

Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais

*Campus* Juiz de Fora

Engenharia Mecatrônica

**Otto Luiz Andrade Glass**

**Comparação de técnicas de aprendizagem de máquina e processamento de  
sinal para interfaces cerebrais**

Juiz de Fora

2020

**Ficha catalográfica elaborada através do Modelo Latex do  
IFSUDESTEMG - *Campus* Juiz de Fora com os dados  
fornecidos pelo autor**

GLASS, Otto Luiz Andrade.

Comparação de técnicas de aprendizagem de máquina e processamento de sinal para interfaces cerebrais / Otto Luiz Andrade Glass, 2020.

49 f.

Orientador: Thiago da Silva Castro

Monografia (Graduação em Engenharia Mecatrônica) –Instituto Federal do Sudeste de Minas Gerais –*Campus* Juiz de Fora, 2020.

1. Interface Cerebral. 2. Aprendizagem de Maquinas. 3. Ondas Mu. I. Instituto Federal do Sudeste de Minas Gerais. II. Título.

Otto Luiz Andrade Glass

Comparação de técnicas de aprendizagem de máquina e processamento de  
sinal para interfaces cerebrais

Trabalho de Conclusão de Curso apresentado  
ao *Campus* Juiz de Fora do Instituto Fede-  
ral de Educação, Ciência e Tecnologia do Su-  
deste de Minas Gerais como requisito parcial  
para obtenção do grau de Bacharel em Enge-  
nharia Mecatrônica.

Orientador: Thiago da Silva Castro

Juiz de Fora

2020

Otto Luiz Andrade Glass

**Comparação de técnicas de aprendizagem de máquina e processamento de sinal para interfaces cerebrais**

Trabalho de Conclusão de Curso apresentado ao *Campus* Juiz de Fora do Instituto Federal de Educação, Ciência e Tecnologia do Sudeste de Minas Gerais como requisito parcial para obtenção do grau de Bacharel em Engenharia Mecatrônica.

Aprovada em: DD/MM/AAAA

BANCA EXAMINADORA

---

Prof. Dr. Thiago da Silva Castro - Orientador  
Instituto Federal Sudeste de MG - Campus Juiz de Fora

---

Prof. Dr. Beltrano  
Instituição

---

Prof. Dr. Beltrano  
Instituição

## AGRADECIMENTOS

“Algo só é impossível até que alguém duvide dele e prove o contrário”.  
(Albert Einstein)

## RESUMO

O objetivo deste trabalho é realizar a comparação de vários métodos de aprendizagem de máquina e processamento de sinais cerebrais. Analisando a sua performance. Verificando como estes fatores afetam a qualidade da classificação facilitando o desenvolvimento de uma Interface cerebral baseado em sinais motores.

Os métodos utilizados foram o Support Vector Machine (SVM) e Redes Neurais Artificiais (*Artificial Neural Networks*) (ANN), métodos já tradicionais em aplicações de classificação. O trabalho vai em conjunto com métodos de processamentos de sinais os utilizados no trabalho foram Multitaper Power Spectral density (PMTM) , Periodograma (PSDp) e Espectro de Welch (PWelch).

Palavras-chave: Interface Cerebral, Aprendizagem de Maquinas, Ondas Mu

## ABSTRACT

The objective of this work is to compare a diverse set of methods of machine learning and signal processing, analyzing the performance. To be able to understand how these factors affect the quality of the classification for a BCI. The machine learning algorithms used for this work were SVM and ANN. The signal processing methods used were PMTM, PSDp and PWelch. The SVM with a linear kernel was superior to all other machine learning methods.

Palavras-chave: Interface Cerebral, Aprendizagem de Maquinas, Ondas Mu

Keywords: Brain Computer Interface, Machine Learning, Mu wave



## LISTA DE FIGURAS

Figura 1 – Estágios de uma interface cerebral (RAO, 2010) . . . . .	15
Figura 2 – Posições do sistema 10-20 sobre um crânio. Obtido em (Robert Plonsey, Jaako Malmivuo, 1995) . . . . .	17
Figura 3 – Comparação do sinal Monopolar (B) vs o Bipolar (A). Obtido em (Robert Plonsey, Jaako Malmivuo, 1995) . . . . .	17
Figura 4 – (A,B) distribuição topográfica no couro cabeludo da diferença calculada do movimento da mão direita (A) real e (B) imaginário de uma pessoa vs ela relaxada com o sinal numa banda 10,5-13,5Hz. (C) O espectro para um outro sujeito em C3 comparando o estado relaxado (linha cortada) vs imaginação motora e (D) o espectro $r^2$ para a imaginação motora vs o estado relaxado deste sujeito (BCI2000, 2018) . . . . .	18
Figura 5 – O Efeito da imaginação motora esquerda vs direita no ritmo $\mu$ tal como captado pelos eletrodos em C3 (lado esquerdo) e C4 (lado direito) (QIN LEI DING, 2004) . . . . .	19
Figura 6 – Diagrama neuronio . . . . .	22
Figura 7 – Função Sinal . . . . .	22
Figura 8 – Função logística . . . . .	23
Figura 9 – Função tangente hiperbólica . . . . .	24
Figura 10 – Diagrama de MLP genérico . . . . .	24
Figura 11 – Exemplo de NN para calcular o Backpropagation . . . . .	26
Figura 12 – NN amp . . . . .	27
Figura 13 – Aproximação linear em (A) vs aproximação quadrática em (B) . . . . .	27
Figura 14 – Fluxograma do SCG baseado na explicação encontrada no artigo (MØLLER, 1993) . . . . .	28
Figura 15 – alguns dos hiperplanos possíveis para a classificação das amostras . . . . .	30
Figura 16 – SVM . . . . .	30
Figura 17 – NN (A) onde o ponto desconhecido (vazio) é classificado como vermelho e (B) onde o ponto desconhecido é classificado como preto devido a um ponto preto irregular. . . . .	32
Figura 18 – Espectro de Welch . . . . .	34
Figura 19 – Multitaper . . . . .	35
Figura 20 – segmento EOG nas sessões do graz-b . . . . .	37
Figura 21 – Os dois tipos de sessão em Graz-b (LEEB C. BRUNNER, 2008). . . . .	38
Figura 22 – diagrama compare methods . . . . .	39
Figura 23 – diagrama do Trials . . . . .	39
Figura 24 – Amostragem e extração de características . . . . .	41
Figura 25 – diagrama do Database . . . . .	41
Figura 26 – diagrama de herança do FeatureExtractionFnc . . . . .	42

Figura 27 – diagrama de herança do Classifier . . . . .	43
Figura 28 – (A) Kappa Overlap vs Window (B) Kappa Overlap vs Window para o experimento completo . . . . .	45
Figura 29 – Performance média dos classificadores utilizados . . . . .	46
Figura 30 – Performance média das Funções de extração utilizadas vs Janela de amostragem . . . . .	46

## LISTA DE TABELAS

Tabela 1	– Métodos da classe Database . . . . .	41
Tabela 2	– Métodos da classe Classifier . . . . .	43
Tabela 3	– Melhores performances na métrica do experimento . . . . .	47
Tabela 4	– Melhores performances por Janela de amostragem de até 2s . . . . .	47
Tabela 5	– Melhores performances por Janela de amostragem de até 0.5s . . . . .	47
Tabela 6	– Resultados da competição BCI-IV 2008 . . . . .	48

## LISTA DE ABREVIATURAS E SIGLAS

<b>ANN</b>	Redes Neurais Artificiais( <i>Artificial Neural Networks</i> )
<b>ANN-2</b>	Rede neural artificial com 2 camadas ocultas contendo 20 neurônios cada
<b>ANN-3</b>	Rede neural artificial com 3 camadas ocultas contendo 20 neurônios cada
<b>BCI</b>	Interface Cerebral( <i>Brain-Computer Interface</i> )
<b>BCIs</b>	Interfaces Cerebrais( <i>Brain-Computer Interfaces</i> )
<b>C-SVM</b>	SVM Gaussiano Grosso
<b>DFT</b>	Transformada de Fourier Discreta ( <i>Discrete Fourier Transform</i> )
<b>ECG</b>	Eletrocardiograficos
<b>EEG</b>	Eletroencefalograma
<b>EEGs</b>	Eletroencefalogramas
<b>EOG</b>	Eletrooculograma
<b>ERD</b>	Event Related Desynchronization
<b>ERS</b>	Event Related Synchronization
<b>F-SVM</b>	SVM Gaussiano Fino
<b>ICA</b>	Anaálise de Componentes Independentes ( <i>Independent Component Analysis</i> )
<b>LDA</b>	Linear Discriminant Analysis
<b>L-SVM</b>	SVM Linear
<b>MLP</b>	perceptrons de múltiplas camadas( <i>Multilayer Perceptron</i> )
<b>OOP</b>	Programação Orientada a Objetos ( <i>Object Oriented Programing</i> )
<b>PCA</b>	Análise de Componente Principal ( <i>Principal Component Analysis</i> )
<b>PMTM</b>	Multitaper Power Spectral density
<b>PSD</b>	Power Spectral Density
<b>PSDp</b>	Periodograma
<b>PWelch</b>	Espectro de Welch

<b>SCG</b>	Scaled Conjugate Gradient
<b>SVM</b>	Support Vector Machine
<b>TCC</b>	Trabalho de Conclusão de Curso

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>14</b>
1.1	Objetivo . . . . .	15
<b>2</b>	<b>Metodologia . . . . .</b>	<b>16</b>
2.1	Eletroencefalograma (EEG) . . . . .	16
2.1.1	Características e Sistemas de Medição . . . . .	16
2.1.2	Ritmo $\mu$ . . . . .	18
2.1.3	Redução de Artefatos . . . . .	18
2.2	Redes Neurais . . . . .	21
2.2.1	Neurônios Artificiais . . . . .	21
2.2.2	Tipos de Funções de ativação . . . . .	22
2.2.3	Perceptron de múltiplas camadas . . . . .	23
2.2.4	Aprendizagem . . . . .	25
2.2.5	Backpropagation(BP) . . . . .	25
2.2.6	Scaled Conjugate Gradient(SCG) . . . . .	26
2.3	Support Vector Machine(SVM) . . . . .	29
2.3.1	Hiperplanos . . . . .	29
2.3.2	Dedução do hiperplano ótimo . . . . .	29
2.3.3	Kernel . . . . .	31
2.4	Avaliação de performance de classificação . . . . .	32
2.4.1	Precisão de classificação . . . . .	32
2.4.2	Coefficiente Kappa . . . . .	32
2.5	Métodos de estimativa de espectro (PSD) . . . . .	33
2.5.1	Periodograma . . . . .	33
2.5.2	Espectro de Welch (PWelch) . . . . .	33
2.5.3	<i>Multitaper Power Spectral density</i> (PMTM) . . . . .	35
<b>3</b>	<b>Desenvolvimento . . . . .</b>	<b>37</b>
3.1	Amostras . . . . .	37
3.2	Framework . . . . .	39
3.2.1	Compare Methods . . . . .	39
3.2.2	Trials . . . . .	39
3.2.3	Database . . . . .	40
3.2.4	FeatureExtractionFnc . . . . .	42
3.2.5	Classifier . . . . .	43
3.3	Métodos de avaliação de performance . . . . .	44

<b>4</b>	<b>Resultados e Análises . . . . .</b>	<b>45</b>
4.1	Performances Médias . . . . .	45
4.1.1	Overlap vs Janela de Amostragem . . . . .	45
4.1.2	Classificadores . . . . .	45
4.1.3	Funções de extração vs Janela de Amostragem . . . . .	46
4.2	Melhores Performances . . . . .	47
<b>5</b>	<b>Conclusão . . . . .</b>	<b>48</b>
	 <b>REFERÊNCIAS . . . . .</b>	 <b>49</b>

## 1 INTRODUÇÃO

Desde a descoberta da atividade elétrica do cérebro e da invenção do Eletroencefalograma (EEG), por Hans Berger em 1924, foram desenvolvidas várias idéias de como explorar esse acesso à fonte de pensamentos, emoções e ações humanas (BLANKERTZ et al., 2016). Interfaces Cerebrais (*Brain-Computer Interfaces*) (BCIs) são dispositivos que se comunicam diretamente com os sinais cerebrais e permitem a interação direta entre com o meio ambiente (RAO, 2010).

Inicialmente esse campo foi desenvolvido com um foco na restauração de sentidos e mobilidade de pacientes, hoje em dia também existe pesquisa em aplicações não-médicas. Por exemplo, como um meio para verificar o alertidão do usuário executando um processo crítico, um dispositivo para jogos e amplificação física através de exoesqueletos (RAO, 2013)(BLANKERTZ et al., 2016).

Para tanto o objetivo de uma BCI é identificar e prever mudanças induzidas pelo comportamento ou o estado cognitivo no sinal cerebral do usuário (RAO, 2010).

Para o desenvolvimento de uma BCI normalmente são implementadas as seguintes etapas, como visto na figura 1 (RAO, 2013):

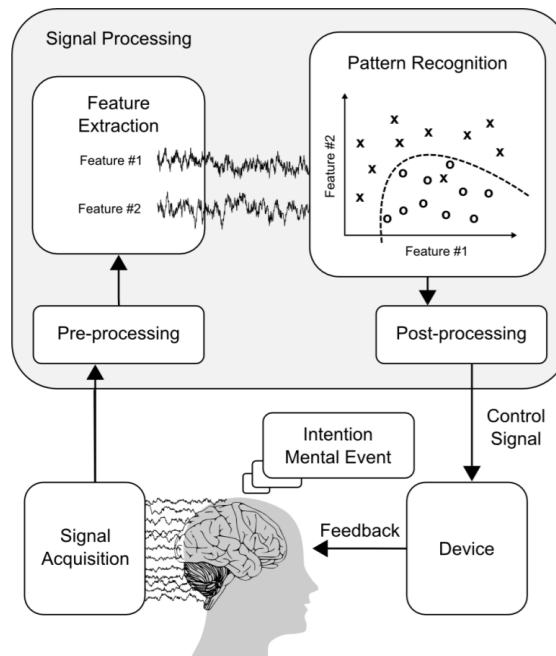
1. **Captação do Sinal cerebral:** Sinais cerebrais podem ser captados usando técnicas invasivas, semi-invasivas ou não-invasivas.
2. **Processamento do Sinal:** Sinais brutos são processados após a aquisição e técnicas de redução de artefatos e extração de características são utilizadas.
3. **Aprendizagem de maquina:** Neste estágio se gera o sinal de controle normalmente utilizando-se de algoritmos de aprendizagem de máquina.
4. **Feedback sensorial:** O sinal de controle gera uma mudança no ambiente. Alguns desses mudanças podem ser vistas, ouvidas ou sentidas pelo usuário.
5. **Processamento de sinais para a estimulação:** Antes de estimular uma região particular no cérebro é importante sintetizar um padrão de atividade que assemelha a atividade normalmente associada àquela região.
6. **Estimulação cerebral:** O padrão de estimulação da etapa anterior é utilizada em conjunto com uma técnica de estimulação invasiva ou não-invasiva para estimular o cérebro.

Durante o desenvolvimento de interfaces cerebrais não-invasivas utilizando-se do EEG encontram-se dois obstáculos principais: O fato do sinal ser não estacionário e sua



variabilidade inerente (RAO, 2010). Dados do mesmo paradigma experimental mas de sessões diferentes provavelmente exibirão diferenças devido a, por exemplo, ligeiras mudanças nas posições dos eletrodos ou mudanças nas propriedades eletromecânicas eletrodos tal como sua impedância (RAO, 2010). Além disto, a superposição ruidosa e não linear da atividade das populações medidas no couro cabeludo podem mascarar os padrões neurais e dificultar sua detecção (RAO, 2010). Devido a esses fatores, técnicas de processamento de sinal estatísticas e de aprendizagem de máquina tem uma importância fundamental durante o processo de reconhecimento de padrões EEG e traduzi-los para um sinal de controle (RAO, 2010).

Figura 1 – Estágios de uma interface cerebral (RAO, 2010)



## 1.1 Objetivo

O objetivo principal deste Trabalho de Conclusão de Curso (TCC) é realizar a comparação de métodos de aprendizagem de máquina e processamento de sinais cerebrais para melhor entender como estes fatores afetam a qualidade da classificação permitindo o desenvolvimento de uma BCI mais precisa.

Considerando estes fatores neste trabalho foram explorados e comparados métodos de aprendizagem de máquina tais como redes neurais e SVMs (kernel linear e gaussiano). Também foram comparados métodos estimativa de espectro (PMTM, Welch e periodograma) com diferentes configurações de comprimento e sobreposição entre as janelas amostradas. Para fazer isto foi necessário obter as amostras de eletroencefalograma para a classificação; desenvolver o framework para realizar as comparações; fazer a análise dos resultados.

## 2 Metodologia

### 2.1 Eletroencefalograma (EEG)

#### 2.1.1 Características e Sistemas de Medição

O EEG é uma das técnicas não invasivas mais populares para o desenvolvimento de BCIs devido à sua alta resolução temporal, baixo custo e por ser de fácil instalação (RAO, 2013).

Sinais EEG, coletados da superfície do couro cabeludo são flutuações de potenciais elétricos que refletem a atividade no cérebro principalmente no córtex cerebral abaixo da superfície do couro cabeludo (VIDAL, 1977). Correntes originárias de regiões mais profundas não são detectadas devido ao fato de que campos elétricos decaem com o quadrado da distância de sua origem. Portanto, o EEG predominantemente captura a atividade no córtex cerebral, cujo o arranjo colunar de neurônios e proximidade ao crânio favorecem a sua captura (RAO, 2013).

O padrão internacional sistema 10-20 é tipicamente utilizado para a gravação de EEG. Nesse sistema 21 eletrodos são posicionados na superfície do couro cabeludo. O posicionamento é determinado da seguinte maneira: Os pontos de referência são o nácion, que está localizado acima do nariz no nível dos olhos e o ínion que é o ressalto ossudo na base da crânio na linha central do traseiro da cabeça. A partir desses pontos, o perímetro da c é medido dividindo nos planos transverso e mediano. Os locais dos eletrodos podem ser determinadas dividindo esses perímetros em intervalos de 10% e 20%. Três outros eletrodos são posicionados em ambos os lados equidistantes dos pontos vizinhos como mostrado na figura 2. (Robert Plonsey, Jaako Malmivuo, 1995)

Eletrodos bipolares e monopolares podem ser utilizados para a medição do EEG (fig. 3). No primeiro método é medido a diferença entre um par de eletrodos (fig. 3-A). No método posterior o potencial de cada eletrodo é comparado a um eletrodo neutro ou à média de todos os eletrodos como pode ser observado na figura 3-B (Robert Plonsey, Jaako Malmivuo, 1995).

Figura 2 – Posições do sistema 10-20 sobre um crânio. Obtido em (Robert Plonsey, Jaako Malmivuo, 1995)

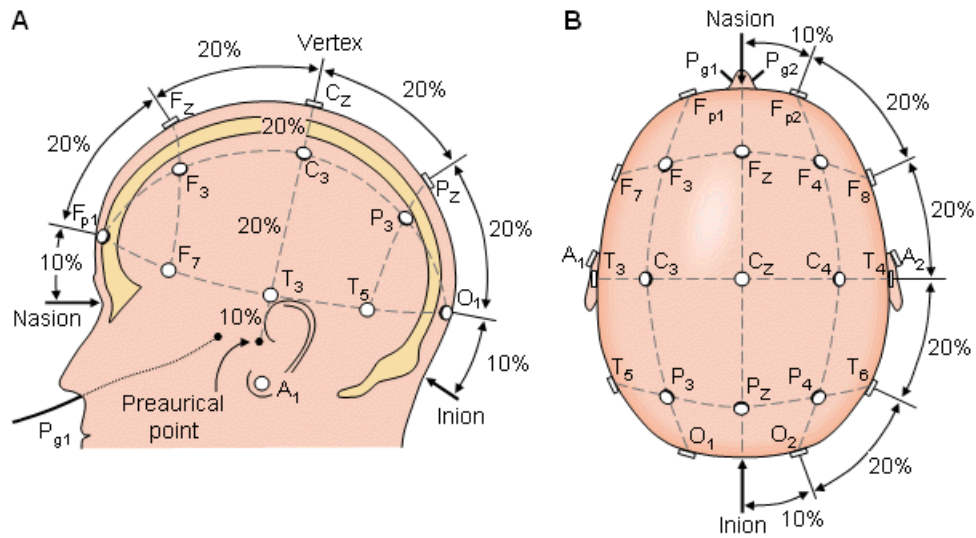
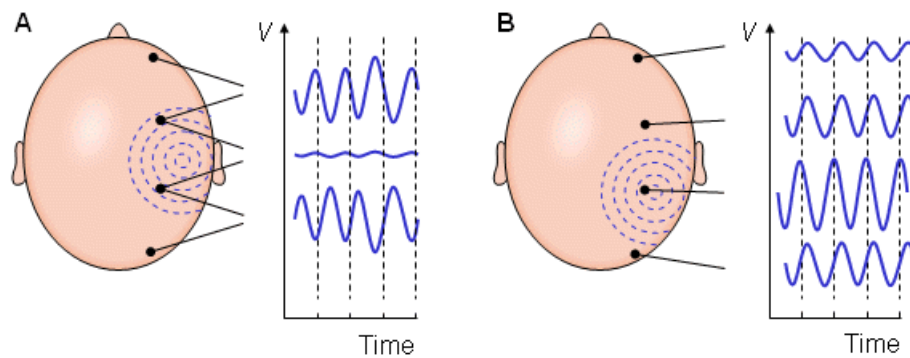


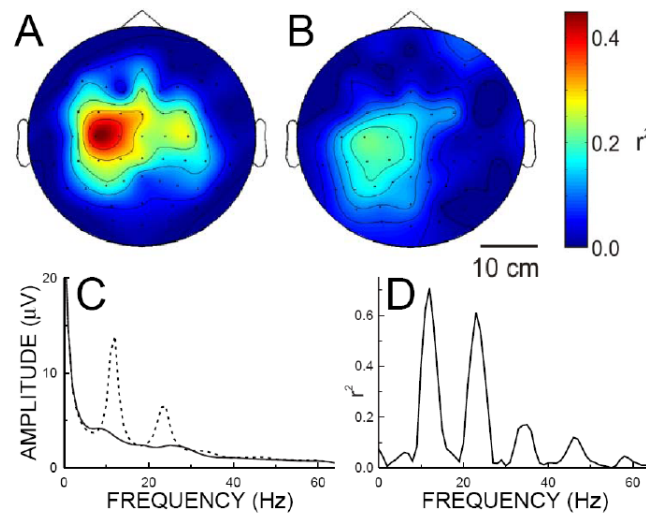
Figura 3 – Comparação do sinal Monopolar (B) vs o Bipolar (A). Obtido em (Robert Plonsey, Jaako Malmivuo, 1995)



### 2.1.2 Ritmo $\mu$

No EEG, em humanos a região próxima ao córtex motor exhibe tipicamente um sinal, de  $10\mu V$ - $50\mu V$  e na banda de frequência de aproximadamente 8-12Hz, enquanto não está produzindo atividade motora, este sinal é denominado de ritmo  $\mu$  (figura 4). Movimento real ou imaginário é normalmente acompanhada de uma redução no ritmo  $\mu$  no lado do cérebro oposto ao movimento como pode ser visto na figura 5. Esta redução de atividade é referida como Event Related Desynchronization (ERD) por (SCHLÖGL C. KEINRATH, 2006) e como Event Related Synchronization (ERS) por (VIDAL, 1977). (BCI2000, 2018) (RAO, 2013)

Figura 4 – (A,B) distribuição topográfica no couro cabeludo da diferença calculada do movimento da mão direita (A) real e (B) imaginário de uma pessoa vs ela relaxada com o sinal numa banda 10,5-13,5Hz. (C) O espectro para um outro sujeito em C3 comparando o estado relaxado (linha cortada) vs imaginação motora e (D) o espectro  $r^2$  para a imaginação motora vs o estado relaxado deste sujeito (BCI2000, 2018)

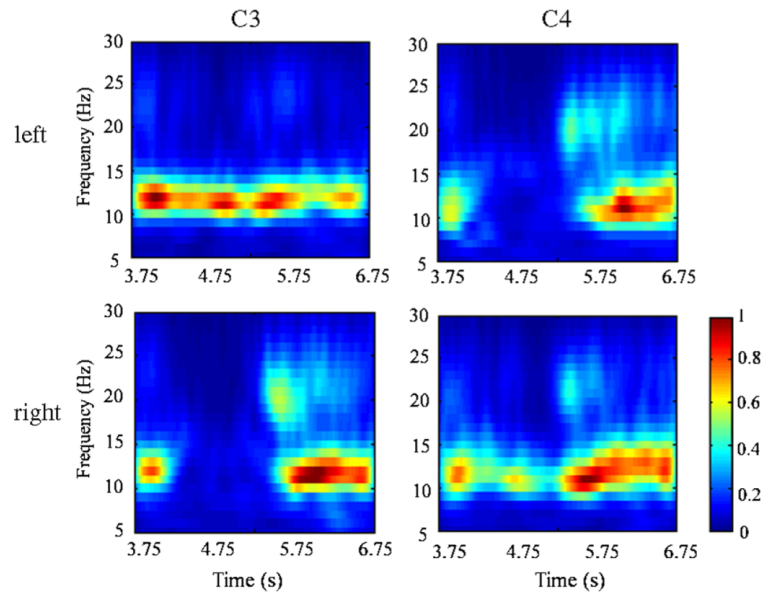


### 2.1.3 Redução de Artefatos

Artefatos são flutuações potências de origem não neural. Essas incluem potências eletro oculares, musculares (do pescoço, couro cabeludo e do rosto), Eletrocardiograficos (ECG) além de fontes externas como o ruído de rede elétrica 50/60Hz (VIDAL, 1977) (RAO, 2013).

Artefatos no geral e especificamente no Eletrooculograma (EOG) são uma grande fonte de ruídos em gravações de Eletroencefalogramas (EEGs). Pode-se assumir que toda gravação EEG está contaminada com artefatos EOG, pois movimentos oculares são difíceis de suprimir por períodos prolongados. Por exemplo um estudo em EEG do sono 9.1% do total gravação estava contaminada com artefatos EOG (SCHLÖGL C. KEINRATH, 2006).

Figura 5 – O Efeito da imaginação motora esquerda vs direita no ritmo  $\mu$  tal como captado pelos eletrodos em C3 (lado esquerdo) e C4 (lado direito) (QIN LEI DING, 2004)



A origem do EOG é devido à atividade elétrica do olho que é propagada pelo corpo e pode ser gravado pelo corpo em sua superfície. Artefatos EOG são causados pelos movimentos do dipolo retinal e das pálpebras. Um modelo simplificado assume um dipolo elétrico dentro dos olhos. A direção do dipolo é alinhada com a linha de visão e a amplitude do dipolo é determinada pela quantidade de luz atingindo a retina. Na maioria dos casos ambos olhos estão na mesma linha de visão e observam a mesma luminância. Portanto ambos os dipolos são paralelos e altamente correlacionados. Por conta disso o EOG pode ser modelado por um único dipolo. Devido aos efeitos da condução de volume, o EOG e EEG são propagados pela superfície da cabeça onde a superposição de ambos é gravado. Os pesos dos componentes dessas superposição são determinados pelas relações espaciais e propriedades elétricas dos tecidos entre as fontes e os eletrodos. Essas propriedades não se alteram durante uma gravação com a exceção do movimento das pálpebras que alteram a geometria do tecido proximo delas. Porém esse efeito do movimento das pálpebras pode ser modelado como uma componente radial do EOG, portanto os pesos dos componentes na superposição podem ser considerados estáticos (SCHLÖGL C. KEINRATH, 2006).

Para a redução de artefatos EOG existem várias técnicas dentre elas Análise de Componente Principal (*Principal Component Analysis*) (PCA), Análise de Componentes Independentes (*Independent Component Analysis*) (ICA) e regressão linear. A seguir será descrito o método discutido em (SCHLÖGL C. KEINRATH, 2006) que é uma regressão linear.

O seguinte modelo linear é assumido contendo 3 componentes espaciais (horizontal,

vertical e radial) de EOG:

$$Y(t, ch) = S(t, ch) + [EOG1(t), EOG2(t), EOG3(t)] \cdot [b_1(ch), b_2(ch), b_3(ch)]^T \quad (2.1)$$

Onde  $Y(t, ch)$  é o valor gravado de cada canal  $ch$  num tempo  $t$ ,  $S$  é a fonte do sinal sem contaminação de artefatos,  $EOG123$  indicam a fonte de ruído  $U$  das 3 componentes espaciais do EOG e  $b(ch)$  indicam os pesos dos componentes EOG no canal EEG  $ch$ . Para podermos obter o sinal corrigido a fonte de ruídos  $U$  e os pesos  $b$  tem que serem conhecidos. O EOG pode ser gravado num canal separado. Para obter  $b$  tem que se assumir de que o sinal  $S$  e o ruído  $U$  são linearmente independentes, assumindo isso.

$$S = Y - U \cdot b \quad (2.2)$$

$$\langle U^T S \rangle = \langle U^T Y \rangle - \langle U^T U \rangle b \quad (2.3)$$

onde  $\langle U^T S \rangle = 0$  resultando em

$$b = \langle U^T U \rangle^{-1} \langle U^T Y \rangle \quad (2.4)$$

permitindo que  $S$  seja calculado.

## 2.2 Redes Neurais

Redes neurais baseadas em Backpropagation tem se mostrado bem sucedido em uma alta variedade de tarefas de classificação, incluindo a classificação de dados de Interface Cerebral (*Brain-Computer Interface*) (BCI). Apesar de serem poderosas tais redes neurais frequentemente sofrem de um problema de *overfitting* aos dados de treinamento, resultando numa generalização fraca. Por consequência disso SVMs são tipicamente favorecidas sobre ANN como o algoritmo de escolha em muitas BCI (RAO, 2013). Mas o seu entendimento ainda é importante, pois novas técnicas estão sendo derivadas de ANN tal como Redes Neurais Convolucionais (??).

ANN são inspiradas pela sua contraparte na biologia e procuram reproduzir algumas das capacidades adaptativas de redes neurais no cérebro em classificar dados de entrada de maneira robusta (Simon Haykin, 2008).

### 2.2.1 Neurônios Artificiais

O modelo de um neurônio que forma as ANN é mostrado na figura 6 e consiste em 3 partes:

1. Um conjunto de sinápses (conexões) cada uma caracterizada por um peso  $w_{l\langle k,j \rangle}$  onde  $k$  é o índice do neurônio na camada  $l$  e  $j$  é o índice do nó de origem na camada anterior ( $l - 1$ ).
2. Um somador para somar todos os sinais de entrada multiplicado pelo seu **peso**  $w_{l\langle k,j \rangle}$  e o valor do **bias**  $b_{(l,k)}$ .
3. Uma função de ativação  $\varphi(s)$  para restringir a amplitude da saída de um neurônio. tipicamente a amplitude da saída de um neurônio varia de  $[0,1]$  ou  $[-1,1]$ .

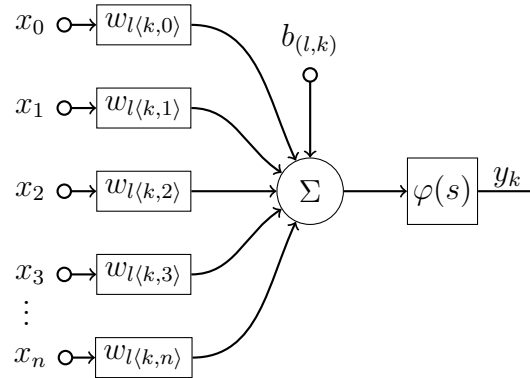
Este modelo também pode ser representado por esse par de equações:

$$s = \sum_{j=1}^n w_{l\langle k,j \rangle} x_j \quad (2.5)$$

e

$$y_k = \varphi(s + b_{(l,k)}) \quad (2.6)$$

Figura 6 – Diagrama neurônio



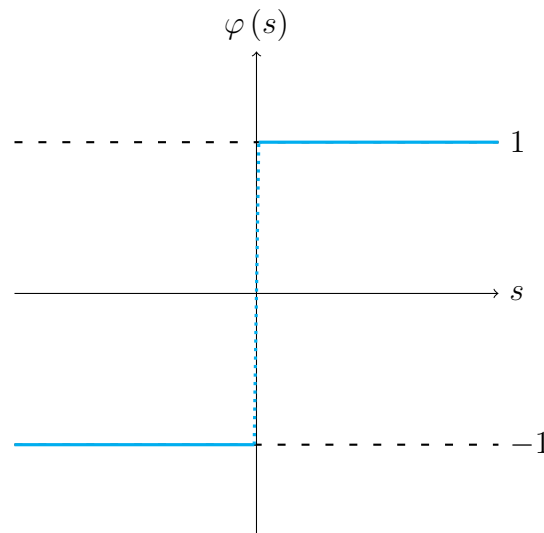
### 2.2.2 Tipos de Funções de ativação

- **Função Sinal:** é uma função de ativação simples descrita pela equação 2.7.

$$\varphi(s) = \begin{cases} 1 & \text{se } s > 0 \\ 0 & \text{se } s = 0 \\ -1 & \text{se } s < 0 \end{cases} \quad (2.7)$$

Ela é uma das funções de ativação mais simples e é utilizada no perceptron de uma camada, Linear Discriminant Analysis (LDA) e SVM entre outros.

Figura 7 – Função Sinal



- **Função Logística** é uma função do tipo sigmóide, cujo gráfico tem forma de S, é a função de ativação mais comum para o uso em ANN. Ela tem uma amplitude de  $[0,1]$  ela é dada pela equação 2.8.

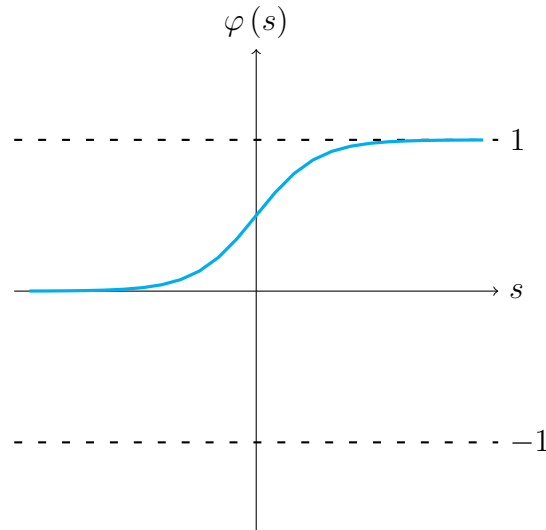
$$\varphi(s) = \frac{1}{1 + \exp(-as)} \quad (2.8)$$



Ela é derivável e sua derivada é extremamente simples e dada por eq. 2.9

$$\frac{d\varphi(s)}{ds} = \varphi(s)(1 - \varphi(s)) \quad (2.9)$$

Figura 8 – Função logística



- **Função tangente hiperbólica** é uma outra função do tipo sigmóide sua amplitude é de  $[-1,1]$  o fato de ser uma função ímpar reduz o numero de iterações nescessárias se comparado à função logística (mais detalhes sobre isso pode-se encontrar em (Simon Haykin, 2008) seção 4.11).

$$\varphi(s) = \tanh(s) \quad (2.10)$$

E sua derivada Eq. 2.11.

$$\frac{d\varphi(s)}{ds} = 1 - \varphi^2(s) \quad (2.11)$$

### 2.2.3 Perceptron de múltiplas camadas

As ANN chamadas de perceptrons de múltiplas camadas (*Multilayer Perceptron*) (MLP) consistem tipicamente de um conjunto de nós de entrada que constituem a *camada de entrada*, uma ou mais *camadas ocultas* e uma *camada de saída* isto pode ser visto na figura 10 (Simon Haykin, 2008).

Um MLP tem três características distintivas (Simon Haykin, 2008):

1. O modelo de cada neurônio inclui uma função de ativação não-linear continuamente diferenciável. A não linearidade é nescessária pois sem ela a rede pode ser reduzida a uma única camada (a camada de saída).

Figura 9 – Função tangente hiperbólica

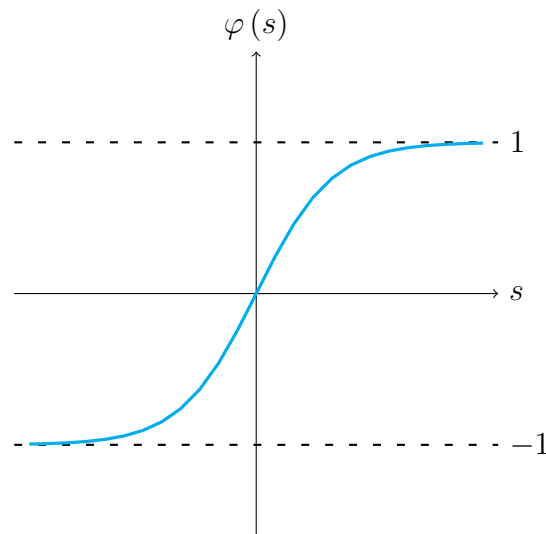
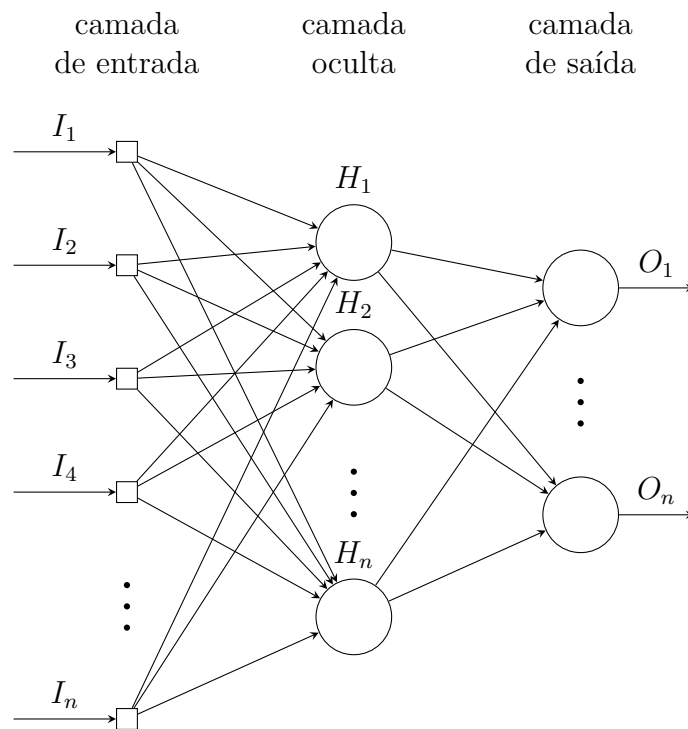


Figura 10 – Diagrama de MLP genérico



2. A rede contém uma ou mais camadas de neurônios ocultos, que não fazem parte da entrada ou saída da rede.
3. A rede exibe um alto grau de **conectividade**, determinado pelas sinapses da rede.

### 2.2.4 Aprendizagem

Dentre das regras de aprendizagem uma das mais antigas é a aprendizagem Hebbiana que simplificada diz (Simon Haykin, 2008):

1. Quando dois neurônios em ambos os lados da sinapse são ativados ao mesmo tempo o peso da conexão entre eles é fortalecido.
2. Quando dois neurônios em ambos lados da sinapse são ativados de maneira dessincronizada o peso da conexão entre eles é enfraquecido.

No caso do perceptrons de múltiplas camadas (*Multilayer Perceptron*) isto pode ser descrito pela eq. 2.12 onde  $\mathbf{w}_n$  é o vetor dos pesos e bias da rede na iteração  $n$ .

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta \mathbf{w}_n \quad (2.12)$$

$$\mathbf{w} = (w_{1\langle 1,1 \rangle}, \dots, w_{1\langle 1,j \rangle}, b_{(1,1)}, \dots, w_{L\langle K,J \rangle}, b_{(L,K)}) \quad (2.13)$$

Para determinar o valor de  $\Delta \mathbf{w}_n$ , uma abordagem é escolher um  $\Delta \mathbf{w}_n$  que minimize o erro de classificação da rede num conjunto de treinamento. Essa abordagem é denominada *aprendizagem por correção de erro* (Simon Haykin, 2008) e pode ser formulada pela eq. 2.14 onde,  $\eta$  é o tamanho do passo, também conhecido como *taxa de aprendizagem* e  $\mathbf{p}_n$  é um vetor com a direção que minimiza a função de erro de classificação  $E(\mathbf{w})$ .

$$\Delta \mathbf{w}_n = \eta \mathbf{p}_n \quad (2.14)$$

### 2.2.5 Backpropagation(BP)

O algoritmo Backpropagation é um algoritmo que lida com os pesos de uma rede de acordo com os erros obtidos em seus neurônios adjacentes. Ele utiliza um método gradiente como tentativa de minimizar o erro  $E_{total}(\mathbf{t}, \mathbf{o})$  entre os valores de saída e os valores alvos. A equação 2.15 mostra o princípio básico do calculo para o erro, sendo a métrica utilizada em algoritmos básicos o MSE (Erro médio quadrático). Sendo  $N$  o número de amostras,  $t_n$  é a reposta alvo e  $o_n$  é a resposta calculada da rede (Simon Haykin, 2008).

$$E_{total}(\mathbf{t}, \mathbf{o}) = \frac{1}{2N} \sum_{n=0}^N (t_n - o_n)^2 \quad (2.15)$$

Para cada entrada  $I_1$  e  $I_2$  e saída  $O_1$  da Figura 11 teremos:

- A propagação das entradas  $I_1$  e  $I_2$  pela rede, e computando a saída  $O_1$  para cada dado de treinamento.

- Para a saída  $O_1$  é calculado então um erro  $E_{total}(\mathbf{t}, \mathbf{o})$  que está relacionado à sua saída e o resultado alvo esperado.
- A partir disso é calculado os erros dos pesos, utilizando a regra da cadeia, aos neurônios ocultos das camadas internas. Retropropagando o erro desde as camadas de saída até a ultima camada interna (figura 12).
- Após o calculo dos erros de cada neurônio, é utilizado um método gradiente para o cálculo das novas sinapses de cada conexão. A equação 2.19 mostra um modo de como pode ser atualizado o peso sendo  $\Delta w_{ij}$  o peso da conexão entre os neurônios  $i$  e  $j$ ,  $\eta$  a Taxa de aprendizado e  $E(w)$  o próprio erro.

Para classificação uma das funções de erro mais utilizadas é entropia cruzada. Ela penaliza fortemente classificações incorretas e é dada pela eq. 2.16 onde  $o_n$  é a saída  $n$  da rede e  $t_n$  é o valor alvo daquela saída.

$$E_n(t_n, o_n) = \frac{-t_n \log(o_n)}{N} \quad (2.16)$$

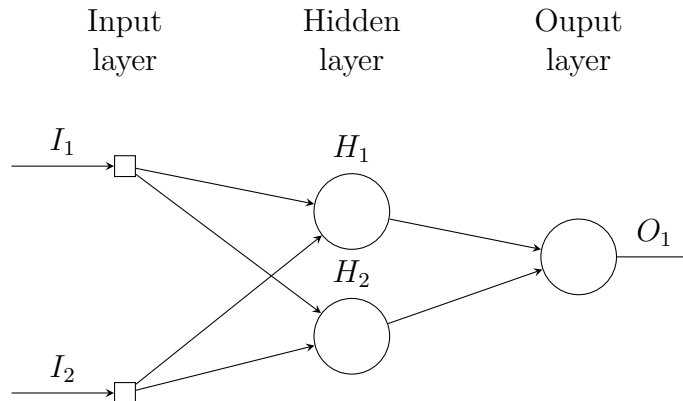
$$E_{total}(\mathbf{t}, \mathbf{o}) = \sum_{n=0}^N \frac{-t_n \log(o_n)}{N} \quad (2.17)$$

e sua derivada é

$$\frac{dE_n(t_n, o_n)}{do_n} = \frac{-t_n}{o_n N} \quad (2.18)$$

$$\Delta w_{ij} = -\eta \frac{\partial E(w)}{\partial w_{ij}} \quad (2.19)$$

Figura 11 – Exemplo de NN para calcular o Backpropagation



### 2.2.6 Scaled Conjugate Gradient(SCG)

Avaliando o procedimento de aprendizagem de uma rede neural como um problema de otimização, ele passa a ser o equivalente a minimizar a função de erro global, que é

Figura 12 – NN amp

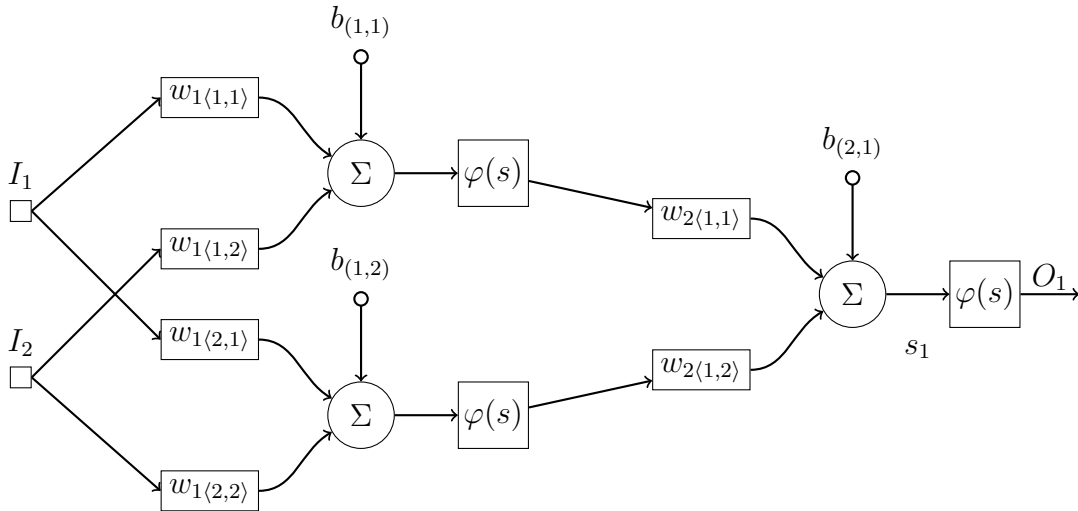
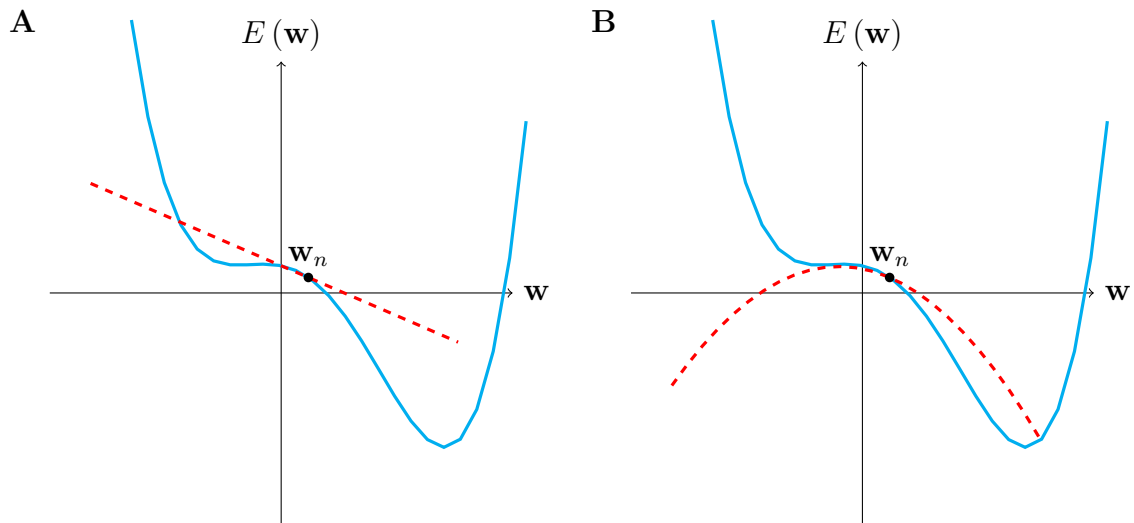
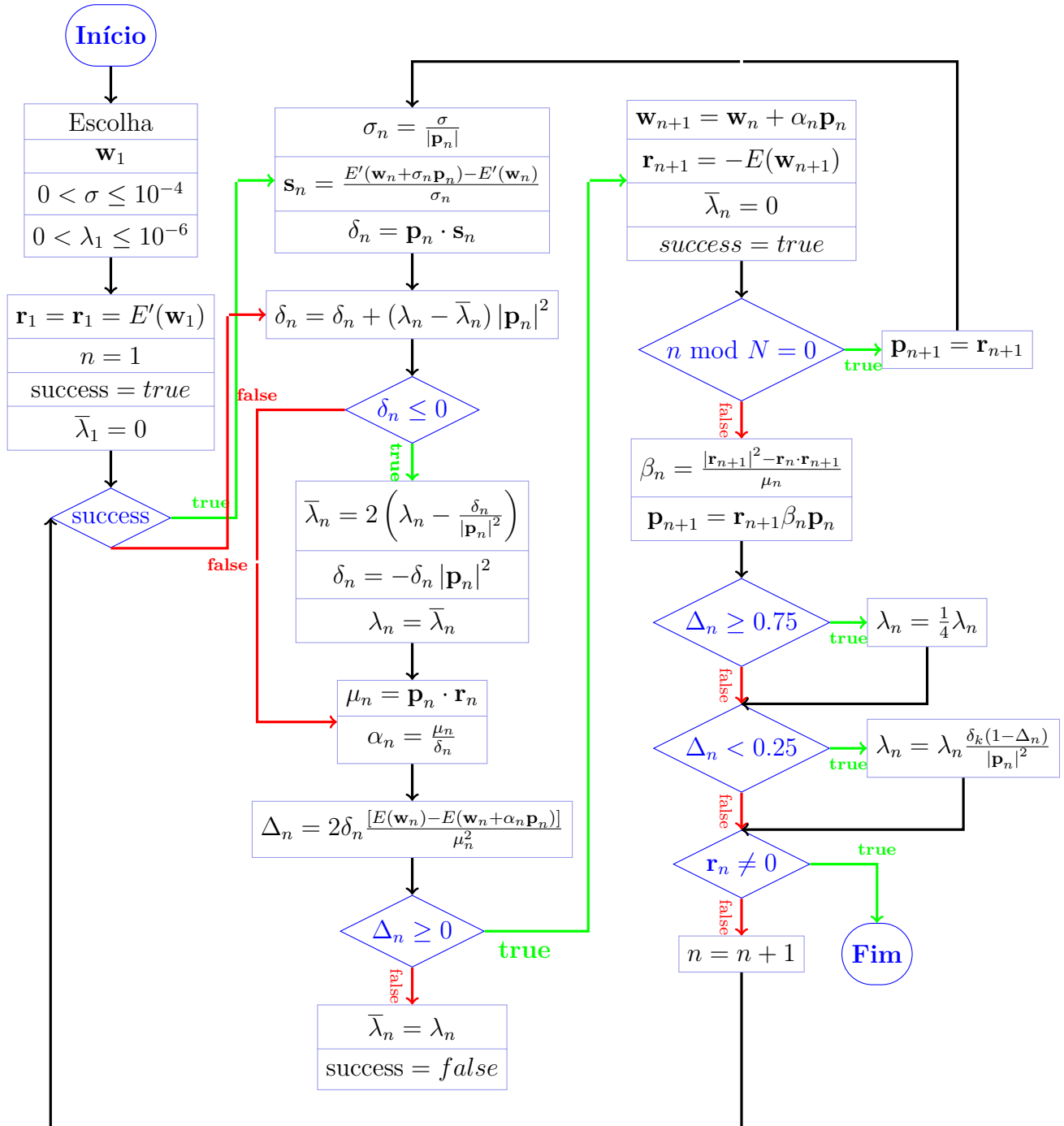


Figura 13 – Aproximação linear em (A) vs aproximação quadrática em (B)



uma função multivariável dependente dos pesos da rede . A maioria dos métodos de minimização utilizam-se da mesma estratégia . A minimização é um processo iterativo. No artigo do Møller foi desenvolvido uma variação do método do gradiente conjugado que evita executar uma pesquisa linear a cada iteração, utilizando o método de Levenberg-Marquardt (MØLLER, 1993). A maioria dos metodos de otimização utilizam a mesma abordagem, em que a minimização é um algoritmo iterativo onde a cada passo é atualizado a direção de busca e a distancia seguindo um algoritmo semelhante à de aprendizagem. No SCG ele utiliza-se de uma aproximação da função erro quadrática ao invéz linear, pois ela consegue convergir de maneira mais rapida como pode-se observar na Figura 13. Este algoritmo foi implementado para as ANN no MATLAB. Um fluxograma construído a partir da explicação do artigo do Møller está na figura 14.

Figura 14 – Fluxograma do SCG baseado na explicação encontrada no artigo (MØLLER, 1993)



## 2.3 Support Vector Machine(SVM)

De modo simples as SVM são hiperplanos que separam os dados de treinamento para uma margem máxima separando os dados de um lado como -1 e de outro lado o valor 1. As instancias de treinamento que ficam proximas ao hiperplano são chamados de Vetores Suporte. De maneira geral, as SVM permitem projetar os dados de treinamento original no espaço  $X$  para um espaço  $F$  via um kernel  $K$  (Simon Haykin, 2008).

SVM lineares tem sido utilizado com sucesso numa grande variedade de aplicações em BCIs (RAO, 2013). Nos casos em que SVM linear não é o suficiente, é possível utilizar-se do *kernel trick* para remapear os dados para um espaço dimensão mais alta onde eles são linearmente separáveis(VAPNIK, 1995)(KOWALCZYK, 2017).

### 2.3.1 Hiperplanos

Na geometria hiperplanos são um subespaço com uma dimensão menor do que o espaço no qual ele esta contido. Por exemplo, um hiperplano num plano é uma reta. A equação que define um hiperplano é eq. 2.20

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \quad (2.20)$$

hiperplanos podem ser utilizados para fazer um classificador binário se utilizarmos da função sinal(eq.2.7) obtemos eq. 2.21 onde os pontos à esquerda da reta são classificados como 1 e os à direita como -1 e observe que esse classificador é idêntico a um perceptron de uma camada usando a função sinal como função de ativação.

$$\varphi(\mathbf{w} \cdot \mathbf{x} + b) = \mathbf{y} \quad (2.21)$$

A desvantagem do perceptron é de que ele aleatoriamente escolhe um hiperplano dentre infinitos possíveis que minimiza o erro (figura 15) (KOWALCZYK, 2017). Isto pode não parecer problemático até se considerar de que o objetivo do algoritmo não é classificar os dados que nós possuímos agora mas sim os dados futuros de que ele irá encontrar.

A solução para esse problema foi encontrada por Vapnick onde ele demonstrou que o hiperplano ótimo (aquele tem que tem a melhor generalização) é aquele que criar a maior separação das amostras conhecidas (KOWALCZYK, 2017).

### 2.3.2 Dedução do hiperplano ótimo

Para encontrar o hiperplano ótimo queremos entrar o hiperplano que maximiza a distância  $D$  que separa as amostras das duas classes (figura 16). Essa distância pode ser obtida através da margem geométrica  $M$ .

$$M = \min_{i=1 \dots m} \gamma_i \quad (2.22)$$

$$\gamma_i = y_i \frac{\mathbf{w}}{\|\mathbf{w}\|} \cdot \mathbf{x}_i + \frac{b}{\|\mathbf{w}\|} \quad (2.23)$$

Figura 15 – alguns dos hiperplanos possíveis para a classificação das amostras

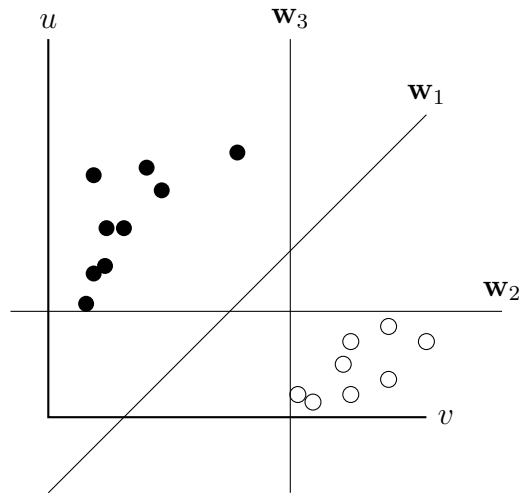
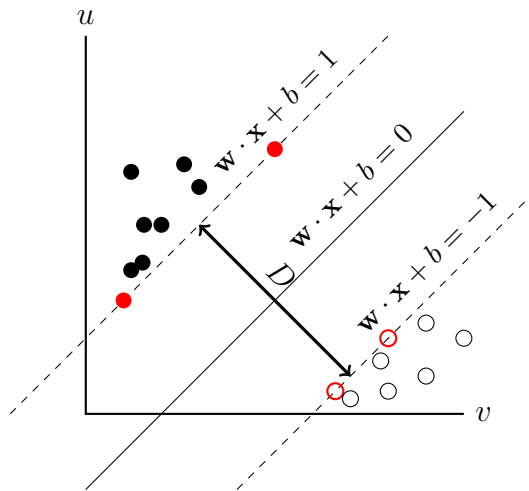


Figura 16 – SVM



onde  $y_i$  é a classificação da amostra  $(+1, -1)$  e  $x_i$  é o vetor de características da amostra.

Para encontrarmos  $\mathbf{w}$  e  $b$  encontrar a maior margem geométrica é equivalente a resolver a equação

$$\begin{aligned} &\underset{\mathbf{w}, b}{\text{minimizar}} && \frac{1}{2} \|\mathbf{w}\|^2 \\ &\text{sujeito a} && y_i(\mathbf{w} \cdot \mathbf{x}_i) + b - 1 \geq 0, \quad i = 1, \dots, m. \end{aligned} \quad (2.24)$$

Utilizando o método de Lagrange que diz que se tiver um problema do tipo

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimizar}} && f(\mathbf{x}) \\ &\text{sujeito a} && y_i(g(\mathbf{x}) = 0. \end{aligned} \quad (2.25)$$

o mínimo de  $f(\mathbf{x})$  é encontrado quando o seu gradiente aponta na mesma direção do



gradiente de  $g(\mathbf{x})$  em outras palavras quando

$$\nabla f(\mathbf{x}) = \alpha \nabla g(\mathbf{x}) \quad (2.26)$$

$$\nabla f(\mathbf{x}) - \alpha \nabla g(\mathbf{x}) = 0 \quad (2.27)$$

$$\mathcal{L}(\mathbf{x}, \alpha) = \nabla f(\mathbf{x}) - \alpha \nabla g(\mathbf{x}) \quad (2.28)$$

Utilizando o método de Lagrange na equação 2.24 obtemos

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^m \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i) + b - 1] \quad (2.29)$$

Como o este é um problema de minimização temos as seguintes condições (Simon Haykin, 2008):

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{0} \quad (2.30)$$

e

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} = 0 \quad (2.31)$$

a partir destas condições podemos obter as seguintes equações:

$$\begin{aligned} \mathbf{w} &= \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \\ \sum_{i=1}^N \alpha_i y_i &= 0 \end{aligned}$$

Utilizando o problema dual e simplificando obtemos

$$\mathbf{w}^2 = \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (2.32)$$

E utilizando a função objetivo  $\mathcal{L}(\mathbf{w}, b, \alpha) = Q(\alpha)$  obtemos a equação final:

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad (2.33)$$

Esta equação pode ser resolvida utilizando métodos de otimização para problemas convexos. Uma explicação mais detalhada de como deduzir pode ser encontrada em (KOWALCZYK, 2017) e (Simon Haykin, 2008).

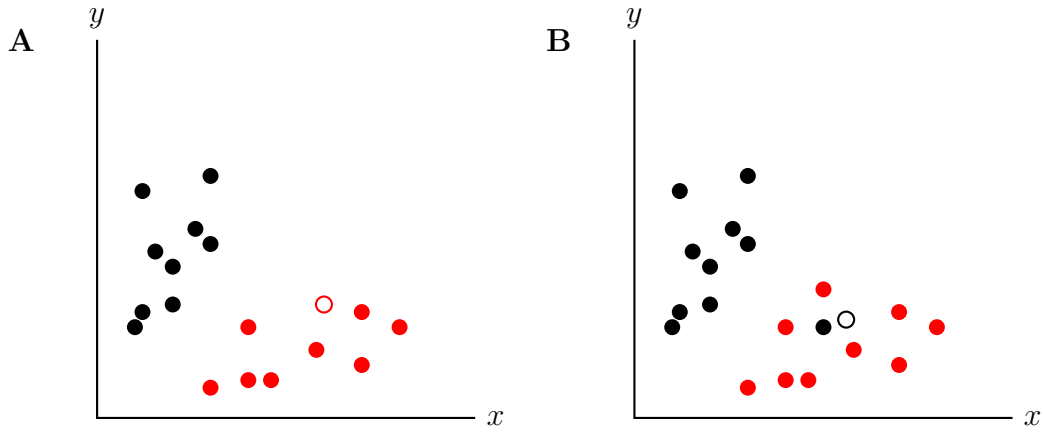
### 2.3.3 Kernel

Para dados que não são linearmente separáveis é possível fazer o uso do *Kernel Trick*. O *Kernel Trick* é um método que consiste em utilizar a propriedade de que para calcular os pesos é apenas necessário o produto escalar dos pontos de treinamento tal como pode ser visto na equação 2.33. Os pontos de treinamento podem ser transformados para um espaço vetorial diferente onde eles são linearmente separáveis. Um Kernel mapeia

o vetor de características para um espaço onde os dados são linearmente separáveis mas onde o produto escalar seja o mesmo. O Kernel gaussiano é um kernel dado pela equação 2.34 que mapeia os pontos de treinamento para um espaço vetorial com uma dimensão infinita

$$K_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \exp[-\gamma \|\mathbf{x} - \mathbf{x}'\|] \quad (2.34)$$

Figura 17 – NN (A) onde o ponto desconhecido (vazio) é classificado como vermelho e (B) onde o ponto desconhecido é classificado como preto devido a um ponto preto irregular.



## 2.4 Avaliação de performance de classificação

### 2.4.1 Precisão de classificação

Precisão de classificação é definida como a razão entre as amostras classificadas corretamente e o número total de amostras (RAO, 2013).

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \quad (2.35)$$

### 2.4.2 Coeficiente Kappa

Uma outra medida de performance é o coeficiente kappa de Cohen:

$$\kappa = \frac{ACC - ACC_0}{1 - ACC_0} \quad (2.36)$$

Onde  $ACC$  é a precisão de classificação e  $ACC_0$  é a probabilidade do classificador acertar a classe escolhendo uma classe aleatoriamente. Isso torna  $\kappa$  uma métrica independente do número de classes e amostras por classe (RAO, 2013). Um  $\kappa = 0$  é a performance onde a probabilidade de acerto é a mesma que a escolha aleatória e  $\kappa = 1$  é a performance perfeita.

## 2.5 Métodos de estimativa de espectro (PSD)

Power Spectral Density (PSD) extraem informações do sinal como um processo estocástico para descrever a distribuição de potência de um sinal no domínio da frequência (Pawel Herman, 2008). A PSD é definido como a transformada de Fourier da função de autocorrelação do sinal contanto que o sinal seja estacionário (Pawel Herman, 2008). Na prática, as características estatísticas do sinal não são conhecidas e só podem ser estimadas de uma sequência de amostras temporais.

### 2.5.1 Periodograma

A técnica de estimativa de espectro  $\hat{S}(f)$  mais comum é multiplicação da transformada de Fourier do sinal  $x(t)$  pela sua complexa conjugada, escalonando isto pelo numero de pontos amostrados  $N$  (DRONGELEN, 2014).

$$\hat{S}(f) = \frac{X(j2\pi f)X^*(j2\pi f)}{N} \quad (2.37)$$

$$\hat{S}(f) = \left| \sum_{t=0}^{N-1} x(t)a(t)e^{-2\pi jft} \right|^2 \quad (2.38)$$

Quando a janela  $a(t)$  é igual a 1 ela é chamada de janela retangular e esta estimativa é chamada de **periodograma** (DRONGELEN, 2014).

A confiabilidade da estimativa é significativamente reduzida quando há variância da estimativa do espectro em cada frequência  $f$  e quando há vazamento de energia em todas frequencias criando um bias (DRONGELEN, 2014). A vazamento é devido ao fato de que utilizamos uma seção do sinal que é o equivalente a utilizar uma janela retangular. (DRONGELEN, 2014). Uma solução a este problema é mutiplicar o sinal no domínio do tempo por uma janela não retangular com uma menor amplitude nas extremidades (DRONGELEN, 2014).

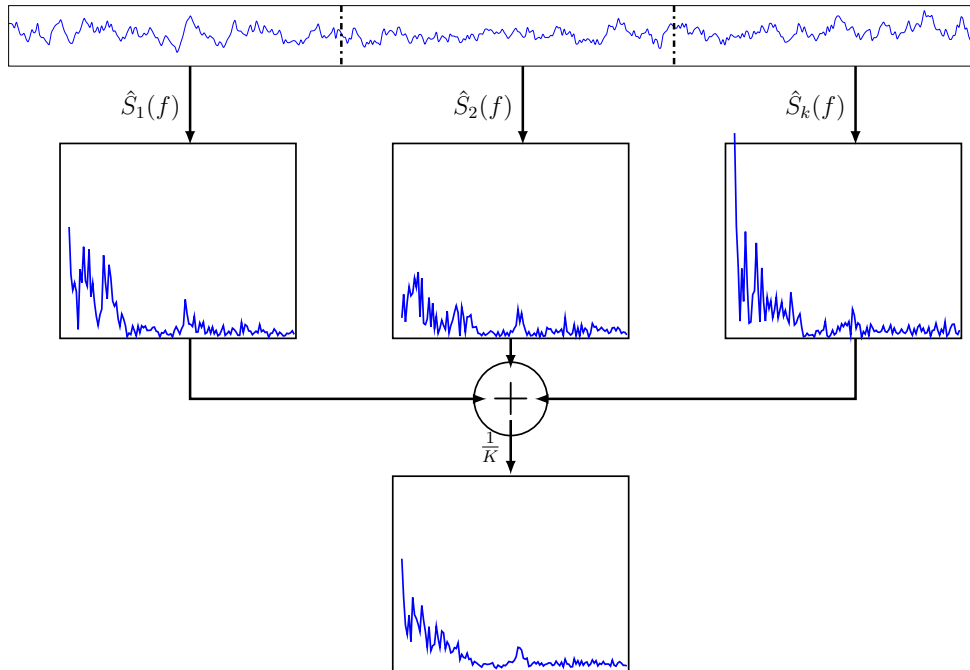
### 2.5.2 Espectro de Welch (PWelch)

O PWelch, também conhecido como método da média do periodograma consiste em dividir o sinal  $x(t)$  em  $K$  segmentos  $x_k(t)$  parcialmente sobrepostos e calcular a média do periodograma dos segmentos (JR., 1991).

$$\hat{S}(f) = \frac{1}{K} \sum_{k=1}^K \hat{S}_k(f) \quad (2.39)$$

Os passos desse algoritmo podem ser descritos assim:

Figura 18 – Espectro de Welch



1. divide o sinal

$$x[0], x[1], \dots, x[N-1]$$

em  $K$  segmentos de comprimento  $M$ :

Segmento 1:  $x[0], x[1], \dots, x[M-1]$

Segmento 2:  $x[S], x[S+1], \dots, x[M+S-1]$

$\vdots$

Segmento K:  $x[N-M], x[N-M+1], \dots, x[N-1]$

onde

$M$  = Número de pontos em cada segmento

$S$  = O deslocamento entre os segmentos

$K$  = Número de segmentos

2. Para cada segmento, calcule a Transformada de Fourier Discreta (*Discrete Fourier Transform*) (DFT) a uma frequência  $\nu = i/M$ :

$$X_k(\nu) = \sum_m x[m]w[m]e^{-j\pi\nu m}$$

onde

$$m = (k-1)S, \dots, M + (k-1)S - 1$$

$$w[m] = \text{ajanela}$$

3. Para cada segmento, calcule o valor do periodograma,  $P_k(f)$ , da transformada de fourier:

$$P_k(\nu) = \frac{1}{W} |X_k(\nu)|^2$$

onde

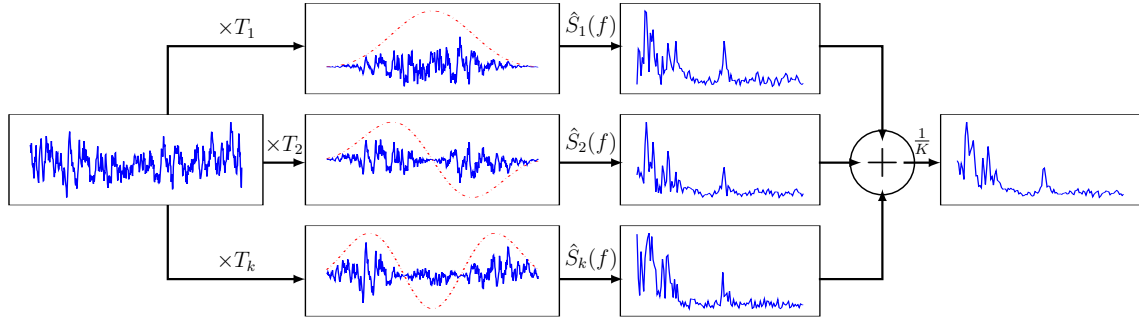
$$W = \sum_{m=0}^M w^2[m]$$

4. Calcule a média dos periodogramas para obter a estimativa de Welch:

$$S_x(\nu) = \frac{1}{K} \sum_{k=1}^K P_k(\nu)$$

### 2.5.3 Multitaper Power Spectral density (PMTM)

Figura 19 – Multitaper



PMTM, inicialmente descrito por Thomson em 1982, melhora a estimativa espectral ao analisar tanto a vazamento quanto a variância na estimativa (DRONGELEN, 2014). Nesta abordagem, cada janela  $v_k$  de um conjunto  $K$  de janelas seja ligeiramente diferente e reduza a vazamento de energia em várias frequências (DRONGELEN, 2014). Além disto, as janelas são ortogonais e são usadas pra providenciar  $K$  amostras ortogonais do sinal  $x(t)$ . Estas amostras são usadas para criar um conjunto de  $K$  estimativas  $\hat{S}_k(f)$  que podem ser utilizadas para calcular uma média  $\bar{S}(f)$  com variância reduzida (DRONGELEN, 2014).

$$\bar{S}(f) = \frac{1}{K} \sum_{k=1}^K \hat{S}_k(f) \quad (2.40)$$

De maneira geral, a aplicação de uma janela segue o seguinte padrão: Cada janela  $a(t)$  é associada com uma estimativa de espectro dado pela equação 2.38 (DRONGELEN, 2014). Para manter os valores de potência totais corretos, nós assumimos que as janelas são normalizadas de tal forma que  $\sum_{t=0}^{N-1} |a(t)|^2 = 1$  (DRONGELEN, 2014). Além disso, a potência do espectro de uma janela  $a(t)$   $|A(f)|^2$  e suas propriedades são importantes porque elas determinam, através da convolução, a estimativa do nosso espectro do sinal  $x(t)$  janelado (DRONGELEN, 2014).

$$\hat{S}(f) = \int_{-\frac{1}{2}}^{\frac{1}{2}} A(f')^2 S(f - f') df' \quad (2.41)$$

Suponha que queremos estimar o nosso espectro com uma banda de resolução  $W$ , que necessariamente configura um valor entre a resolução do espectro  $1/N$  e a frequência de Nyquist (DRONGELEN, 2014). Para simplificar a notação assumimos que as amostras entre a unidade e a frequência de seja normalizado em  $\frac{1}{2}$ . A fração  $\lambda$  da energia da janela dentro da banda selecionada é dada por (DRONGELEN, 2014):

$$\lambda(N, W) = \frac{\int_{-W}^W |A(f)|^2 df}{\int_{\frac{1}{2}}^{\frac{1}{2}} |A(f)|^2 df} \quad (2.42)$$

A ideia basicamente é de que queremos encontrar as janelas com o mínimo de vazamento ao maximizar  $\lambda$  a fração de energia dentro da banda  $W$  é maximizada (DRONGELEN, 2014). Em suma,  $\lambda$  é maximizado ao configurar a derivada da expressão da equação 2.42 com o vetor  $a(t)$  igual a zero (DRONGELEN, 2014). Isto é o equivalente a encontrar os auto valores de uma matriz  $N \times N$   $D$  com os componentes  $D_{t,t'} = \frac{2\pi W(t-t')}{\pi(t-t')}$  (DRONGELEN, 2014). Note de que esta função é simétrica portanto  $D$  é uma matriz simétrica.

$$D \cdot \mathbf{a} = \lambda \mathbf{a}$$

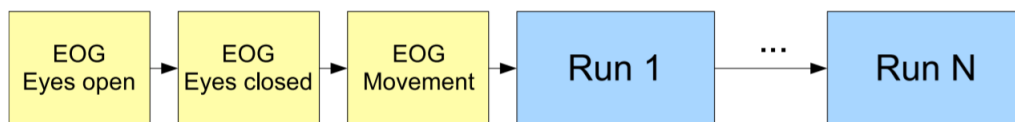
A solução tem  $N$  autovalores  $\lambda_k$  e autovetores  $(\nu_1, \nu_2, \dots, \nu_{N-1})$  ortonormais (DRONGELEN, 2014). O primeiro conjunto de componentes da matriz capturam a maioria das propriedades dessa janela ideal, enquanto os componentes seguintes capturam menos, a questão é quais componentes incluir quando se utiliza eles como janelas (DRONGELEN, 2014). O primeiro autovalor é próximo de um, portanto ele está associado a um excelente autovetor, em outras palavras uma janela que minimiza o vazamento (DRONGELEN, 2014).

### 3 Desenvolvimento

#### 3.1 Amostras

As amostras utilizadas neste trabalho foram obtidas do BCI Competition IV - Graz data set 2B e de acordo com (LEEB C. BRUNNER, 2008) ele se consiste em dados de EEG de 9 sujeitos em um estudo. Para cada um dos sujeitos 5 sessões foram gravadas em quais primeiras duas sessões contem dados sem feedback (screening), e as ultimas três sessões foram gravadas com feedback. Nestas sessões os sujeitos tiveram que executar uma imaginação motora de duas categorias diferentes (esquerda ou direita) dependendo de uma deixa.

Figura 20 – segmento EOG nas sessões do graz-b



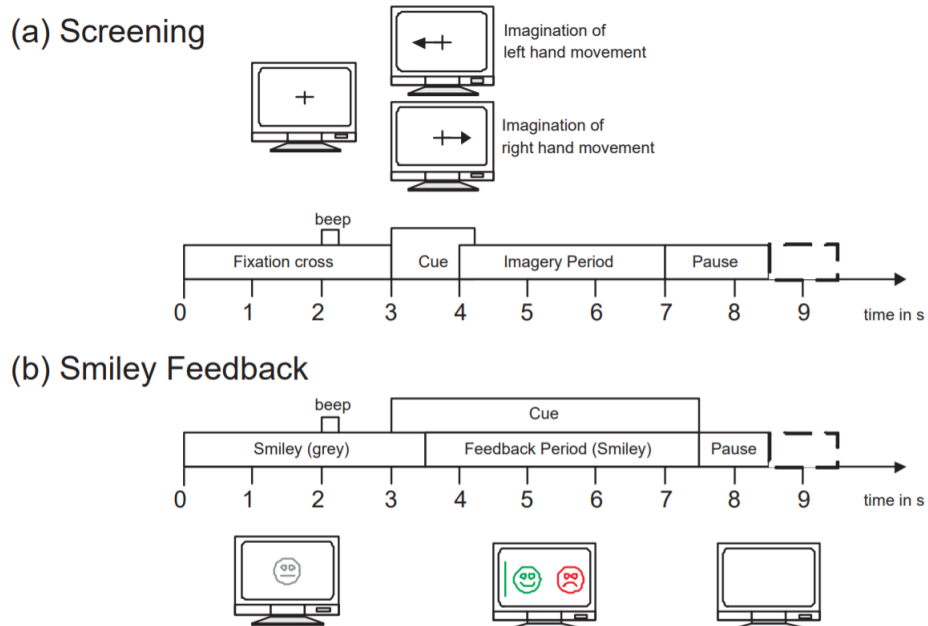
Os primeiros 5 minutos de cada sessão foram gravados para estimar o efeito do EOG no sinal EEG. Este período de gravação foi dividido em 3 blocos: (1) dois minutos com os olhos abertos (olhando num marcador fixo) (2) um minuto com os olhos fechados e (3) um minuto com os olhos em movimento. Este bloco de EOG não está disponível para as sessões B0102T e B0504E, de acordo com a fonte, devido a problemas técnicos.

No processo de gravação foram utilizados 3 canais bipolares (C3,Cz e C4) com uma frequência de amostragem de 250Hz. As gravações têm faixa dinâmica de  $\pm 100\mu V$  para as sessões sem feedback e  $\pm 50\mu V$  para as sessões com feedback. Elas foram filtrados com um passa-faixa de 0.5Hz a 100Hz e um filtro notch em 50Hz (para reduzir o ruído da rede elétrica). O posicionamento para cada um dos eletrodos variou ligeiramente em cada um dos sujeitos. Também foi gravado o EOG com três eletrodos monopolares usando a mesmas configurações de amplificação mas com uma faixa dinâmica maior de  $\pm 1mV$ .

As sessões podem divididas em dois tipos

- (a) **Sessões sem feedback:** Estas sessões consistem em 6 séries contendo 10 experimentos de cada uma das duas categorias, dando um total de 120 experimentos. Neste experimento, nos primeiros 3 segundos foi exibido uma cruz de fixação com um curto aviso sonoro. Depois disso uma seta apontando para direita ou esquerda (dependendo do tipo de ação motora imaginária) foi utilizado como deixa durando 1.25s . Posteriormente o sujeito teve um período de 4 segundos para imaginar o movimento da mão correspondente ao lado da seta. Para finalizar cada experimento

Figura 21 – Os dois tipos de sessão em Graz-b (LEEB C. BRUNNER, 2008).



foi seguido por uma parada aleatória de 1.5 a 2.5 segundos para evitar de que o sujeito se adapte.

- (b) **Sessões com feedback:** Estas sessões consistiram em 4 séries de 20 experimentos para cada tipo de imaginação motora dando um total de 160 experimentos. No início deste tipo de sessão foi centralizado um smiley cinza na tela e no segundo posterior foi acionado um aviso sonoro. Após mais um segundo a deíxa foi apresentada dependendo da deíxa os sujeitos foram pedidos para moverem o smiley para a direita ou esquerda dando um período de 4.5 segundos para imaginar esta ação. Caso o smiley fosse movido para o lado correto ele ficaria verde caso contrario vermelho. Depois disso, um periodo aleatório de 1-2 segundos foi adicionado a cada experimento.



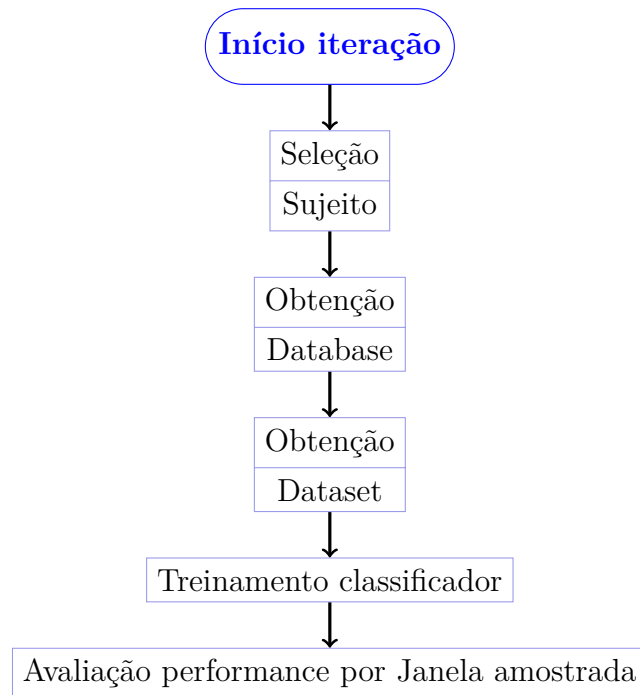
### 3.2 Framework

Para fazer a comparação entre as metodologias de extração de característica e classificadores foi desenvolvido um *framework* para automatizar o processo de treinamento e comparação. Este *framework* foi projetado com programação Programação Orientada a Objetos (*Object Oriented Programming*) (OOP) em mente devido à sua modularidade e fácil expansão.

#### 3.2.1 Compare Methods

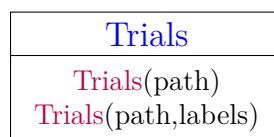
A função *CompareMethods* compara todas as configurações fornecidas, gera uma tabela com as configurações e retorna os classificadores treinados.

Figura 22 – diagrama compare methods



#### 3.2.2 Trials

Figura 23 – diagrama do Trials



**Trials** durante a criação pega o arquivo contendo a sessão de EEG filtra os canais de artefatos EOG nos canais EEG (C3,Cz,C4) usando a *regressão linear* de (SCHLÖGL C. KEINRATH, 2006), posteriormente ele extrai os experimentos e associa a categoria de

imaginação motora (1) esquerda e (2) direita ao experimento. Essa informação é obtida do arquivo da sessão ou é fornecida através do argumento *labels*.

### 3.2.3 Database

O **Database** é a classe responsável por fazer a amostragem, extração de características e de maneira geral preparar os dados para serem utilizados por um classificador. O processo de preparação dos dados para treinamento e validação é o seguinte (figura 24)

1. Durante criação do objeto do tipo **Database**, cada experimento armazenado em *Trials* é dividido em várias janelas de amostragem com duração *windowLength* e cada janela tem uma porcentagem de sobreposição com a janela anterior determinada por *overlap*.
2. As janelas de amostragem são armazenadas junto com as classificações obtidas do objeto *Trials*.
3. É gerado um vetor contendo os índices de amostras aleatórias contendo um número igual de amostras para cada categoria.
4. De cada janela de amostragem são extraído os vetores de características usando a função de extração de características configurada. É retornado uma matriz contendo o Dataset seguindo o padrão em que os vetores de características estão nas linhas e na última coluna consta a categoria da amostra.

$$\begin{bmatrix} F(C3_1) & F(Cz_1) & F(C4_1) & Cat_1 \\ F(C3_2) & F(Cz_2) & F(C4_2) & Cat_2 \\ \vdots & \vdots & \vdots & \vdots \\ F(C3_n) & F(Cz_n) & F(C4_n) & Cat_n \end{bmatrix} \quad (3.1)$$

Figura 24 – Amostragem e extração de características

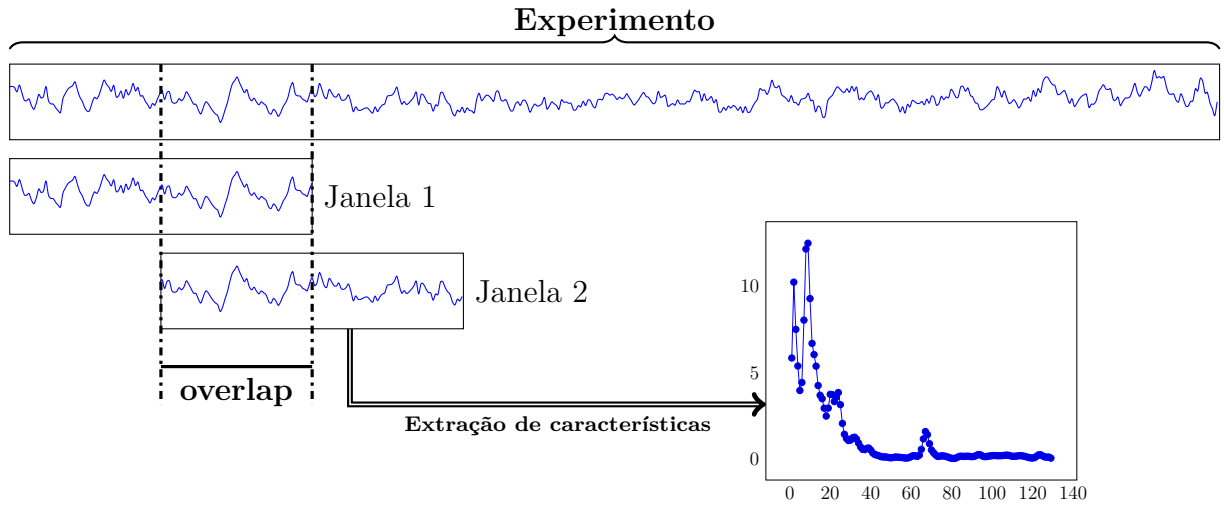


Figura 25 – diagrama do Database

Database
Database(overlap,windowLength,Trials) getSampleCountPerLabel() generateDatasetIndex(numSamples) generateDataset(TrainingDataset) getSample(Range) getSampleCount() setFeatureExtractionFcn(FEFunction)

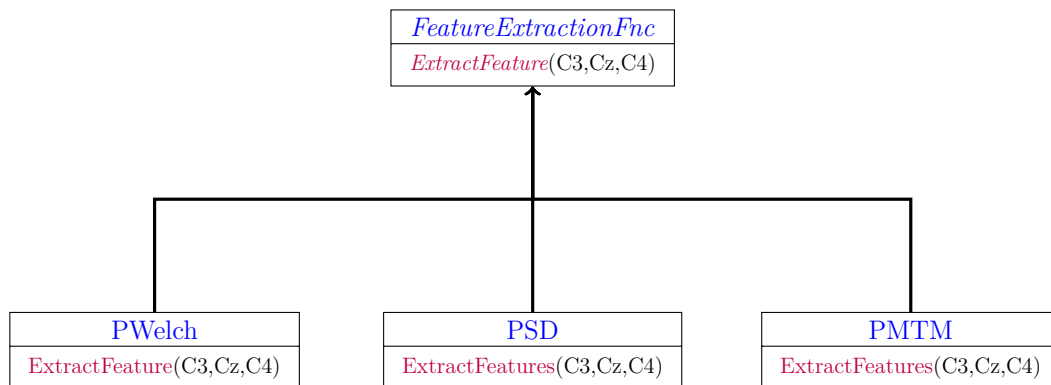
Tabela 1 – Métodos da classe Database

METODO	DESCRIÇÃO
Database	Cria um objeto da classe Database contendo as amostras do objeto da classe Trials usando os parâmetros <i>overlap</i> e <i>windowLength</i> para gerar as amostras.
getSampleCountPerLabel	Retorna a quantidade de janelas de amostragem de cada categoria
generateDatasetIndex	Gera o índice do dataset contendo o número de janelas de amostragem <i>numSamples</i>
generateDataset	Retorna o dataset contendo os vetores de características seguido pelas suas categorias
getSample	Retorna as janelas de amostragem com o seus índices determinado por <i>Range</i>
getSampleCount	Retorna o número total de janelas de amostragem
setFeatureExtractionFcn	Configura a função de extração de características

### 3.2.4 FeatureExtractionFnc

É a classe contendo a função de extração de características. A função *ExtractFeature*, virtual na classe base, foi implementada em três classes derivadas. São elas: o Espectro de Welch (PWelch), Multitaper Power Spectral density (PMTM) e PSDp. Para o PSDp e PWelch foram utilizadas janelas retangulares. Todas as configurações restantes são as padrões do **MATLAB**.

Figura 26 – diagrama de herança do FeatureExtractionFnc



## 3.2.5 Classifier

Figura 27 – diagrama de herança do Classifier

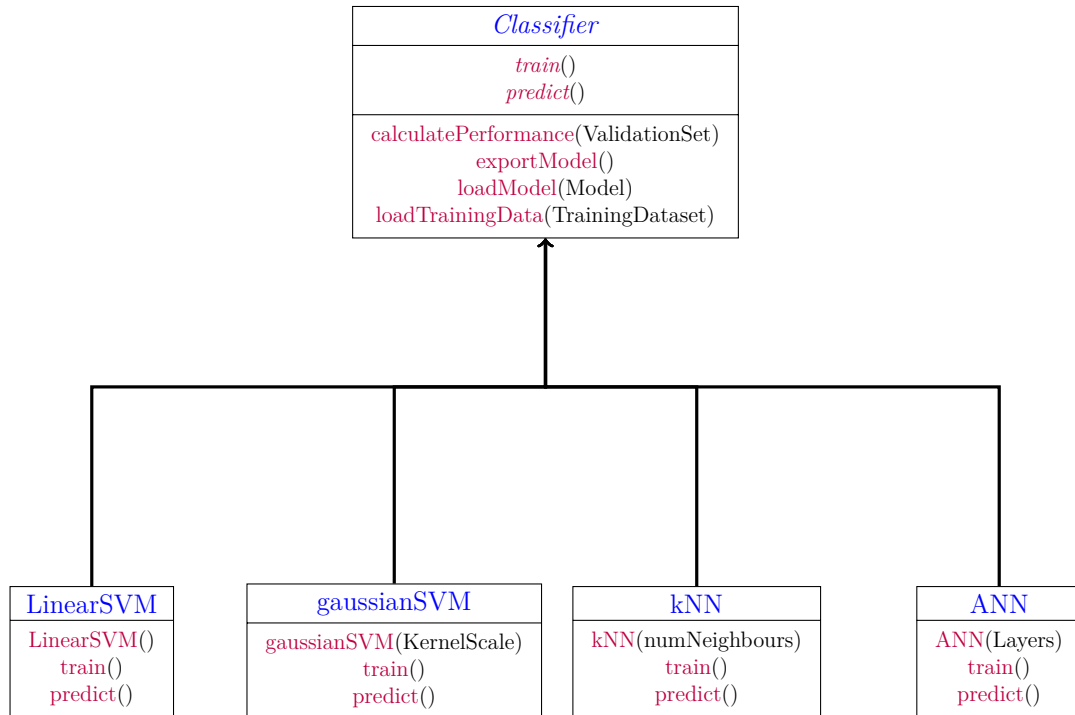


Tabela 2 – Métodos da classe Classifier

METODO	DESCRIÇÃO
<code>train</code>	Treina o classificador
<code>predict</code>	Determina a classificação dos dados de entrada
<code>calculatePerformance</code>	Aceita o parâmetro <code>ValidationSet</code> , contendo o conjunto de validação e retorna a performance do classificador em percentual de acertos e $\kappa$ de Cohen.
<code>exportModel</code>	Exporta o modelo do classificador
<code>loadTrainingData</code>	Carrega o conjunto de treinamento <code>TrainingDataset</code> .

**Classifier** é uma classe abstrata que descreve os classificadores e contém os métodos descritos na tabela 2. Os métodos `train` e `predict` são virtuais na classe base e realizam o treinamento e a classificação dos dados nas seguintes classes derivadas:

1. **LinearSVM** implementa um SVM com kernel linear.
2. **gaussianSVM** implementa um SVM com kernel gaussiano durante a inicialização é possível especificar o parâmetro  $\gamma$  do kernel usando o argumento `KernelScale`
3. **k-NN** é um classificador **kNN!** com métrica euclidiana que permite, no seu construtor, a especificação da quantidade de vizinhos próximos ( $k$ ).
4. **ANN** é uma rede neural utilizando-se da função de ativação do tipo *tanh*, Scaled Conjugate Gradient (SCG) como o método de treinamento e a entropia cruzada

como função de erro. O número de neurônios e a quantidade de camadas ocultas pode ser passada pelo argumento `Layers` durante a inicialização (por padrão, os objetos desta classe são configurados com duas camadas ocultas de 20 neurônios cada).

### 3.3 Métodos de avaliação de performance

Foram utilizadas 2 métricas de avaliação de performance. A primeira envolve classificar janelas de amostragem isoladas e calcular o  $\kappa$  de Cohen e a precisão de classificação. Esta métrica foi escolhida para poder analisar o impacto da redução da largura da janela de amostragem.

O segundo método consiste em obter todas as janelas de amostragem de um experimento, classificar elas e considerar a classificação do experimento como a moda das janelas de amostragem. Depois disso calculado o  $\kappa$  de Cohen e a precisão de classificação do experimento. Essa métrica foi usada para comparar a performance destes métodos com os utilizados na competição BCI-IV 2008 que utilizou o mesmo dataset.

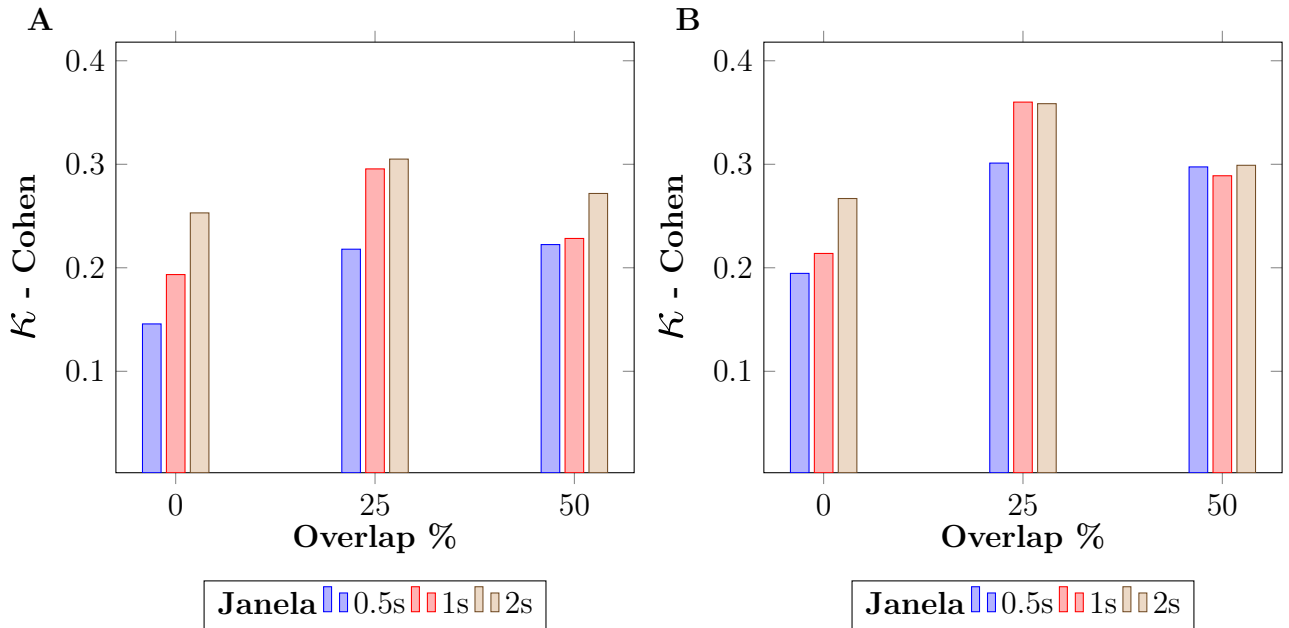
## 4 Resultados e Análises

### 4.1 Performances Médias

#### 4.1.1 Overlap vs Janela de Amostragem

A figura 28 sugere usar um valor de 25% de overlap, que produz o melhor resultado entre os valores verificados. Esta observação pode ser explicada com uma semelhança maior entre as observações se as janelas tiverem um overlap de 50%, o que tem impacto negativo no desempenho do classificador. Também podemos notar que a performance se beneficia de janelas de amostragem maior. Entretanto, esse efeito se reduz a partir de 25% de overlap e janelas de um segundo.

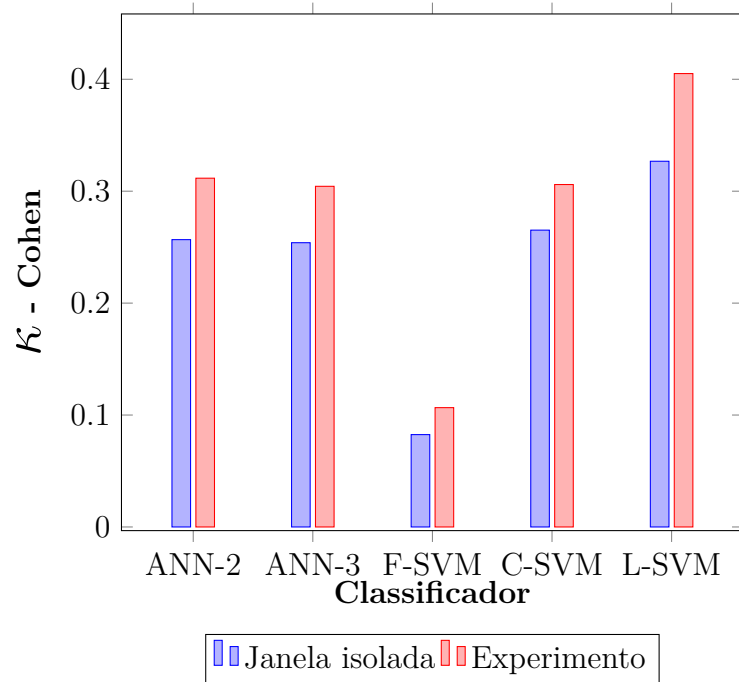
Figura 28 – (A) Kappa Overlap vs Window (B) Kappa Overlap vs Window para o experimento completo



#### 4.1.2 Classificadores

Na figura 29 podemos observar que por uma margem significativa o classificador com a melhor performance foi o SVM Linear (L-SVM). As redes neurais (ANN-2 e ANN-3) tiveram performances medianas junto com o SVM Gaussiano Grosso (C-SVM). A pior performance por uma margem significativa foi do SVM Gaussiano Fino (F-SVM) em que o parâmetro  $K$  do kernel teve um impacto negativo na generalização do classificador.

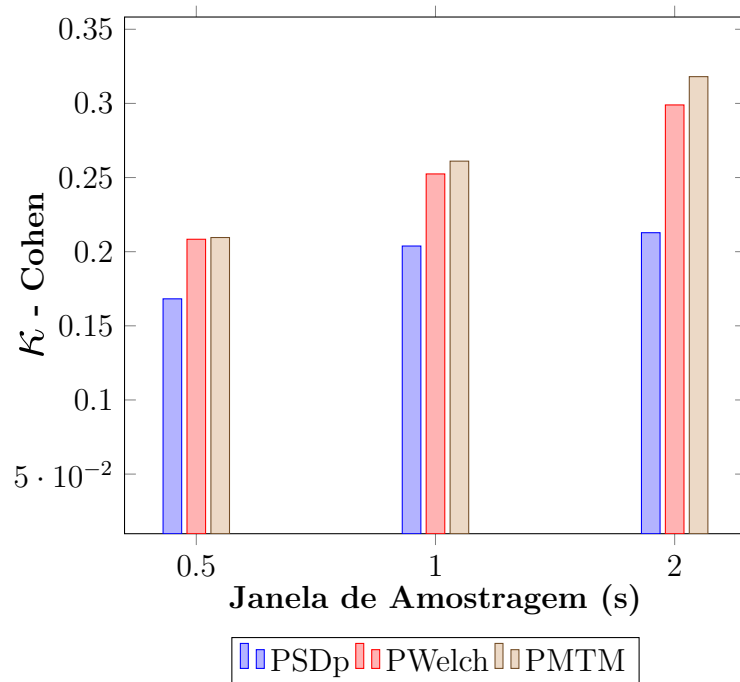
Figura 29 – Performance média dos classificadores utilizados



#### 4.1.3 Funções de extração vs Janela de Amostragem

A figura 30 sugere que com o aumento da janela de amostragem a performance melhora igualmente para os PWelch e PMTM sendo que o desempenho do PMTM foi ligeiramente maior. O PSDp teve claramente a pior performance isso é possivelmente devido à sua falta de capacidade de lidar com ruídos.

Figura 30 – Performance média das Funções de extração utilizadas vs Janela de amostragem





## 4.2 Melhores Performances

Pode-se observar nas tabelas abaixo a prevalência do L-SVM nas melhores posições entre os classificadores e do PMTM nas funções de extrações. Na performance utilizando o intervalo completo e na performance de janela de amostragem de até 2 segundos o melhor overlap foi de 25% enquanto para a performance de até 0.5 segundos o melhor overlap foi o de 50%.

Tabela 3 – Melhores performances na métrica do experimento

Pos	$\kappa_1$	ACC <sub>1</sub>	$\kappa_2$	ACC <sub>2</sub>	Janela	Overlap	Função	Classificador
1 <sup>o</sup>	0.422	71.14%	0.526	76.32%	2s	25%	PWelch	L-SVM
2 <sup>o</sup>	0.426	71.35%	0.512	75.62%	2s	25%	PMTM	L-SVM
3 <sup>o</sup>	0.421	71.06%	0.491	74.58%	2s	25%	PMTM	ANN-3
4 <sup>o</sup>	0.407	70.36%	0.490	74.51%	1s	25%	PMTM	L-SVM
5 <sup>o</sup>	0.402	70.11%	0.484	74.23%	1s	25%	PMTM	ANN-2

Tabela 4 – Melhores performances por Janela de amostragem de até 2s

Pos	$\kappa_1$	ACC <sub>1</sub>	$\kappa_2$	ACC <sub>2</sub>	Janela	Overlap	Função	Classificador
1 <sup>o</sup>	0.426	71.35%	0.512	75.62%	2s	25%	PMTM	L-SVM
2 <sup>o</sup>	0.422	71.14%	0.526	76.32%	2s	25%	PWelch	L-SVM
3 <sup>o</sup>	0.421	71.07%	0.452	72.61%	2s	0%	PMTM	L-SVM
4 <sup>o</sup>	0.421	71.06%	0.491	74.58%	2s	25%	PMTM	ANN-3
5 <sup>o</sup>	0.418	70.95%	0.466	73.33%	2s	50%	PMTM	L-SVM

Tabela 5 – Melhores performances por Janela de amostragem de até 0.5s

Pos	$\kappa_1$	ACC <sub>1</sub>	$\kappa_2$	ACC <sub>2</sub>	Janela	Overlap	Função	Classificador
1 <sup>o</sup>	0.293	64.69%	0.401	70.09%	0.5s	50%	PMTM	L-SVM
2 <sup>o</sup>	0.286	64.31%	0.388	69.45%	0.5s	50%	PWelch	L-SVM
3 <sup>o</sup>	0.284	64,21%	0.399	69.98%	0.5s	25%	PMTM	L-SVM
4 <sup>o</sup>	0.280	64,01%	0.397	69.88%	0.5s	25%	PWelch	L-SVM
5 <sup>o</sup>	0.275	63.78%	0.391	69.57%	0.5s	25%	PMTM	ANN-2

## 5 Conclusão

Em conclusão, o melhor classificador foi o SVM com um kernel linear. O melhor overlap é de 25% e, dependendo da aplicação, qualquer tamanho de janela pode ser utilizado com um pequeno ganho de performance com uma largura de 2s.

Por fim, a melhor configuração utilizada teve uma performance próxima aos classificadores utilizados durante a competição, apesar de ser claramente inferior à primeira posição.

Tabela 6 – Resultados da competição BCI-IV 2008

Pos	contributor	$\kappa$	1	2	3	4	5	6	7	8	9
1 <sup>o</sup>	Z. Y. Chin	<b>0.60</b>	0.40	0.21	0.22	0.95	0.86	0.61	0.56	0.85	0.74
2 <sup>o</sup>	H. Gan	<b>0.58</b>	0.42	0.21	0.14	0.94	0.71	0.62	0.61	0.84	0.78
3 <sup>o</sup>	D. Coyle	<b>0.46</b>	0.19	0.12	0.12	0.77	0.57	0.49	0.38	0.85	0.61
4 <sup>o</sup>	S. Lodder	<b>0.43</b>	0.23	0.31	0.07	0.91	0.24	0.42	0.41	0.74	0.53
5 <sup>o</sup>	J. F. D. Saa	<b>0.37</b>	0.20	0.16	0.16	0.73	0.21	0.19	0.39	0.86	0.44
6 <sup>o</sup>	Y. Ping	<b>0.25</b>	0.02	0.09	0.07	0.43	0.25	0.00	0.14	0.76	0.47
-	L-SVM PMTM	<b>0.51</b>	0.41	0	0.09	0.92	0.58	0.63	0.4	0.83	0.71

## REFERÊNCIAS

- BCI2000. *Introduction to the Mu Rhythm*. 2018. Disponível em: <[https://www.bci2000.org/mediawiki/index.php/User\\_Tutorial:Introduction\\_to\\_the\\_Mu\\_Rhythm](https://www.bci2000.org/mediawiki/index.php/User_Tutorial:Introduction_to_the_Mu_Rhythm)>. Acesso em: 29 jul. 2018.
- BLANKERTZ, B. et al. The berlin brain-computer interface: Progress beyond communication and control. *Frontiers in Neuroscience*, v. 10, p. 530, 2016. ISSN 1662-453X. Disponível em: <<https://www.frontiersin.org/article/10.3389/fnins.2016.00530>>.
- DRONGELEN, W. van. *Multitaper Power Spectrum Estimation*. [s.n.], 2014. Disponível em: <<https://pdfs.semanticscholar.org/752d/1a551b96559458064323eb3de7faaaef4c4e.pdf>>.
- JR., O. M. S. *PSD Computations Using Welch's Method*. 1991.
- KOWALCZYK, A. *Support Vector Machines: Succinctly*. Morrisville, NC: Syncfusion, 2017.
- LEEB C. BRUNNER, G. R. M.-P. A. S. G. P. R. *BCI Competition 2008 Graz data set B*. 2008. Disponível em: <[http://www.bbci.de/competition/iv/desc\\_2b.pdf](http://www.bbci.de/competition/iv/desc_2b.pdf)>.
- MØLLER, M. F. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, v. 6, p. 525–533, Dec 1993.
- Pawel Herman. Comparative analysis of spectral approaches to feature extraction for eeg-based motor imagery classification. *IEEE Transactions on neural systems and rehabilitation*, v. 16, n. 4, p. 317–326, August 2008.
- QIN LEI DING, B. H. L. Motor imagery classification by means of source analysis for braincomputer interface application. *JOURNAL OF NEURAL ENGINEERING*, v. 1, p. 135–141, 2004.
- RAO, R. P. N. *Brain-Computer Interfacing: An Introduction*. 1. ed. New York: Cambridge University Press, 2013.
- RAO, R. S. R. P. N. *Brain-computer interfacing*. 2010.
- Robert Plonsey, Jaako Malmivuo. *Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields*. New York: Johns Hopkins University Press, 1995.
- SCHLÖGL C. KEINRATH, D. Z. R. S. R. L. G. A. A fully automated correction method of eeg artifacts in eeg recordings. *Clinical Neurophysiology*, v. 118, p. 98–104, November 2006.
- Simon Haykin. *Redes Neurais: Princípios e prática*. 2. ed. Porto Alegre: Bookman, 2008.
- VAPNIK, C. C. V. Support-vector networks. *Machine Learning*, ATT Bell Labs., Holmdel, NJ, USA, v. 20, p. 273–297, 1995.
- VIDAL, J. J. Real-time detection of brain events in eeg. *Proceedings of the IEEE*, v. 65, n. 5, p. 633–641, May 1977.