UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA BIOMEDICA

Metodi di estrazione di feature e classificazione per BCI basati su attività sensomotoria

Relatore
Prof. Tavazzi Erica

Laureando
Zhou Jun Hao

ANNO ACCADEMICO 2024-2025

Data di laurea 24/07/2025

Alla mia famiglia,
che mi ha permesso di inseguire i miei sogni.
Ai miei cari amici,
che hanno reso questo percorso unico.

Sommario

La *brain-computer interface* (BCI) è una nuova tecnologia che ha la capacità di analizzare i segnali prodotti dal nostro cervello e di interpretare le nostre intenzioni. Le BCI rappresentano potenzialmente una soluzione innovativa per persone affette da disabilità motorie, dato che hanno la capacità di restaurare un canale di comunicazione con il mondo esterno. Queste particolari BCI sono basate sull'immaginazione motoria e questo fenomeno è individuabile nel segnale elettroencefalico. Dopo una introduzione sulla struttura generale delle BCI e sull'immaginazione motoria, questo elaborato offre, tramite ricerca bibliografica, una panoramica generale su due fasi importanti nel percorso che compie l'informazione nelle BCI: l'estrazione delle feature e la loro classificazione.

L'estrazione delle feature ha come scopo l'individuazione delle componenti informative del segnale cerebrale (*feature*). Le tecniche di interesse sono la trasformata di Fourier e i modelli autoregressivi per il dominio delle frequenze, la trasformata Wavelet per il dominio tempo-frequenze, e il *Common Spatial Pattern* per il dominio spaziale.

I *classificatori* sono componenti fondamentali per decodificare l'intenzione del movimento del paziente. I modelli d'interesse sono l'analisi discriminante lineare e i *Support Vector Machine* per i classificatori lineari, e le rete neurali per i classificatori non lineari.

Essendo queste tecniche molto varie, che spaziano dall'elaborazione dei segnali al machine learning, l'elaborato fornisce una breve definizione matematica ai metodi presentati, evidenziandone le principali caratteristiche e limitazioni.

Abstract

The brain-computer interface (BCI) is a novel technology that is able to analyse the electrical signals coming from the brain and it is also able to interpret our intentions. BCIs are a potential innovative solution for people affected with motor disabilities, since they have the ability to build a line of communication with the outside world. These kinds of BCIs are based on motor imagery and this phenomenon can be seen in electroencephalograms. After an introduction on the general structure of BCIs and motor imagery, this dissertation offers through literature review an overview of two fundamental steps in the information pipeline in BCIs: feature extraction and their classification.

Feature extraction has the aim to identify the most informative components of the cerebral signal and these are therefore called features. The techniques of interest are Fourier Transform and autoregressive models in the frequency domain, Wavelet Transform in the time-frequency domain, and Common Spatial Patterns in the spatial domain.

Classifiers are essential to decode the thought of movement of a patient. The models of interest are Linear Discriminant Analysis and Support Vector Machines for linear classifiers, and Neural Networks for non-linear classifiers.

Since these topics range from signal processing to machine learning, this dissertation gives a brief mathematical definition of each technique, underlining their strengths and limitations.

Indice

I Introduzione **1**

1.1 La *brain-computer interface* (BCI) 2

1.2 Il segnale d'interesse e l'immaginazione motoria 3

1.3 Applicazioni 5

II Estrazione delle feature **7**

2.1 *Fast Fourier Transform* (FFT) 7

2.2 Modelli autoregressivi (AR) 10

2.3 Trasformata Wavelet (WT) 12

2.4 *Common Spatial Pattern* (CSP) 15

III Classificatori **19**

3.1 Analisi discriminante lineare (LDA) 19

3.2 *Support Vector Machine* (SVM) 22

3.3 Reti neurali e *Deep Learning* 24

Conclusioni **30**

Bibliografia **31**

Capitolo I

Introduzione

Una delle funzioni del sistema nervoso è la propagazione del segnale elettrico attraverso i neuroni ed è ciò che consente la comunicazione tra il cervello e il resto del corpo. Questa capacità è alla base della nostra interazione con l'ambiente circostante e con gli altri individui, elementi fondamentali per il benessere di animali sociali come l'essere umano. Tuttavia, il normale funzionamento neurofisiologico del sistema nervoso può essere alterato da malattie neurodegenerative, come la sclerosi multipla che danneggia la guaina mielinica dei neuroni, o da lesioni cerebrali, rendendo talvolta impossibile l'interazione con l'ambiente esterno.

Negli ultimi decenni, i progressi nelle neuroscienze hanno portato allo sviluppo di nuove tecnologie per ripristinare tali capacità e una tra queste è la *brain-computer interface* (BCI). L'idea alla base della BCI è quella di analizzare i segnali prodotti dai neuroni del cervello, discriminare le componenti caratteristiche del segnale d'interesse, dette anche *feature*, e di decodificare l'intenzione del paziente, detto anche *pilota*, al fine di controllare dispositivi esterni senza utilizzare le vie nervose danneggiate. Quindi, è di fondamentale importanza l'utilizzo di un sistema fisico di acquisizione preciso, l'utilizzo di algoritmi che riescano ad estrarre i dati significativi e sviluppo di modelli di classificazione di *machine learning* per decodificare al meglio le intenzioni del pilota. I sistemi di acquisizione possono variare da sistemi mobili, come elettrodi fissati in delle cuffiette, a chip di piccole dimensioni impiantabili direttamente nell'encefalo; mentre i modelli di classificazione possono variare da paradigmi lineari a basso costo computazionale, fino alle più recenti tecniche di *deep learning*.

In questo capitolo verranno introdotte le nozioni generali sul funzionamento delle BCI, in particolare di BCI non invasivi che si basano sull' attività della corteccia motoria del cervello, e sulla natura del segnale elettrofisiologico d'interesse. Questo è necessario per affrontare le specifiche tecniche dei prossimi capitoli e per fornire potenziali applicazioni di questa tecnologia, in particolare nell'ambito della riabilitazione.

1.1 La *brain-computer interface* (BCI)

La *brain-computer interface* (BCI) è una tecnologia composta principalmente da cinque componenti funzionali come mostra la figura I.1: modulo d'acquisizione, modulo di condizionamento del segnale, modulo di estrazione delle feature, modulo di classificazione e dispositivi di uscita.

Il *modulo d'acquisizione* delle BCI ha il compito di acquisire il segnale elettroencefalico e di digitalizzarlo. Esistono due tipologie di tecniche di acquisizione: di tipo invasivo e di tipo non invasivo. Nelle prime, gli elettrodi vengono impiantati chirurgicamente nell'encefalo del paziente o sulla sua superficie. Tale posizionamento permette generalmente di ottenere una risoluzione maggiore del segnale rispetto alle tecniche non invasive; tuttavia, l'intervento è costoso e comporta dei rischi significativi per il paziente, ad esempio lesioni o emorragie. Tali tecniche, pertanto, vengono usate solamente in pazienti affetti da particolari condizioni neurologiche. Le tecniche di acquisizione non invasive, usate largamente nella ricerca accademica, utilizzano invece tecnologie di imaging come la risonanza magnetica funzionale (*functional magnetic resonance imaging*, fMRI) oppure acquisiscono il segnale tramite elettrodi posti superficialmente sullo scalpo, non richiedendo così ad interventi chirurgici e permettendo di ottenere un segnale elettroencefalico (EEG) [1].

Il segnale viene successivamente pre-processato dal *modulo di condizionamento* per aumentare il rapporto segnale-rumore (*signal-to-noise ratio*, SNR) e, in particolare nelle BCI basate su EEG, rimuovere artefatti causati da altri processi, come il movimento del capo o il movimento oculare, attraverso filtri che agiscono sia sulla risoluzione temporale che sullo spettro del segnale. Questo passaggio è importante per la corretta estrazione delle *feature* di interesse e conseguentemente della loro corretta classificazione [2].

Il *modulo di estrazione* delle feature ha il ruolo di decodificare i dati ricevuti a monte in comandi che rappresentano l'intenzione del pilota. Nel caso di BCI basati su EEG, il modulo di estrazione delle feature utilizza algoritmi che analizzano il segnale sia nel dominio del tempo che nel dominio delle frequenze.

Il *modulo di classificazione* ha il compito di determinare quali feature estratte corrispondano a dei potenziali comandi da inviare al sistema di controllo. Nelle BCI basate su EEG vengono usate maggiormente algoritmi di tipo lineare per il loro basso consumo computazionale [3].

Infine, i *dispositivi di uscita* è il modulo che raggruppa i dispositivi che il pilota ha intenzione di controllare. La complessità di questi dispositivi è molto variabile: alcuni semplici sistemi BCI possono essere progettati per far muovere un cursore in un computer, mentre quelli più complessi possono interagire con protesi robotiche [4].

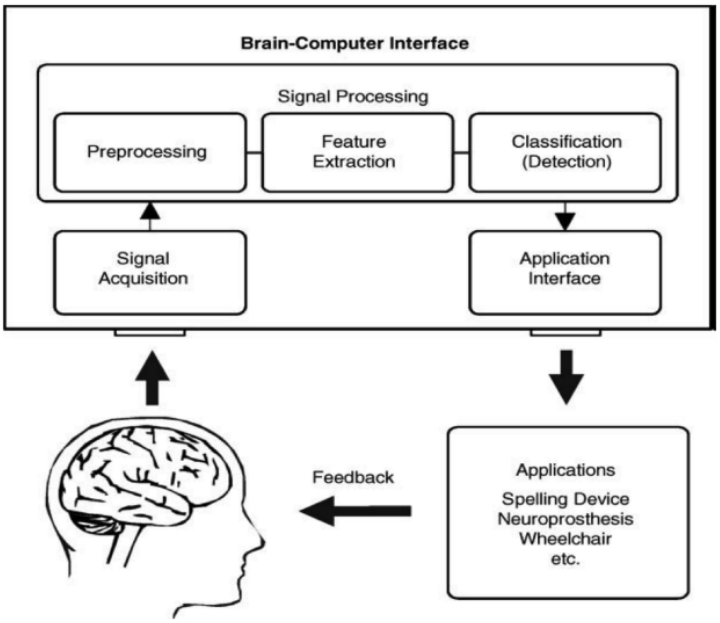

Figura I.1: Componenti principali di un BCI [2]

1.2 Il segnale d'interesse e l'immaginazione motoria

Esistono diverse tipologie di segnali utilizzabili per permettere al pilota di comunicare con dispositivi esterni: potenziali evocati visivi, potenziali evocati P300, potenziali corticali lenti e ritmi sensomotori. I *potenziali evocati visivi* sono potenziali generati nella corteccia visiva in risposta a stimoli visivi e aumentano in ampiezza quando gli stimoli si avvicinano al campo visivo centrale. I *potenziali evocati P300* sono potenziali generati in risposta a stimoli infrequenti (*oddball*) visivi, uditori o tattili; questi potenziali sono caratterizzati da un picco positivo approssimativamente 300 ms dopo la ricezione dello stimolo oddball e la loro ampiezza rappresenta il grado di attenzione del paziente. I *potenziali corticali lenti* sono potenziali generati nella corteccia frontale e dipendono dal grado di eccitabilità della zona; vengono definite *lenti* per la frequenza delle loro oscillazioni (< 1 Hz). I *ritmi sensomotori* sono particolari oscillazioni generate dalla corteccia motoria; la loro modulazione in risposta a movimenti reali o immaginati li rende particolarmente adatti all'impiego nelle BCI, e costituirà pertanto la tipologia principale di segnale su cui è focalizzato questo elaborato [4][5].

Semplificando i meccanismi neurofisiologici, il comando motorio è gestito nella corteccia motoria dell'encefalo; l'attivazione dei neuroni piramidali in questa regione generea oscillazioni rilevabili tramite EEG, note come *ritmi mu*. Queste oscillazioni hanno la caratteristica di avere una diminuzione del loro spettro di potenza; questo fenomeno è noto come desincronizzazione legata ad eventi (*event-related desynchronization*, ERD). Una volta terminato il processo motorio, i neuroni recuperano la loro sincronizzazione e questo fatto è rilevabile come un aumento del

loro spettro di potenza nelle stesse frequenze coinvolte nella precedente ERD; questo fenomeno è noto come sincronizzazione legata ad eventi (*event-related synchronization*, ERS). Questa alternanza tra ERD e ERS è osservabile anche durante l'immaginazione di un movimento, senza che questo venga realmente eseguito, indicando come la pianificazione e l'esecuzione del processo motorio coinvolgano gli stessi circuiti neurali. Il fenomeno appena descritto è comunemente definito come immaginazione motoria (*motor imagery*, MI) [6][7]. La corteccia premotoria, infatti, ha la funzione di gestire la pianificazione di movimenti complessi e la sua attività non è dipendente dall'esecuzione effettiva del movimento volontario [8]. Data la limitata risoluzione spaziale degli elettrodi e la prossimità anatomica delle regioni corticali coinvolte, l'attività registrata riflette il contributo complessivo dell'area motoria, come illustrato nella figura I.2.

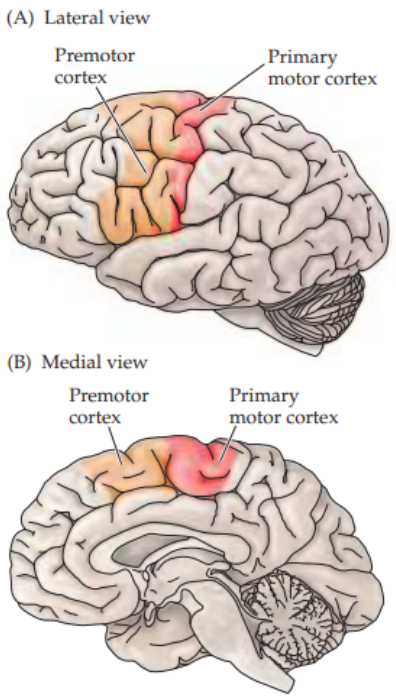

Figura I.2: Illustrazione anatomica della corteccia premotoria e della corteccia motoria.
(A) Vista laterale. (B) Vista mediale [8].

Analizzando in dettaglio il fenomeno ERD/ERS, la banda di interesse è la fascia alpha, che contiene le frequenze $8 - 12$ Hz. Si osserva un ERD circa 2 s prima dell'esecuzione del movimento reale volontario nell'emisfero controlaterale, seguito da un'ulteriore ERD nell'emisfero ipsilaterale poco prima dell'esecuzione del movimento. Nelle *task* di MI, i quali sono esperimenti controllati dove il pilota immagina un movimento successivamente ad uno stimolo acustico, si è osservato invece un marcato ERS nell'emisfero ipsilaterale nella fascia superiore della banda alpha ($10 - 12$ Hz) oltre all'atteso ERD dell'emisfero controlaterale, indicando il comportamento antagonista del ERD che induce un ERS nelle zone circostanti della corteccia come mostra la

figura I.3 [7].

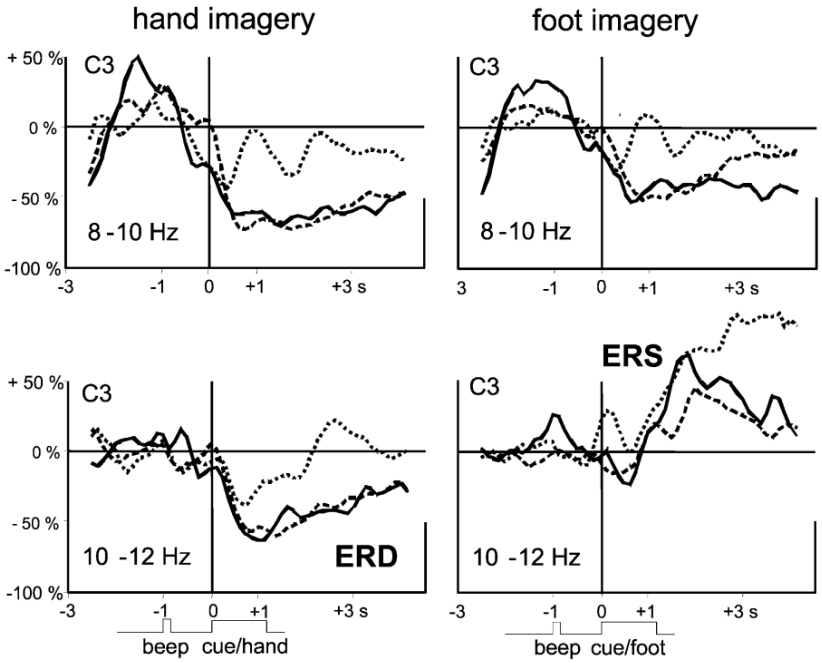

Figura I.3: Sovrapposizione degli andamenti temporali di ERD/ERS di tre pazienti differenti durante task di MI per discriminare movimenti della mano destra contro quella del piede [7].

Questo fenomeno è facilmente riproducibile ed è abbastanza stabile da poter essere usato come segnale di comando per BCI nei pazienti capaci di generarlo, dato che per avere un buon grado di operatività il paziente deve essere addestrato per le task di MI [4].

1.3 Applicazioni

In ambito clinico, le BCI hanno mostrato risultati promettenti nella riabilitazione motoria: durante le task di MI, i piloti devono generare un segnale di comando e attraverso *neurofeedback*, nella forma di propriocezione o meccanismi esterni, riescono a modulare la propria attività cerebrale formando così una catena chiusa con feedback. Questo protocollo migliora la *neuroplasticità*, cioè la capacità dell'encefalo di apprendere ed adattarsi a stimoli esterni, e porta alla ripresa funzionale o alla ricollocazione del controllo delle funzionalità perse in altre aree del cervello[3].

In particolare, uno degli studi attuati è quello di Mrachacz 2016 *et al.*[9], a cui hanno partecipato ventidue pazienti affetti da disabilità motorie causate da ictus cronico. Il protocollo è consistito nell'utilizzo di BCI per misurare l'attività corticale del soggetto durante i tentativi di dorsiflessione del piede controlaterale alla lesione. L'attività misurata è stata utilizzata per temporizzare in modo preciso delle stimolazioni nervose periferiche del nervo peroneo comune

durante il picco negativo massimale del ERD, assicurandosi così che il segnale esterno arrivasse nel momento di massima attivazione corticale. Dopo solo tre sessioni di $30 - 50$ ripetizioni, si sono osservati nei pazienti miglioramenti nei test fisici rispetto ad un gruppo di controllo in cui le stimolazioni nervose erano temporizzate in modo casuale.

La BCI è utilizzabile come uno strumento di comunicazione nel senso più lato del termine: per pazienti completamente paralizzati, detti *locked-in*, la BCI rappresenta l'unico strumento per poter interagire con altre persone. Uno dei sistemi più comuni è l'utilizzo della BCI per il controllo del cursore di un mouse in uno schermo esterno; un esempio è il *Graz-BCI* [10] che ha un protocollo che prevede l'utilizzo della BCI nella selezione di lettere in una tastiera virtuale. Pur avendo generalmente un trasferimento di dati basso, questo sistema è preferibile ad altre tecnologie come il tracciamento oculare per la libertà di utilizzo in qualsiasi ambiente e per l'assenza di gravi effetti di affaticamento [4]. Recentemente, uno dei pazienti dell'azienda *Neuralink*, la quale sviluppa BCI invasivi basati su MI, ha pubblicato sulla piattaforma *Youtube* un contenuto multimediale creato e modificato totalmente attraverso l'utilizzo del suo impianto neurale, mostrando così un utilizzo ibrido tra comunicazione e intrattenimento di una BCI [11].

Figura I.4: Pilota che si sta allenando in uno dei *tracks* della BCI race del *CYBLATHON* [12]

La BCI e i suoi progressi nella ricerca, come anche altre tecnologie della neuro-robotica, vengono esposti al pubblico attraverso competizioni tra varie squadre di ricerca come il *CYBLATHON* organizzato da ETH Zürich [12] (in figura I.4 un momento dell'allenamento). Questa competizione è ideale per testare queste nuove tecnologie in un ambiente non controllato, come può esserlo invece il laboratorio, e per mostrare gli avanzamenti della ricerca e le aree su cui migliorare. Nella competizione tra BCI, il *Team WHi*, capitanato dal Professor Tonin dell'Università degli Studi di Padova, ha ottenuto ottimi risultati, dimostrando così l'efficacia delle BCI non invasivi a basso costo computazionale nell'interazione simbiotica tra pilota e macchina [13].

Capitolo II

Estrazione delle feature

Il segnale elettroencefalico rappresenta l'attività complessiva dell'encefalo e include componenti informativi (feature) riconducibili a diversi task mentali, non limitandosi quindi alla sola attività della corteccia motoria. Una BCI deve essere pertanto in grado di discriminare il segnale d'interesse da quello complessivo attraverso l'*estrazione delle feature*, come descritto nella sezione 1.1. Tale compito può risultare complesso dato che le feature dei vari segnali possono manifestarsi simultaneamente in diversi domini: il segnale cerebrale può variare nel tempo, posizioni differenti nello scalpo possono avere ampiezze del segnale differenti oppure le sue componenti in frequenza possono variare. Questo fatto comporta che l'utilizzo di semplici filtri digitali risulta inadeguato, rendendo necessario l'impiego di algoritmi specifici per estrarre le feature nei diversi domini [4].

In questo capitolo verranno approfonditi i principali algoritmi impiegati nella letteratura scientifica: la trasformata di Fourier (sez. 2.1) e i modelli autoregressivi (sez. 2.2) per l'estrazione delle feature nel dominio delle frequenze, la trasformata Wavelet (sez. 2.3) per l'analisi congiunta delle feature nel dominio tempo-frequenza, e infine il metodo *Common Spatial Pattern* (sez. 2.4) per l'estrazione delle feature nel dominio dello spazio [2][4][14].

2.1 *Fast Fourier Transform* (FFT)

Quando la feature di interesse riguarda la variazione di frequenza di un segnale, lo strumento tradizionalmente usato è la trasformata di Fourier. La trasformata di Fourier è uno strumento utilizzato frequentemente nell'analisi spettrale, in particolare nel calcolo dello spettro di potenza (*power spectral density*, PSD) nelle varie frequenze [14]. Come già descritto nella sezione 1.2, la banda di frequenza d'interesse è la banda alpha ($8 - 12\,\text{Hz}$).

Invece della classifica definizione della trasformata di Fourier, si utilizza la *Fast Fourier Transform* (FFT) per il ridotto costo computazionale. Uno degli algoritmi FFT più usati è quello

di Cooley-Turkey che riduce le $O(n^2)$ operazioni della trasformata di Fourier a $O(N \log_2 N)$. Questo è possibile grazie all'approccio *divide et impera* e consiste nel frammentare le operazioni richieste per la trasformata discreta di Fourier (*discreet fourier transform*, DFT) dividendola ricorsivamente a metà:

Sia $N = 2^n$ il numero di dati. Si divide la DFT iniziale in due DFT più corte di lunghezza N_1 e N_2 tale che $N = N_1 N_2$. Una divisione intuitiva è $N_1 = 2$, quindi $N_2 = N/2 = 2^{n-1}$. Definendo per comodità $\omega_N := e^{-2\pi i/N}$, si può ridefinire la DFT di lunghezza $N/2$ attraverso la somma dei suoi termini $f(m)$ pari e dei suoi termini dispari.

$$\begin{aligned}\hat{f}(n) &= \frac{1}{\sqrt{N}} \left[\sum_{m=0}^{N/2-1} f(2m)\omega_N^{n(2m)} + \sum_{m=0}^{N/2-1} f(2m+1)\omega_N^{n(2m+1)} \right] \\ &= \frac{1}{\sqrt{N}} \left[\sum_{m=0}^{N/2-1} f(2m)\omega_{N/2}^{n\cdot m} + \omega_N^n \sum_{m=0}^{N/2-1} f(2m+1)\omega_{N/2}^{n\cdot m} \right] \end{aligned} \tag{II.1}$$

Quindi si è ridotto il numero di ogni $\hat{f}(n)$ a due moltiplicazioni e una addizione, un totale di tre passaggi [15].

Il calcolo della PSD della banda alpha P_α può essere ottenuto utilizzando la sua definizione classica:

$$P_\alpha = \frac{FFT^*_{(EEG\alpha)} FFT_{(EEG\alpha)}}{p^2} \tag{II.2}$$

dove p è il numero di punti della finestra temporale, $FFT_{(EEG\alpha)}$ denota la FFT del segnale EEG nella banda alpha e il simbolo * ne rappresenta il complesso coniugato [16].

La FFT offre una buona stima della PSD quando il segnale EEG è stazionario. Purtroppo, il segnale EEG è di base un segnale altamente non stazionario e le prestazioni della FFT sono impattate negativamente da questo fatto. Infatti, questo metodo è poco utilizzato per la sua bassa risoluzione in brevi segmenti temporali e viene generalmente preferito l'utilizzo dei modelli autoregressivi per la loro risoluzione maggiore, come mostra la figura II.1 [4][2].

Un metodo alternativo per calcolare la PSD è il metodo di Welch [18][17][19], conosciuto anche come *Short-Time Fourier Transform* (STFT). Il metodo consiste nell'analisi in frequenza della sovrapposizione dei singoli segmenti del segnale diviso per epoche di breve durata:
Sia la sequenza di informazione $x_i(n)$ espressa come

$$\begin{aligned} x_i(n) = x(n + iD) \quad & n = 0,1,2 \ldots M-1 \\ \text{con} \quad & i = 0,1,2 \ldots L-1 \end{aligned} \tag{II.3}$$

prendendo iD come il punto di partenza della sequenza i-esima di lunghezza $L = 2M$, mante-

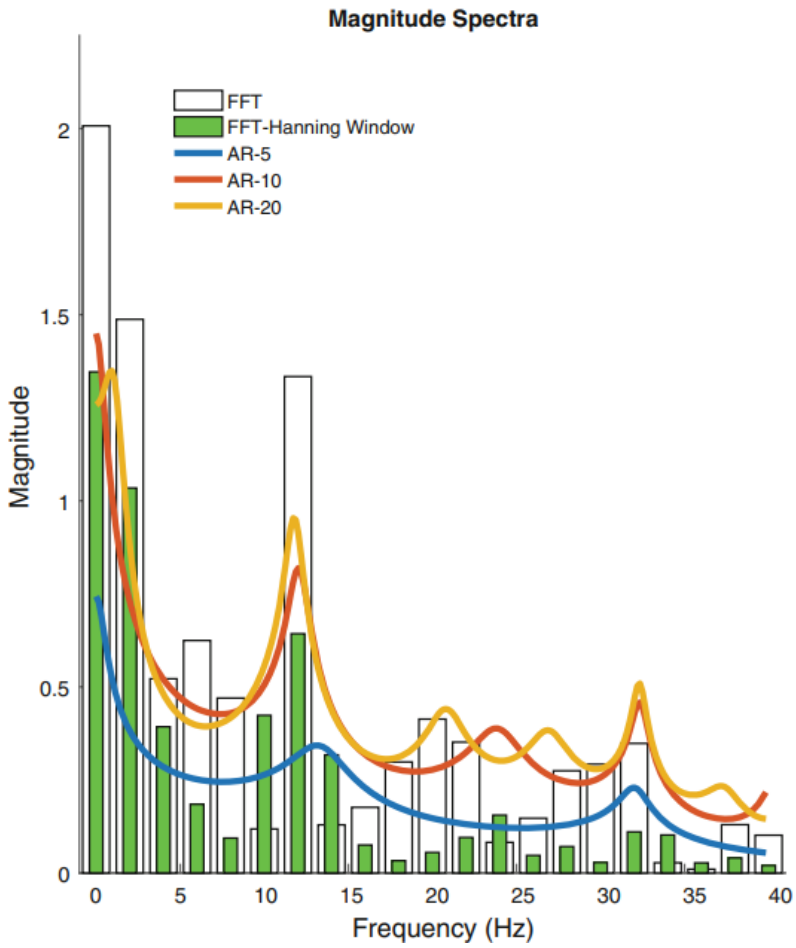

Figura II.1: Confronto dello spettro di un segnale utilizzando la FFT, la FFT usando la finestra di Hanning e modelli autoregressivi di ordini 5, 10 e 20 [17].

nendo così una sovrapposizione del 50%.

Il periodogramma risultante è:

$$P_i(f) = \frac{1}{U} \left| FFT\left(x_i(n)\ w(n)\right) \right|^2 \tag{II.4}$$

dove $w(n)$ è la funzione di finestratura, che solitamente è la finestra di Hanning, e U è il fattore di normalizzazione tale che

$$U = \frac{1}{M} \sum_{n=0}^{M-1} w^2(n) \tag{II.5}$$

La media di questi periodogrammi è la stima finale del PSD secondo Welch:

$$P^W(f) = \frac{1}{L} \sum_{i=0}^{L-1} P_i(f) \tag{II.6}$$

Questa segmentazione del segnale riduce la varianza del PSD e ne riduce il *spectral leakage* (fenomeno in cui l'energia di una certa frequenza è "dispersa" nelle frequenze adiacenti), risolvendo così parzialmente il problema della bassa risoluzione temporale per segnali non stazionari [2] [17].

2.2 Modelli autoregressivi (AR)

Un'alternativa della trasformata di Fourier per l'analisi spettrale del segnale sono i modelli autoregressivi (*autoregressive*, AR). I modelli AR possono stimare la PSD del segnale $x(t)$ come l'uscita di un filtro lineare tempo-invariante:

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots + a_p x(t-p) + e(t) \tag{II.7}$$

dove $[a_1, a_2 \ldots a_p]$ sono i coefficienti del filtro, p l'ordine del filtro $h(t)$ e $e(t)$ la componente di rumore bianco del segnale. Un aspetto molto insidioso dei modelli AR è la scelta dell'ordine p del filtro. Se l'ordine del modello è troppo basso, i coefficienti rappresenteranno lo spettro di un segnale troppo *smooth*; da contro, la scelta di un ordine troppo alto del modello comporta la presenza di picchi spurie nello spettro del segnale, come mostra la figura II.1 [4].

Esistono diverse metodologie per stimare i coefficienti del filtro e la varianza σ^2 (interpretabile come l'errore di predizione del filtro rispetto al segnale reale, deriva dalla componente di rumore), ad esempio tramite la risoluzione delle equazioni di Yule-Walker [18][20]:
Sia la funzione di autocorrelazione $R_{xx}(k)$ definita come

$$\begin{aligned} R_{xx}(k) =& E\left[x_{n+m} x_n^*\right] = E\left[x_n^* \left(\sum_{l=1}^{p} a_l x_{n-l+k} + e_{n+k}\right)\right] \\ =& \sum_{k=1}^{p} a_l R_{xx}(k-l) + E\left[e_{n+k} x_n^*\right]. \end{aligned} \tag{II.8}$$

La formula (II.8) instaura una relazione tra la funzione di autocorrelazione e i coefficienti del filtro. Dato che il filtro $H(z) = \frac{1}{\sum_{m=0}^{p} a_m z^{-m}}$ è supposto causale e stabile, si ha

$$\begin{aligned} E\left[e_{n+k} x_n^*\right] =& E\left[e_{n+k} \sum_{l=0}^{\infty} h_l^* e_{n-l}^*\right] = \sum_{l=0}^{\infty} h_l^* \sigma^2 \delta_{k+l} = \sigma^2 h_{-k}^* \\ =& \begin{cases} 0 & \text{per } k > 0 \\ h_0^* \sigma^2 & \text{per } k = 0 \end{cases} \end{aligned} \tag{II.9}$$

con δ_n delta di Dirac e h_k la risposta impulsiva. Dato che $h_0 = \lim_{z\to\infty} H(z) = 1$, si ha

$$R_{xx}(k) = \begin{cases} \sum_{l=1}^{p} a_l R_{xx}(k-l) & \text{per } k > 0 \\ \sum_{l=1}^{p} a_l R_{xx}(k-l) + \sigma^2 & \text{per } k = 0 \end{cases} \tag{II.10}$$

dove l'equazione (II.10) è l'espressione delle equazioni di Yule-Walker. Per determinare i coefficienti $[a_1, a_2 \dots a_p]$ è sufficiente risolvere il sistema con p equazioni per $k > 0$ e si può esprimere matricialmente come

$$\begin{bmatrix} R_{xx}(0) & R_{xx}(-1) & \dots & R_{xx}(-(p-1)) \\ R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(-(p-2)) \\ \vdots & \vdots & \vdots & \vdots \\ R_{xx}(p-1) & R_{xx}(p-2) & \dots & R_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ \vdots \\ R_{xx}(p) \end{bmatrix} \tag{II.11}$$

Osservando la forma dell'equazione (II.10), si può espandere (II.11) per incorporare σ^2

$$\begin{bmatrix} R_{xx}(0) & R_{xx}(-1) & \dots & R_{xx}(-p) \\ R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(-(p-1)) \\ \vdots & \vdots & \vdots & \vdots \\ R_{xx}(p) & R_{xx}(p-1) & \dots & R_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{II.12}$$

Quindi si ricava la PSD del segnale attraverso i coefficienti attraverso il seguente risultato

$$P^{YW} = \frac{\sigma^2 \Delta t}{\left|1 + \sum_{k=1}^{p} a_k \exp\left(-j2\pi f k \Delta t\right)\right|^2} \tag{II.13}$$

La risoluzione di (II.12) avviene per $O(p^3)$ operazioni attraverso l'eliminazione gaussiana, però attraverso l'algoritmo di Levinson-Durbin [20] i passaggi necessari si riducono ad $O(p^2)$. L'algoritmo processa ricorsivamente gli insiemi $\{a_{11}, \sigma_1^2\}, \{a_{21}, a_{22}, \sigma_2^2\} \dots \{a_{p1}, a_{p2} \dots a_{pp}, \sigma_p^2\}$, dove il primo pedice rappresenta l'ordine p-esimo del coefficiente e il risultato desiderato è l'insieme di ordine p. Questo fatto permette di avere anche un confronto delle prestazioni del modello AR al variare dell'ordine p, ed è utile soprattutto se non si ha informazioni a priori della scelta ottimale dell'ordine del modello.

L'algoritmo è inizializzato nel seguente modo:

$$a_{11} = R_{xx}(1)/R_{xx}(0) \tag{II.14}$$

$$\sigma_1^2 = (1 - |a_{11}|^2) R_{xx}(0) \tag{II.15}$$

e per la ricorsione per $k = 2.3 \ldots p$ si ha

$$a_{kk} = \left[R_{xx}(k) + \sum_{l=1}^{k-1} a_{k-1,l} R_{xx}(k-l) \right] / \sigma_{k-1}^2 \tag{II.16}$$

$$a_{ki} = a_{k-1,i} + a_{kk} a_{k-1,k-1}^* \tag{II.17}$$

$$\sigma_k^2 = (1 - |a_{kk}|^2) \sigma_{k-1}^2 \tag{II.18}$$

e la PSD del segnale si ottiene analogamente alla formula (II.13)

$$P^{YW} = \frac{\sigma_p^2 \Delta t}{\left|1 + \sum_{k=1}^{p} a_{pk} \exp\left(-j2\pi f k \Delta t\right)\right|^2} \tag{II.19}$$

Analogamente all'analisi spettrale tramite FFT, la prestazione dei modelli AR diminuiscono a causa della natura non stazionaria del segnale EEG. I modelli *MultiVariate Adaptive AutoRegressive* (MVAAR) hanno lo scopo di risolvere parzialmente il problema e di essere un metodo più efficace per l'analisi in tempo reale per BCI. I modelli MVAAR, oltre all'analisi del segnale di autocorrelazione, analizzano i segnali di cross-correlazione per ogni istante di tempo e i coefficienti del modello vengono stimati in modo adattivo dall'utilizzo di un filtro speciale di Kalman [21]:

$$Y(n) = \sum_{i=1}^{p} A_i(n) Y(n-1) + E(n) \tag{II.20}$$

dove $Y(n) = [y_1(n), y_2(n) \ldots y_k(n)]^T$, y_j rappresenta la j-esima sorgente di segnale e $n = 1{,}2 \ldots L$ è l'indice di tempo.

2.3 Trasformata Wavelet (WT)

Le tecniche di estrazione di feature fino ad ora menzionate forniscono informazioni solamente nel dominio delle frequenze. Tuttavia, il segnale cerebrale è un segnale generalmente non stazionario ed è pertanto importante ricavare informazioni nel dominio del tempo: l'utilizzo di strumenti matematici come la trasformata Wavelet (*Wavelet Transform*, WT) permette di analizzare il segnale EEG senza trascurarne la componente temporale, attraverso l'implementazione di funzioni denominate *wavelet* [4].

Le funzioni wavelet sono funzioni di frequenza variabile e di durata temporale limitata di una finestra principale (*mother wavelet*) e permettono lo studio di un segnale sia nel dominio delle frequenze che nel dominio del tempo. In particolare, la WT analizza il segnale in modo tale da applicare finestre temporali (wavelet) ristrette per le alte frequenze, mentre per le basse frequenze vengono applicate finestre temporali più larghe, come mostra la figura II.2 [18][17].

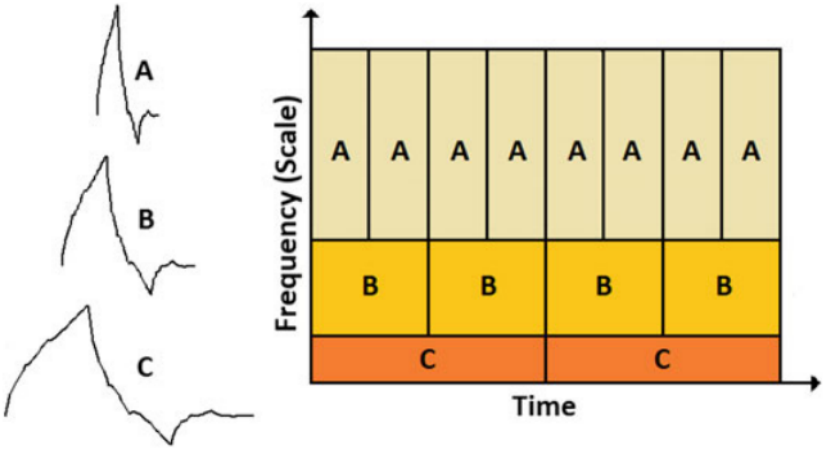

Figura II.2: Il diagramma nel lato destro rappresenta la risoluzione temporale ottenuta dalla WT al variare delle frequenze. Nel lato sinistro vengono rappresentati esempi dei corrispettivi mother wavelet nel dominio del tempo scalati secondo la WT [17].

Nella trasformata Wavelet continua (*Continuos Wavelet Transform*, CWT) l'analisi congiunta del segnale nel dominio tempo-frequenza può essere espressa come la seguente convoluzione [4][18]:

$$w(a,\tau) = \int_{-\infty}^{\infty} x(t)\psi^*_{a,\tau}(t)dt \tag{II.21}$$

dove $x(t)$ è il segnale EEG e $\psi_{a,\tau}$ è la funzione wavelet di una mother wavelet $\psi(t)$ dilatata per un fattore a e traslata per τ. Le feature di interesse sono quindi i coefficienti $w(a,\tau)$. Inoltre, la funzione wavelet si può esprimere come:

$$\psi_{a,\tau}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-\tau}{a}\right) \tag{II.22}$$

La mother wavelet può assumere varie forme e da essa dipende la tipologia di segmentazione del segnale nel dominio tempo-frequenza, ma rispetta sempre la condizione [4][17]:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{II.23}$$

Un esempio di mother wavelet è la *Mexican Hat* wavelet (figura II.3), che è utilizzata nell'analisi di potenziali sinaptici per la sua alta risoluzione nel tempo.

L'utilizzo della CWT introduce un alto costo computazionale derivato dalle infinite possibili combinazioni del fattore di scala a e del fattore di traslazione τ. Questo fatto ha introdotto l'utilizzo della trasformata Wavelet discreta (*Discreet Wavelet Transform*, DWT), la quale utilizza solamente un insieme finito di dilatazioni e traslazioni della mother wavelet in base alla frequenza del segnale d'interesse. Generalmente la DWT viene decomposta, come in figura II.4,

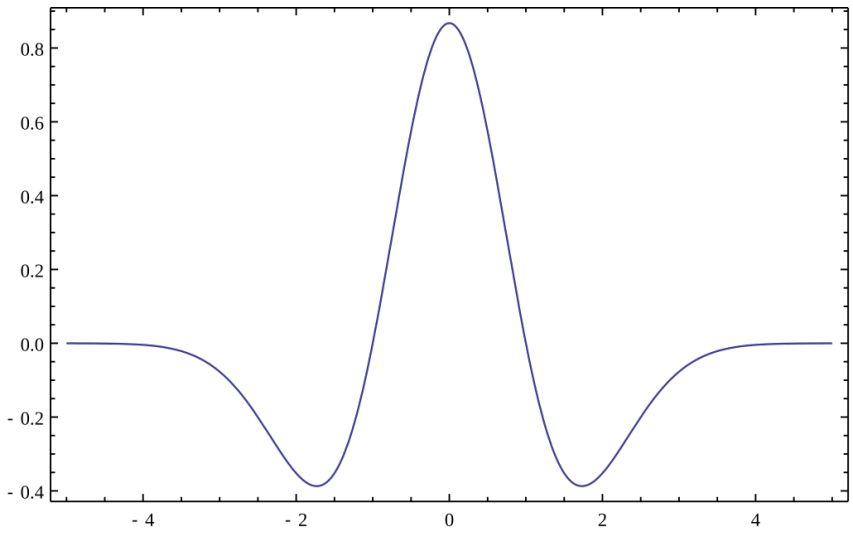

Figura II.3: Mexican mother wavelet [22].

attraverso l'utilizzo di due filtri digitali (mother wavelet) complementari [18][4].

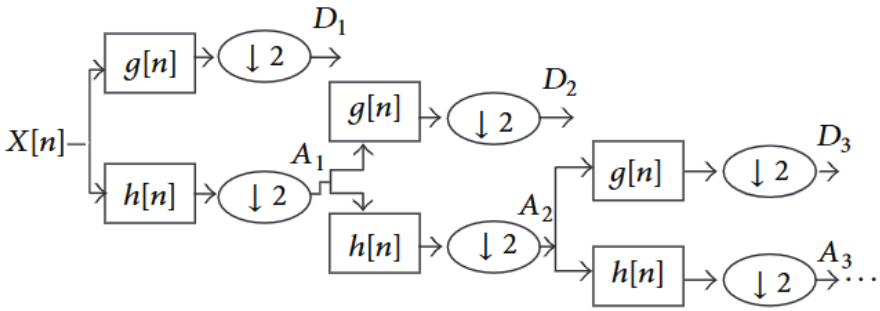

Figura II.4: Schema della decomposizione della DWT [18].

Il protocollo di decomposizione divide il segnale EEG d'interesse nelle sue componenti in alta frequenza attraverso la mother wavelet $g(n)$, mentre la mother wavelet complementare $h(n)$ ne estrae le componenti in bassa frequenza. Ad ogni passaggio intermedio, il segnale di uscita viene decimato di un fattore 2 e la componente in bassa frequenza A_i viene decomposta ulteriormente, ottenendo così una suddivisione più fine rispetto al segnale in alta frequenza D_i [18].

Pur avendo un costo computazionale maggiore rispetto alla DWT, la CWT è generalmente preferita per la sua alta risoluzione nel dominio tempo-spazio, soprattutto nelle situazioni in cui la DWT non riesce ad estrarre con precisione le feature di interesse [4][18].

2.4 *Common Spatial Pattern* (CSP)

Il segnale d'interesse e il fenomeno di MI sono caratterizzati, come descritto nella sezione 1.2, dal variare dello spettro di potenza nel tempo, ma soprattutto per la loro caratterizzazione spaziale in un certo istante di tempo. Nello specifico, la corteccia motoria è suddivisa lateralmente in base a specifiche porzioni del corpo e l'attivazione di una zona della corteccia motoria indica l'attività della rispettiva porzione del corpo (figura II.5) [8].

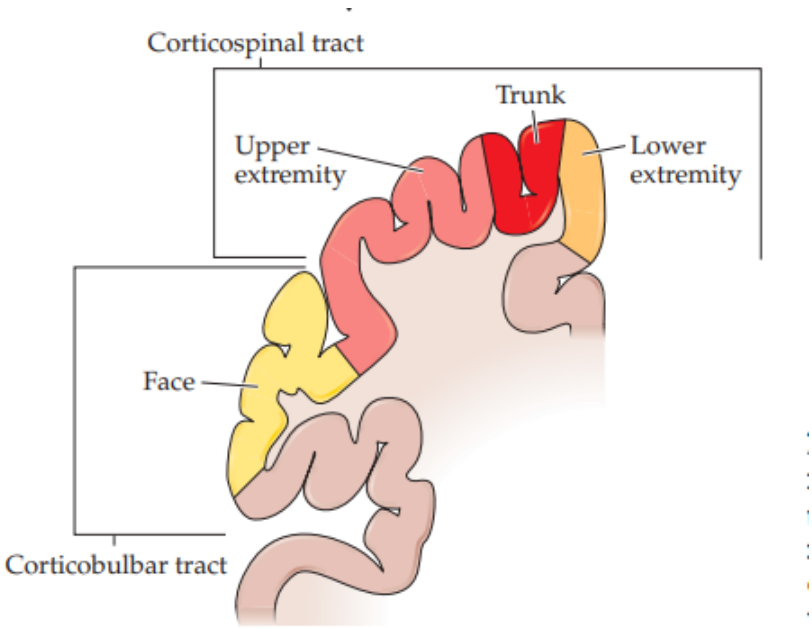

Figura II.5: Sezione trasversale della corteccia motoria dell'emisfero sinistro suddivisa nei suoi gruppi funzionali [8].

Il metodo *Common Spatial Pattern* (CSP) sfrutta questa variazione del segnale nello spazio nell'istante di tempo, denominato *spatial pattern*, proiettando l'EEG multicanale in uno sottospazio dove vengono individuate le classi che generano l'EEG e massimizza le loro differenze. Generalmente il metodo si basa sull'individuazione di due classi, ad esempio discriminando il movimento dell'arto destro rispetto a quello dell'arto sinistro, ma può essere ampliato a problemi multi-classe [4].

Per ottimizzare il metodo CSP, il segnale EEG di partenza viene filtrato in modo tale da eliminarne la componente costante, rendendo la parte informativa del segnale la sua matrice di covarianza spaziale [23][24]:
Si assuma di avere due gruppi principali A e B che rappresentano il segnale EEG d'interesse e che ogni gruppo si possa esprimere come una matrice X_i di dimensione $N \times T$, dove N rappresenta il numero di canali d'ingresso del EEG e T le istanze di tempo. Le matrici normalizzate

di covarianza spaziale sono:

$$R_A = \frac{X_A X_A^T}{\mathbf{tr}(X_A X_A^T)} \qquad R_B = \frac{X_B X_B^T}{\mathbf{tr}(X_B X_B^T)} \tag{II.24}$$

dove X_i^T rappresenta la matrice trasposta di X_i e la funzione traccia tr(A) calcola la somma della diagonale principale di A. Si procede a costruire la matrice composita di covarianza:

$$R_c = R_A + R_B \tag{II.25}$$

che si può fattorizzare nei suoi autovalori in

$$R_c = B_c \lambda B_c^T \tag{II.26}$$

dove B_c è la matrice $N \times N$ degli autovettori normalizzati, $B_c B_c^T = I$ matrice identità e λ la matrice diagonale contenente gli autovalori. Utilizzando la *whitening transform*, possiamo equalizzare la varianza attraverso

$$W = \lambda^{-1/2} B_c^T \tag{II.27}$$

dove W è la matrice di trasformazione per il sottospazio descritto da B_c. Le matrici di covarianze trasformate sono:

$$S_A = W R_A W^T \qquad S_B = W R_B W^T \tag{II.28}$$

osservando che per linearità vale $S_A + S_B = I$. Quindi, le matrici condividono gli stessi autovettori e valgono:

$$S_A = U \psi_A U^T \;\; S_B = U \psi_B U^T \;\; \psi_A + \psi_B = I \tag{II.29}$$

dove U è la matrice ortonormale di autovettori e ψ_i le rispettive matrici diagonali di autovalori. Questo comporta che gli autovettori degli autovalori più grandi per la matrice S_A sono anche gli autovettori degli autovalori più piccoli per la matrice S_B e viceversa. La proiezione della *whitened* EEG sul sottospazio degli autovettori di U permette la separazione ottimale delle matrici di covarianza (figura II.6):

$$P = U^T W \tag{II.30}$$

dove P è la matrice di proiezione ed è possibile ricavare le componenti non correlate del segnale EEG

$$Z = PX \tag{II.31}$$

dove la matrice Z si può interpretare come le sorgenti dei task differenti per l'EEG. Si può

ricostruire l'EEG originale invertendo l'operazione:

$$X = P^{-1}Z \tag{II.32}$$

dove P^{-1} è l'inversa di P. Le colonne di P^{-1} contengono le spatial pattern di interesse e possono essere considerate come vettori generativi per l'EEG: la prima e l'ultima colonna sono in particolare le spatial pattern più importanti, dato che contengono rispettivamente la varianza massima per un task e la varianza minima per l'altro task.

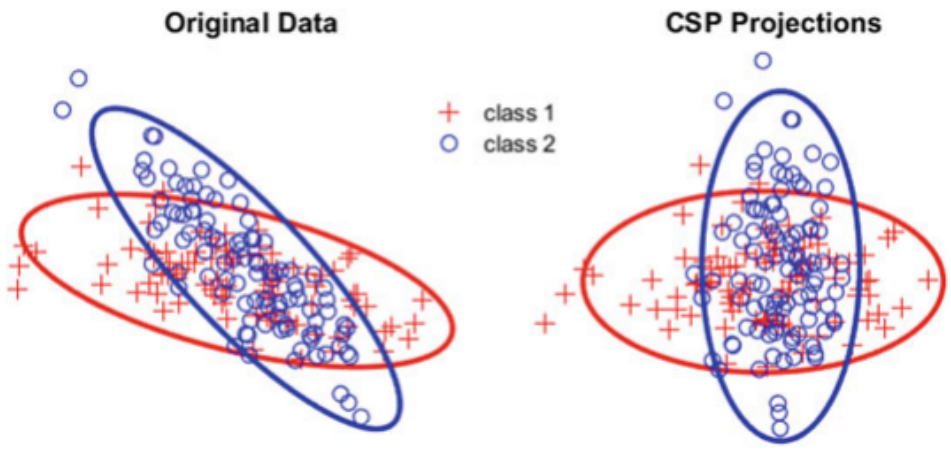

Figura II.6: Esempio di proiezione del segnale EEG di due classi principali in uno sottospazio di due feature [17].

L'analisi mediante il metodo CSP permette di discriminare gli elettrodi più rilevanti, in modo tale da ridurre il numero di canali di ingresso senza compromettere sull'accuratezza della classificazione del task di MI. Ad esempio, nella classificazione di task MI della mano destra e del piede destro, è possibile analizzare i spatial pattern più informativi. Per un determinato pilota, si osserva che il movimento immaginato della mano destra genera un ERD predominante nell'emisfero sinistro, causando simultaneamente un aumento di potenza (ERS) nella zona centrale della corteccia. Invece, il movimento immaginato del piede destro causa un ERD nella zona centrale della corteccia, accompagnata da un ERS nell'emisfero sinistro nella zona che rappresenta la mano. Questi spatial pattern si possono interpretare come la distribuzione ERD del task opposto; nel caso della figura II.7, l'elettrodo $C3$ rappresenta l'elettrodo discriminante per il piede destro e l'elettrodo CPz quello per la mano destra [24].

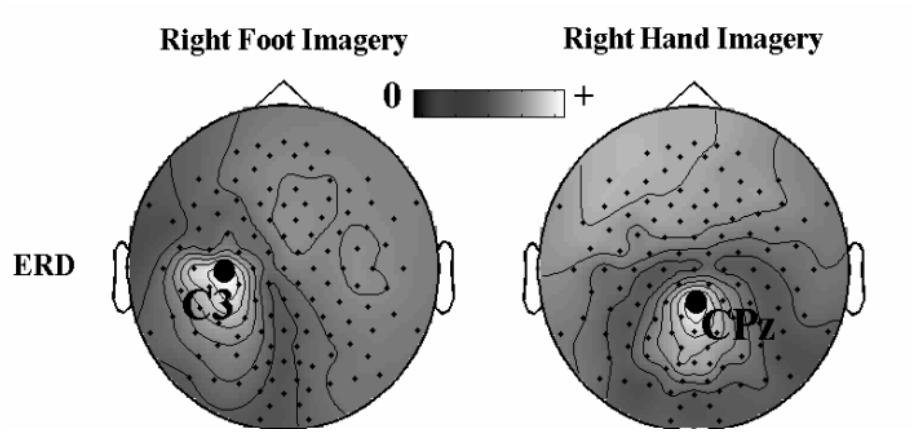

Figura II.7: Mappatura spaziale di due spatial pattern corrispondenti a task MI della mano destra e del piede destro [24].

Capitolo III

Classificatori

Uno degli aspetti più importanti nelle BCI è la loro capacità di decodificare l'intenzione del pilota, compito che viene svolto tramite l'impiego di una classe di metodologie di apprendimento supervisionato (machine learning). Nello specifico, il paradigma usato per risolvere questo problema è l'apprendimento supervisionato. Nell'apprendimento supervisionato, si dispone di un *dataset* $S = \{(x_1, y_1), (x_2, y_2) \dots (x_m, y_m)\}$, dove x_i è una componente della matrice di feature $X \in \mathbf{R}^d$, d la dimensione delle feature, e y_i è la sua corrispettiva etichetta (*label*), che in questo contesto rappresenta la presenza o l'assenza del fenomeno di MI. Il compito del classificatore è quindi quello di produrre una regola di predizione h, a partire dal dataset S durante la fase di addestramento, in modo tale da riuscire ad associare correttamente le feature alle sue label, svolgendo il compito efficientemente sia nel dataset usato per apprendere h, che per nuove osservazioni di task di MI [25][26].

In questo capitolo si tratteranno i classificatori più utilizzati nelle BCI basate su MI, e si possono suddividere in classificatori lineari e classificatori non lineari. I *classificatori lineari*, come l'analisi discriminante lineare (sez. 3.1) e i *Support Vector Machine* (sez. 3.2), sono popolari per il loro basso costo computazionale e la loro discreta accuratezza di classificazione anche per task in tempo reale. Da contro, i *classificatori non lineari*, divenuti molto popolari nella ricerca recente, utilizzano reti neurali profonde (*deep learning*) e le sue derivazioni (sez. 3.3). Queste tecniche forniscono generalmente classificazioni più accurate rispetto ai classificatori lineari, a fronte però di una complessità computazionale superiore e l'impossibilità di ricostruire in modo trasparente i passaggi eseguiti, venendo così definiti modelli a "scatola chiusa" [2][4][27].

3.1 Analisi discriminante lineare (LDA)

L'analisi discriminante lineare (*Linear Discrimant Analysis*, LDA) è un classificatore lineare che si basa su un modello *generativo*, cioè classifica i dati assumendo che siano generati da

una distribuzione di probabilità; nelle applicazioni per BCI si assume che la distribuzione sia normale o normale multivariata nel caso di dimensione $d > 1$. In particolare, è d'interesse la stima della probabilità che una feature appartenga ad una particolare classe ed è calcolabile attraverso il teorema di Bayes [26]:

$$\mathrm{P}\left(Y = k | X = x\right) = \frac{\mathrm{P}\left(Y = k\right) f_k(x)}{\sum_{l=1}^{K} \mathrm{P}\left(Y = l\right) f_l(x)} \tag{III.1}$$

dove $Y = k$ è la k-esima classe di $1 \ldots K$ possibili classi, $\mathrm{P}\left(Y = k\right)$ è la probabilità (*a priori*) che una osservazione appartenga alla classe k-esima e $f_k(x)$ è la densità di probabilità che una specifica classe rappresenti una feature. Nel caso di dimensione $d > 1$, la densità di probabilità è:

$$\mathrm{P}\left(X = x | Y = k\right) = f_k(x) = \frac{1}{2\pi^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_k)^T \Sigma^{-1} (x - \mu_k)\right) \tag{III.2}$$

dove μ_k è il vettore media della k-esima classe e Σ è la matrice di covarianza ed è assunta comune a tutte le K classi.

I parametri reali del modello μ_k e Σ sono ignoti e devono essere stimati attraverso strumenti statistici come il *Maximum Likelihood Estimator* (MLE). La MLE massimizza la probabilità di un parametro θ, che definisce una distribuzione di probabilità, di generare una particolare osservazione e in particolare massimizza il logaritmo di questa probabilità (*log-likelihood*) [25]:

$$L(S, \theta) = \log\left(\prod_{i=1}^{m} \mathrm{P}(\theta | X = x_i)\right) \tag{III.3}$$

dove $S = \{x_1 \ldots x_m\}$ è il dataset formato da feature x_i con m quantità di dati. Un metodo semplice per massimizzare la log-likelihood è quello di studiarne la derivata. Si ricava che, per una distribuzione gaussiana multivariata, vale

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \hat{\Sigma} = \frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu})(x_i - \hat{\mu})^T. \tag{III.4}$$

Mettendo insieme le equazioni (III.4), (III.2) e (III.1), si assegna ciascuna feature alla propria classe quando la funzione

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2}\mu_k^T \Sigma^{-1} \mu_k + \log \mathrm{P}(Y = k) \tag{III.5}$$

è massima; questa funzione deriva da una riscrittura del logaritmo dell'equazione (III.1). Per dividere le classi (per semplicità si analizza il caso della classificazione binaria $K = 2$), la LDA

individua l'iperpiano (*decision boundary*) ottimale che soddisfa la condizione $\delta_{k1}(x) = \delta_{k2}(x)$, cioè:

$$x^T\Sigma^{-1}\mu_{k1} - \frac{1}{2}\mu_{k1}^T\Sigma^{-1}\mu_{k1} = x^T\Sigma^{-1}\mu_{k2} - \frac{1}{2}\mu_{k2}^T\Sigma^{-1}\mu_{k2}. \tag{III.6}$$

Si può esplicitare l'equazione (III.6) come una funzione lineare $\langle w, x\rangle + b$:

$$w = (\mu_{k2} - \mu_{k1})^T\Sigma^{-1} \quad b = \frac{1}{2}\left(\mu_{k1}^T\Sigma^{-1}\mu_{k1} - \mu_{k2}^T\Sigma^{-1}\mu_{k2}\right) \tag{III.7}$$

dove $\langle w, x\rangle$ rappresenta il prodotto interno tra la feature e i coefficienti w, mentre b rappresenta la componente di *bias*. Il classificatore LDA associa le feature in base al segno di $\langle w, x\rangle + b$, come mostrato in figura III.1 [26][25].

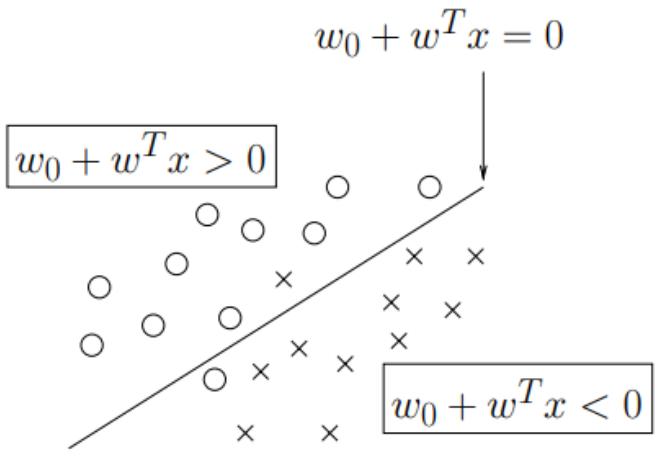

Figura III.1: Un iperpiano in $\mathbf{R}^2$ che divide due classi: i "cerchi" e le "croci" [27]. In questa rappresentazione, il termine di bias (b nel testo) è indicato come w_0.

Questo modello di classificazione ha un costo computazionale molto basso, dato che utilizza univocamente operazioni lineari. Questo fatto rende la LDA un classificatore molto popolare nella classificazione binaria di task in tempo reale, ovvero eseguiti contestualmente all'acquisizione del segnale, riscuotendo anche un elevato grado di accuratezza. Tuttavia, proprio per la sua linearità, la LDA non riesce a gestire correttamente dati EEG non lineari o dove sono presenti *outlier*, cioè dati che si discostano di molto dai dati su cui è stato allenato il classificatore e quindi dal modello generativo stimato [27].

I classificatori LDA regolarizzati di Fisher (*Regularized Fisher's LDA*, RFLDA) introducono un parametro di regolarizzazione C che permette di penalizzare in modo arbitrario gli errori di classificazione durante la fase di addestramento. Questo fatto permette una migliore generalizzazione del classificatore e di riscuotere prestazioni migliori anche in casi di dati non lineari

o outlier. Tuttavia, questo particolare modello non è molto utilizzato ed è molto simile ai più utilizzati *Support Vector Machine* [27].

3.2 *Support Vector Machine* (SVM)

I *Support Vector Machine* (SVM) sono classificatori lineari che, analogamente ai classificatori LDA, individuano l'iperpiano ottimale che separa le classi massimizzando la loro separazione. Dato che i classificatori SVM non ricavano un modello generativo dei dati, vengono denominati classificatori *discriminativi* e ricavano l'iperpiano ottimale direttamente dai dati [26][27].

Sia il dataset S definito come nelle precedenti sezioni e ci si ponga in un problema di classificazione binaria $y_i \in \{+1; -1\}$. Il dataset si dice *linearmente separabile* se

$$\forall i \in 1{,}2 \dots m \quad y_i(\langle w, x_i \rangle + b) > 0, \tag{III.8}$$

cioè se esiste l'iperpiano (w, b) che associa correttamente ogni istanza x_i alla sua label y_i attraverso la regola di predizione $h(x_i) = \text{sign}(\langle w, x_i \rangle + b)$, dove sign$(\cdot)$ è la funzione segno. Se il dataset è linearmente separabile, si definisce allora il *margine* dell'iperpiano come la distanza minima tra l'iperpiano e i punti del dataset; i punti più vicini al dataset sono definiti vettori di supporto (*support vector*), e da essi dipende la definizione dell'iperpiano [25][26].

Esistono infiniti iperpiani che dividono un dataset linearmente separabile, ma una scelta intuitiva è quella di selezionare l'iperpiano che massimizza il margine. I modelli *Hard-SVM* massimizzano il margine nell'intuizione che più grande è il margine, minore è l'errore di classificazione. Formalmente [25]:
Sia l'iperpiano (w, b) dove la norma $||w|| = 1$, allora la distanza tra un punto x_i e l'iperpiano è $|\langle w, x \rangle + b|$. Quindi, il modello Hard-SVM ottimizza:

$$\underset{(w,b):||w||=1}{\text{argmax}} \quad \min_{i \in 1,2 \dots m} |\langle w, x \rangle + b|, \tag{III.9}$$

dove argmax è la funzione che restituisce i parametri che massimizzano il suo argomento. Il problema di ottimizzazione si può riformulare nel trovare i coefficienti w di minima norma quadratica tale che la distanza tra il margine e ogni punto sia almeno 1:

$$(w_0, b_0) = \underset{(w,b)}{\text{argmin}} ||w||^2 \quad \text{t.c.} \quad \forall i \quad y_i(\langle w, x_i \rangle + b) \geq 1 \tag{III.10}$$

$$\hat{w} = \frac{w_o}{||w_o||} \quad \hat{b} = \frac{b_0}{||w_0||} \tag{III.11}$$

dove argmin è la funzione che restituisce i parametri che minimizzano il suo argomento, e $(\hat{w}, \hat{b})$

è l'iperpiano ottimale. Questo ci permette di individuare l'iperpiano con il margine massimo e la distanza minima tra l'iperpiano e ogni punto corrisponde al margine (figura III.2).

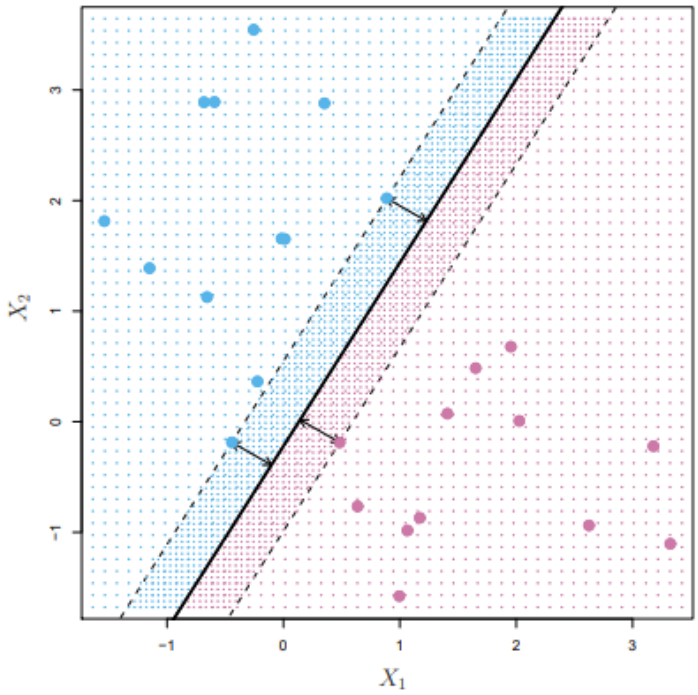

Figura III.2: Esempio di classificazione binaria utilizzando un SVM che massimizza il margine. Le due classi sono i punti "blu" e i punti "rossi" in $\mathbf{R}^2$. La linea nera continua rappresenta l'iperpiano e la linea tratteggiata i margini. Le frecce rappresentano la distanza minima dei punti all'iperpiano [26].

Nella realtà, il dataset non si può assumere sempre come linearmente separabile e il modello Hard-SVM non è più definibile. I modelli *Soft-SVM* si possono interpretare come rilassamenti dei modelli Hard-SVM grazie all'introduzione di un parametro *slack variable* $\xi_1 \dots \xi_m$ con $\xi_i \geq 0$ che rilassa la condizione della formula (III.10) a [25]:

$$y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i. \tag{III.12}$$

Intuitivamente ξ_i indica la misura di quanto ciascun punto possa entrare nella zona delineata dal margine massimale (figura III.3). I modelli Soft-SVM minimizzano la norma di w e la media dei parametri ξ_i; analogamente ai classificatori RFLDA, i modelli Soft-SVM utilizzano un fattore di regolarizzazione λ per gestire l'ottimizzazione:

$$\underset{w,b,\xi}{\text{argmin}} \left(\lambda ||w||^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right) \quad \text{t.c.} \quad y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i. \tag{III.13}$$

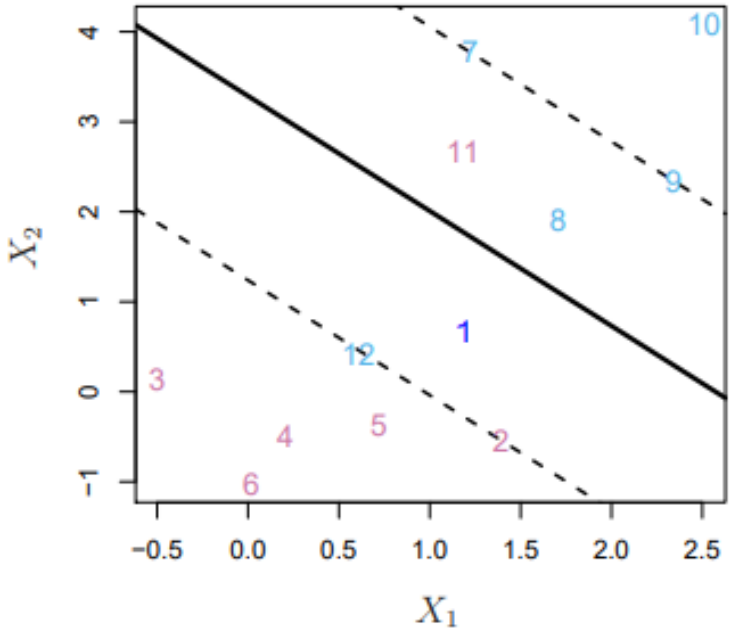

Figura III.3: Esempio di classificazione binaria in $\mathbf{R}^2$ con Soft-SVM. Le classi sono ”blu” e ”rosso”. I support vector sono le cifre 2 e 9. L’ osservazione 8 è classificata correttamente, ma ha una distanza dall’iperpiano minore rispetto al margine. Le osservazioni 11, 1 e 12 sono state classificate erroneamente [26].

In generale, i SVM sono classificatori con un buon grado di generalizzazione, resistenti all’*overfitting* (quando il classificatore predice perfettamente le istanze del dataset, ma con scarsi risultati nei test su dati nuovi) e agli effetti negativi dell’alta dimensionalità della matrice delle feature [27].

I classificatori SVM possono utilizzare iperpiani non lineari per separare con più precisione dati EEG non lineari attraverso l’utilizzo di funzioni *kernel*. Le funzioni kernel si possono interpretare come una mappatura dei dati da un sottospazio ad un altro di dimensionalità maggiore; nell’ambito delle BCI è usato spesso il kernel gaussiano, una funzione che misura la similarità tra due punti in base alla loro distanza euclidea, applicando un’esponenziale a una funzione quadratica negativa. Questo corrisponde a una trasformazione nello spazio delle funzioni in cui i punti vicini sono fortemente correlati e quelli lontani lo sono esponenzialmente meno, secondo la forma tipica di una distribuzione gaussiana. [27][4].

3.3 Reti neurali e *Deep Learning*

Le reti neurali (*Neural Networks*, NN), le quali hanno dato vita al ramo del *deep learning*, sono una particolare struttura computazionale ispirata alle reti biologiche formate dai neuroni nell’en-

cefalo. Le NN sono strutturati come un grafo: i nodi rappresentano le componenti computazionali di base (neuroni) e sono connesse tra di loro attraverso una complessa rete di archi (*edge*) da cui ricevono e inviano informazione. Esistono diverse architetture della rete, ma nella ricerca per BCI si utilizzano maggiormente le *feedforward* NN in cui non esistono collegamenti ciclici nel grafo [25][2].

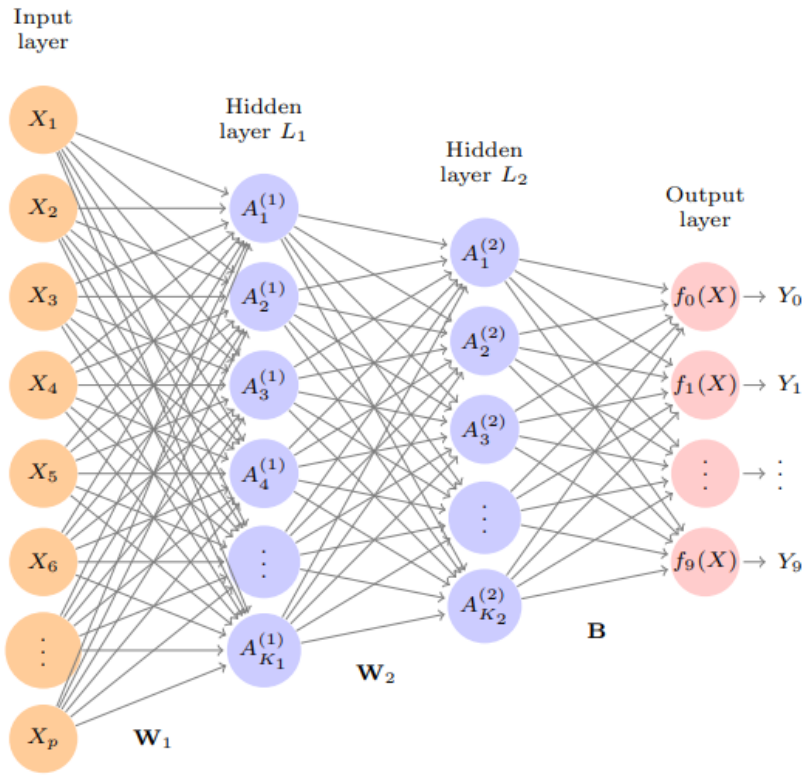

Figura III.4: Esempio di feedforward NN con due hidden layer usato nel riconoscimento di cifre scritte a penna. Il layer di ingresso è il layer $L = 0 \rightarrow A_k^{(0)} = X_k$, mentre il layer di uscita è il layer $L = D = 3 \rightarrow A_k^{(D)} = f_k(X)$ [26].

Una delle prime architetture di NN implementate è il *Multilayer perceptron* (MLP) e i neuroni sono suddivisi in vari *layer*: un layer di ingresso, un layer di uscita e un numero "arbitrario" di layer nascosti (*hidden*) (figura III.4). Il layer di ingresso è composto dai neuroni che prendono i valori del vettore di p feature d'ingresso $X = (X_1, X_2 \dots X_p)$ con lo scopo finale di costruire una regola di predizione $f(X)$ per descrivere le classi Y. Le frecce (*directed edges*) indicano un collegamento tra i neuroni di un layer L e i neuroni del layer precedente $L-1$, in particolare i K neuroni del layer L possono essere "attivati" tramite la ricezione di informazioni da questi collegamenti [26]:

$$A_k^{(L)} := h_k^{(L)}(X) = g\left(b_k^{(L)} + \sum_{j=1}^{p} w_{kj}^{(L)} A_j^{(L-1)}\right) \tag{III.14}$$

dove l'indice $L = 1,2 \ldots D$ rappresenta l'appartenenza al L-esimo layer su un totale di D layer (*depth*), b_k rappresenta la componente di bias del neurone $A_k^{(L)}$, $w_{kj}^{(L)}$ sono i coefficienti che determinano il contributo per l'attivazione del neurone $A_k^{(L)}$ di ciascun neurone $A_j^{(L-1)}$ del layer precedente $L-1$ e $g(\cdot)$ è la *funzione di attivazione* ed è una funzione non lineare. Una delle prime funzioni di attivazione proposte nell'ambito del deep learning è la funzione sigmoide:

$$g(z) = \frac{1}{1+e^{-z}} \tag{III.15}$$

ed è particolare per la sua capacità di mappare qualsiasi valore in un intervallo $[0,1]$ (figura III.5). Nelle applicazioni odierne e nelle BCI, si preferisce usare la *Rectified Linear Unit* (ReLU) dato che fornisce una maggiore efficienza nell'addestramento della NN e ne riduce il costo computazionale, essendo la funzione lineare a tratti:

$$g(z) = \begin{cases} 0 & \text{se } z < 0 \\ z & \text{se } z \geq 0. \end{cases} \tag{III.16}$$

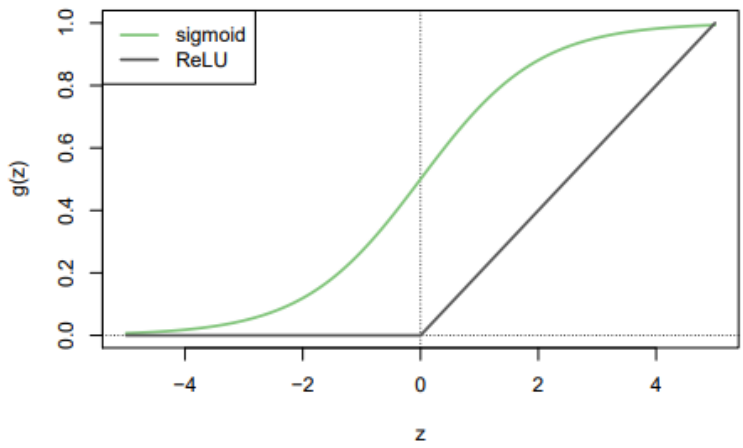

Figura III.5: Confronto tra le funzioni di attivazione sigmoide (verde) e ReLU (grigio). La funzione ReLU è stata modificata per facilitare il confronto [26].

Intuitivamente, un neurone $A_k^{(L)}$ riceve informazione da ciascun neurone $A_j^{(L-1)}$ sotto forma di una media pesata z. Questo valore viene successivamente posto sotto una funzione di attivazione e il grado di attivazione del neurone $A_k^{(L)}$ dipende dal valore di $g(z)$. Un neurone è inattivo se $g(z)$ si avvicina al valore 0 e il suo grado di attivazione aumenta all'aumentare del valore $g(z)$; per le funzioni di attivazione sigmoide e ReLU equivale al superamento di un valore di soglia di z. Questo comportamento della NN è simile al comportamento di un neurone biologico: un neurone trasmette informazione se la somma dei segnali sinaptici che riceve supera una certa *soglia di attivazione*.

Per misurare quantitativamente la prestazione della NN, si implementa l'uso della *squared-error loss* [26]:

$$C(w,b) = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{III.17}$$

dove y_i sono i valori desiderati dei neuroni che rappresentano le varie n classi. Nella fase di addestramento, la NN minimizza la squared-error loss attraverso l'analisi del gradiente per individuare la direzione di un minimo locale e modifica conseguentemente i coefficienti (w,b) per raggiungerlo attraverso varie iterazioni (*gradient descent*):

$$\nabla C(\theta_m) = \left.\frac{\partial C(\theta)}{\partial \theta}\right|_{\theta=\theta_m} \tag{III.18}$$

$$\theta_{m+1} \leftarrow \theta_m - \rho \nabla C(\theta_m) \tag{III.19}$$

dove θ è il vettore contenente i vari parametri da derivare w_{kj} e b_k, l'indice m indica il passo di iterazione e ρ indica la velocità di apprendimento. Per passi sufficientemente piccoli nella direzione opposta al gradiente, si garantisce $C(\theta_{m+1}) \leq C(\theta_m)$.

Dato che $C(\theta) = \sum_{i=1}^{n} C_i(\theta) = \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2$, possiamo computare il gradiente ci ciascun neurone di uscita e lo si generalizza come un neurone generico $f_\theta(x_i) = A_k^{(L)}$. Si definisce per comodità $z_k^{(L)} = b_k^{(L)} + \sum_{j=1}^{p} w_{kj}^{(L)} A_j^{(L-1)}$, allora il calcolo del gradiente può essere esplicitato tramite la regola della catena [26][28]:

$$\frac{\partial C_i}{\partial w_{kj}^{(L)}} = \frac{\partial C_i}{\partial A_k^{(L)}} \frac{\partial A_k^{(L)}}{\partial z_k^{(L)}} \frac{\partial z_k^{(L)}}{\partial w_{kj}^{(L)}} = -2(y_i - A_k^{(L)}) g'(z_k^{(L)}) A_j^{(L-1)} \tag{III.20}$$

$$\frac{\partial C_i}{\partial b_k^{(L)}} = \frac{\partial C_i}{\partial A_k^{(L)}} \frac{\partial A_k^{(L)}}{\partial z_k^{(L)}} \frac{\partial z_k^{(L)}}{\partial b_k^{(L)}} = -2(y_i - A_k^{(L)}) g'(z_k^{(L)}). \tag{III.21}$$

Si può osservare che il valore del neurone del layer precedente $A_j^{(L-1)}$ compare nella derivata parziale (III.20), influenzando così sul valore del neurone $A_k^{(L)}$. Infatti, analizzando la derivata parziale $\frac{\partial C_i}{\partial A_j^{(L-1)}}$ si ricava che:

$$\frac{\partial C_i}{\partial A_j^{(L-1)}} = \frac{\partial C_i}{\partial A_k^{(L)}} \frac{\partial A_k^{(L)}}{\partial z_k^{(L)}} \frac{\partial z_k^{(L)}}{\partial A_j^{(L-1)}} = -2(y_i - A_k^{(L)}) g'(z_k^{(L)}) w_{kj}^{(L)}. \tag{III.22}$$

Questo fatto ci permette di iterare la regola della catena a neuroni di layer precedenti e studiarne l'influenza sul comportamento globale della NN; questo processo viene definito *backpropagation*.

Calcolare analiticamente il gradient descent può risultare computazionalmente oneroso, spe-

cialmente nei casi di NN grandi. Un algoritmo di ottimizzazione molto usato è il *Stochastic Gradient Descent* (figura III.6), un algoritmo che calcola iterativamente il gradiente a ogni osservazione, rendendo il processo meno ottimale nell'individuazione del locale minimo, ma più efficiente dal punto di vista computazionale [25].

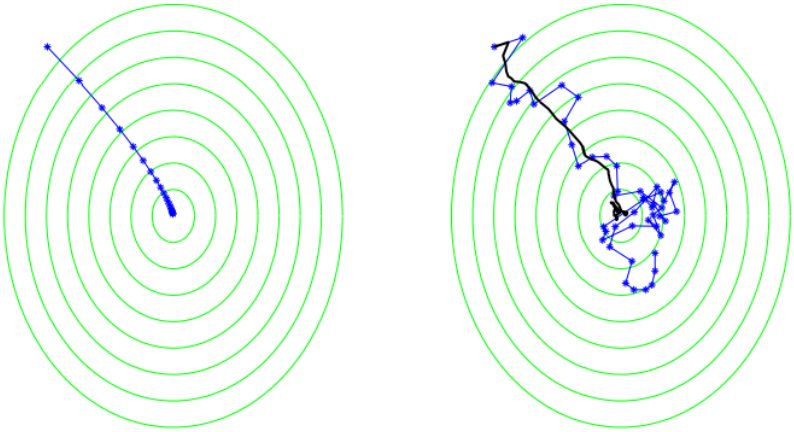

Figura III.6: A sinistra un esempio di gradient descent, mentre a destra il calcolo del gradiente attraverso Stochastic Gradient Descent. La linea blu è una singola osservazione, mentre la linea nera corrisponde alla media [25].

I MLP sono classificatori "universali", capaci di classificare dati EEG non lineari con alta precisione e non sono influenzati dal numero di classi da analizzare quanto i classificatori delle sezioni precedenti; questo avviene tuttavia a discapito di un costo computazionale molto alto, rendendoli non idonei per la classificazione in tempo reale per BCI [2].

Un'altra particolare architettura delle NN utilizzate nel contesto delle BCI sono le reti neurali convoluzionali (*Convolutional Neural Networks*, CNN), che utilizzano come input un'immagine dell'analisi nel dominio frequenza-tempo del segnale EEG [2].

In generale, le NN forniscono un alto grado di accuratezza, ma, come già accennato, hanno una bassa interpretabilità: l'unica metodologia di ottimizzazione proposta è l'ottimizzazione della squared-error loss, il quale non ha alcun significato fisico con le feature all'ingresso, ma è solo un problema di ottimizzazione matematica. La scelta dei pesi, e quindi le feature di maggiore importanza, può sembrare totalmente arbitraria senza poter ricavarne un processo logico; per questo vengono definiti spesso modelli a "scatola chiusa" [25].

Conclusioni

Vi è molta letteratura sulle metodologie di estrazione delle feature e sulla loro classificazione, con una preferenza generale verso tecniche a basso computazionale spesso usate in combinazione.

La tabella III.1 riporta una panoramica delle combinazioni fra le tecniche di estrazione delle feature e i classificatori proposti. In particolare, la tabella riporta gli autori e l'anno in cui è stato pubblicato l'articolo, la combinazione della tecnica di estrazione e il classificatore proposto, il numero di classi (definito multi-classe se il numero di classi è maggiore di tre), e il dataset su cui è stato fatto il test con il relativo grado di accuratezza.

Autori	Anno	Tecnica di estrazione	Classificazione	Classi	Dataset	Accuratezza
Chen *et al.* [29]	2014	CSP	LDA	2	Preparato dall'autore	91.25%
Linding-Leon *et al.* [30]	2015	CSP	LDA	Multi-classe	Preparato dall'autore	51.67%
Kim *et al.* [31]	2018	CSP	RFLDA	2	BCI competition III 4A	74.28%
Aler *et al.* [32]	2010	CSP	SVM	3	BCI competition III	69.94%
Yi *et al.* [33]	2013	CSP	SVM	Multi-classe	Preparato dall'autore	70%
Abdalsalam *et al.* [34]	2014	WT	SVM	3	BCI competition III	85.54%
Abdalsalam *et al.* [34]	2014	WT	NN	3	BCI competition III	82.43%
Sagee *et al.* [35]	2017	Decomposizione WT	NN	Multi-classe	Physionet	93.05%
Zhang *et al.* [36]	2017	STFT	CNN	2	Preparato dall'autore	89.46%

Tabella III.1: Panoramica delle prestazioni di BCI basate su MI; estratta e adattata da [2].

Da questa analisi emerge come la tecnica di estrazione delle feature più diffusa siano la CSP e le sue varianti, soprattutto per quanto riguarda la classificazione binaria. Essa separa le feature in base alla matrice di covarianza, facilitandone così la classificazione.

Il classificatore più popolare è il SVM, grazie alla sua robustezza e per la sua insensibilità all'alta dimensionalità dei dati; un aspetto di fondamentale importanza nell'ambito del machine learning. Se il segnale EEG è sufficientemente lineare, la LDA può fornire risultati anche più accurati per un costo computazionale minore rispetto ai SVM.

Le NN e il deep learning sono le tematiche più esplorate nella ricerca recente. Essi offrono un elevato grado di accuratezza ed è stato anche proposto di utilizzarli non solo per classificare le feature, ma anche per estrarle. Questa elevata accuratezza e flessibilità derivano principalmente da analisi a posteriori attraverso l'analisi di dataset, i quali presentano una criticità: i dataset ufficiali, come quelli delle *BCI competition*, sono pochi e per questa ragione ogni laboratorio addestra le NN su dataset propri; questo rende un confronto diretto tra modelli difficile, dato che sono stati addestrati con dati molto differenti. Inoltre, questo fatto rende queste tecniche di deep learning non idonee per la classificazione in tempo reale di task di MI, un aspetto di vitale importanza per la praticità dell'utilizzo di una BCI.

Una sfida importante per la tecnologia BCI non è tanto l'avanzamento di queste tecniche, ma ottenere un SNR basso durante l'acquisizione del segnale: dato che il segnale EEG è misurato attraverso tecniche di acquisizione non invasive, è essenziale mantenere un'alta qualità di acquisizione e accompagnarla con tecniche di condizionamento del segnale allo stato dell'arte. Come evidenziato da Janapati *et al.* [37], la letteratura scientifica si è concentrata principalmente sull'avanzamento delle tecniche di estrazione e di classificazione, ma i filtri usati nel condizionamento del segnale sono quelli usati nell'elaborazione del segnale tradizionale. Vi è una quasi totale mancanza di innovazione in questo aspetto della ricerca per le BCI e questo fatto è peculiare dato che la qualità del segnale rappresenta uno dei "colli di bottiglia" più evidenti. Investire su tecniche innovative di condizionamento potrebbe alleggerire il carico computazionale del classificatore e migliorare di conseguenza la qualità di utilizzo per i piloti. Questo aspetto non è stato approfondito in questo elaborato, ma rappresenta un'interessante direzione di ricerca futura che merita una maggiore attenzione.

Bibliografia

[1] O. George, R. Smith, P. Madiraju, N. Yahyasoltani e S. I. Ahamed, «Motor imagery: A review of existing techniques, challenges and potentials,» in *Proceedings - 2021 IEEE 45th Annual Computers, Software, and Applications Conference, COMPSAC 2021*, Institute of Electrical e Electronics Engineers Inc., lug. 2021, pp. 1893–1899, isbn: 9781665424639. doi: 10.1109/COMPSAC51774.2021.00286.

[2] S. Aggarwal e N. Chugh, «Signal processing techniques for motor imagery brain computer interface: A review,» *Array*, vol. 1-2, gen. 2019, issn: 25900056. doi: 10.1016/j.array.2019.100003.

[3] M. Zhuang, Q. Wu, F. Wan e Y. Hu, «State-of-the-art non-invasive brain–computer interface for neural rehabilitation: A review,» *Journal of Neurorestoratology*, vol. 8, pp. 12–25, 1 mar. 2020, issn: 23242426. doi: 10.26599/jnr.2020.9040001.

[4] L. F. Nicolas-Alonso e J. Gomez-Gil, *Brain computer interfaces, a review*, feb. 2012. doi: 10.3390/s120201211.

[5] J. R. Wolpaw, N. Birbaumer, D. J. Mcfarland, G. Pfurtscheller e T. M. Vaughan, «Brain-computer interfaces for communication and control,» rapp. tecn., 2002, pp. 767–791. indirizzo: www.elsevier.com/locate/clinphCLINPH2001764.

[6] G Pfurtscheller e F. H. L. D. Silva, «Invited review Event-related EEG/MEG synchronization and desynchronization: basic principles,» rapp. tecn. indirizzo: www.elsevier.com/locate/clinph.

[7] C Neuper e G Pfurtscheller, «Event-related dynamics of cortical rhythms: frequency-specific features and functional correlates,» rapp. tecn., 2001, p. 4158.

[8] Purves, *Neuroscience 6th Edition*. Oxford University Press, Incorporated, 2017, isbn: 9781605356372. indirizzo: https://books.google.it/books?id=oWWnzgEACAAJ.

[9] N. Mrachacz-Kersting, X. N. Jiang, A. James et al., «Efficient neuroplasticity induction in chronic stroke patients by an associative brain-computer interface,» *J Neuro-physiol*, vol. 115, pp. 1410–1421, 2016. doi: 10.1152/jn.00918.2015.-Brain-computer. indirizzo: www.jn.org.

[10] G. Pfurtscheller, C. Neuper, G. Muller et al., «Graz-BCI: state of the art and clinical applications,» *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 1–4, 2 giu. 2003, issn: 1534-4320. doi: 10.1109/TNSRE.2003.814454.

[11] B. Smith, *Elon Musk makes ALS TALK AGAIN || Nonverbal ALS Patient Uses Neuralink to Create & Narrate Video*, 2025. indirizzo: https://www.youtube.com/watch?v=CJnOWRKwg34.

[12] CYBLATHON, *CYBLATHON 2024*, 2024. indirizzo: https://cybathlon.com.

[13] CYBLATHON, «Luca Tonin: Treating Brain-Computer Interfaces as a neuroprosthesis,» *ETH Zürich*, 2024.

[14] D. Jaipriya e K. C. Sriharipriya, *Brain Computer Interface-Based Signal Processing Techniques for Feature Extraction and Classification of Motor Imagery Using EEG: A Literature Review*, set. 2024. doi: 10.1007/s44174-023-00082-z.

[15] A. Bounchaleun, «AN ELEMENTARY INTRODUCTION TO FAST FOURIER TRANSFORM ALGORITHMS,» 2019.

[16] G. Rodŕiguez-Beŕmudez e P. J. Gárcia-Laencina, «Automatic and adaptive classification of electroencephalographic signals for brain computer interfaces,» *Journal of Medical Systems*, vol. 36, SUPPL.1 2012, issn: 01485598. doi: 10.1007/s10916-012-9893-4.

[17] C.-H. Im, *Computational EEG Analysis: Methods and Applications*. Springer, 2018. indirizzo: http://www.springer.com/series/3740.

[18] A. S. Al-Fahoum e A. A. Al-Fraihat, «Methods of EEG Signal Features Extraction Using Linear Analysis in Frequency and Time-Frequency Domains,» *ISRN Neuroscience*, vol. 2014, pp. 1–7, feb. 2014, issn: 2314-4661. doi: 10.1155/2014/730218.

[19] P. D. Welch, «The Use of Fast Fourier Transform for the Estimation of Power Spectra,» rapp. tecn., 1967.

[20] S. M. Kay e S. L. Marple, «Spectrum Analysis-A Modern Perspective,» rapp. tecn., 1981.

[21] I. T. Hettiarachchi, T. T. Nguyen e S. Nahavandi, «Multivariate Adaptive Autoregressive Modeling and Kalman Filtering for Motor Imagery BCI,» in *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, Institute of Electrical e Electronics Engineers Inc., gen. 2016, pp. 3164–3168, isbn: 9781479986965. doi: 10.1109/SMC.2015.549.

[22] JonMcLoone. indirizzo: https://commons.wikimedia.org/w/index.php?curid=18679602.

[23] J. Müller-Gerking, G. Pfurtscheller e H. Flyvbjerg, «Designing optimal spatial filters for single-trial EEG classification in a movement task,» *Clinical Neurophysiology*, vol. 110, n. 5, pp. 787–798, 1999, issn: 1388-2457. doi: https://doi.org/10.1016/S1388-2457(98)00038-8. indirizzo: https://www.sciencedirect.com/science/article/pii/S1388245798000388.

[24] Y. Wang, S. Gao e X. Gao, «Common Spatial Pattern Method for Channel Selelction in Motor Imagery Based Brain-computer Interface,» in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 5392–5395. doi: 10.1109/IEMBS.2005.1615701.

[25] S. Shalev-Shwartz e S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms* (Understanding Machine Learning: From Theory to Algorithms). Cambridge University Press, 2014, isbn: 9781107057135. indirizzo: https://books.google.it/books?id=ttJkAwAAQBAJ.

[26] G. James, D. Witten, T. Hastie, R. Tibshirani e J. Taylor, *An Introduction to Statistical Learning: with Applications in Python* (Springer Texts in Statistics). Springer International Publishing, 2023, isbn: 9783031387463. indirizzo: https://books.google.it/books?id=bQgD0AEACAAJ.

[27] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche e B. Arnaldi, *A review of classification algorithms for EEG-based brain-computer interfaces*, giu. 2007. doi: 10.1088/1741-2560/4/2/R01.

[28] 3Blue1Brown, *Backpropagation calculus | Deep Learning Chapter 4*, 2018. indirizzo: https://www.youtube.com/watch?v=tIeHLnjs5U8.

[29] C.-Y. Chen, C.-W. Wu, C.-T. Lin e S.-A. Chen, «A novel classification method for motor imagery based on Brain-Computer Interface,» in *2014 International Joint Conference on Neural Networks (IJCNN)*, 2014, pp. 4099–4102. doi: 10.1109/IJCNN.2014.6889535.

[30] C. Lindig-Leon e L. Bougrain, «A multi-label classification method for detection of combined motor imageries,» in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 2015, pp. 3128–3133.

[31] Y.-J. Kim, N.-S. Kwak e S.-W. Lee, «Classification of motor imagery for Ear-EEG based brain-computer interface,» in *2018 6th International Conference on Brain-Computer Interface (BCI)*, IEEE, 2018, pp. 1–2.

[32] R. Aler, I. M. Galván e J. M. Valls, «Evolving spatial and frequency selection filters for brain-computer interfaces,» in *IEEE congress on evolutionary computation*, IEEE, 2010, pp. 1–7.

[33] W. Yi, S. Qiu, H. Qi, L. Zhang, B. Wan e D. Ming, «EEG feature comparison and classification of simple and compound limb motor imagery,» *Journal of NeuroEngineering and Rehabilitation*, vol. 10, p. 106, 1 dic. 2013, issn: 1743-0003. doi: 10.1186/1743-0003-10-106.

[34] E. A. Mohamed, M. Z. B. Yusoff, N. K. Selman e A. S. Malik, «Enhancing EEG signals in brain computer interface using wavelet transform,» *International Journal of Information and Electronics Engineering*, vol. 4, n. 3, p. 234, 2014.

[35] G. S. Sagee e S. Hema, «EEG feature extraction and classification in multiclass multiuser motor imagery brain computer interface u sing Bayesian Network and ANN,» in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2017, pp. 938–943. doi: 10.1109/ICICICT1.2017.8342691.

[36] J. Zhang, C. Yan e X. Gong, «Deep convolutional neural network for decoding motor imagery based brain computer interface,» in *2017 IEEE international conference on signal processing, communications and computing (ICSPCC)*, IEEE, 2017, pp. 1–5.

[37] R. Janapati, V. Dalal e R. Sengupta, «Advances in modern EEG-BCI signal processing: A review,» *Materials Today: Proceedings*, vol. 80, pp. 2563–2566, gen. 2023, issn: 22147853. doi: 10.1016/j.matpr.2021.06.409.

.