applied.

2.1.3. Feature Extraction

Feature extraction aims to generate patterns that belong to the same class and allows to discriminate the samples that will be used, grouping these in a certain region of the space to their subsequent classification, using algorithms that transform the initial spatial domain of the data. Feature extraction algorithms for EEG signals are commonly found in three groups [76], those based on temporal analysis, based on frequency analysis and those that combine both doing a time-frequency analysis.

Of this three groups some of the most used algorithms in EEG signals for BCI [73] are described below: First one, based on temporal analysis, Fractal Dimension (FD) method is very common, FD can be interpreted as the degree of irregularity of a signal. In FD method, the signal is reconstructed by smaller signals of itself [31], there are many techniques with which it is possible to estimate FD but the common ones are the Higuchi and Katz methods. Higuchi method creates new waveforms of the original signal in smaller sequences, then length of the signal curve is calculated to obtain the averages of the data , with these values the length averages can find and convert the signal into fractal dimensions and calculate its linear approximations in minimum data [28]. Katz algorithm finds the fractal dimension by calculating the length of the signal curve, making a sum of the Euclidean distances (The Euclidean dimensions are all orthogonal) among several successive data being divided by the longest distances of the data samples [36]. Second based on time-frequency analysis, Wavelet Decomposition is a very good method that combines frequency-time information having the information embedded in frequency and time [49]. A wavelet transform uses a variable-size window, which allows it to perform higher mappings of the signals in those segments where greater precision is required; then, the signal initiates a process of data separation in portions of frequencies, doing this process repeatedly until the signal has been broken down into three levels of signals that are function of frequency, time and amplitude. These data represent the original signal allowing better discrimination of the information, to later be grouped into ranges corresponding to each of them resulting in the reconstruction of the original signal [64].

Although these methods are very popular, they present serious disadvantages in the face of EEG signals, Higuchi's method decrease accuracy when noise ratio increases [31] and Katz' method could be affected by reduntant information [64] that benefits Deep Learning Methods.

Finally, based on frequency analysis, that is the method used in this work, Power Spectral Density (PSD) shows to be a great algorithm due to its capacity to split the signals means its power density. PSD allows finding peaks in certain frequency bands, these peaks are the power density of the signal, such as the Alpha and Betha bands, in which the movements of the right and left hand are found. Which suggests that the peaks for the two other classes, movement of feet and tongue are also clearly differentiable [66].

Power Spectral Density

A Power Spectral Density (PSD) is the measure of signal's power content versus frequency. A PSD is typically used to characterize broadband random signals and describes the distribution of power into frequency components composing that signal. According to Fourier analysis, any physical signal can be decomposed into a number of discrete frequencies, or a spectrum of frequencies over a continuous range. The statistical average of a certain signal or sort of signal (including noise) as analyzed in terms of its frequency content, is called its spectrum. The amplitude of the PSD is normalized by the spectral resolution employed to digitize the signal[71].

There are different methods to calculate the PSD of a signal, among which are Periodograms and the Welch [80] function among others. PSD gives an estimation about the power of a signal at different frequencies for any temporal signal.

Periodogram Function Periodogram is based on the Fourier Transform, according to which a series that meets certain requirements can be decomposed as the sum of a finite or infinite number of frequencies. The periodogram measures contributions to the total variance of periodic components series of a given frequency, if the periodogram presents a peak at a certain frequency, this indicates that this frequency has greater relevance than the rest.
This measure is computed over a signal to find the spectrum in different parts. The square of these results are known as periodograms [48], defined by the expression 2.1:

$$P = \frac{1}{N} A^2[\omega] \tag{2.1}$$

Where $A^2[\omega]$ is the square of the FFT for a signal $a[k]$ with N observations.
If a signal has a non-zero arithmetic mean, its power spectrum has a pulse at zero frequency. If the value of the mean is large, this component will have a magnitude that will dominate the estimate of the power spectrum. The magnitude of the periodogram has approximately the magnitude of the power spectrum.

According with [48], where stochastic quantities are represent with bold letters, the Periodogram of a time-series $\boldsymbol{y}$(n), n=1,...,N is defined as:

$$I_y(\omega_k) = \frac{1}{N} | \sum_{n=1}^{N} \boldsymbol{y}(n) e^{-j\omega_k n} |^2 \tag{2.2}$$

with $\omega_k = 2\pi\, k/N, k = 1, ..., N$. Periodogram ordinates are symmetric around N/2, assuming N even, i.e., $\mathbf{I}_y(\omega_k) = \mathbf{I}_y(\omega_{N-k})$. The value $\mathbf{I}_y(\omega_k)$ is a measure of the energy contribution of the frequency ω_k to the "signal effect".

If $\{\boldsymbol{y}(\mathrm{n}),\ \mathrm{n}=1,...,\mathrm{N}\}$ are samples of a stationary time-series with autocorrelation sequence $\{r_y(i)$,-$\infty < i < \infty\}$, then periodogram ordinates, $\mathbf{I}_y(\omega_k)$, are asymptotically independent exponential random variables, with mean $S_y(\omega_k)$. The function $\mathbf{s}_y(\omega)$ is the power spectral density (PSD) of $\{\boldsymbol{y}(\mathbf{n})\}$ and is given by the Fourier transform of the autocorrelation sequence $\{r_y(i)\}$. Notice that the PSD is a deterministic function of ω.

Welch Function The method is described in the original paper [80] as follows: let $X(j),\ j = 0, \ldots,\ N-1$ as a sample from a stationary, second order stochastic sequence. Assume for simplicity that $E(X) = 0$. Let $X(j)$ have spectral density $P(f),\ |f| \leq \frac{1}{2}$. Segments possibly overlapped with length L were taken, with starting points of these segments D units apart. Let $X_1(j),\ j = 0, \ldots,\ L-1$ being the first such segment. Then:

$$X_1(j) = X(j)\ with\ j = 0, \ldots,\ L-1 \tag{2.3}$$

$$\tag{2.4}$$

$$X_2(j) = X(j+D)\ with\ j = 0, \ldots,\ L-1 \tag{2.5}$$

$$\tag{2.6}$$

$$X_k(j) = X(j+(k-1)D)\ with\ j = 0, \ldots,\ L-1 \tag{2.7}$$

It is suppose to have k such segments; $X_1(j), \ldots,\ X_k(j)$, and that they cover the entired record, i.e., that $(K-1)D + L = N$.The method of estimation is as follows. For each segment of lenght L a modified periodogram is calculated. That is, is selected a data window $W(j),\ j = 0, \ldots,\ L-1$ and form the sequences $X_1(j)W(j), \ldots,\ X_k(j)W(j)$. Then the finite Fourier Transform $A_1(n), \ldots, A_k(n)$ of these sequences was calculated here:

$$A_k(n) = \frac{1}{L}\sum_{j=0}^{L-1} X_k(j)W(j)e^{-}2kijn/L,\ and\ i = (-1)^{1}/2. \tag{2.8}$$

Finally K modified periodograms are obtained by:

$$I_k(fn) = \frac{L}{U}\ |A_k(n)|^2,\ with\ k = 1, 2, \ldots, K \tag{2.9}$$

Where,

$$fn = \frac{n}{L},\ with\ n = 0, \ldots, L/2, and \tag{2.10}$$

$$U = \frac{1}{L}\sum_{j=0}^{L-1} W^2(j). \tag{2.11}$$

The spectral estimated is the average of these periodograms, i.e.,

$$\hat{P}(fn) = \frac{1}{K}\sum_{k=1}^{K} I_k(fn) \tag{2.12}$$

Then

$$E\{\hat{P}(fn)\} = \int_{-1/2}^{1/2} h(f)P(f - fn)df \tag{2.13}$$

Where,

$$h(f) = \frac{1}{LU}\left|\sum_{j=0}^{L-1} W(j)e^{2\pi ifj}\right|^2,\ and \tag{2.14}$$

$$\int_{-1/2}^{1/2} h(f)df = 1. \tag{2.15}$$

With this the estimator $\hat{P}(f)$ with a resultant spectral window whose area is unity and whose width is of the order $1/L$ was obtained.

This nonparametric approach for PSD estimation is regularly employed, its main advantage is the no assumption about the distribution of data[60]. Due to this Welch function was used in the feature extraction stage of this thesis.

2.1.4. Signal Classification

In a BCI system, the classification stage allows to identify or decoding the movement intention of the subject based on the characteristics of the data that is being collected from the signals and associated with mental tasks. The commonly used classification techniques in BCI investigations include both Linear Classifiers and the Nonlinear Classifiers [56].

Among the methods that have been proposed for this purpose, the most used are Support Vector Machines (SVM) [41], Artificial Neural Networks (ANN) [75], and other systems based on machine learning and statistical theory [61]. The methodologies based on SVM and ANN have performed well [82]. However, development of new methodologies seemed to have remained stalled until the emergence of deep learning based technologies.

Recently, Helal et al. [27] developed an LDA based method with Autoencoders, which was composed of five stages: pre-processing, feature extraction, dimensionality reduction, classification, and evaluation. In the preprocessing, signal artifacts were removed by applying whitening, CAR and Z-Score normalization to the raw data. Band-power method was implemented as feature extraction, PCA and Autoencoders were used for dimensionality reduction, and finally, LDA was employed in the classification task. Reported results showed that Autoencoders with non-linear activation function (Sigmoid) achieves better performance compared to PCA, getting a mean classification accuracy of 67%.

In 2015, Bashashati et al. [7] presented a comprehensive comparison of classification models for identifying movement intentions in several EEG datasets, including the BCI Competition IV - 2a. The methodology used starts with a filter bank composed by fifth order Butterworth band pass filters array. Then, a spatial filter (common spatial pattern) is applied before extracting signal features that were then used for evaluating the classification models. In the BCI IV competition 2a dataset, the logistic regression and Multi Layer Perceptron (MLP) classifiers outperformed others, getting a mean accuracy of 74.33% and 74.42% respectively. However, a high dispersion between the accuracy of each subject makes the results unreliable.

Due to the success of deep learning in different fields and applications, it was presented as an alternative in the classification of EEG signals of motor imagery for BCI. Recently, many works have been developed involving deep learning for this purpose, however, it is necessary to explore, test and compare different architectures of deep learning in search of surpassing the threshold of performance that is currently in the classification of motor imagery signals.

Neural network architectures based on back-propagation and forward-propagation have been able to improve the results obtained by SVM-based systems. Wang et al. [79] proposed a binary classification method for a computer-based interface based on EEG signals using the packet of the wavelet transform and neural networks. For the feature extraction, a new method was introduced, which combines the slow cortical potentials (SCPs) and the specific energy of the time-frequency domain in beta bands through the wavelet transform. The 3-layer perceptron architecture established by backward propagation was compared to the performance of a Gaussian kernel based SVM. Among the tests, it was found that the 4-5-1 architecture reached a precision of 91.47% in the test set, against a 91.13% obtained by the SVM with a Gaussian kernel, however the neural network architecture needed a lot of time to be trained, being being very susceptible to inital weights and biases and dependent of the preprocessing techniques and feature extraction algorithms. Subsequent studies compared ANN techniques to determine their performance. Khare et al. [37] compare the performances of the following architectures, Gradient Descent Back Propagation, Levenberg-Marquardt architecture, Resilient Back propagation, Conjugate Learning Gradient Back Propagation, and Gradient Descent Back Propagation, in the classification of EEG signals obtained from the imagination of the right hand movement with respect to an awakened state of relaxation, feature extraction was performed using the Wavelet Transform (WT). As a result it was obtained that, for this classification problem, the ANN with Resilient Back Propagation obtained the highest yield with 95% accuracy.

Amarasinghe eta al. [5] proposed a novel methodology for the recognition of self-organized mapping patterns (SOM), the recognition methodology that was presented consisted of a three-step process that uses SOM for the unsupervised grouping of EEG signals Preprocessed images that were taken as input to an ANN without feedback loops where the outputs of the neurons go to the next layer but not to the previous layer. Data was taken with the non-invasive Emotiv Epoc device. The presented method was compared with the classification of the EEG data using ANN only. The experimental results of the 5 chosen users showed an improvement of 8% with respect to the classification based on ANN without SOM.

Likewise, Lekshmi [47] presents a comparison between two feature extraction approaches applied to forward propagation ANN to classify EEG signals at the different frequency levels of the beta, alpha, theta and delta waves produced by human emotions , The first is principal component analysis (PCA) and the second WT. The study showed that PCA with ANN is more accurate compared to WT. In addition, it is concluded that the accuracy of the artificial neural network can be improved by the use of various optimization techniques, but they do not mentioned which can be.

The aforementioned works were evaluated with the dataset BCI Competition IV 2a, on the other hand, He et. al. proposed a methodology based on a Bayesian Network architecture for feature extraction, a SVM for classification and and Gaussian Distribution to simulate the probability density function of the continuous EEG signals over the BCI III 3a dataset. The proposed methodology obtained an accuracy of 0.93 but with a high variability between the subjects [26], [55] proposed a novel feature weighting and regularization method that utilizes all CSP features to avoid information loss. Results on BCI Competition III Dataset IIIa demonstrate that the proposed method enhances the classification accuracy comparing to the conventional feature selection approaches, the accuracy obtained with a LDA classifier was 0.92, hhowever, the proposed approach did not show the consistency to obtain good performance in all subjects.

Notwithstanding, currently, methods based on deep learning has shown better performance among the works that are in the state of the art [82], [62].

2.2. Deep Neural Networks

The deep learning paradigm was developed in recent years thanks to recent advances in hardware, specifically in graphics processing units (GPUs) that allows the parallelization of the Deep Learning Algorithms through the GPU's cores. It is based on existing algorithms such as ANN, but densifying their architectures (number of hidden layers, number of neurons per layer, among others). Thanks to this depth, interconnections are achieved that allow high-level abstractions that extract discriminating characteristics for the learning process in the classification task performed by the network. Currently there are several types of network such as Recurrent Neural Networks [50], Deep Belief Networks [85], Neural Networks with Time Delays [54], Convolutional Neural Networks [84], among others.

With the advent of Deep Learning techniques thanks to new hardware capabilities, considerable performance has been achieved in other areas of knowledge such as computer vision, in [19] a comparative analysis is presented, comparing different types and Architectures of Deep Neural Networks (DNN), using a technique called Pooling to reduce dimensions, being Max-pooling the most used in the state of the art, although other functions of pooling can improve the performance. As conclusions it was observed that the choice of the activation function is important for the neural network to handle the problem of gradient decrease, in addition the key to a good activation function is that it should cause shortage and limit the decreasing gradient flow.

Thanks to this performance, works has begun on the classification of EEG signals taken from emotional states. Jia et all. [32] propose a semi-supervised learning framework for the recognition of affective states from EEG signals. For this they used a two-level procedure, which includes both information from unsupervised tags and information from supervised structures to make the channel decision, then add a Boltzmann Machine with Restriction to sort. The experiments performed on the actual EEG dataset showed a good result in the critical selection of the channel and the superiority of the proposed method in the recognition of the affective state.

In [85] they present an advanced model of deep learning to classify two categories from data taken from signals of emotions obtained through EEG. A Deep Belief Network (DBN) with differential entropic characteristics extracted from several channels was trained, and Hidden Markov Models (HMM) were added to accurately capture a more reliable emotional state change. The performance of the proposed system was compared against techniques such as Nearest K-Neighbors (KNN), SVM, and Graphic Extreme Learning Machine with regularization and (GELM). The results showed that the DBN and DBN models with HMM improve the accuracy in the classification compared to those found in the state of the art.

Methodologies based on deep learning with filter banks and CSP have been also proposed with promising results. Merinov et al. [52] proposed a spatial filter network (SFN). Their approach was evaluated using the BCI competition III dataset 3a and the BCI competition IV dataset 2a. In the preprocessing step, time segments between 0.5-4 seconds from the instruction cue on set, are taken for obtaining segments of 3.5 seconds as training set, then, the signals were passed through the frequency bands range of the original CSP algorithm. In the SFN, authors utilized cross-entropy loss function, a batch learning scheme and data augmentation. Each learning epoch is composed by 100 batches, in a 5-fold cross validation loop. This approach reported a best accuracy of 0.65. Sakhavi et al.[67] used the Filter-Bank CSP (FBCSP) proposed in [6]. A bank of 9 filters from 4 to 40Hz, with a width of 4Hz was initially applied for extracting a set of features that were then selected by a mutual information feature selection algorithm. For the CSP process, four pairs of spatial filters were picked for each frequency band, finally the proposed parallel CNN and linear architecture were implemented to decode the final movement intention class. Mean accuracy for all subjects of 70.60% in the BCI IV Competition 2a dataset was obtained. Yang et.al. [82] used Convolutional Neural Networks (CNNs) to classify multiple classes in EEG signals obtained from motor imagery using Augmented CSP (ACSP) features based on a varying the frequency bands with different bandwidths to cover as many bands as possible testing over the BCI IV competition 2a dataset. They proposed a way to select the feature maps, namely frequency complementary map selection (FCMS), and compared it with random map selection (RMS) and with the selection of all feature maps (SFM). Average cross-validation accuracy of 68.45% for FCMS and 69.27% for SFM was achieved. Nonetheless, approaches based on Filter banks with CSP highly depends on the filter bands selection and this can cause loss of relevant information.

In [33], Jingwei et al. propose Multi-Scalar Convolutional Neural Networks, extracting high-level features called Deep Motor Features (DMF) that can be learned by the network through non-linear hierarchical mapping. As a result a 100% accuracy was obtained in the classification of tasks of motor imagination.

On the other hand, Walker proposes a method to define the convolution filters of a CNN for separate groups of electrodes in similar regions of the brain in the classification of signals of motor imagination in real time [78]. The activation function chosen for this was the Tanh in the intermediate layers obtaining an accuracy of 80.08% in one of the test subjects, significantly higher than 50% obtained by a random model.

One of the networks that has obtained very good performance in the classification of EEG signals is the Recurrent Neural Networks due to they are designed to work with time series and allow capturing this temporary information [25].

2.2.1. Recurrent Neural Networks

In certain recognition tasks it is necessary to have prior information of an event that is required to classify, this is the case of speech recognition, language modeling and videos in which it is necessary to have temporary information of previous frames to determine what is happening in the present frame [25]. The classic artificial neural networks do not have the capacity to take into account previous information, however the Recurrent Neural Networks solve this problem since these are networks that internally have a loop that allows the information to persist in time, endowing the memory network. In the diagram 2.2 a basic scheme of a RNN operation with the loop that allows the network to have a small memory can be see.

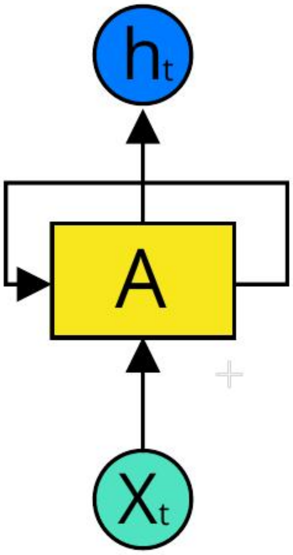

Figure 2.2.: Recurrent Neural Network, being A, a chunk of neural network, h_t, the output and X_t the input.

Internally, a RNN can be considered as a set of interconnected networks, so the first network has an input $\mathbf{x}_0$, being this the signal in the time t_0 and having its respective output h_0, consecutively this network is connected to an another network with an input x_1 and output t_1 in the next time step, having the same number of connected networks as x_t entries. As can be see, RNN are highly compatible with time series and sequences. Figure 2.3 allows to visualize the process.

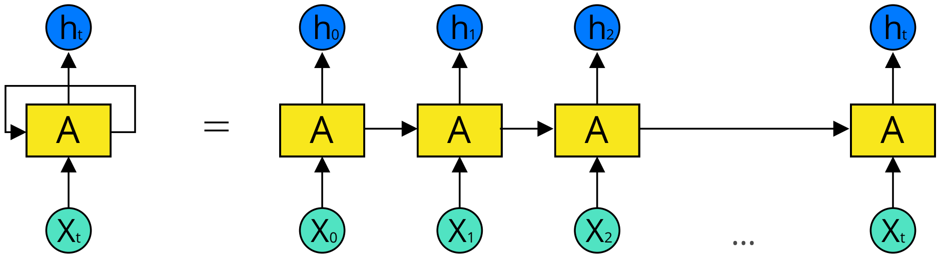

Figure 2.3.: Inputs over time of a Recurrent Neural Network. As can be see a Recurrent Neural Network can be interpreted as a chain of networks. Being a determined output for a specific input in time.

The problem with the RNN is that the longer the chain of connected networks, the more difficult it will be to relate a certain output to an input that is too far away temporarily. To solve this, a modification to the basic RNN was proposed, this modification gave way to the Long Short Term Memory networks (LSTM) [29].

Although RRN are naturally appropriate for time series, this paper proposes the use of CNN since these have had great performance in the task of classifying EEG signals as shown by [78]. To this we must add that the filter nature of CNN allows signals to be worked without the need to use too complex pre-processing algorithms.

2.2.2. Convolutional Neural Networks

A convolutional network is a type of DNN with forward feeding based on the multilayer perceptron, they are based in the mathematical convolution. It takes one or several inputs and passes them through several layers of neurons, where each neuron represents a linear combination of its input, then pass through a function of non-linear activation, which in the case of this work is the function ReLU, to pass again to another layer of neurons, this happens successively until reaching the output layer. They have the ability to learn highly nonlinear functions and were introduced for the first time by [45] showing great performance in face detection and classification of handwritten characters.

Although they are based on the multilayer perceptron, being considered as deep learning, their operation is initially given by multiple convolutional layers that allow the extraction of characteristics given by the filter nature of the convolutional neurons, which alternate with pooling layers that are in charge of grouping the characteristics correlated with some grouping function like the average or the maximum, finally the classification stage is presented.

In a convolutional network, the connection of the neurons is given by a sparsely connected architecture (Sparsely Connected), where the neurons in each layer are not connected to each neuron of the previous layer, thus reducing the probability of overtraining [78]. The output of each convolutional neuron is given by the equation:

$$Y_j = g(b_j + \sum k_{ij} \otimes Y_i) \tag{2.16}$$

Since Y_j is the output of the neuron j, a matrix generated by the linear combination of the outputs Y_i of the neurons in the back layer each multiplied by the convolution core k_{ij} corresponding to that connection, this in turn added to the influence factor b_j, g is the non-linear activation function [46]. Figure 2.4 shows the CNN behavior.

Due to the high abstraction capacity of CDNN is difficult to visualize which features were taken into account for the classification process, being hard to have clearly a physiological intepretation of the inputs' of the neurons (weights, biases). However, for the filter nature of a CNN the highest weights of the network will be linked with the most relevant features in the classification problem.

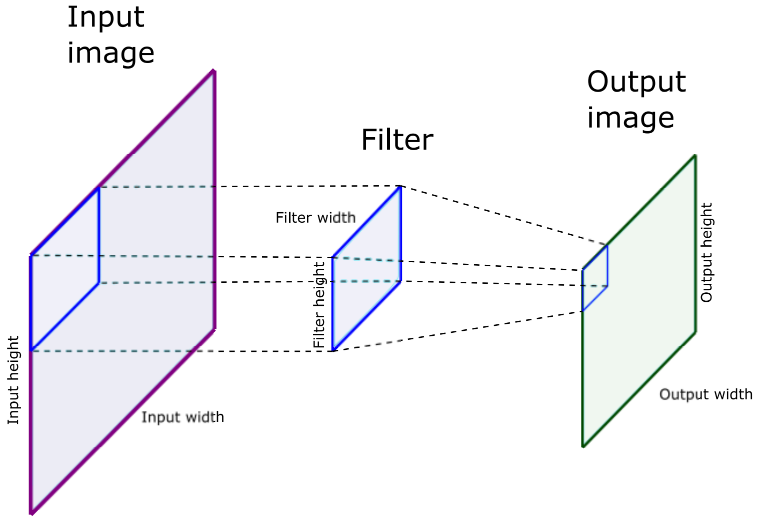

Figure 2.4.: Convolutional Neural Network Behavior. The original input is passed through a convolutional filter, which reduces the dimensions of the input by matching them with the filter dimensions.

Back Propagation

Let $\delta^{(l+1)}$ be the error term for the $(l+1)-st$ layer in a network with a cost function $J(W,b;x,y)$ where (W,b) are the parameters and (x,y) are the training data and label pairs. If the $l-th$ layer is densely connected to the $(l+1)-st$ layer, then the error for the $l-th$ layer is cumputed as:

$$\delta^{(l)} = ((W^{(l)})^T\delta^{(l+1)}) \bullet f\,'(z^{(l)}) \tag{2.17}$$

and the gradients are:

$$\nabla^{(l)}_{W}J(W,b;x,y) = \delta^{(l+1)}(a^{(l)})^T, \tag{2.18}$$

$$\nabla^{(l)}_{b}J(W,b;x,y) = \delta^{(l+1)} \tag{2.19}$$

f the $l-th$ layer is a convolutional and subsampling layer then the error is propagated through as:

$$\delta^{(l)}_k = upsample((W^{(l)}_k)^T\delta^{(l+1)}_k) \bullet f\,'(z^{(l)}_k) \tag{2.20}$$

Where k indexes the filter number and $f\,'(z^{(l)}_k)$ is the derivative of the activation function. The upsample operation has to propagate the error through the pooling layer by calculating the error with respect to each unit incoming to the pooling layer. Finally, to calculate the gradient with respect to the filter maps, border handling convolution operation is used to rotate the error matrix $\delta^{(l)}_k$:

$$\nabla_{W^{(l)}_k}J(W,b;x,y) = \sum_{i=1}^{m}(a^{(l)}_i) * rot90(\delta^{(l+1)}_k, 2), \tag{2.21}$$

$$\nabla_{b^{(l)}_k}J(W,b;x,y) = \sum_{a,b}(\delta^{(l+1)}_k)a,b\cdot \tag{2.22}$$

Where $a^{(l)}$ is the input to the $l-th$ layer, and $a^{(1)}$ is the input array. The operation $(a^{(l)}_i)*\delta^{(l+1)}_k$ is the valid convolution between $i-th$ input in the $l-th$ layer and the error respect to the $k-th$ filter.

Rectified Linear Units (ReLU)

Normally the output of a neuron as a function of its input is F (x) = tanh (x). Due to saturating nonlinearity, downward gradient training becomes much slower compared to a function F (x) = max (0, x) which has non-saturating non-linearity. This type of function is known as ReLU.

The ReLU function has advantages with respect to the activation functions traditionally used, among them is its speed to be calculated and do not suffer from the problem of gradient leakage [57].

$$ReLU(z) = max(0, z) \tag{2.23}$$

In [40] Graphically show that convolutional neural networks equipped with ReLU activation functions train much faster than with tanh for image classification.

Dropout

Dropout is a technique that consists in canceling certain neurons by putting their output in 0. The neurons that are to be annulled are those that have a probability of 0.5. As neurons are nullified, they do not contribute to the backpropagation of the network. This way every time you change the network entry will change the architecture because different neurons will be overridden by the Dropout, without ignoring the already determined weights.

In this way, the network is forced to learn more robust characteristics that are useful when correlating with the random information obtained from other neurons. [70].

2.2.3. Parameterization

ANN hyper-parameters tuning defines the network architecture, this process has great importance since one of the weaknesses of ANNs is its great dependence on the correct selection of the free parameters of the same and there is no fixed rule or method defined for this because depending on the data set used depends on the hyper-parameters selection, as is well known, an architecture designed for a specific task may not perform well in another different classification task and it is necessary to retrain the network to adjust the weights to this new classification task.

The hyper-parameters selection process is complex and must be carried out by a specialist who based on their experience and intuition will make the selection of them, many times rules of thumbs and brute force search are required.

Each selected hyper-parameter set has a defined architecture, if some parameter changes a different architecture was obtained, different from the initial one, that is why every time a set of parameters is selected by a specialist, the network should be trained again and tested to calculate its performance. As it can see, the process becomes complex and requires a lot of time because there are deep architectures that can take days and even weeks to be trained [40].

It is due to these difficulties that it was proposed to use an optimization algorithm that automatically selects the hyper-parameters of the network in order to obtain the highest accuracy.

2.3. Optimization Strategies

Mathematical optimization consists of selecting from a given set of elements the most adequate one that allows obtaining the best possible result according to some specific criterion. For this, what is done is to optimize or minimize a function by selecting the best input values and calculating its result.

Algorithms for optimization can be roughly divided into two categories: exact algorithms and heuristics. The difference between the two categories is that exact algorithms are designed in such a way that it is guaranteed that they will find the optimal solution in a finite amount of time. Heuristics do not have this guarantee, and therefore generally return solutions that are worse than optimal [69].

In deep learning optimization methods shows to be promising approaches, [9] proposed a greedy algorithm to initialize the weights of a Deep belief network using a version of the wake-sleep algorithm for the classification of digits. As results it was observed that the greedy algorithm can quickly find a good set of parameters, even in deep networks with millions of parameters and hidden layers, although the algorithm is unsupervised learning can be applied to labeled data, in addition the algorithm had the ability to learn a generative model using the MNIST database that exceeded the discriminative methods found in the state of the art.

Bayesian methods, although have shown good performance in the hyper-parameters optimization of machine learning algorithms, use probabilistic surrogate models like Gaussian proccess to approximate and minimize the validation error function, however probabilistic methods require a low covariance as well as many functions evaluations, reason why that are inefficient in the task of hyper-parameters tuning of deep networks since these are highly expensive to evaluate. Because of this, [30] proposed a deterministic method called HORD for the optimization of hyperparameters that uses Radial Basis Function (RBF) as error surrogate which only takes into account the most promising parameters through dynamic coordinate search which requires much less function evaluations, the proposed method outperforms Bayesian methods that have currently performed as well as GP, SMAC and TPE.

The method was tested in the MNIST and CIFAR-10 databases, showing to be 6 times faster than GP-EI in the optimization of 19 hyper-parameters.

Nevertheless, the use of optimization algorithms for the training of a DANN is still a challenging task because the training of the network is already a process that takes a lot of time and resources, furthermore it is necessary to prove a lots of architectures. Due to this, metaheuristics algorithms thanks to its free-derivative nature allows to train and fine tune the free parameters of deeper networks without being highly expensive or time-consuming.

2.3.1. Meta-heuristics

Metaheuristics Algorithms add rules at a higher level that not only use a heuristic function (trial and error) but also guide the search to increase its efficiency. In their original definition, are solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space [21]. All algorithms that use a 'surrogate' or auxiliary objective function fall into this category.

Meta-heuristic algorithms are approximate optimization algorithms based on behaviors found in nature. They are characterized by two fundamental concepts, first the intensification that is to search in a local and intense way, second the diversification, responsible for ensuring that the algorithm explores the search space globally [83].

2.3.2. Derivative-Free Algorithms

There are many problems for which the true objective function is quite costly to evaluate. When this occurs, the evaluation of moves may become prohibitive, even if sampling is used. An effective approach to handle this issue is to evaluate neighbors using a surrogate objective, i.e., a function that is correlated to the true objective, but is less computationally demanding, in order to identify a (small) set of promising candidates (potential solutions achieving the best values for the surrogate). Among this algorithms there are Gutmann's Radial Basis Function (RBF) [24], the stochastic RBF method [65], and the kriging-based Efficient Global Optimization method (EGO) [34], among others.

2.3.3. Derivative-free Optimization Based on Radial Basis Function

One of the main characteristics of certain machine learning models, such as Deep Learning is the requirement of the maximization of a performance criterion in which a description in analytic form is not available. For this reason, the use of classical optimization algorithms are discarded since the first or higher order derivatives cannot be computed. In some cases, first and second order derivatives can be estimated using finite differences, although this requires a large number of evaluations; e.g. in case of a n-dimensional space at one point it would be required to evaluate the function at $n+1$ points at least. However, when the evaluation of the objective function is computationally expensive (main feature of Deep Learning strategies), it becomes in a non-viable application in practice.

Derivative-free optimization (DFO) is an area of mathematical optimization dedicated to create optimization approaches depending on zero-order information exclusively [18]. With these approaches, the interest is based on trying to optimize an objective function f performing a small number of evaluations, avoiding results in a prohibitive computing time. Since f is optimized in a space domain described only by upper and lower bounding constraints and some decision bounds are restricted to integer values; these models could be considered as a box-constrained problem with integrality constraints. An extensive description of the large amount and main features of DFO algorithms could be found in [16].

To implement the DFO algorithms for hyper parameter optimization in a Deep Learning model, it should be considered a main drawback. The parameters values changes during optimization process should describe a valid network architecture e.g. typically, a network architecture does not contain empty layers, so this must be imposed. The way to apply optimization to a network architecture is described as follows. for an upper bound l concerning the maximum size of a single layer belonging to a fixed network architecture, in the formulation of the DFO algorithm $l+1$ decision variables must be considered to determine the size of the hidden layers, saying $x_1...x_l$ without loss of generality. The decision variable x_{l+1}; which aims to determine the number of units in the layers, is constrained to be an integer belonging to $[1, l]$, since network layers cannot have non-decimal number of units. For example, if $(x_1...x_l) = (20, 10, 30, 10, 40, 50)$ for a network architecture of 6 layers, the model must be constructed with hidden layers of size 20, 10, 30, 10, 40 and 50 respectively. Additionally, it is important to notice that parameters values are symmetric for the structure of the network, since several values of $(x_1..x_l)$ corresponding to the same structure can be obtained by several permutation of the decision parameters. Symmetry in the DFO model guarantees a solution in a large space domain using few objective function evaluations. The approach described below could be implemented using the open source RBFOpt toolbox [17] available online. The tool allows to implement all experiments modeling the network as a black-box (box-constrained) problem delimiting the number of units for each layer belonging to the network as the space domain. The use of the toolbox for the proposed model will be described in the next chapters.

2.4. Gutmann's RBF Method for Black-Box Optimization

In [17] the radial basis function is defined as: Let $\Omega := [x^L, x^U] \subset \mathbb{R}^n, \Omega_I := \Omega \cap (\mathbb{Z}^q \times \mathbb{R}^{n-q})$ and its assume that the box constraint on the first q variables have integer endpoints. Given k distinct points $x_1, ..., x_k \in \Omega$, the radial basis function interpolant s_k is defined as:

$$s_k(x) := \sum_{i=1}^{k} \lambda_i \phi(||x - x_i||) + p(x) \tag{2.24}$$

where $\phi : \mathbb{R}_+ \to \mathbb{R}, \lambda_1, ..., \lambda_k \in \mathbb{R}$ and p is a polynomial of degree d. The minimum degree $d_m in$ to guarantee existence of the interpolant depends on the form of the function ϕ. Typically, the polynomial is picked to be of degree exactly $d_m in$ in practical implementations. If $\phi(r)$ is cubic or thin plate spline, $d_m in = 1$ and with $d = d_m in$ the next interpolant form is obtain:

$$s_k(x) := \sum_{i=1}^{k} \lambda_i \phi(||x - x_i||) + h^T \begin{pmatrix} x \\ 1 \end{pmatrix} \tag{2.25}$$

where $h \in \mathbb{R}^{\ltimes+\nVdash}$. The values of λ_i, h can be determined by solving the following linear system:

$$\begin{pmatrix} \Phi & P \\ P^T & 0_{(n+1)\times(n+1)} \end{pmatrix} \begin{pmatrix} \lambda \\ h \end{pmatrix} = \begin{pmatrix} F \\ 0_{n+1} \end{pmatrix} \tag{2.26}$$

with:

$$\Phi = (\Phi(||x_i - x_j||))_{i,j=1,...,k}, P = \begin{pmatrix} x_1^T & 1 \\ \vdots & \vdots \\ x_k^T & 1 \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_k \end{pmatrix}, P = \begin{pmatrix} f(x_1) \\ \vdots \\ f(x_k \end{pmatrix} \tag{2.27}$$

The algorithm follows the next general scheme:

- Initial step Choose affinely independent points $x_1, ..., x_n+1 \in \Omega_I$ using an initialization strategy. Set $k \leftarrow n + 1$
- Iteration step Repeat the following steps.

  i Compute the RBF interpolant s_k to the points $x_1, ..., x_k$, solving.

ii Choose a trade-off between exploration and exploitation.

iii Determine the next point x_{k+1} based on the choice at last step.

iv Evaluate f at x_{k+1}.

v If we exceed a prescribed number of function evaluations, stop. Otherwise, set $k \leftarrow k + 1$.

At Iteration step (ii), exploration implies trying to improve the surrogate model in unknown parts of the domain, whereas exploitation implies trying to find the best objective function value based on the current surrogate model. To turn the general scheme above into fully specified algorithm, we must describe how to choose points in the initial step, and an implementation of Iteration steps (ii) and (iii).

3. Methods

This chapter describes the main details of the proposed method for classifying motor imagery EEG signals based on deep Convolutional Neural Networks with RELU activation functions, which was optimized by using the Gutmann's RBF method for black-box optimization described in the chapter 2.

As will be seen later in this chapter, for addressing the specific objective 1, the electrodes space was defined following the methodology proposed in [78], which allowed to establish a spatial correlation between the signals. Then, a feature extraction algorithm based on PSD was implemented to obtain a representation of the signals that was used as input to a deep convolutional neural network designed to extract relevant deep features used for the end classification by the last network layer.

Once the classifier was designed, the specific objective 2 was addressed by the analysis of the state of the art on meta-heuristic hyper-parameter optimization algorithms in deep learning. The Gutmann's RBF Method for Black-Box Optimization was selected, which was also implemented in an open source toolbox that automatically parallelize and tune the free hyper-parameters of the network. Thus, both optimization approach and learning CNN model were implemented on a framework based on TensorFlow library and CUDA for python, which allowed the parallelization of the process in the GPU cores. This implementation allowed to obtain an optimal CNN architecture respect to the accuracy classification.

The third objective was two-way addressed. First, the performance of the unoptimized learning CNN model, with hiper-parameters selected heuristically, was evaluated using the multiclass BCI competition IV-2a dataset; then, the optimized network, designed by using the Gutmann's RBF Method for Black-Box Optimization, was two-fold evaluated, using the same BCI competition IV-2a dataset and the BCI competition III-2a, which corresponds also to a multiclass problem.

3.1. Dataset Description

Performance evaluation was initially carried out using the benchmark BCI Competition IV dataset 2a [10] due to its complexity and extended use in the state of the art. This dataset consists of EEG signals from 9 subjects performing four motor imagery tasks namely left hand, right hand, feet and tongue movements. The database is originally in .GDF format, which is used for biomedical signals. To access the signals, the MATLAB Biosig toolbox was used. The signals are in an array of 672528 x 25, the last three rows correspond to electrodes intended to acquire EOG signals, which are not necessary in the classification task, so they have been eliminated. Since the matrix contains values that are not typical of the classification task, it was necessary to extract only the classes, drawing a matrix for each trial of each class, resulting in 72 matrices of 22x314 for each class in each test, obtaining a total of 2592 matrices. Each of the matrices was transposed so as to have the electrodes in the rows and later to make a permutation that allowed the electrodes to be grouped by distances, thus seeking that the electrodes more adjacent to each other are in a group, and that each electrode does not belong to more than one group. In this way a spatial correlation i obtained between the electrodes that are in the same region of the brain that is activated in front of certain tasks of motor imagination.

With the aim of evaluating the generalization of the proposed learning model, the validation of the model was again carried out over the BCI Competition III 3a data [1]. This dataset consists of EEG signals from 3 subjects performing four motor imagery tasks namely left hand, right hand, feet and tongue movements. The database is originally in .GDF format too. The dataset has 60 EEG channels, using the left mastoid for reference and the right mastoid as ground. The EEG was sampled with 250 Hz, it was filtered between 1 and 50Hz with Notchfilter on. This dataset was made exactly the same treatment as the BCI Competition IV dataset 2a but without realizing the channels grouping.

It is important to clarify that the tests performed on this dataset were made with the architecture found with the automatic hyper-parameter Tuning, which was run on the subject with the worst performance obtained by the manual hyper-parameter tuning.

3.2. Preprocessing

Since CNN requires a large amount of data, and spatial relationships between this data, which are not necessarily found in one-dimensional MI signals, the preprocessing stage is herein composed of two steps, the first one seeks to expand the number of training and testing data, for that purpose, we employed a window-based approach proposed in [78] to create pseudo images to feed the CNN and the second, reorganize the order of the electrodes according to its proximity into the scalp, seeking to establish a spatial correlation between the signals.

Windows Extraction

A window-based approach over temporal signals was carried out, in order to extract the enough amount of pseudo images as [78] proposed, but with few differences in the overlapping. For the sake of generality, we introduce some notation; for each subject suppose a matrix $\mathbf{X} \in \mathbb{R}^{T \times N}$ where T is the amount of temporal observations and N is the number of channels in the recording process. The main purpose is to use a sliding window of size τ to slice the temporal axis in several overlapping windows given by $\mathbf{x}_i \in \mathbb{R}^{\tau \times N}$ $\forall$ $i = 1, \ldots, T - \tau + 1$. In order to avoid over fitting, the overlap size was 95%(this value was selected as rule-of-thumb) instead of the maximum overlapping size that gives an almost identical pseudo image, redefining the classification task. This process can be seen on figure 3.1.

Channels Grouping

Commonly, CNNs are trained using 2D data, where spatial correlation is guaranteed. However, motor imagery signals are not spatially organized. To overcome this problem, we ensure spatial correlations among channels by applying k-means clustering over them, in line with its position in the scalp, unlike [78], where K Nearest Neighbor was applied. Position in the scalp was based on the electrode montage corresponding to the international 10-20 system used on [10] as can be see in 3.2, with the value of the coordinates representing the distance between the electrodes, having an inter-electrode distance of 3.5 cm, obtaining the spatial matrix:

SP=[0 0;-7 -3.5;-3.5 -3.5;0 -3.5;3.5 -3.5;7 -3.5;-10.5 -7;-7 -7;-3.5 -7;0 -7;3.5 -7;7 -7;10.5 -7;-7 -10.5;-3.5 -10.5;0 -10.5;3.5 -10.5;7 -10.5;-3.5 -14;0 -14;3.5 -14;0 -17.5]

Electrode 1 takes the coordinates (0,0) and the others take its coordinates according to its

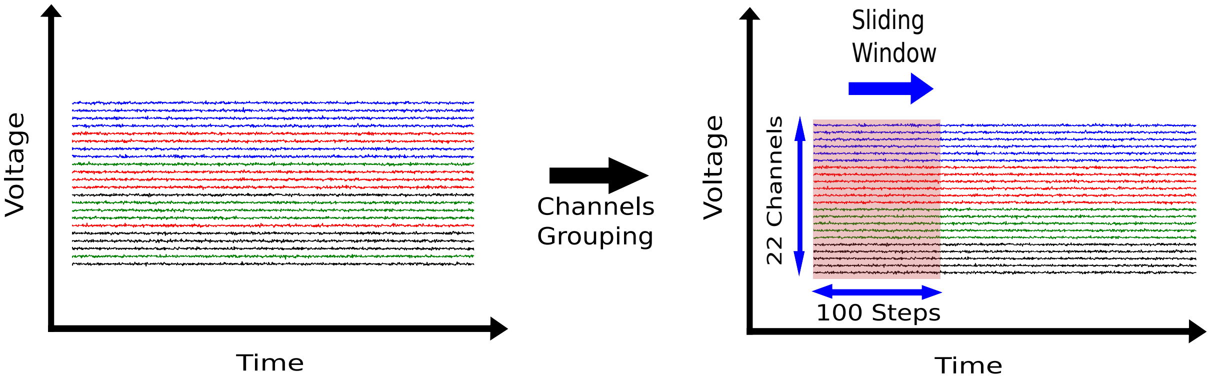

Figure 3.1.: Preprocessing of motor imagery signals. Each element of the vector corresponds to a EEG channel, being in turn the rows of each pseudo image with the first element of the vector being the first row and the last element the last row.

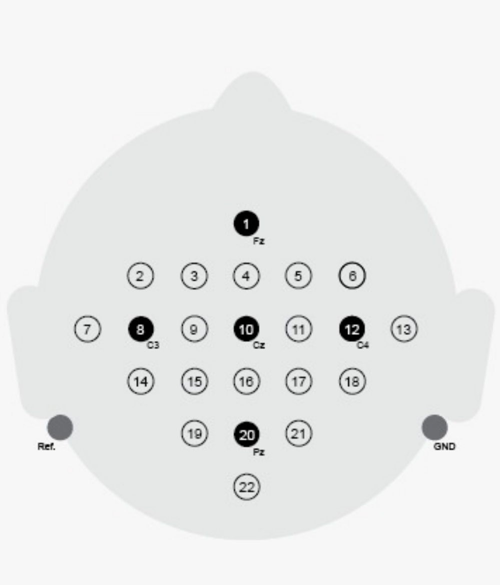

Figure 3.2.: Location of the electrodes on the skull. Electrode 1 takes the coordinates (0,0) and the others take its coordinates according to its distance in cms to electrode 1. taken from [10].

distance in cms to electrode 1. Then, the 22 channels were organized in four groups, within each group channels were ranked according to the distance to the center of its group, finally, all groups are stacked into a single matrix. This method was used by Walker et.al. [78].

To this spatial matrix was applied a K-means clustering algorithm, selecting 4 clusters for this and finding the Euclidean distance between the electrodes.The K-means algorithm grouped the electrodes according to its distance making sure that each electrode belong to only one group at a time. The order of the electrodes found by the K-means was = [4, 1, 3, 5, 10, 8, 9, 14, 2, 7, 15, 12, 6, 11, 13, 18, 20, 21, 16, 22, 19, 17], each element of the vector corresponds to a EEG channel, being in turn the rows of each pseudo image with the first element of the vector being the first row and the last element the last row. See figure 3.1.

As can be deduced, it is achieved in this way to obtain information from each electrode for each class, while at the same time obtaining a spatial relationship between the electrodes, thereby emulating the correlation given by the adjacency of pixels in an image.

3.3. Feature Extraction Based on Power Spectral Density

The main drawback of the periodigram estimator is the high variance [11]. Conversely, the Welch method presented in [80] is a more efficient estimator which is based on overlapping sections, where a sliding window is used to find the periodigram in those segments. Finally the result is obtained by averaging the estimations of all sections as is explained in [60]. Figure 3.3 shows this PSD estimation approache.

In this work, PSD was employed for feature extraction in EEG signals using FFT, this is done for each channel; that is, given a record $\mathbf{x}_i \in \mathbb{R}^{\tau \times N}$ we computed periodigrams for the channel $j \; \forall j = 1, \ldots, N$. PSD implememtation available in the SciPY package [35] was used. The parameters selected were those configured by default in the function of the package but with the sampling frecuency according to the signals, established in 250.

3.4. Deep Convolutional Neural Network Architectures

In the presented architectures, the input layer is configured with the dimensions of the input data. The convolutional layers used are responsible for performing the mathematical process of convolution on the pseudo images generated. Each of these layers is activated by the ReLU activation function which determines the output value of each neuron. While the Max pooling Layers are responsible for grouping the original input data, finding the maximum value between the values that are inside the pool size, to later pass this value as a summary of characteristics on that area. As a result, the size of the data is reduced by a factor equal to the size of the sample window on which it is operated. The dropout layers are

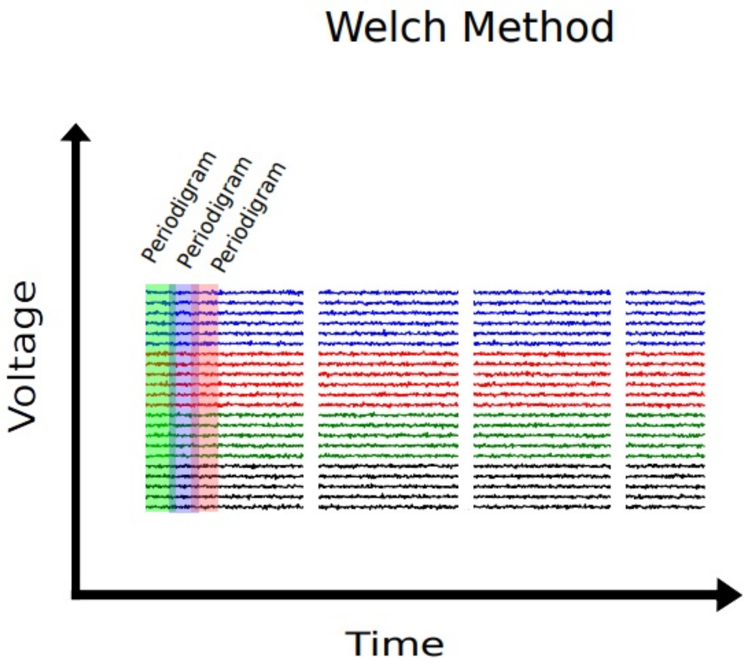

Figure 3.3.: PSD estimation approaches. A sliding window is used to find the periodigram in those segments. Finally the result is obtained by averaging the estimations of all sections.

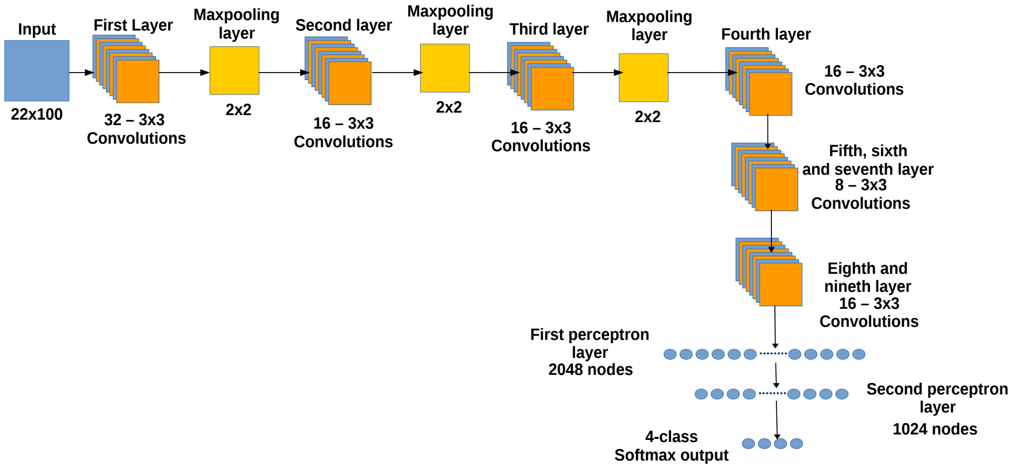

Figure 3.4.: Configuration of the Deep convolutional neural network. Nine convolutional networks followed by two dense layers and a softmax layer were the base for the manual and automatic hyper-parameter tuning. Dropout layers were not included in this scheme.

responsible for disconnecting a portion of neurons from the previous layer to avoid overfitting in this way. The layers that are between the input layer and the output layer are called hidden layers, these as well as the number of neurons by hidden layers determine how deep the architecture is, being evident that the greater the depth of the network, the greater the abstraction capacity of the network, however greater is the computational cost to train it [40].

An pipeline of the architecture with the manual hyper-parameter tuning can be seen in figure 3.4. It is important to note that in this scheme Dropout layers were not included. Convolutional Neural Networks and PSD are able to extract feature maps from non-preprocessed data obtaining high level abstractions over input data by using trained filters[44]. The weights of these filters are updated during the training process by stochastic gradient descent or any other variation of the gradient descent algorithm.

3.5. RBFOpt

The DFO algorithm used in this work is based on a surrogate model of the objective function f, in which random initial points are evaluated with the aim of balancing the exploration of unknown areas of the domain. This exploitation allows to identify the most feasible areas in which the global optimum for an specific set of parameters could be found. To do so, a radial basis function is employed combined with a polynomial tail of degree 1. The algorithm evaluates the function f at $n+1$ points randomly selected according to a generated Latin Hypercube design within the domain (S). Then, the algorithm fits the surrogate model for

f interpolating the set of points S and chooses the next evaluation point (N) according to two main criteria: the value of the surrogate model at N and the euclidean distance to the closest point in S. Thus, the problem becomes a bi-objective optimization containing in general, a possibly infinite set of optimal points; however, since the main purpose of the algorithm is determining a single point in N; both objectives are normalized and reduced to a single one using a weighted combination of weights w that determines the trade-off. Besides, the weights w are chosen according to a cyclic strategy that oscillates between favoring the maximum-minimum distance criterion. This feature allow to emphasize the exploration of unknown parts of the domain, and favors the surrogate model value criterion by extracting the choice of points that are supposed to have the largest values for f (maximization problem). The resulting objective function for the choice of N is be solved using a simple default genetic algorithm. Thus, it is important to remark that the genetic algorithm is not applied to the original problem but the auxiliary optimization problem which determines the correct choice of N. Besides, evaluating f in the original space domain is significantly more computationally expensive, compared to the evaluation of the value of the surrogate model and the minimum distance from N to S. Finally, once N is determined, f is evaluated at N, results are computed with S, and the iteration would be completed. The number of iterations is selected according to a satisfaction criteria such as "maximum number of iterations".

For the hyper-parameter optimization toolbox implementation the steps that were taken were the following:

- First, a function that constructs the network architecture model was defined based on the parameters to be optimized (number of neurons per convolutional layer, number of neurons per dense layer, learning rate). These parameters are taken from a vector $\mathbf{x}$ defined for the selection made by the toolbox according to the search space defined by a lower bound and an upper bound.

- Second, an evaluation function was defined that receives as input the current network architecture model and gave as output an estimation of the current network performance regarding the accuracy with these parameters through 5-fold cross validation.

- Third, performance function was defined, this function is responsible for call the network architecture constructor defined in the first step and the evaluation function defined in the second step. It gave as output the higher accuracy obtained.

- Fourth, the parameters of the optimization are selected in the rbfopt configuration method. The parameters are: the number of network hype-parameters to be optimize, two vectors with the lower and upper bounds, the data type of each value in the bounds vectors and finally the performance function defined in third step.

- Finally, the black-box optimization method was invoked. In this stage the algorithm select a set of hyper-parameters from the search space, calls the rbfopt configuration method, calls the performance function defined in the third step, the latter build the network with the current set of hyper-parameters and computed the accuracy through the 5-fold cross validation. The algorithm only trains completely the 50 most promising architectures and the rest are discarded. In the output a vector with the optimal parameters and the accuracy performance with these are obtained.

The tuning of the hyper-parameters of the network for the training was done in Keras, this allowed the algorithm of construction and testing of the network based on the parameters defined by the toolbox RBFOpt as well as the process of selection of the hyper-parameters to be done in parallel over the cores of the GPU.

3.6. Performance Metrics and Evaluation

Performance evaluation is carried out using the benchmark BCI Competition IV dataset 2a and BCI Competition III dataset 3a. The data preparation was described in section 3.1 Dataset Description. As result of that process, for the BCI Competition IV dataset 2a 2592 matrices were obtained. After the data augmentation scheme based on windows extraction, approximately (few variations per subject) 13550 pseudo images of 100x22 per subject were obtained. After the feature extraction process the size of the pseudo images changed to 51x22. For each case, a learning model was trained per subject, and performance was computed using a 5-fold cross-validation strategy, i.e. 80% (10840) of subject samples for training and 20% (2710) for testing. Thus, nine evaluations, one per subject, were carried out. Performance measurements are computed as the mean of the five folds per subject, and the mean for all subjects as the model accuracy and kappa index.

For the BCI Competition III dataset 3a, the following pseudo images were obtained per subject, subject 1, 6277, subject 2, 4264, subject 3, 4101 with the the same size mentioned above because the same process was performed on both datasets, all other conditions remained the same for this dataset.

3.6.1. K-fold Cross Validation

In k-fold cross-validation, sometimes called rotation estimation, the dataset D is randomly split into k mutually exclusive subsets (the folds) $D1, D2, ..., Dk$ of approximately equal size. The inducer is trained and tested k times; each time $t \in \{1, 2, ..., k\}$, it is trained on $D \backslash Dt$ and tested on Dt. The cross-validation estimate of accuracy is the overall number of correct classifications divided byt the number of instances in the dataset. Formally, let $D(i)$ be the test set that includes instance $xi =< vi, yi >$, then the cross-validation estimate of accuracy [39]:

$$acc_c v = \frac{1}{n} \sum_{<vi,yi> \in D} \delta(L(D \setminus D(i), v_i), y_i) \tag{3.1}$$

The cross-validation estimate is a random number that depends on the division into folds.

3.6.2. Accuracy

The accuracy of a classifier C is the probability of correctly classifying a randomly selected instance, i.e., $acc = Pr(C(v) = y)$ for a randomly selected instance $(v, y) \in X$,, where the probability distribution over the instance space is the same as the distribution that was used to select instances for the inducer's training set [39].

3.6.3. Cohen's Kappa Coefficient

The kappa coefficient (K) measures pairwise agreement among set of coders making category judgments, correcting for expected chance agreement:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \tag{3.2}$$

where $P(A)$ is the propotion of times that the coders agree and P(E) is the proportion of times that we would expect them to agree by chance, calculated along the lines of the intuitive argument presented above [12].

4. Results

As was mentioned in the previous chapter, the proposed approach was evaluated by using two public and well-known databases, which are available to validate signal processing and classification methods for Brain-Computer Interfaces (BCIs), i.e., the BCI Competition IV-2a and the BCI competition III datasets. Both cases contain EEG signals captured in a multi-class motor imagery task, in which the participants were asked to imagine one of the four activities: left hand, right hand, foot or tongue movements according to a cue. Thus, three main experiments were performed, the first one evaluated the accuracy of the proposed classification method to the identification of the action imagined by participants in the BCI competition IV-2a dataset when network hyper-parameters were manually searched. Then, a second experiment, evaluated the performance obtained for the same classification task when the network hyper-parameters were automatically defined by the RBFOpt optimization algorithm. Finally, to evaluate the robustness of the optimal CNN architecture, It was used for training a learning model able to classify instances from the BCI III 3a competition dataset. The performance results obtained for these experiments are presented below.

4.1. Manually Tuned CNN Architecture with BCI Competition IV-2a Dataset

The proposed network architecture consists of nine convolutional layers, which are responsible for extracting the necessary features from the preprocessed signals. All convolutional layers are composed of 3×3 kernels. With these kernels, features maps are computed on each layer, respectively. The first three layers are followed by MaxPooling sub-sampling applied in 2×2 regions to preserve only the most relevant information, Rectified Linear Unit (ReLU) activation was chosen to avoid nonlinearities. Each convolutional layer was initialized by Glorot uniform kernel initializer [22]. Finally, was included a multilayer perceptron (MLP) that receive the information of the last convolutional layer for MI detection. The output of the perceptron is connected to an additional dropout function layer, to avoid overfitting, and finally fed a softmax function layer to classify. The softmax is the output layer of the neural network, which contains a neuron corresponding to each type of MI (class), for a total of 4 neurons. Adaptive Moment Estimation (Adam) optimizer [38] was selected as the optimization algorithm. This method is based on adaptive estimates of lower-order moments and was selected due to its computational efficiency and its capacity to work with non-stationary objectives. The parameters of the optimizer follow those provided in Kingma et.al. original paper [38], except for the initial learning rate, which was fixed to 0.0001.

The proposed CNN architecture was initially trained by setting the network hyper-parameters as follows: Kernels size of the convolutional layers: 32, 16, 16, 16, 8, 8, 8, 16, 16; the number of neurons of each dense layer: 2048 and 1024; and the learning rate to 0.0001. These values were defined then several heuristic tests. Through experimentation, it was found that the best performance in terms of accuracy was achieved with this architecture. Heuristic experimentation showed that the reduction of convolutional layers could give rise to low classification performance. Additionally, the two PSD estimators, i.e., Periodogram and Welch function, were also evaluated.

The results obtained by the PSD estimators with the manually tuned network are shown in Table 4.1, which report the mean testing accuracy and Cohen's kappa coefficient per subject in the database, for both Periodogram and Welch PSD estimators, from a 5-fold cross-validation strategy. As can be observed, the Welch function obtained a higher performance compared to the Periodogram function (accuracies of 0.88 and 0.82, respectively).

It is also important to note the small variability (standard deviation) of the proposed approach in comparison with state of the art methods, as it is shown in Table 4.2, even to subjects with poor results in previous works such as the subjects 2, 5 and 6, which shown

Table 4.1.: Performance of the PSD estimators with the manually tuned architecture for each subject in the BCI Competition IV-2a database. Welch function outperforms Periodogram PSD estimator.

Subject	**Welch**		**Periodogram**	
	Accuracy	**Kappa**	**Accuracy**	**Kappa**
1	0.86	0.82	0.78	0.78
2	0.87	0.83	0.81	0.75
3	0.93	0.91	0.86	0.82
4	0.85	0.81	0.80	0.73
5	0.84	0.79	0.77	0.69
6	0.86	0.82	0.78	0.71
7	0.88	0.84	0.81	0.75
8	0.89	0.86	0.84	0.78
9	0.92	0.89	0.89	0.86
Mean ± std	0.88 ± 0.030	0.84 ± 0.039	0.82 ± 0.042	0.75 ± 0.056

Table 4.2.: Accuracy performance comparison with state of the art methods reporting results for the BCI Competition IV-2a dataset

Subject	**Yang et al. (2015)**	**Sakhavi et al. (2015)**	**Bashashati et al. (2015)**	**Periodogram (Proposed)**	**Welch (Proposed)**
1	0.77	0.81	0.79	0.78	**0.86**
2	0.50	0.54	0.61	0.81	**0.87**
3	0.80	0.85	0.86	0.86	**0.93**
4	0.54	0.65	0.74	0.80	**0.85**
5	0.65	0.59	0.60	0.77	**0.84**
6	0.49	0.44	0.57	0.78	**0.86**
7	0.81	0.84	0.87	0.81	**0.88**
8	0.84	0.87	0.81	0.84	**0.89**
9	0.82	0.78	0.84	0.89	**0.92**
Mean Acc. ± std	0.69 ± 0.15	0.71 ± 0.16	0.74 ± 0.12	0.82 ± 0.04	$\mathbf{0.88 \pm 0.03}$

that the use of CNN allows to obtain high level representations able to overcome the natural psychophysiological variability in the signals generated by each subject. Additionally, as can be observed the proposed approach improves the accuracy reported in state of the art by Bashashati et al. [7], Yang et al.[82] and Sakhavi et al. [67], for the same dataset.

4.2. Automatically Tuned CNN Architecture with BCI Competition IV-2a Dataset

For the automatic hyper-parameter tuning, the number of layers and the kernel size were fixed, as well as, activation functions and maxpooling layers. While, the optimal number of neurons for each convolutional and dense layer and the optimal learning rate were found using the RBFOpt algorithm. The lower and upper bounds of the hyper-parameter search space were from 8 to 256 to the number of neurons for each convolutional layer, from 128 to 256 to the number of neurons for the dense layers, and from 0.0001 to 0.01 to the learning rate. For greater clarity, The boundaries of the search space for each hyper-parameter can be seen in the table 4.3. This search space was selected based on to the previous experimental manual tuning tests and the hardware limitations.

Table 4.3.: Lower and upper bounds of the hyper-parameter searching space determined through experimentation.

Hyper-parameter	Lower Bound	Upper Bound
Conv Layer 1	8	256
Conv Layer 2	8	256
Conv Layer 3	8	256
Conv Layer 4	8	256
Conv Layer 5	8	256
Conv Layer 6	8	256
Conv Layer 7	8	256
Conv Layer 8	8	256
Conv Layer 9	8	256
Dense Layer 1	128	1024
Dense Layer 2	128	1024
Learning rate	0.0001	0.01

Table 4.4.: Comparison of the hyper-parameters of the proposed network architecture selected by the manual and the automatical optimization based strategies

Hyper-parameter	Manual Hyper-parameter Tuning	Automatic Hyper-parameter Tuning
Conv Layer 1	32	8
Conv Layer 2	16	30
Conv Layer 3	16	256
Conv Layer 4	16	255
Conv Layer 5	8	10
Conv Layer 6	8	10
Conv Layer 7	8	254
Conv Layer 8	16	21
Conv Layer 9	16	13
Dense Layer 1	2048	1010
Dense Layer 2	1024	1018
Learning rate	0.0001	0.0001

The RBFopt toolbox was run only one time on the subject with the worst performance obtained by the manual tuning. A total of 50 architectures were tested, converging towards

Table 4.5.: Accuracy performance of the manual hyper-parameter tuning compared against hyper-parameter tuning optimization.

Subject	**Manual Hyper-parameter Tuning**	**Automatic Hyper-parameter Tuning**
1	0.86	**0.97**
2	0.87	**0.99**
3	0.93	**0.99**
4	0.85	**0.98**
5	0.84	**0.98**
6	0.86	**0.99**
7	0.88	**0.99**
8	0.89	**0.98**
9	0.92	**0.99**
Mean Acc. ± std	0.88 ± 0.03	$\mathbf{0.98 \pm 0.0068}$

the optimal values which sought to maximize the accuracy classification performance. Once optimal hyper-parameters were found, the optimal CNN architecture was used to train the learning models that identifying the movements imagined by the other subjects. The optimal hyper-parameter vector found by the optimization algorithm was [8, 30, 256, 255, 10, 10, 254, 21, 13, 1010, 1018, 0.0001]. Table 4.4 shows a comparison between the hyper-parameters definedn by the manually tuned architecture and they obtained by the optimization algorithm.

Table 4.5 shows the comparison between the accuracy performance of the CNN with hyper-parameter defined by the manual tuning and those found by the optimization algorithm. As was expected the optimal hyper-parameters make the network achieve better performance than the selected by the heuristic strategy. Specifically, the optimal CNN architecture improves the manually tuned in a $11,36\%$, further reducing the variability observed between subjects.

Table 4.6 presents a comparison of the best mean accuracy obtained by the proposed approach, i.e., by using of the Welch function as PSD estimator, for both the automatic and the manual hyper-parameter tuning, against state of the art works, including some that not reported results per subject. According with the results, the proposed approach outperforms the best result reported up to 23.7% in both tuning strategies. Comparison was performed only with the state of the art works that reported the best results using the BCI Competition IV-2a dataset, i.e., the same dataset used in the present work.

Table 4.6.: Accuracy performance comparison with state of the art methods with reported result to the BCI Competition IV-2a dataset.

Methodology	Mean accuracy
Merinov et al., 2016	0.650
Helal et al., 2017	0.670
Yang et al. 2015	0.692
Sakhavi et al. 2015	0.706
Bashashati et al. 2015	0.743
Manual Tuning	**0.88**
Automatic Tuning	**0.98**

4.3. Optimal CNN Architecture with BCI Competition III 3a Dataset

To evaluate the robustness of the optimal CNN architecture obtained by the RBFOpt algorithm, This was used to train a classification model for distinguishing the imagine movements in the BCI Competition III 3a database. In this database subjects were also asked to imagine the same actions that the BCI Competition IV-2a dataset, but in this case each sample contained signals captured from 60 electrodes and not from 22, such as BCI IV-2a. However, this issu does not affect the architecture of the designed network, because simply the input image goes from an array of 22×100 to one of 60×100 values. Comparative results can be observed in 4.7, these results show the generalization capacity of the proposed method in comparison with two state of the art approaches [26], [55].

Table 4.7.: Accuracy performance obtained by the optimal CNN to the BCI Competition III 3a database against two state of the art approaches

Subject	He et. al. 2016	Mishuhina et. al. 2018	Proposed Approach accuracy
k3bb	0.85	0.97	0.992
k6b	0.94	0.84	0.989
l1b	1	0.94	0.983
Mean Acc. ± std	0.93 ± 0.061	0.92 ± 0.0555	0.98 ± 0.0037

In comparison with the state of the art, it is observed that approaches based on Filter banks with CSP as Merinov et. al., Sakhavi et. al. and Yang et. al., highly depends on the filter bands selection and this can cause loss of relevant information. Even though Bashashati et. al. presents one of the best results, but the high dispersion between the accuracy of each subject makes the results unreliable, the reason is the significant difference among the accuracy obtained for each subject due to the psychophysiological variability among subjects [63].

4.4. Computational Cost

The selected framework consisted of the TensorFlow library and Cuda for Python. TensorFlow [2] is a library created for the implementation of deep learning in Python, whereas CUDA allows the parallelization of the process in the GPU cores. The evaluated deep learning architectures were implemented in Keras version 2.0.6 [15], a deep learning framework that uses TensorFlow version 1.5.0 as backend.

Experiments were carried out in Kubuntu Linux distribution, in a Dell Precision T5810 CPU equipped with Intel Xeon processors of 3 GHz, 8 cores, 64-bit architecture, 16 GB of RAM and a Nvidia GeForce 1080ti GPU. The preprocessing stage of the database using Matlab was performed on the same computer.

4.4.1. BCI Competition IV Dataset 2a

The computational cost during training of the manually tuned network was 23 seconds per epoch, the network was trained during 120 epochs, and this was done 5 times due to the 5-fold cross-validation. It should be noted that this was for each one of the subjects, therefore, for the 9 subjects that compose the dataset, the total training time was 34.5 hours. In a similar way, the computational cost of training the optimal network architecture was 24 seconds per epoch, as well as, this was trained by 120 epochs with 5-fold cross-validation for each of the subjects, obtaining a total training time of 36 hours.

The optimization process was performed only on the subject with the worse performance obtained with the parameters tuned manually to this dataset. In the process approximately 50 different hyper-parameters configurations were tested, with a total time of 72 hours, this thanks to the fact that the network does not fully train the 50 different architectures, only the most promising ones.

4.4.2. BCI Competition III Dataset 3a

For the BCI competition III 3a dataset, the computational cost was calculated only to the automatically tuned hyper-parameters. The result was 10 seconds per epoch, and 120 epochs were also trained with 5-fold cross-validation. This process was also performed for each of the 3 subjects that compose the database, having this test a total duration of 5 hours.

5. Conclusions and Future Works

This thesis addressed the problem of classifying EEG signals of motor imagery for BCI systems through implementation of a convolutional neural network with ReLU activation functions. Two main challenges were here tackled, the design of a learning model based on CNN that would allow to differentiate intention of movement in a subject based on the analysis of EEG signals generated by a motor imagery process, and the optimal selection of the network hyper-parameters that allows to raise the classification performance of the proposed model. For the design of the CNN based learning model, the use of the power spectral density (PSD) of the signals as input of a CNN, allows to extract relevant characteristics of the raw EEG to obtain a model that identified the activity imagined by a subject to two public databases named the BCI Competition IV Dataset 2a and the BCI Competition III Dataset 3a. Regarding to the optimization of network hyper-parameters, the use of a meta-heuristic optimization algorithm, based on radial-based functions, allowed to design an optimal network that achieved a performance that exceeds not only the model initially proposed, but also the state of the art methods. It showed how the use of optimization approaches is a very important task to improve the performance of a deep neural network.

Since EEG signals have not a spatial relationship between channels, two light preprocessing steps were implemented to achieve the design of the CNN learning model in the EEG electrode space as was proposed in the first specific objective. As first step, a window-based approach over temporal signals was used, which allowed the signals to be modeled as pseudo-images converting 1D signals into 2D signals. This strategy, allowed also to generate a greater number of training data thanks to the window overlapping technique used, because large amounts of examples are required for training deep learning models. In the second step, a clustering algorithm was used to generate a spatial matrix that allows to reordering the position of the EEG channels according to their location in the scalp. It allowed to obtain the spatial relationship between the electrodes. After these light preprocessing steps, for feature extraction, the data were modeled in the frequency domain using power spectral density estimators. In this case, two estimators were evaluated, Periodogram and Welch function (a modified Periodogram method), according to the reported results the last one (Welch function) reach a better representation of the signal variations allowing a better signal classification.

Several manual tuning network architectures were evaluated by changing number of layers and convolutional filters per layer (results not shown). As was expected, we found that manual tuning of network parameters is unfeasible due to the multiple combinations that must be evaluated and the lot of time that each combination takes. Even though, the parameters selected by this way allows to obtain results that outperform state of the art methods to both evaluated datasets.

To obtain the best CNN architecture based on the hiper-parameter optimization algorithm, three parameters were modified with variations in a sufficient range of values to evaluate the performance of the different possibilities contemplated in the search space, among which are the number of neurons per convolutional layer, the number of neurons per dense layer and the learning rate. Although the evaluation of other parameters is feasible, it was not considered due to the computational cost of the model, which took 72 hours per an unique subject. It is important to remember that the optimization process was performed only with the subject with the worst performance obtained with the manually tuned parameters, which shows that the method has enough robustness to be generalized in the other subjects, and even in another datasets, making it unnecessary to carry out the process for each subject.

An advantage of the proposed approach is that due to the electrodes are input to the neural network as an "image" formed up by the alignment of signals from each of them as the rows of the image, the network architecture not require be modified if a different number of electrodes is used, because it would be introduced as a larger image to the network. Thus, electrode selection is not required. Additionally, the majority of the previous works against which the obtained results were compared, are highly dependent on the stages of preprocessing and feature extraction steps. Many of them are based on filter banks with CSP doing those highly dependant of the filter bands selection. This preprocessing can cause a loss of relevant information that could to affects the classification accuracy. In the proposed approach the use of preprocessing step was not implemented and eventough results outperform those methods using these filtering stages.

The training process of the manually tuned network, as well as the automatically tuned network, was carried out over Keras, using the RBFOpt toolbox to run the metaheuristic optimization algorithm that allowed to find the optimal network hyper-parameters. The use of those tools allowed to run the process in parallel on the GPU's cores, in this way it was possible to evaluate more than 50 different networks with the toolbox used in approximately 72 hours. Process that would have taken weeks if it had been executed in series or over a CPU. Additionally, It was observed that the computational cost in the training stage is high due to the optimization process. However, this allows to find optimal network architectures that reduce the computational cost during the classification with the benefit of increasing the classification accuracy. It was also shown that the proposed optimization method considerably reduces the execution time compared to manual hyper-parameter tuning, by not having to thoroughly evaluate each architecture formed by a certain set of hyper-parameters.

Another advantage of the proposed approach is the capacity of dealing with the high variability in the performance obtained between subjects due to the psycho-physiological differences related to the processes of perception and memory of each subject, as well as the training to which each subject must be subjected to correctly perform the task of motor imagination. This generates great variability in the signals of each subject, because the state of concentration in which each one is, affects directly the performance of the task. This fact, has caused a high dispersion between the accuracy of the state of the art approaches for each subject, making the results unreliable. However, the proposed approach shows independence to this variability among subjects thanks to the ability of abstraction of the convolutional filters that allows to find relevant characteristics to discriminate between the four classes addressed, as well as the PSD algorithm that can find patterns to discriminate the power densities found in the mu and betha bands.

The proposed approach addressed the classification of EEG signals problem for a motor imagery BCI. However, as has been considered by the research comunity, Deep Neural Networks are a black box model, by which it is not easy to obtain a physiological interpretation of the neuron inputs (weights, biases), that is to say it is difficult to visualize which features were taken into account for the classification process. However, because the filter nature of a CNN the highest weights of the network will be linked with the most relevant features in the classification problem, which could be used for try to know which characteristics are being taken into account plotting the outputs of each of the network layers. However, the physiological interpretability of the results were out of the scope of this thesis and could be approaches in future works.

According with the reported results, the proposed method provides a reliable strategy for differentiating the movement intention in a motor imagery based BCI, outperforming state of the art methods that were evaluated using the same datasets. In this way, It was demonstrated that the proposed scheme provides reproducibility and robustness, showing to be a valuable and promising strategy for the design of Brain Computer Interfaces.

It is evident that in this thesis it was not possible to address all the possible aspects of the different stages worked, such as the use of spatial and/or frequency filters, the implementation of different feature extraction algorithms, the extension of the approach to more complex problems, as well as the inclusion of more hyper-parameters in the process of meta-heuristic optimization. Thanks to this, as future works other spatial filters could be explored to evaluate the advantages and drawbacks of the proposed scheme. Also, the implementation of the proposed method for the development of learning models with other tasks of motor imagination could be considered. Furthermore the metaheuristic optimization could include unproven parameters such as the number of hidden layers, the dropout rate, the size of the Convolutional filters, the type of activation functions and even the number of training epochs that would make the process even less expensive and less time-consuming.

A. Appendix 1: Deep Convolutional Neural Networks and Power Spectral Density Features for Motor Imagery Classification of EEG Signals.

Pérez-Zapata A.F., Cardona-Escobar A.F., Jaramillo-Garzón J.A., Díaz G.M.
In: Schmorrow D., Fidopiastis C. (eds) Augmented Cognition: Intelligent Technologies.
Lecture Notes in Computer Science, vol 10915. Springer, Cham. 2018.

B. Appendix 2: MSpecFace: A Dataset for Facial Recognition in the Visible, Ultra Violet and Infrared Spectra.

Rubén D. Fonnegra and Alexander Molina and Andrés F. Pérez-Zapata and Gloria M. Díaz
In: Botto-Tobar M., Esparza-Cruz N., León-Acurio J., Crespo-Torres N., Beltrán-Mora M. (eds) Technology Trends.
Comunications in Computer and Information Science, Springer, Cham. 2017.

Bibliography

[1] *Dataset IIIa: 4-class EEG data.* http://www.bbci.de/competition/iii/#data_set_iiia. – Accessed: 2019-03-12

[2] Abadi, Martín ; Agarwal, Ashish ; Barham, Paul ; Brevdo, Eugene ; Chen, Zhifeng ; Citro, Craig ; Corrado, Greg S. ; Davis, Andy ; Dean, Jeffrey ; Devin, Matthieu [u. a.]: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. En: *arXiv preprint arXiv:1603.04467* (2016)

[3] Abaitua, Torre [u. a.]: *Procesado de señales EEG para un interfaz cerebro-máquina (BCI)*, Tesis de Grado, 2012

[4] Acharya, U R. ; Oh, Shu L. ; Hagiwara, Yuki ; Tan, Jen H. ; Adeli, Hojjat: Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. En: *Computers in biology and medicine* 100 (2018), p. 270–278

[5] Amarasinghe, K ; Wijayasekara, D ; Manic, M: EEG based brain activity monitoring using Artificial Neural Networks. En: *Human System Interactions (HSI), 2014 7th International Conference on* IEEE, 2014, p. 61–66

[6] Ang, Kai K. ; Chin, Zheng Y. ; Wang, Chuanchu ; Guan, Cuntai ; Zhang, Haihong: Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. En: *Frontiers in neuroscience* 6 (2012)

[7] Bashashati, Hossein ; Ward, Rabab K. ; Birch, Gary E. ; Bashashati, Ali: Comparing different classifiers in sensory motor brain computer interfaces. En: *PloS one* 10 (2015), Nr. 6, p. e0129435

[8] Bengio, Y. ; Boulanger-Lewandowski, N. ; Pascanu, R.: Advances in optimizing recurrent networks. En: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013. – ISSN 1520–6149, p. 8624–8628

[9] Bergstra, James S. ; Bardenet, Rémi ; Bengio, Yoshua ; Kégl, Balázs: Algorithms for hyper-parameter optimization. En: *Advances in neural information processing systems*, 2011, p. 2546–2554

[10] Brunner, C ; Leeb, R ; Müller-Putz, G ; Schlögl, A ; Pfurtscheller, G: BCI Competition 2008–Graz data set A. En: *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology* (2008), p. 136–142

[11] Cannon, Michael J. ; Percival, Donald B. ; Caccia, David C. ; Raymond, Gary M. ; Bassingthwaighte, James B.: Evaluating scaled windowed variance methods for estimating the Hurst coefficient of time series. En: *Physica A: Statistical Mechanics and its Applications* 241 (1997), Nr. 3-4, p. 606–626

[12] Carletta, Jean: Assessing agreement on classification tasks: the kappa statistic. En: *Computational linguistics* 22 (1996), Nr. 2, p. 249–254

[13] Chaudhary, Ujwal ; Birbaumer, Niels ; Ramos-Murguialday, Ander: Brain–computer interfaces for communication and rehabilitation. En: *Nature Reviews Neurology* 12 (2016), Nr. 9, p. 513

[14] Cho, Hohyun ; Ahn, Minkyu ; Ahn, Sangtae ; Kwon, Moonyoung ; Jun, Sung C.: EEG datasets for motor imagery brain computer interface. En: *Gigascience* (2017)

[15] Chollet, François [u. a.]. *Keras.* 2015

[16] Conn, Andrew R. ; Scheinberg, Katya ; Vicente, Luis N.: *Introduction to derivative-free optimization.* Vol. 8. Siam, 2009

[17] Costa, Alberto ; Nannicini, Giacomo: RBFOpt: an open-source library for black-box optimization with costly function evaluations. En: *Mathematical Programming Computation* 10 (2018), Nr. 4, p. 597–629

[18] Diaz, Gonzalo I. ; Fokoue-Nkoutche, Achille ; Nannicini, Giacomo ; Samulowitz, Horst: An effective algorithm for hyperparameter optimization of neural networks. En: *IBM Journal of Research and Development* 61 (2017), Nr. 4, p. 9–1

[19] van Doorn, Joost: Analysis of deep convolutional neural network architectures. (2014)

[20] Fraga, Tania: MindFluctuations: Poetic, Aesthetic and Technical Considerations of a Dance Spectacle Exploring Neural Connections. En: *Tecno Lógicas* 21 (2018), Nr. 41, p. 81–102

[21] Gendreau, Michel ; Potvin, Jean-Yves [u. a.]: *Handbook of metaheuristics.* Vol. 2. Springer, 2010

[22] Glorot, Xavier ; Bengio, Yoshua: Understanding the difficulty of training deep feed-forward neural networks. En: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, p. 249–256

[23] Göhring, Daniel ; Latotzky, David ; Wang, Miao ; Rojas, Raúl: Semi-autonomous car control using brain computer interfaces. En: *Intelligent Autonomous Systems 12*. Springer, 2013, p. 393–408

[24] Gutmann, H-M: A radial basis function method for global optimization. En: *Journal of global optimization* 19 (2001), Nr. 3, p. 201–227

[25] Hammer, Barbara: On the approximation capability of recurrent neural networks. En: *Neurocomputing* 31 (2000), Nr. 1-4, p. 107–123

[26] He, Lianghua ; Liu, Bin ; Hu, Die ; Wen, Ying ; Wan, Meng ; Long, Jun: Motor imagery EEG signals analysis based on Bayesian network with Gaussian distribution. En: *Neurocomputing* 188 (2016), p. 217–224

[27] Helal, Mahmoud A. ; Eldawlatly, Seif ; Taher, Mohamed: Using autoencoders for feature enhancement in motor imagery Brain-Computer Interfaces. En: *Biomedical Engineering (BioMed), 2017 13th IASTED International Conference on* IEEE, 2017, p. 89–93

[28] Higuchi, Tomoyuki: Approach to an irregular time series on the basis of the fractal theory. En: *Physica D: Nonlinear Phenomena* 31 (1988), Nr. 2, p. 277–283

[29] Hochreiter, Sepp ; Schmidhuber, Jürgen: Long short-term memory. En: *Neural computation* 9 (1997), Nr. 8, p. 1735–1780

[30] Ilievski, Ilija ; Akhtar, Taimoor ; Feng, Jiashi ; Shoemaker, Christine A.: Efficient Hyperparameter Optimization for Deep Learning Algorithms Using Deterministic RBF Surrogates. En: *AAAI*, 2017, p. 822–829

[31] Jackie, T ; Paulraj, MP ; Adom, AH ; Majid, MS A.: Kinesthetic Motor Imagery Based Brain-Computer Interface for Power Wheelchair Manoeuvring. En: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 10 (2018), Nr. 1-15, p. 23–27

[32] Jia, Xiaowei ; Li, Kang ; Li, Xiaoyi ; Zhang, Aidong: A novel semi-supervised deep learning framework for affective state recognition on eeg signals. En: *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on* IEEE, 2014, p. 30–37

[33] Jingwei, Liu ; Yin, Cheng ; Weidong, Zhang: Deep learning EEG response representation for brain computer interface. En: *Control Conference (CCC), 2015 34th Chinese* IEEE, 2015, p. 3518–3523

[34] Jones, Donald R. ; Schonlau, Matthias ; Welch, William J.: Efficient global optimization of expensive black-box functions. En: *Journal of Global optimization* 13 (1998), Nr. 4, p. 455–492

[35] Jones, Eric ; Oliphant, Travis ; Peterson, Pearu [u. a.]: *SciPy: Open source scientific tools for Python*. 2001–. – [Online; accessed ¡today¿]

[36] Katz, Michael J.: Fractals and the analysis of waveforms. En: *Computers in biology and medicine* 18 (1988), Nr. 3, p. 145–156

[37] Khare, Vijay ; Santhosh, Jayashree ; Anand, Sneh: Performance comparison using five ANN methods for classification of EEG signals of two mental states. En: *India Conference, 2008. INDICON 2008. Annual IEEE* Vol. 1 IEEE, 2008, p. 7–10

[38] Kingma, Diederik ; Ba, Jimmy: Adam: A method for stochastic optimization. En: *arXiv preprint arXiv:1412.6980* (2014)

[39] Kohavi, Ron [u. a.]: A study of cross-validation and bootstrap for accuracy estimation and model selection. En: *Ijcai* Vol. 14 Montreal, Canada, 1995, p. 1137–1145

[40] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey E.: Imagenet classification with deep convolutional neural networks. En: *Advances in neural information processing systems*, 2012, p. 1097–1105

[41] Kumar, Abhishek ; Kolekar, Maheshkumar H.: Machine learning approach for epileptic seizure detection using wavelet analysis of EEG signals. En: *Medical Imaging, m-Health and Emerging Communication Systems (MedCom), 2014 International Conference on* IEEE, 2014, p. 412–416

[42] Kumar, Suresh [u. a.]: Author productivity in the field Human Computer Interaction (HCI) research. En: *Annals of Library and Information Studies (ALIS)* 61 (2015), Nr. 4, p. 273–285

[43] Lance, Brent J. ; Kerick, Scott E. ; Ries, Anthony J. ; Oie, Kelvin S. ; McDowell, Kaleb: Brain–computer interface technologies in the coming decades. En: *Proceedings of the IEEE* 100 (2012), Nr. Special Centennial Issue, p. 1585–1599

[44] LeCun, Yann ; Bengio, Yoshua ; Hinton, Geoffrey: Deep learning. En: *Nature* 521 (2015), Nr. 7553, p. 436–444

[45] LeCun, Yann ; Boser, Bernhard ; Denker, John S. ; Henderson, Donnie ; Howard, Richard E. ; Hubbard, Wayne ; Jackel, Lawrence D.: Backpropagation applied to handwritten zip code recognition. En: *Neural computation* 1 (1989), Nr. 4, p. 541–551

[46] LeCun, Yann ; Bottou, Léon ; Bengio, Yoshua ; Haffner, Patrick: Gradient-based learning applied to document recognition. En: *Proceedings of the IEEE* 86 (1998), Nr. 11, p. 2278–2324

[47] Lekshmi, SS ; Selvam, V ; Rajasekaran, M P.: EEG signal classification using Principal Component Analysis and Wavelet Transform with Neural Network. En: *Communications and Signal Processing (ICCSP), 2014 International Conference on* IEEE, 2014, p. 687–690

[48] Liavas, Athanasios P. ; Moustakides, George V. ; Henning, Günter ; Psarakis, Emmanuil Z. ; Husar, Peter: A periodogram-based method for the detection of steady-state visually evoked potentials. En: *IEEE Transactions on Biomedical Engineering* 45 (1998), Nr. 2, p. 242–248

[49] Lin, Jzau-Sheng ; Shihb, Ray: A Motor-Imagery BCI System Based on Deep Learning Networks and Its Applications. En: *Evolving BCI Therapy-Engaging Brain State Dynamics*. IntechOpen, 2018

[50] Liu, Yu-Ting ; Lin, Yang-Yin ; Wu, Shang-Lin ; Chuang, Chun-Hsiang ; Lin, Chin-Teng: Brain dynamics in predicting driving fatigue using a recurrent self-evolving fuzzy neural network. En: *IEEE transactions on neural networks and learning systems* 27 (2016), Nr. 2, p. 347–360

[51] McFarland, Dennis J. ; Wolpaw, Jonathan R.: Brain-computer interface operation of robotic and prosthetic devices. En: *Computer* 41 (2008), Nr. 10

[52] Merinov, Pavel ; Belyaev, Mikhail ; Krivov, Egor: Filter bank extension for neural network-based motor imagery classification. En: *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on* IEEE, 2016, p. 1–6

[53] Mirowski, Piotr W. ; LeCun, Yann ; Madhavan, Deepak ; Kuzniecky, Ruben: Comparing SVM and convolutional networks for epileptic seizure prediction from intracranial EEG. En: *2008 IEEE Workshop on Machine Learning for Signal Processing* IEEE, 2008, p. 244–249

[54] Mirowski, Piotr W. ; Madhavan, Deepak ; LeCun, Yann: Time-delay neural networks and independent component analysis for eeg-based prediction of epileptic seizures propagation. En: *Proceedings of the national conference on artificial intelligence* Vol. 22 Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007, p. 1892

[55] Mishuhina, Vasilisa ; Jiang, Xudong: Feature weighting and regularization of common spatial patterns in EEG-based motor imagery BCI. En: *IEEE Signal Processing Letters* 25 (2018), Nr. 6, p. 783–787

[56] Muñoz, Henríquez ; Nureibis, Claudia: *Estudio de Técnicas de análisis y clasificación de senales EEG en el contexto de Sistemas BCI (Brain Computer Interface)*, Tesis de Grado, 2014

[57] Nair, Vinod ; Hinton, Geoffrey E.: Rectified linear units improve restricted boltzmann machines. En: *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, p. 807–814

[58] Nicolas-Alonso, Luis F. ; Gomez-Gil, Jaime: Brain computer interfaces, a review. En: *Sensors* 12 (2012), Nr. 2, p. 1211–1279

[59] Nijholt, Anton: BCI for games: A 'state of the art'survey. En: *International Conference on Entertainment Computing* Springer, 2008, p. 225–228

[60] Parhi, Keshab K. ; Ayinala, Manohar: Low-complexity Welch power spectral density computation. En: *IEEE Transactions on Circuits and Systems I: Regular Papers* 61 (2014), Nr. 1, p. 172–182

[61] Paul, Sananda ; Mazumder, Ankita ; Ghosh, Poulami ; Tibarewala, DN ; Vimalarani, G: EEG based emotion recognition system using MFDFA as feature extractor. En: *Robotics, Automation, Control and Embedded Systems (RACE), 2015 International Conference on* IEEE, 2015, p. 1–5

[62] Pérez-Zapata, AF ; Cardona-Escobar, Andrés F. ; Jaramillo-Garzón, Jorge A. ; Díaz, Gloria M.: Deep convolutional neural networks and power spectral density features for motor imagery classification of EEG signals. En: *International Conference on Augmented Cognition* Springer, 2018, p. 158–169

[63] Pfurtscheller, Gert ; Brunner, Clemens ; Schlögl, Alois ; Da Silva, FH L.: Mu rhythm (de) synchronization and EEG single-trial classification of different motor imagery tasks. En: *NeuroImage* 31 (2006), Nr. 1, p. 153–159

[64] Rahman, MKM ; Joadder, Md A M.: A review on the components of EEG-based motor imagery classification with quantitative comparison. En: *Application and Theory of Computer Technology* 2 (2017), Nr. 2, p. 1–15

[65] Regis, Rommel G. ; Shoemaker, Christine A.: A stochastic radial basis function method for the global optimization of expensive functions. En: *INFORMS Journal on Computing* 19 (2007), Nr. 4, p. 497–509

[66] Saa, Jaime F D. ; Gutierrez, Miguel S.: EEG signal classification using power spectral features and linear discriminant analysis: a brain computer interface application. En: *Eighth Latin American and Caribbean Conference for Engineering and Technology* LACCEI Arequipa, 2010, p. 1–7

[67] Sakhavi, Siavash ; Guan, Cuntai ; Yan, Shuicheng: Parallel convolutional-linear neural network for motor imagery classification. En: *Signal Processing Conference (EUSIPCO), 2015 23rd European* IEEE, 2015, p. 2736–2740

[68] Solanke, Ms Prerna B. ; Patil, Ms Smruti P. ; Shende, Ms Priyanka S.: Mind-Driven Vehicle for Disabled Person using Intelligent System. En: *International Journal of Innovative Studies in Sciences and Engineering Technology* 3 (2017), Nr. 5

[69] Sörensen, Kenneth: Metaheuristics—the metaphor exposed. En: *International Transactions in Operational Research* 22 (2015), Nr. 1, p. 3–18

[70] Srivastava, Nitish ; Hinton, Geoffrey ; Krizhevsky, Alex ; Sutskever, Ilya ; Salakhutdinov, Ruslan: Dropout: a simple way to prevent neural networks from overfitting. En: *The Journal of Machine Learning Research* 15 (2014), Nr. 1, p. 1929–1958

[71] Stoica, Petre ; Moses, Randolph L. [u. a.]: Spectral analysis of signals. (2005)

[72] Sturm, Irene ; Lapuschkin, Sebastian ; Samek, Wojciech ; Müller, Klaus-Robert: Interpretable deep neural networks for single-trial EEG classification. En: *Journal of neuroscience methods* 274 (2016), p. 141–145

[73] Taghizadeh-Sarabi, Mitra ; Daliri, Mohammad R. ; Niksirat, Kavous S.: Decoding objects of basic categories from electroencephalographic signals using wavelet transform and support vector machines. En: *Brain topography* 28 (2015), Nr. 1, p. 33–46

[74] Thodoroff, Pierre ; Pineau, Joelle ; Lim, Andrew: Learning robust features using deep learning for automatic seizure detection. En: *Machine learning for healthcare conference*, 2016, p. 178–190

[75] Tibdewal, Manish N. ; Fate, RR ; Mahadevappa, M ; Ray, AjoyKumar: Detection and classification of eye blink artifact in electroencephalogram through discrete Wavelet Transform and neural network. En: *Pervasive Computing (ICPC), 2015 International Conference on* IEEE, 2015, p. 1–6

[76] Vaid, Swati ; Singh, Preeti ; Kaur, Chamandeep: EEG signal analysis for BCI interface: a review. En: *2015 Fifth International Conference on Advanced Computing & Communication Technologies* IEEE, 2015, p. 143–147

[77] Vidal, Jacques J.: Toward direct brain-computer communication. En: *Annual review of Biophysics and Bioengineering* 2 (1973), Nr. 1, p. 157–180

[78] Walker, Ian ; Deisenroth, Marc ; Faisal, Aldo: Deep convolutional neural networks for brain computer interface using motor imagery. En: *IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE DEPARTMENT OF COMPUTING* (2015)

[79] Wang, Baojun ; Jun, Liu ; Bai, Jing ; Peng, Le ; Li, Guang ; Li, Yan: EEG recognition based on multiple types of information by using wavelet packet transform and neural networks. En: *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the* IEEE, 2006, p. 5377–5380

[80] Welch, Peter: The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. En: *IEEE Transactions on audio and electroacoustics* 15 (1967), Nr. 2, p. 70–73

[81] Xu, Bing ; Wang, Naiyan ; Chen, Tianqi ; Li, Mu: Empirical evaluation of rectified activations in convolutional network. En: *arXiv preprint arXiv:1505.00853* (2015)

[82] Yang, Huijuan ; Sakhavi, Siavash ; Ang, Kai K. ; Guan, Cuntai: On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification. En: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE* IEEE, 2015, p. 2620–2623

[83] Yang, Xin-She: *Nature-inspired metaheuristic algorithms*. Luniver press, 2010

[84] Zeiler, Matthew D. ; Fergus, Rob: Visualizing and understanding convolutional networks. En: *European conference on computer vision* Springer, 2014, p. 818–833

[85] Zheng, Wei-Long ; Zhu, Jia-Yi ; Peng, Yong ; Lu, Bao-Liang: EEG-based emotion classification using deep belief networks. En: *Multimedia and Expo (ICME), 2014 IEEE International Conference on* IEEE, 2014, p. 1–6