

**UNIVERSIDADE FEDERAL DO PAMPA**

**LUCAS VISINTAINER SEVERO**

**ANÁLISE COMPARATIVA DE  
CLASSIFICADORES DE PADRÕES  
PARA CLASSIFICAÇÃO DE TAREFAS  
MENTAIS**

**Alegrete  
2020**

**LUCAS VISINTAINER SEVERO**

**ANÁLISE COMPARATIVA DE  
CLASSIFICADORES DE PADRÕES  
PARA CLASSIFICAÇÃO DE TAREFAS  
MENTAIS**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Engenharia Elétrica como requisito parcial para a obtenção do grau de Bacharel em Engenharia Elétrica.

Orientador: Prof. Dr. Alessandro Botti Benevides

**Alegrete  
2020**



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
Universidade Federal do Pampa

**LUCAS VISINTAINER SEVERO**

**ANÁLISE COMPARATIVA DE CLASSIFICADORES DE PADRÕES PARA CLASSIFICAÇÃO DE TAREFAS  
MENTAIS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica da Universidade Federal do Pampa, como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica.

Trabalho de Conclusão de Curso defendido e aprovado em: 03 de Dezembro de 2020.

Banca examinadora:

---

Prof. Dr. Alessandro Botti Benevides

Orientador

UNIPAMPA

---

Prof. Me. Eliezer Soares Flores

UNIPAMPA

---

Prof. Dr. Giovani Guarienti Pozzebon

UNIPAMPA

---



Documento assinado eletronicamente por **GIOVANI GUARIENTI POZZEBON, PROFESSOR DO MAGISTERIO SUPERIOR**, em 03/12/2020, às 11:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **ELIEZER SOARES FLORES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 03/12/2020, às 11:59, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **ALESSANDRO BOTTI BENEVIDES, PROFESSOR DO MAGISTERIO SUPERIOR**, em 03/12/2020, às 12:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://sei.unipampa.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0417454** e o código CRC **3E78600F**.

---

Universidade Federal do Pampa, Campus Alegrete  
Av. Tiarajú, 810 – Bairro: Ibirapuitã – Alegrete – RS CEP: 97.546-550

Telefone: (55) 3422-8400

## **AGRADECIMENTO**

A Deus pela saúde, sabedoria e força nos momentos difíceis

Ao Prof. Dr. Alessandro Botti Benevides por compartilhar seu conhecimento e pela orientação no trabalho

À minha esposa pelo companheirismo, paciência e amor

À minha família pelo apoio

Aos colegas que partilharam dessa jornada

“É fazendo que se aprende a fazer aquilo que se deve aprender a fazer.”

— Aristóteles

## RESUMO

O eletroencefalograma é uma técnica que utiliza eletrodos posicionados na superfície da cabeça para se obter os sinais biológicos presentes no cérebro de forma gráfica. Com esses gráficos, é possível diagnosticar e monitorar doenças que atingem o sistema nervoso. Ainda, o eletroencefalograma permite monitorar o que ocorre com os sinais cerebrais nos momentos de execução de tarefas mentais, como imaginar o movimento dos membros superiores. Essas tarefas mentais podem ser utilizadas para aplicações como interfaces cérebro computador, para isso os sinais devem ser adquiridos em tempo real através do eletroencefalograma e classificados por classificadores de padrões. Nesse intuito foram utilizados sinais de eletroencefalograma obtidos através de experimentos para testar o desempenho dos classificadores k-vizinhos mais próximos e redes neurais artificiais para classificar as atividades presentes nestes sinais. E com isso, comparar os resultados obtidos através dos classificadores para definir qual foi o classificador com melhor desempenho nessa atividade.

**Palavras-chave:** Eletroencefalograma. K-vizinhos mais próximos. Redes neurais artificiais. Classificadores de padrões.

## ABSTRACT

The electroencephalogram is a technique that uses electrodes positioned on the surface of the head to obtain the biological signals present in the brain in a graphic way. With these charts, it is possible to diagnose and monitor diseases that affect the nervous system. In addition, the electroencephalogram allows monitoring what happens to brain signals when performing mental tasks, such as imagining the movement of the upper limbs. These mental tasks can be used for applications such as brain computer interfaces, for which the signals must be acquired in real time through the electroencephalogram and classified by pattern classifiers. For this purpose, electroencephalogram signals obtained through experiments were used to test the performance of the k-nearest neighbor classifiers and artificial neural networks to classify the activities present in these signals. And with that, compare the results obtained through the classifiers to define which classifier had the best performance in this activity.

**Keywords:** Electroencephalogram. K-nearest neighbors. Artificial neural networks. Classifiers of patterns.



## LISTA DE FIGURAS

Figura 1	Neurônio humano (BEAR et al., 2008).	19
Figura 2	Potencial de ação, PEPS e PIPS (adaptado de BEAR et al.,2008).	20
Figura 3	Camadas entre o eletrodo e o cérebro (BEAR et al., 2008).	21
Figura 4	Posicionamento dos eletrodos (Elaborada pelo autor).	23
Figura 5	Artefato instrumental de 50 Hz (Elaborada pelo autor).	24
Figura 6	Artefato fisiológicos (BEAR et al., 2008).	24
Figura 7	Classificação do k-nn (Elaborado pelo autor).	27
Figura 8	Neurônio artificial (Elaborado pelo autor).	29
Figura 9	Função degrau unitário (Elaborado pelo autor).	30
Figura 10	Função linear (Elaborado pelo autor).	30
Figura 11	Função rampa simétrica (Elaborado pelo autor).	30
Figura 12	Função logística (Elaborado pelo autor).	31
Figura 13	Função tangente hiperbólica (Elaborado pelo autor).	31
Figura 14	Arquitetura perceptron multicamadas (Elaborado pelo autor).	32
Figura 15	Representação da validação cruzada (Elaborado pelo autor).	34
Figura 16	Representação do experimento realizado para a obtenção da base de dados BCI <i>competition</i> II (BLANKERTZ et al., 2004).	37
Figura 17	Representação dos sinais no domínio do tempo (Elaborado pelo autor).	39
Figura 18	Representação dos sinais no domínio da frequência (Elaborado pelo autor).	39
Figura 19	Representação dos sinais no domínio da frequência antes e durante a imaginação das atividades motoras (Elaborado pelo autor).	40
Figura 20	Representação do sinal do eletrodo C3 (Elaborado pelo autor).	42
Figura 21	Representação da RNA para o banco de dados de Fisher (Elaborado pelo autor).	45
Figura 22	Representação da RNA para o banco de dados BCI <i>competition</i> II (Elaborado pelo autor).	47
Figura 23	Representação da RNA para o banco de dados BCI <i>competition</i> III (Elaborado pelo autor).	50
Figura 24	Resultado da análise do classificador k-nn com banco de dados de Fisher (Elaborado pelo autor).	51
Figura 25	Resultado da análise da rede neural artificial com algoritmo de treinamento Levenberg-Marquardt com banco de dados de Fisher (Elaborado pelo autor).	53
Figura 26	Resultado da análise da rede neural artificial com algoritmo de treinamento regularização bayesiana com banco de dados de Fisher (Elaborado pelo autor).	54
Figura 27	Resultado da classificação com o classificador k-nn para o banco de dados BCI <i>competition</i> II (Elaborado pelo autor).	56
Figura 28	Resultados da classificação do banco de dados BCI <i>competition</i> II com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor).	58
Figura 29	Resultados da classificação do banco de dados BCI <i>competition</i> II com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).	59
Figura 30	Resultado da classificação com o classificador k-nn para o banco de dados BCI <i>competition</i> III (Elaborado pelo autor).	61

Figura 31 Resultados da classificação do banco de dados BCI <i>competition</i> III com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor). .....	62
Figura 32 Resultados da classificação do banco de dados BCI <i>competition</i> III com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor). .....	62

## LISTA DE TABELAS

Tabela 1 Representação das saídas da RNA para o banco de dados de Fisher (Elaborado pelo autor). .....	45
Tabela 2 Representação das saídas da RNA para o banco de dados BCI <i>competition</i> II (Elaborado pelo autor). .....	47
Tabela 3 Representação das saídas da RNA para o banco de dados BCI <i>competition</i> III (Elaborado pelo autor). .....	49
Tabela 4 Resultado da análise do classificador k-nn com banco de dados de Fisher (Elaborado pelo autor). .....	52
Tabela 5 Resultado do treino para a análise da rede neural artificial com algoritmo de treinamento Levenberg-Marquardt com banco de dados de Fisher (Elaborado pelo autor). .....	53
Tabela 6 Resultado da classificação para a análise da rede neural artificial com algoritmo de treinamento Levenberg-Marquardt com banco de dados de Fisher (Elaborado pelo autor). .....	54
Tabela 7 Resultado do treino para a análise da rede neural artificial com algoritmo de treinamento regularização bayesiana com banco de dados de Fisher (Elaborado pelo autor). .....	55
Tabela 8 Resultado da classificação para a análise da rede neural artificial com algoritmo de treinamento regularização bayesiana com banco de dados de Fisher (Elaborado pelo autor). .....	55
Tabela 9 Resultado da classificação com o classificador k-nn para o banco de dados BCI <i>competition</i> II (Elaborado pelo autor). .....	57
Tabela 10 Resultados do treino para classificação do banco de dados BCI <i>competition</i> II com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor). .....	58
Tabela 11 Resultados da classificação do do banco de dados BCI <i>competition</i> II com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor). .....	59
Tabela 12 Resultados do treino para classificação do banco de dados BCI <i>competition</i> II com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor). .....	60
Tabela 13 Resultados da classificação do do banco de dados BCI <i>competition</i> II com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor). .....	60
Tabela 14 Resultado da classificação com o classificador k-nn para o banco de dados BCI <i>competition</i> III (Elaborado pelo autor). .....	63
Tabela 15 Resultados do treino para classificação do banco de dados BCI <i>competition</i> III com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor). .....	64
Tabela 16 Resultados da classificação do do banco de dados BCI <i>competition</i> III com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor). .....	64
Tabela 17 Resultados do treino para classificação do banco de dados BCI <i>competition</i> III com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor). .....	65

Tabela 18 Resultados da classificação do do banco de dados BCI <i>competition</i> III com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).....	65
Tabela 19 Comparação dos resultados da classificação do banco de dados BCI <i>competition</i> III (adaptado de BENEVIDES et al., 2012). .....	66

## LISTA DE ABREVIATURAS E SIGLAS

BCI	<i>Brain-computer interface</i> (Interface cérebro-computador)
CAR	<i>Common Average Reference</i> (Referência Média Comum)
ECG	Eletrocardiograma
EEG	Eletroencefalograma
EMG	Eletromiograma
EOG	Eletrooculograma
FFT	<i>Fast Fourier transform</i> (Transformada rápida de Fourier)
ICC	Interface cérebro-computador
K-NN	<i>K-nearest neighbors</i> (K-vizinhos mais próximos)
LM	Levenberg-Marquardt
PEPS	Potencial excitatório pós-sináptico
PIPS	Potencial inibitório pós-sináptico
PMC	Perceptron multicamadas
RB	Regularização bayesiana
RNA	Redes Neurais Artificiais

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
<b>1.1 Objetivos gerais</b> .....	<b>15</b>
<b>1.2 Objetivos específicos</b> .....	<b>16</b>
<b>1.3 Divisão do atual trabalho</b> .....	<b>16</b>
<b>2 REVISÃO TEÓRICA</b> .....	<b>18</b>
<b>2.1 Sinais biomédicos</b> .....	<b>18</b>
<b>2.2 Eletroencefalograma</b> .....	<b>20</b>
2.2.1 Aquisição do sinal.....	22
2.2.2 Artefatos.....	23
2.2.3 Filtragem do sinal .....	25
<b>2.3 Transformada de Fourier</b> .....	<b>25</b>
<b>2.4 Classificadores</b> .....	<b>26</b>
2.4.1 K-vizinhos mais próximos .....	26
2.4.2 Redes Neurais Artificiais .....	28
2.4.3 Funções de Ativação .....	29
2.4.4 Redes Perceptron Multicamadas.....	31
2.4.5 Validação cruzada .....	33
<b>3 METODOLOGIA</b> .....	<b>35</b>
<b>3.1 Banco de dados</b> .....	<b>35</b>
3.1.1 Banco de dados de Fisher .....	35
3.1.2 Banco de dados BCI <i>competition</i> II.....	36
3.1.3 Banco de dados BCI <i>competition</i> III.....	37
<b>3.2 Pré-processamento dos dados de EEG</b> .....	<b>38</b>
3.2.1 Pré-processamento do Banco de dados BCI <i>competition</i> II.....	38
3.2.2 Pré-processamento do Banco de dados BCI <i>competition</i> III .....	41
<b>3.3 Análise dos classificadores</b> .....	<b>42</b>
3.3.1 Análise do k-vizinhos mais próximos.....	43
3.3.2 Análise da rede neural artificial .....	43
<b>3.4 Classificação dos dados BCI <i>competition</i> II</b> .....	<b>45</b>
3.4.1 Classificação dos dados BCI <i>competition</i> II com k-vizinhos mais próximos.....	45
3.4.2 Classificação do BCI <i>competition</i> II com redes neurais artificiais .....	46
<b>3.5 Classificação dos dados BCI <i>competition</i> III</b> .....	<b>48</b>
3.5.1 Classificação do BCI <i>competition</i> III com k-vizinhos mais próximos .....	48
3.5.2 Classificação do BCI <i>competition</i> III com redes neurais artificiais .....	49
<b>4 RESULTADOS</b> .....	<b>51</b>
<b>4.1 Resultado da análise dos classificadores com banco de dados Iris de Fisher</b> ....	<b>51</b>
<b>4.2 Resultado da classificação do banco de dados BCI <i>competition</i> II</b> .....	<b>54</b>
<b>4.3 Resultado da classificação do banco de dados BCI <i>competition</i> III</b> .....	<b>57</b>
<b>5 CONSIDERAÇÕES FINAIS</b> .....	<b>67</b>
<b>REFERÊNCIAS</b> .....	<b>69</b>
<b>ANEXO A — CÓDIGO PARA SE OBTER O SINAL EEG DO BANCO DE DADOS BCI II NO DOMÍNIO DO TEMPO</b> .....	<b>72</b>
<b>ANEXO B — CÓDIGO PARA SE OBTER O SINAL EEG DO BANCO DE DADOS BCI II NO DOMÍNIO DA FREQUÊNCIA</b> .....	<b>73</b>
<b>ANEXO C — CÓDIGO PARA SALVAR OS DADOS DO EEG DO BANCO DE DADOS BCI II</b> .....	<b>74</b>
<b>ANEXO D — CÓDIGO REPRESENTA O GRÁFICO DA IMAGINAÇÃO DE ATIVIDADE CEREBRAIS DO EEG DO BANCO DE DADOS BCI II</b> .....	<b>75</b>

ANEXO E — CÓDIGO QUE SALVA OS DADOS DO BCI III .....	77
ANEXO F — CÓDIGO QUE GERA O SINAL DO ELETRODO C3 DO BANCO DE DADOS BCI III .....	79
ANEXO G — CÓDIGO DO K-NN PARA CLASSIFICAÇÃO DO BANCO DE DADOS DE FISHER .....	80
ANEXO H — CÓDIGO DO LEVENBERG-MARQUARDT PARA CLASSI- FICAÇÃO DO BANCO DE DADOS DE FISHER .....	82
ANEXO I — CÓDIGO DO DA REGULARIZAÇÃO BAYESIANA PARA CLASSIFICAÇÃO DO BANCO DE DADOS DE FISHER.....	84
ANEXO J — CÓDIGO DO K-NN PARA CLASSIFICAÇÃO DO BCI <i>COM- PETITION</i> II .....	86
ANEXO K — CÓDIGO DE LEVENBERG-MARQUARDT PARA CLASSI- FICAÇÃO DO BCI <i>COMPETITION</i> II.....	88
ANEXO L — CÓDIGO DA REGULARIZAÇÃO BAYESIANA PARA CLAS- SIFICAÇÃO DO BCI <i>COMPETITION</i> II .....	90
ANEXO M — CÓDIGO DO K-NN PARA CLASSIFICAÇÃO DO BCI <i>COM- PETITION</i> III.....	92
ANEXO N — CÓDIGO DE LEVENBERG-MARQUARDT PARA CLASSI- FICAÇÃO DO BCI <i>COMPETITION</i> III .....	94
ANEXO O — CÓDIGO DA REGULARIZAÇÃO BAYESIANA PARA CLAS- SIFICAÇÃO DO BCI <i>COMPETITION</i> III.....	96

## 1 INTRODUÇÃO

O corpo humano é uma grande fonte de sinais elétricos gerados por células nervosas e musculares. Esses sinais podem ser chamados de sinais biológicos ou biomédicos. Para que haja um melhor entendimento sobre os sinais biomédicos, é necessário extrair e processar esses sinais. A meta do processamento de sinais biomédicos é extrair informações de um sinal biológico que nos ajude a melhorar ainda mais nossa compreensão dos mecanismos básicos da função biológica e nos auxilie no diagnóstico ou tratamento de uma condição médica (HAYKIN; VEEN, 2001).

Devido à presença de uma grande massa de células nervosas, o cérebro humano é uma grande fonte de sinais biomédicos. Uma das formas de se obter o sinal cerebral é através do eletroencefalograma (EEG). O EEG é obtido a partir de eletrodos posicionados na parte superior da cabeça. Esse método é o mais amplamente utilizado em laboratórios clínicos e psicológicos para monitorar não invasivamente a atividade do cérebro (GOMES, 2015). As informações presentes no EEG estão contidas em um intervalo de frequências que se estende até aproximadamente 80 Hz. Este intervalo de frequências é subdividido em intervalos menores denominados ritmos os quais por sua vez estão relacionados a diferentes atividades no cérebro como, por exemplo, a ação de imaginar o movimento de um dos membros superiores. No momento desta imaginação o EEG vai capturar uma alteração na frequência do ritmo relacionado às atividades motoras.

Outra característica do EEG é possuir uma amplitude baixa, devido a isso seu sinal está sujeito a diferentes tipos de ruídos ou artefatos. Esses artefatos são divididos em dois tipos, os artefatos técnicos os artefatos biológicos. Os artefatos técnicos podem ser oriundos do mau uso de equipamentos e da interferência da rede elétrica gerada nos aparelhos ligados em tomadas, e os de origem biológica são os artefatos produzidos pelo corpo humano como o piscar dos olhos. Os artefatos podem ser removidos com a utilização de filtros devidamente ajustados, sem que ocorra perda significativa na informação presente no sinal de EEG (BENEVIDES, 2009).

Além de extrair informações sobre o funcionamento cerebral, o EEG vem sendo utilizado para controlar máquinas com a imaginação através de interfaces cérebro-máquina. As interfaces cérebro-máquina podem ser soletradores, cadeiras de rodas e interfaces cérebro-computador (ICCs). Já em 2004 J. Millán propôs a partir de testes envolvendo um robô comandado por sinais cerebrais com funcionamento similar a uma cadeira de rodas, a partir dos testes foi concluído-se que é possível mover uma cadeira de



rodas com sinais cerebrais. Para que isso seja possível o sinal deve ser processado e enviado em tempo real para a máquina. Nessa etapa são utilizados classificadores de padrões para separar os sinais de informação enviados para a máquina dos sinais pertencentes a outras atividades, como a atividades espontâneas do cérebro. Os métodos de classificação que foram utilizados neste trabalho são o k-vizinhos mais próximos (k-nn, do inglês *k-nierst neighbor*) e as redes neurais artificiais (RNAs).

O método k-nn faz a classificação dos dados baseada na distância entre amostras de teste e treinamento. As redes neurais são máquinas de aprendizagem que imitam o funcionamento do cérebro humano. Elas possuem diferentes tipos de arquiteturas, a arquitetura utilizada neste trabalho foi a de perceptron multicamadas. Essa arquitetura tem como característica possuir no mínimo uma camada escondida. O objetivo primário das Redes Neurais de Múltiplas Camadas, assim como das demais máquinas de aprendizagem, é, sem dúvida, o correto mapeamento do sinal entrada-saída apresentado durante a etapa de treinamento, de modo que os critérios previamente definidos sejam atendidos (TAVARES, 2015).

Como o sinal de EEG apresenta muito ruído, e a extração de características é problemática, testou-se primeiramente os 2 classificadores em problemas mais simples: primeiramente o problema de classificação de três espécies de flor dadas pelo banco de dados de Fisher; depois o problema de classificação de duas tarefas mentais utilizando o sinal EEG (obtido com realimentação do usuário) fornecido pelo banco de dados da BCI *competition II*; e por fim o problema de classificação de três tarefas mentais utilizando o sinal EEG (obtido sem realimentação do usuário) fornecido pelo banco de dados da BCI *competition III*. Os resultados obtidos através da classificação dos bancos de dados foram comparados para ver qual dos classificadores teve o melhor desempenho.

## 1.1 Objetivos gerais

Este trabalho tem como objetivo geral a classificação de tarefas mentais presentes no eletroencefalograma. Para isso foram utilizados dois tipos de classificadores de padrões, são eles: o classificador k-vizinhos mais próximos e a rede neural artificial com arquitetura perceptron multicamadas (PMC).

## 1.2 Objetivos específicos

O presente trabalho visa o estudo do sinal eletroencefalográfico (EEG) trazendo um apanhado teórico sobre a aquisição do sinal através de eletrodos posicionados na parte superior da cabeça, suas características de baixa amplitude devido à resistência exercida sobre o eletrodo das camadas que envolvem o cérebro e a presença de informação no espectro de frequência do sinal.

O apanhado teórico traz o conhecimento necessário para efetuar os objetivos que esse trabalho visa apresentar. Foram utilizados dois bancos de dados contendo sinais de EEG com tarefas mentais obtidos através de experimentos. Para se obter as tarefas presentes no EEG é necessário fazer a representação desse sinal no domínio da frequência, para isso foi utilizada a transformada discreta de Fourier para esse fim.

Com o sinal representado no domínio da frequência, foi feito o pré-processamento utilizando um método de filtragem denominado referência média comum com o objetivo de eliminar as interferências presentes no sinal. Além de eliminar as interferências o pré-processamento visa realçar a presença de informações referentes as tarefas mentais nas faixas de frequência do EEG.

Por fim utilizar o sinal pré-processado para classificar as tarefas mentais utilizando os classificadores k-vizinhos mais próximos e a rede neural artificial com arquitetura perceptron multicamadas, assim comparando os resultados obtidos para observar qual método apresenta o melhor resultado.

## 1.3 Divisão do atual trabalho

Este trabalho está dividido em cinco capítulos, tendo a introdução no primeiro capítulo. O segundo capítulo traz a revisão teórica necessário para o entendimento da origem dos sinais biológicos, da aquisição, características e como se obter as informações do eletroencefalograma (EEG). As informações presentes no EEG estão contidas em faixas de frequência. Assim, a seção 2.3, trata da transformada de Fourier, que é o método matemático para representa o EEG no domínio da frequência. Por fim, na seção 2.4 Ã© feito um apanhado teórico dos classificadores utilizados para classificar o EEG.

O terceiro capítulo possui a metodologia utilizada neste trabalho. Esse capítulo contém as características dos bancos de dados utilizados e o pré-processamento dos mesmos. Ainda, é demonstrado como foi feita a análise dos classificadores e a classificação

dos sinais referentes aos bancos de dados *BCI competition II* e *BCI competition III*.

No capítulo quatro são apresentados os resultados obtidos a partir da análise dos classificadores e da classificação dos sinais referentes aos bancos de dados *BCI competition II* e *BCI competition III*. Os resultados são apresentados em tabelas e gráficos.

Por fim, no capítulo cinco são apresentadas as considerações finais, referentes a comparação dos resultados obtidos.

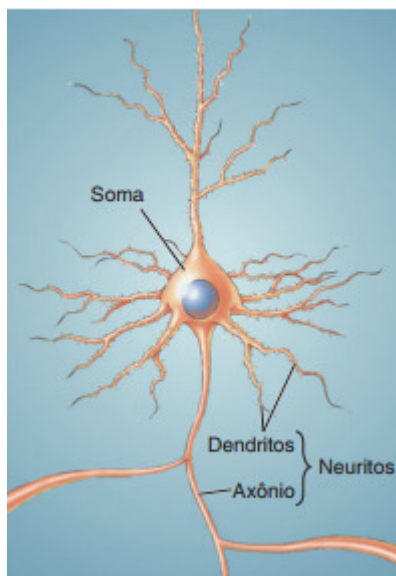
## 2 REVISÃO TEÓRICA

Neste capítulo são abordados primeiramente os sinais biomédicos com foco no sinal de estudo, o eletroencefalograma (EEG). É abordado na seção 2.1 a origem deste sinal com o intuito de esclarecer suas limitações, propensão a contaminação por variados tipos de ruídos, e assim, esclarecer as decisões necessárias na etapa de pré-processamento digital do sinal. Informações relacionadas ao procedimento de aquisição e posicionamento dos eletrodos são abordadas na seção 2.2. Este conhecimento implicará na decisão da escolha dos eletrodos e frequências, etapa que corresponde a extração das características que são passadas ao classificador. A etapa de extração de características corresponde à correta aplicação da transformada de Fourier no sinal EEG, em que somente algumas bandas de frequência são tratadas e repassadas para o classificador. Assim, a transformada de Fourier é brevemente introduzida na seção 2.3 e a seção 2.4 abordará os diferentes tipos de classificadores utilizados neste trabalho, são eles: o *k*-vizinhos mais próximos (*k*-nn, do inglês *k-nearest neighbour*), a rede neural artificial perceptron multicamadas.

### 2.1 Sinais biomédicos

Os sinais biológicos presentes no corpo humano são gerados por um grande grupo de células nervosas ou musculares. Os efeitos acumulados de todas as células ativadas em uma vizinhança produzem um campo elétrico que se propaga no volume do condutor constituído pelos vários tecidos do corpo (BORN, 2000). Dentre os sinais biológicos os que possuem maior importância são o Eletrocardiograma (ECG - atividade elétrica do coração), Eletromiograma (EMG - atividade elétrica muscular), Eletro-oculograma (EOG - relacionado ao movimento dos olhos) e o Eletroencefalograma (EEG - atividade elétrica do cérebro). Dos sinais citados anteriormente, o foco principal do trabalho se dá em torno do Eletroencefalograma, os demais são tratados como interferências no EEG. Para dar início ao estudo do EEG é importante saber a origem dos sinais cerebrais. Como citado anteriormente os sinais biológicos são gerados por um grande grupo de células nervosas. O cérebro é formado por bilhões de células nervosas, cada qual recebendo sinais elétricos de outros neurônios e os retransmitindo a outras células nervosas (JARDIM, 2017). A estrutura de um neurônio humano está descrita na Figura 1. A transmissão de informações entre neurônios e outras células nervosas é feita por um sinal denominado potencial de ação. O potencial de ação é gerado através das sinapses neuronais. A sinapse é a região

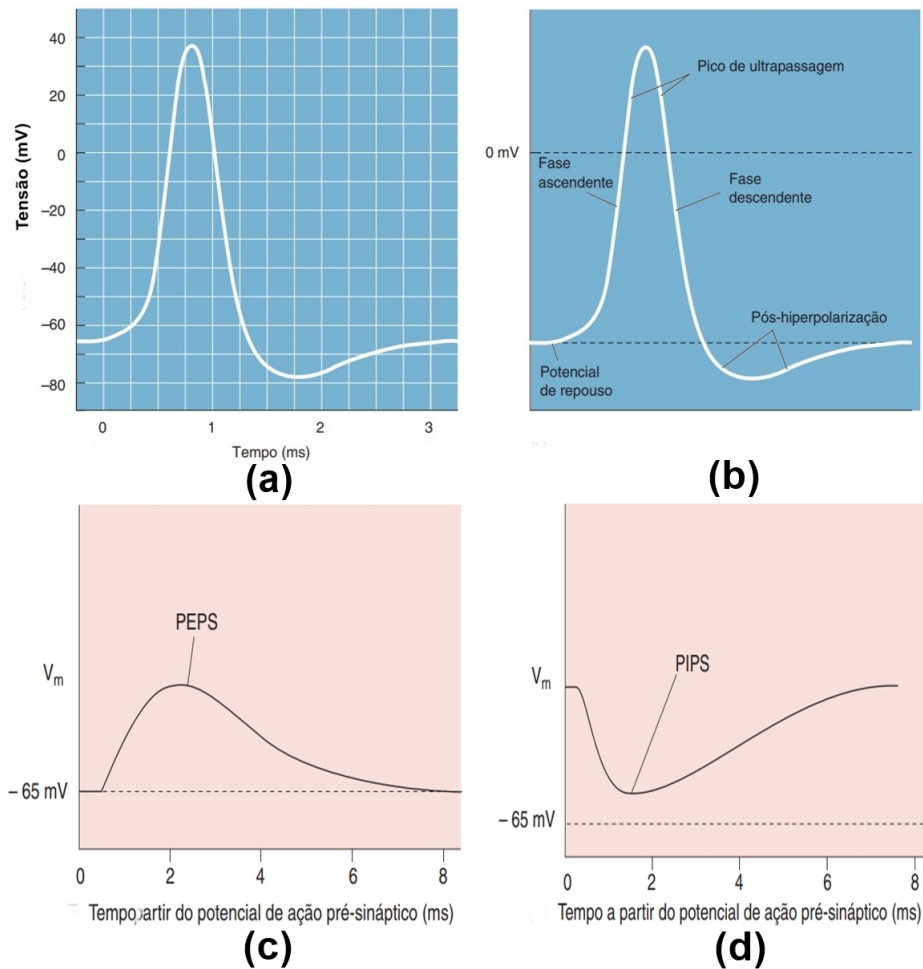
Figura 1 – Neurônio humano (BEAR et al., 2008).



onde o terminal de um axônio de neurônio faz contato com outro neurônio ou célula. Sabe-se que nas sinapses a direção de propagação dos sinais é anterógrada, ou seja, do axônio de um neurônio para os dendritos de outros neurônios subsequentes (GUYTON; HALL, 2006 apud RAMOS, 2017). O potencial do axônio é denominado potencial pré-sináptico e o potencial do neurônio alvo é denominado pós-sináptico. O potencial pós-sináptico pode ser de dois tipos: pode ser um potencial excitatório pós-sináptico (PEPS), que desencadeia um potencial de ação ou um potencial inibitório pós-sináptico (PIPS), que diminui a possibilidade de desencadear um potencial de ação.

O potencial de ação é composto por etapas. O mesmo tem início quando membrana do neurônio está em repouso possui, neste instante a tensão na mesma é de -65 mV. Devido a reações químicas que ocorrem na célula, a membrana começa a se despolarizar até alcançar um limiar denominado limiar de ativação, esse limiar é o nível crítico de despolarização que deve ser alcançado para disparar um potencial de ação (BEAR et al., 2008). Ao passar o limiar inicia-se a fase denominada ascendente, está fase dura até atingir uma tensão máxima de 40 mV denominada pico de ultrapassagem. Após esse pico a membrana começa se repolarizar entrando em uma fase descendente até chegar a hiperpolarização. Este ciclo leva 2 ms para ser concluído, assim tem-se o sinal do potencial de ação descritos nos gráficos a e b da Figura 2, ainda, na mesma figura estão demonstrados o PEPS no gráfico c e o PIPS no gráfico d.

Figura 2 – Potencial de ação, PEPS e PIPS (adaptado de BEAR et al.,2008).



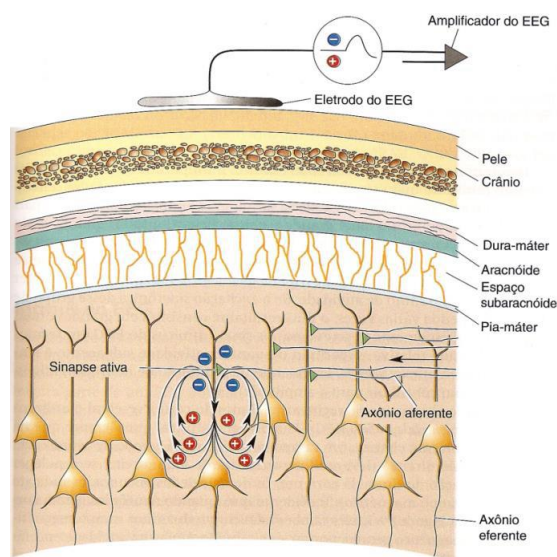
## 2.2 Eletroencefalograma

O EEG é uma técnica de aquisição de sinais cerebrais. Esses sinais podem ser obtidos de forma não invasiva com eletrodos posicionados no couro cabeludo e de forma invasiva com eletrodos intracranianos. O método não invasivo é mais empregado, pois, o método intracraniano necessita de cirurgia e com o passar do tempo o equipamento é rejeitado pelo corpo. Os eletrodos posicionados na superfície da cabeça captam sinais elétricos gerados por grandes grupos de neurônios. A atividade das ondas captadas no EEG é causada pelo fluxo de corrente devido à atividade somada do PEPS e PIPS nas sinapses dendríticas, mas não pelos potenciais de ação (GOMES, 2015).

Ao realizar a aquisição de sinais da atividade cerebral há uma grande dificuldade devido à estrutura de tecido presente entre a localização dos sensores e o córtex cerebral (couro cabeludo, ossos, superfície externa do córtex) (SILVA, 2017). A Figura 3 descreve as camadas entre o eletrodo e o cérebro. Devido à resistência exercida pelas camadas do

crânio o sinal do EEG é atenuado e espalhado. Com isso o potencial elétrico registrado pelos eletrodos no escalpo, no momento em que uma massa de neurônios é excitada em sincronia, isto é, recebe sinapses ao mesmo tempo para haver um somatório de potenciais tem uma intensidade muito menor que a intensidade do PIPS e do PEPS, que estão na ordem de dezena de milivolts. O sinal captado no EEG tem amplitude de 1 a 50  $\mu\text{V}$  e frequência de 0,1 a 80 Hz. A essa faixa de frequências é subdividida em grupos ou ritmos

Figura 3 – Camadas entre o eletrodo e o cérebro (BEAR et al., 2008).



( $\delta$ ,  $\theta$ ,  $\alpha$ ,  $\beta$ ,  $\mu$  e  $\gamma$ ). Esses ritmos são:

- i. Delta ( $\delta$ ) - Ritmo lento, registrado no estágio do sono profundo, situado na faixa de frequência de 1 a 4 Hz.
- ii. Teta ( $\theta$ ) - Ritmo lento, registrado quando o indivíduo realiza atividades mentais, como resolução de problemas, situado na faixa de frequência de 4 a 8 Hz
- iii. Alfa ( $\alpha$ ) - Ritmo lento, registrado em indivíduos acordados, melhor visto em estado de relaxamento, situado na faixa de frequência de 8 a 12 Hz.
- iv. Beta ( $\beta$ ) - Ritmo rápido, registrado em estado de vigia ou sono REM (sono de movimento rápido dos olhos), situado na faixa de frequência de 14 a 30 Hz.
- v. Mu ( $\mu$ ) - Ritmo lento, associado às atividades motoras, situado na faixa de frequência de 8 a 12 Hz.
- vi. Gama ( $\gamma$ ) - Ritmo muito rápido, com frequências entre 30 a 80 Hz. Ocorre no córtex somatosensorial durante o processamento sensorial de modalidade cruzada (percepção que combina dois sentidos diferentes, tais como audição e visão).

Observa-se que a faixa de frequências dos ritmos  $\alpha$  e  $\mu$  se sobrepõem, no entanto estes ritmos diferem entre si no que concerne a sua função e origem no encéfalo. O ritmo  $\alpha$  pode ser medido por eletrodos posicionados em locais variados do escalpo, como os lobos frontal, parietal e occipital. No lobo occipital este ritmo está associado ao estado síncrono de repouso de neurônios no córtex estriado e se manifesta enquanto não há processamento de informação visual, ou seja, quando o indivíduo permanece de olhos fechados. Ao abrir os olhos os neurônios responsáveis pelo processamento da informação visual são ativados e sua atividade global é dessincronizada resultando no desaparecimento deste potencial oscilatório no EEG. Já o ritmo  $\mu$  pode ser medido por eletrodos posicionados sobre o córtex motor e está associado ao estado de repouso dos neurônios motores. Então, o ritmo  $\mu$  pode ser observado enquanto o indivíduo permanece em repouso e sem se movimentar. No entanto, durante a execução de um movimento ou mesmo o planejamento, e imaginação da execução deste movimento, os neurônios motores são ativados e sua atividade global torna-se dessincronizada resultando no desaparecimento deste potencial oscilatório no EEG medido nesta área do escalpo (PFURTSCHELLER; SILVA, 1999).

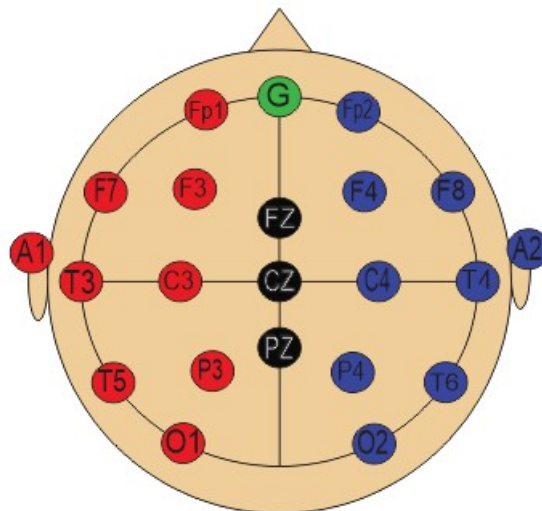
Como a origem dos ritmos citados anteriormente são conhecidas, é possível retirar informações do EEG que podem ser úteis para diferentes aplicações. Utilizando o ritmo  $\mu$ , podemos identificar os diferentes sinais gerados durante a imaginação do movimento dos braços. Assim, podemos obter dois sinais diferentes, um para o movimento do braço direito e outro do braço esquerdo. Processando esses sinais é possível criar um comando binário para uma interface cérebro máquina.

### 2.2.1 Aquisição do sinal

Como visto anteriormente a aquisição do sinal é feita através de eletrodos posicionados no couro cabeludo. O posicionamento dos eletrodos é feito de acordo com o sistema internacional 10-20 de posicionamento de eletrodos, que está descrito na Figura 4. Na Figura 4 podemos observar que os eletrodos apresentam letras que representam os lobos cerebrais e o ponto de referência G. Os lobos são responsáveis por funções cerebrais, o lobo frontal é responsável pelas funções motoras, pensamentos e emoções. O lobo parietal responde pela sensibilidade da pele, músculos e tendões. O lobo occipital corresponde ao controle da visão e o lobo temporal controla a audição. E os números representam os hemisférios, onde números pares representam o lado direito e os números



Figura 4 – Posicionamento dos eletrodos (Elaborada pelo autor).



impares o hemisfério esquerdo. Como o sinal no couro cabeludo é fraco, junto com os eletrodos é empregado um amplificador para aumentar a amplitude do sinal.

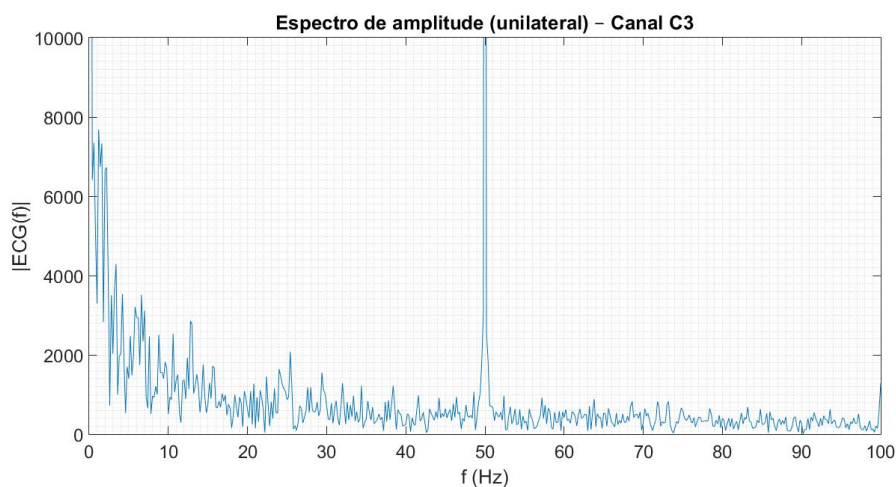
O EEG captado do cérebro é de dois tipos. O primeiro é o registro da atividade espontânea do cérebro o qual é o resultado do campo elétrico gerado pelo cérebro sem tarefas específicas que o determine (adaptado de ALMEIDA, 1997). E o segundo sinal ocorre quando a pessoa imagina uma ação ou antes praticar a mesma. Assim, podemos definir onde está a informação de interesse, ou seja, quando o sinal cerebral é alterado por uma ação. Podemos considerar a ação de imaginar a movimentação dos braços, neste caso os eletrodos posicionados sobre o lobo frontal vão captar uma mudança no sinal. Como o cérebro é dividido em dois hemisférios, direito e esquerdo. Sabe-se que o hemisfério direito do cérebro comanda o lado esquerdo do corpo e o hemisfério esquerdo comanda o lado direito. Assim, quando uma pessoa imaginar o movimento do braço direito, o eletrodo do lado esquerdo vai captar um sinal com uma energia menor na frequência de 10 Hz do que o eletrodo da direita. Ou seja, quando existe uma ação o cérebro emite sinais com energia menor na região que está executando essa ação.

### 2.2.2 Artefatos

Devido à baixa amplitude que o EEG apresenta, o mesmo é bastante suscetível a interferências e ruídos, denominados de artefatos. Os artefatos são sinais elétricos que corrompem o EEG. Os artefatos provocados de ocorrência mais comum são os instrumentais, como os de eletrodo e fisiológicos ou biológicos (ANGHINAH et al., 2006). Muitas

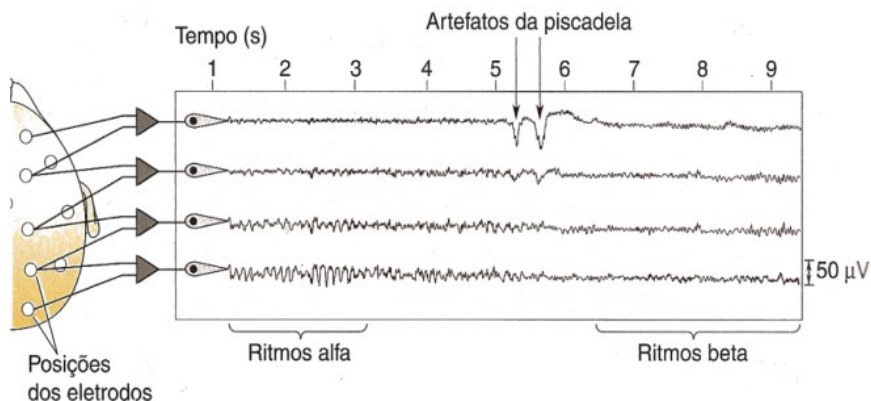
vezes os eletrodos não fazem contato corretamente com a superfície analisada gerando artefatos e flutuações do sinal. Em alguns casos, a interferência é gerada pela má higienização dos eletrodos formando uma superfície com maior impedância (JARDIM, 2017). Além de artefatos gerados pelos eletrodos, existe outro artefato de origem instrumental que é frequência fornecidas pela rede elétrica aos aparelhos conectados em tomadas que podem ser de 50 Hz ou 60 Hz, a Figura 5 demonstra um artefato de 50 Hz de uma rede elétrica. Os artefatos fisiológicos são referentes aos outros sinais biológicos presentes no

Figura 5 – Artefato instrumental de 50 Hz (Elaborada pelo autor).



corpo humano. Esses sinais são de atividade elétrica muscular, movimento dos olhos e atividade elétrica do coração. Na Figura 6 é demonstrada a captação de um sinal fisiológico no EEG. Esse artefato é originado pelo movimento dos olhos. Na Figura 6 podemos observar o artefato entre os instantes 5 e 6.

Figura 6 – Artefato fisiológicos (BEAR et al., 2008).



### 2.2.3 Filtragem do sinal

Antes de enviar os dados para classificação foi utilizada uma técnica de filtragem simples para diminuir os ruídos do sinal, denominada CAR (do inglês, *Common Average Reference*). Essa técnica se baseia no fato de que os sinais biológicos externos ao cérebro afetam aproximadamente da mesma maneira todos os eletrodos de EEG, devido à grande distância da fonte de ruído aos eletrodos sob o escalpo (BENEVIDES, 2009). Assim, para efetuar o CAR, deve-se realizar a média dos sinais de todos os eletrodos. Com a média calculada deve-se fazer a subtração entre a média e o sinal dos eletrodos que se deseja filtrar.

### 2.3 Transformada de Fourier

O sinal eletroencefalográfico é obtido há uma taxa de amostragem constante no domínio do tempo. Como citado anteriormente, a informação retirada do EEG está contida em diferentes faixas de frequência, assim, existe a necessidade de representar o seu espectro no domínio da frequência. Sabe-se que o EEG é um sinal não periódico, assim, a forma correta de representa-lo no domínio da frequência é utilizando a transformada de Fourier. A chamada Transformada de Fourier pode ser considerada como um limite de uma combinação linear infinita de ondas senoidais e que encontra grandes aplicações no tratamento de sinais estacionários (BOLZAN, 2004).

A transformada de Fourier ocorre em pares de transformada, ou seja, quando transformamos um sinal no domínio do tempo em seu respectivo espectro de frequência utilizamos a chamada integral de contorno representada na Equação 1. E para desfazer a transformação usamos o seu par denominado integral de inversão ou transformada inversa representado pela Equação 2.

$$g(t) = \int_{-\infty}^{\infty} G(f) \cdot e^{j2\pi ft} df \quad (1)$$

$$G(f) = \int_{-\infty}^{\infty} g(t) \cdot e^{-j2\pi ft} dt \quad (2)$$

A transformada de Fourier citada anteriormente é denominada transformada de Fourier de tempo contínuo, é importante salientar que a função utilizada para fazer a representação do sinal no domínio da frequência é a função FFT do *software* Matlab. Essa

função é representada pela série de Fourier de tempo discreto, que representa o tempo em amostras discretas. Segundo Haykin e Veen (2001) a série de Fourier de tempo discreto é a única representação que tem valores discretos no tempo e na frequência e é adequada para a implementação direta no MATLAB. Os pares de série de Fourier são representados pelas Equações 3 e 4.

$$x[n] = \sum_{k=N} X[k] e^{jk\omega n} \quad (3)$$

$$X[k] = \frac{1}{N} \sum_{k=N} x[n] e^{-jk\omega n} \quad (4)$$

Onde  $x[n]$  tem período fundamental  $N$  e  $\omega = \frac{2\pi}{N}$ .

## 2.4 Classificadores

Os classificadores podem ser definidos como um conjunto de técnicas que conseguem definir de qual classe um conjunto de dados pertence. Nessa seção são descritos os classificadores que foram utilizados para fazer a classificação das informações presentes no EEG.

### 2.4.1 K-vizinhos mais próximos

O k-nn foi proposto por Fukunaga e Narendra (1975), este é um classificador onde o aprendizado de um novo “objeto” é feito com base nos exemplos de treinamento (aprendizagem supervisionada) (ANDRADE, 2018). A classificação no k-nn é baseada na distância entre amostras de teste e treinamento. Esse aspecto faz com que o método seja simples de implementar. O algoritmo k-nn completa sua execução em duas etapas, primeiro encontrando o número de vizinhos mais próximos e, segundo, classificando o ponto de dados em uma classe particular usando a primeira etapa (BABLANI et al., 2018). Para encontrar os vizinhos mais próximo é necessário calcular a distância entre o grupo de treinamento e o grupo de teste. Existem diferentes métodos para o cálculo desta distância, um dos mais comuns é a distância euclidiana, sendo essa a forma que foi utilizada no

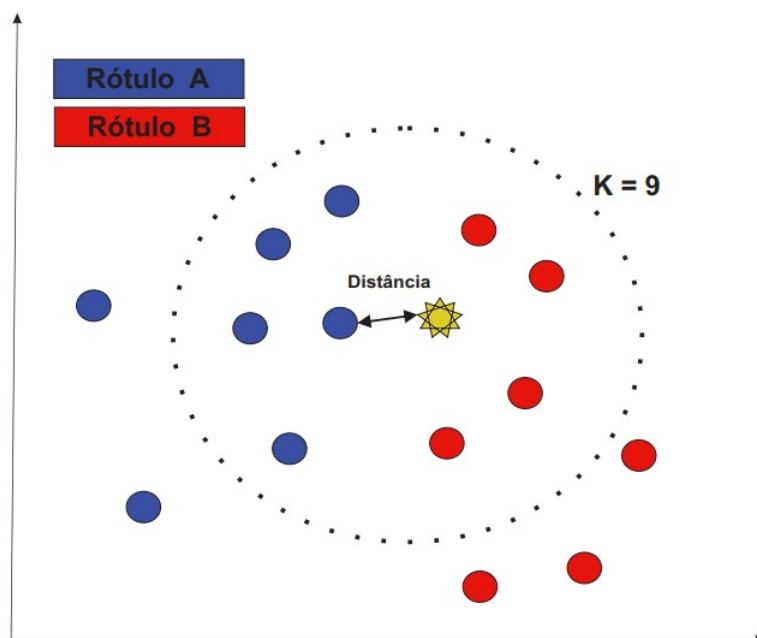
trabalho. O cálculo da distância euclidiana é definido pela Equação 5.

$$De(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (5)$$

Onde x e y são pontos no plano cartesiano, com o somatório dos pontos é possível trassar um reta entre eles para definir sua distância.

Para a etapa de classificação deve-se definir o valor de k. O k representa quantas amostras do grupo de treinamento são utilizadas para classificar uma amostra de teste, esse processo pode ser observado na Figura 7.

Figura 7 – Classificação do k-nn (Elaborado pelo autor).



Na Figura 7 pode-se observar algumas amostras do grupo de treinamento separadas em dois rótulos, sendo eles rótulo a e rótulo b, representados respectivamente pelas cores azul e vermelho. Como dito anteriormente, o valor de k define o número de amostras que serão analisadas, ou seja, o número de vizinhos mais próximos, neste caso o valor de k é nove.

Com a amostra de teste representada pela estrela na figura, podemos definir os nove vizinhos mais próximos de acordo com o cálculo da distância euclidiana. Dentro da circunferência pontilhada observamos que dos nove vizinhos mais próximos 5 pertencem ao rótulo a e quatro a rótulo b. Como o número de vizinhos do rótulo a é maior que o do rótulo b, a amostra de teste é classificada como pertencente ao grupo do rótulo a.

### 2.4.2 Redes Neurais Artificiais

Uma rede neural artificial é um sistema paralelo distribuído composto por unidades de processamento que executam funções matemáticas, cujo objetivo é solucionar problemas, formando um sistema representativo de um cérebro humano, capaz de aprender através de exemplos e generaliza a informação aprendida (JUNIOR, 2005). As unidades de processamento das RNAs são os neurônios artificiais. Como nos neurônios biológicos, os neurônios artificiais possuem a função de receber e transmitir informações. A transmissão em um neurônio artificial é dada a partir do recebimento de sinais de entrada ( $X_1, X_2, \dots, X_n$ ) vindos de fontes externas propagados através das sinapses artificiais. As sinapses artificiais são interconexões controladas por pesos sinápticos ( $W_1, W_2, \dots, W_n$ ). Os pesos sinápticos são valores multiplicados aos valores de entrada que permitem a avaliação de cada uma das entradas. Assim, é possível avaliar a relevância de cada entrada para o funcionamento do respectivo neurônio. O resultado do produto entre as entradas e os pesos sinápticos é somado com o limiar de ativação ( $\theta$ ) através de um combinador linear. O resultado deste somatório é o potencial de ativação ( $u$ ), se  $u > (\sum - \theta)$  o potencial resultante é excitatório, ou seja, a saída do neurônio é ativada, se  $u < (\sum - \theta)$  o potencial resultante é inibitório, isto é, a saída do neurônio é inibida. O potencial de ativação é descrito pela Equação 6. Os resultados do potencial de ação serão limitados pela função de transferência, gerando a saída ( $y$ ) do neurônio artificial descrita pela Equação 7. A representação de um neurônio artificial é dada pela Figura 8.

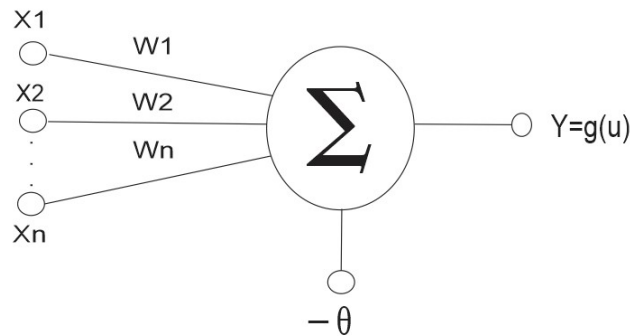
A RNA é utilizada para a classificação de padrões. A classificação é feita através das operações matemáticas que compõe o neurônio artificial e permitem o desenvolvimento da aprendizagem. A rede neural artificial tem por benefício a aprendizagem (supervisionada ou não), que seria o mapeamento de entradas e saídas em que se fornecem os pesos sinápticos para adequar as saídas com as entradas (PERALES, 2011).

$$u = \sum_{i=1}^n (x_i \cdot w_i) - \theta \quad (6)$$

Onde  $W_i$  são os pesos sinápticos,  $X_i$  as variáveis de entrada e  $\theta$  o limiar de ativação.

$$Y = g(u) \quad (7)$$

Figura 8 – Neurônio artificial (Elaborado pelo autor).



### 2.4.3 Funções de Ativação

As funções de ativação calculam a saída de um neurônio, limitando o valor das saídas conforme o intervalo da função escolhida. As funções de ativação são divididas em dois tipos, funções de ativação parcialmente diferenciais e funções de ativação totalmente diferenciais. As funções parcialmente diferenciais são aquelas que possuem pontos onde sua derivada de primeira ordem é inexistente. E as diferenciais possuem derivada de primeira ordem em todos os pontos pertencentes a mesma. A seguir são mostradas algumas funções que podem ser utilizadas em redes neurais artificiais:

- i. Função degrau unitário: Representada na Figura 9, é uma função que limita a amplitude da saída do neurônio em um intervalo entre zero e um, quando o valor da função potência for maior ou igual que zero a função assume o valor um, quando o valor for menor que zero a mesma possui resultado nulo, esse comportamento pode ser visto na Equação 8.

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases} \quad (8)$$

- ii. Função linear: É uma função representada por uma reta que produz um resultado de saída igual ao potencial de ativação ( $u$ ). Esta função é representada pela Figura 10 e pela Equação 9.

$$g(u) = u \quad (9)$$

- iii. Função rampa simétrica: É uma função que limita a saída em um intervalo  $[1,-1]$ , a

Figura 9 – Função degrau unitário (Elaborado pelo autor).

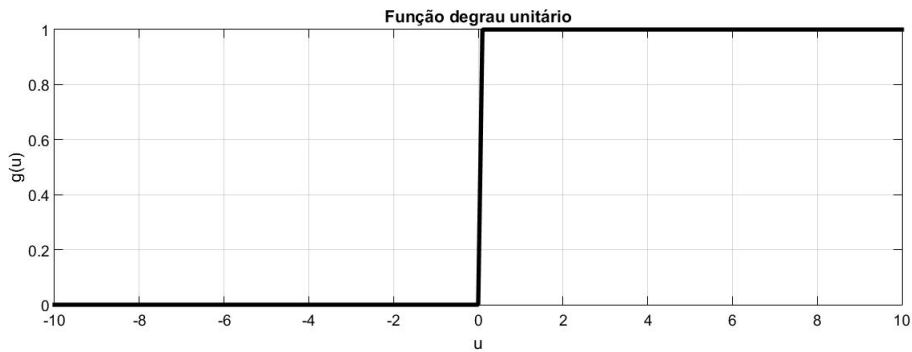
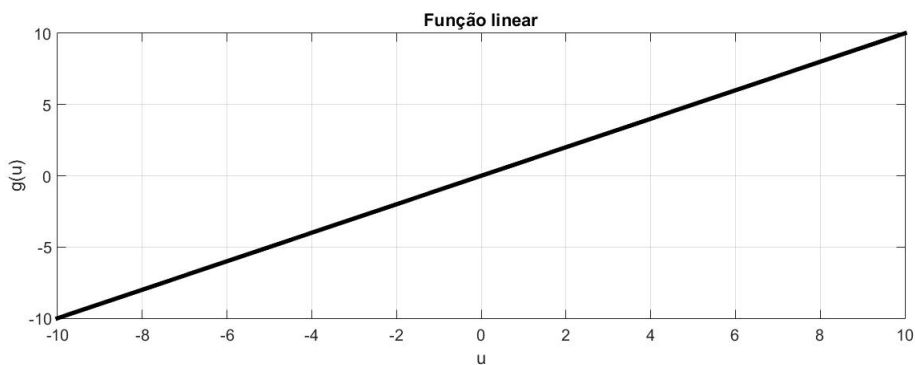


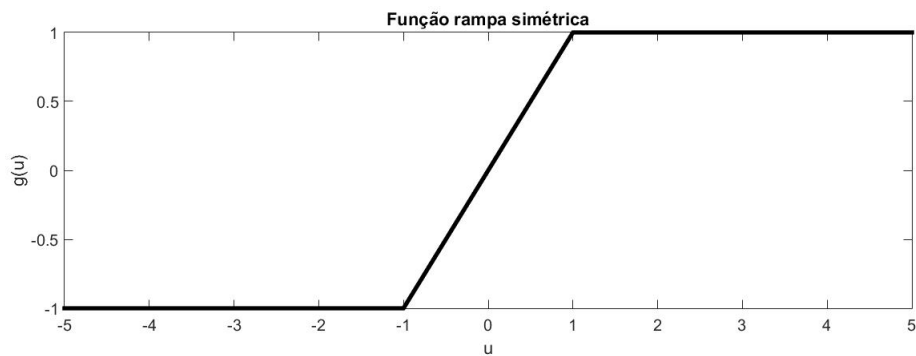
Figura 10 – Função linear (Elaborado pelo autor).



Equação 10 descreve esse intervalo e a Figura 11 representa o gráfico da função.

$$g(u) = \begin{cases} 1, & \text{se } u > 1 \\ u, & \text{se } -1 \leq u \leq 1 \\ -1, & \text{se } u < -1 \end{cases} \quad (10)$$

Figura 11 – Função rampa simétrica (Elaborado pelo autor).



- iv. Função logística ou sigmoide: Descrita na Figura 12 limitara o valor da saída em valores reais entre zero e um, a expressão matemática que define a mesma é dada

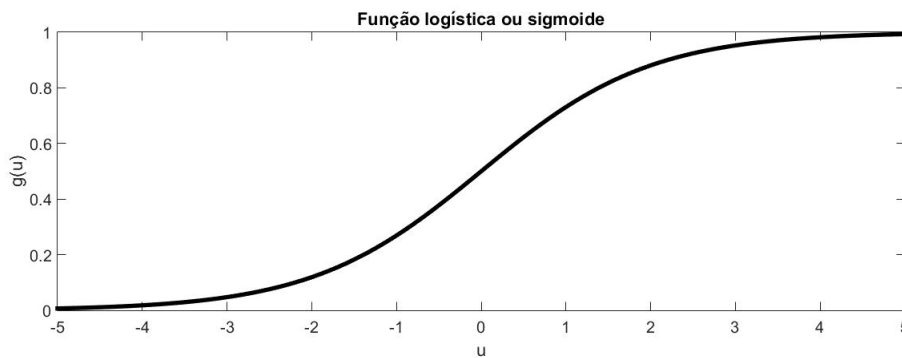


pela Equação 11.

$$g(u) = \frac{1}{1 + e^{-\beta u}} \quad (11)$$

Onde  $\beta$  é uma constante real que define a inclinação da função referente ao seu ponto de inflexão.

Figura 12 – Função logística (Elaborado pelo autor).

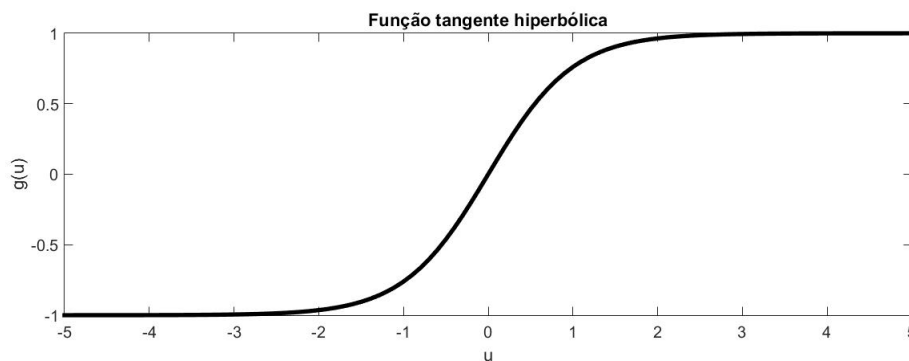


- v. Função tangente hiperbólica: É uma função que sempre assume valores reais entre 1 e -1. Essa função tem como representação matemática a Equação 12 e o gráfico da Figura 13.

$$g(u) = \frac{1 - e^{-\beta u}}{1 + e^{-\beta u}} \quad (12)$$

Onde  $\beta$  é uma constante real que define a inclinação da função referente ao seu ponto de inflexão.

Figura 13 – Função tangente hiperbólica (Elaborado pelo autor).



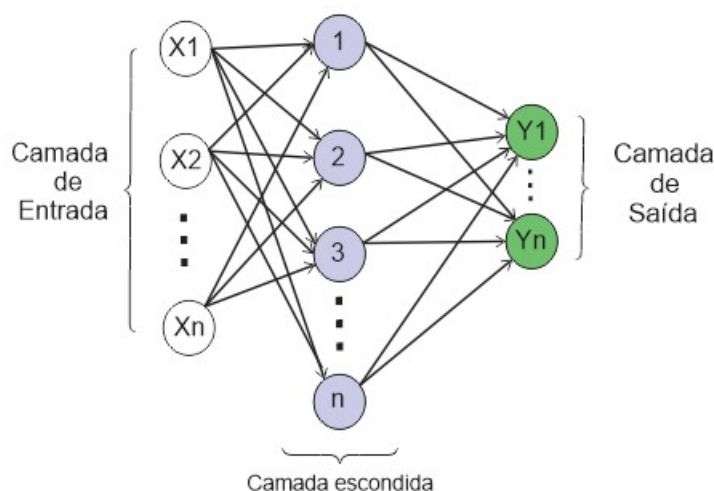
#### 2.4.4 Redes Perceptron Multicamadas

A arquitetura de uma RNA é a forma como os neurônios estão dispostos na rede (RUFINO, 2017). Existem vários tipos de arquiteturas de rede neural. Uma delas é a

rede perceptron multicamadas (PMC). Essa arquitetura tem como característica arranjar os neurônios em camada de entrada, uma ou mais camadas escondidas e uma camada de saída. A função dos neurônios ocultos é intervir entre a entrada externa e a saída da rede de uma maneira útil, tornando a rede capaz de extrair estatísticas de ordem elevada (HAYKIN, 2001 apud CORREIA, 2015).

O princípio de funcionamento do PMC se dá com o fluxo de sinal que se propaga de camada em camada a partir da entrada e chegando ao fim na camada de saída, assim as entradas das camadas são as saídas de suas respectivas camadas anteriores. A representação da arquitetura da PMC está descrita na Figura 14. Para que o sinal se propague deve haver um sinal de ativação na entrada de cada camada, este é denominado limiar de ativação ou função de ativação.

Figura 14 – Arquitetura perceptron multicamadas (Elaborado pelo autor).



O método de treinamento desta arquitetura é feito de forma supervisionada. O método define que a amostra de treinamento deve ser composta por sinais de entrada e sua respectiva saída desejada. A primeira etapa de treinamento é denominada de *forward*, onde o sinal vai de camada em camada até gerar uma resposta de saída. Nessa etapa os pesos sinápticos e os limiares se mantêm inalterados. Com o sinal de saída gerado nesta etapa, faz-se a comparação com os sinais da resposta desejada disponibilizados pelo aprendizado supervisionado. A partir da comparação dos dados é possível calcular os desvios ou erros de resposta da rede, o método utilizado para calcular o erro é a função de erro quadrático descrito pela Equação 13. Assim, deve-se passar para a segunda etapa denominada *backward*. Nesta etapa os sinais se propagam no sentido contrário da *forward*, os sinais vão da camada de saída até a camada de entrada. A diferença é que

nesta etapa os pesos sinápticos e limiares são ajustados para eliminar o erro. O ajuste dos pesos sinápticos é dado pela Equação 14. As duas etapas citadas anteriormente compõem o algoritmo *backpropagation*.

$$E(k) = \frac{1}{2} \sum_{j=1}^n (d_i(k) - y_j(k))^2 \quad (13)$$

Onde  $y_j(k)$  é o valor produzido pelo  $j$ -ésimo neurônio de saída da rede considerando a  $K$ -ésima amostra de treinamento e  $d_i(k)$  é o seu respectivo valor desejado.

$$W_{ji}(t+1) = W_{ji} + \eta \cdot \delta_j \cdot y_j \quad (14)$$

Onde  $\eta$  é a taxa de aprendizagem,  $\delta_j$  é o gradiente do  $j$ -ésimo neurônio e  $y_j$  é à saída do  $j$ -ésimo neurônio.

Neste trabalho foram utilizados dois algoritmos de treinamento para a RNA. O primeiro foi o algoritmo de treinamento Levenberg-Marquardt, que é o treinamento padrão da função utilizada no Matlab. Este algoritmo de treinamento atualiza os pesos sinápticos com base no método de Levenberg-Marquardt. Inicialmente o método foi proposto por Lvenberg (LEVENBERG, 1944) como uma alternativa dos métodos padrões para resolver problemas de mínimos quadrados não-lineares. Anos depois, Marquardt (MARQUARDT, 1963) avançou nas pesquisas acerca de tal técnica, originando o denominado método de Levenberg-Marquardt. Frequentemente é considerado como o mais rápido dos algoritmos de treinamento retropropagação do erro, porém requer mais memória computacional do que os outros algoritmos (GONÇALVES et al., 2010).

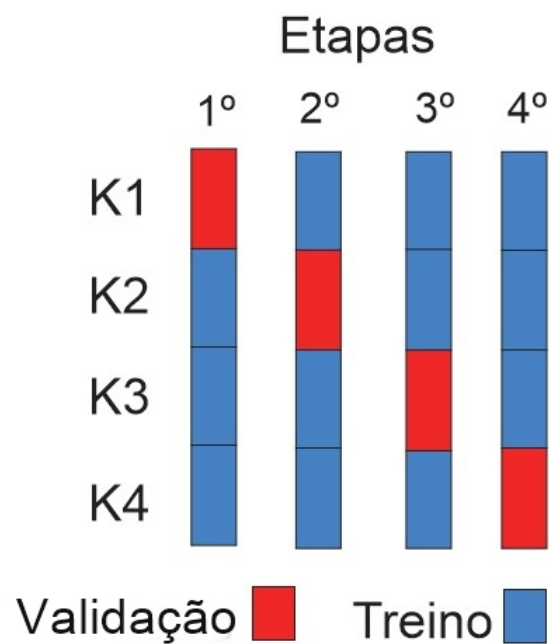
O segundo algoritmo de treinamento utilizado é a regularização Bayesiana, essa função atualiza os valores dos pesos de acordo com o método de Levenberg-Marquardt. Tipicamente os algoritmos de treinamento têm como objetivo reduzir a soma dos erros quadráticos. A regularização Bayesiana minimiza a combinação dos erros quadráticos e os pesos, para então produzir uma rede generalizada e com respostas mais suaves (MAC-KAY, 1992; FORESSE e HAGAN, 1997 aput GONÇALVES et al., 2010).

#### 2.4.5 Validação cruzada

Para todos os classificadores foi utilizado o método de validação cruzada do tipo  $k$ -fold para validar a capacidade de generalização dos classificadores. O método se baseia em dividir o conjunto de dados em  $k$ -partições. A variável  $k$  pode assumir qualquer valor

inteiro entre 1 e o número de amostras presentes no conjunto de dados. Supondo que  $k=4$ , os dados são divididos em 4 partes  $k_1, k_2, k_3$  e  $k_4$ , sendo utilizado  $3/4$  desses dados para o subconjunto de treinamento e  $1/4$  para validação. Essa etapa deve se repetir até que todas as 4 partes façam parte do subconjunto de validação. Assim, se  $k_1$  for do subconjunto de validação,  $k_2, k_3$  e  $k_4$  são utilizados para treinar o classificador, em seguida se  $k_2$  for utilizado para o subconjunto de validação, então  $k_1, k_3$  e  $k_4$  são do subconjunto de treino e assim sucessivamente. A Figura 15 representa a validação cruzada.

Figura 15 – Representação da validação cruzada (Elaborado pelo autor).



### 3 METODOLOGIA

Neste capítulo são abordados os dados utilizados neste trabalho, assim como a metodologia escolhida para processá-los e classificá-los. Escolheu-se trabalhar com três bancos de dados cuja classificação de padrões fosse gradualmente mais difícil e foram testados dois tipos de classificadores em cada um dos dados: um classificador matematicamente simples, do tipo k-nn e um classificador mais complexo, que é a rede neural perceptron multicamadas. Os bancos de dados são detalhados nas seções 3.1.1, 3.1.2 e 3.1.3. Além, disso, conforme detalhado na revisão bibliográfica, a execução de tarefas mentais altera frequências específicas de sinais EEG medidos por canais posicionados em locais também específicos. Então, na seção 3.2.1 foi feita a verificação de que esta informação estava presente nos banco de dados *BCI competition II* e *BCI competition III*.

#### 3.1 Banco de dados

##### 3.1.1 Banco de dados de Fisher

O banco de dados de Fisher (FISHER, 1936), ou conjunto de dados flor de Iris, é um conjunto de medições relacionadas a três tipos de flores. Estas medições foram realizadas pelo estatístico e biólogo britânico Ronald Fisher em seu artigo de 1936. Posteriormente estes dados se tornaram uma base para testes de desempenho de diferentes classificadores. O banco de dados de Fisher é composto por dimensões de uma flor de Iris. Esse banco de dados é composto por uma matriz que possui dados referentes a largura e comprimento de pétalas e sépalas de três espécies diferentes de flor de Iris. A matriz possui 5 colunas e 150 linhas. As quatro primeiras colunas são referentes as larguras e comprimentos de sépalas e pétalas e a quinta coluna possui os rótulos das espécies. As linhas são divididas em três partes iguais, cada parte representa um dos rótulos, os mesmos são "setosa", "versicolor" e "virginica". Esse banco de dados foi utilizado para fazer a análise dos classificadores utilizados neste trabalho.

### 3.1.2 Banco de dados BCI *competition II*

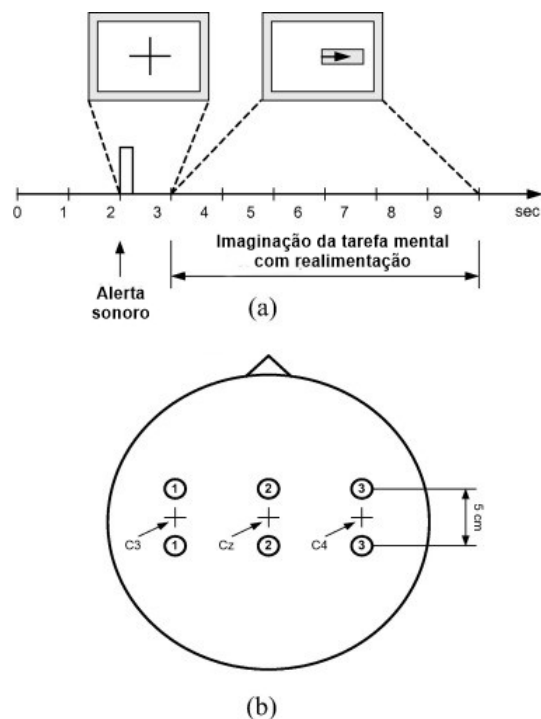
O banco de dados BCI *competition II* foi disponibilizado para uma competição de interfaces cérebro-máquina (BLANKERTZ et al., 2004). Este banco de dados é composto por uma matriz  $y_{train}$  com 140 linhas, possuindo valores entre 1 e 2, dois arranjos tridimensionais  $x_{train}$  e  $x_{test}$ , ambas possuem 1152 linhas 3 colunas e 140 camadas. Por serem dados utilizados para competição,  $y_{train}$  possui os rótulos dos dados de treinamento  $x_{train}$  e  $x_{teste}$  não possui rótulos, assim, para o trabalho a matriz  $x_{teste}$  não foi utilizada.

Os dados citados anteriormente foram obtidos através de um experimento com o EEG. O experimento constituía em obter o EEG através de 3 eletrodos posicionados no topo da cabeça de uma pessoa em uma região denominada córtex motor, que é responsável pelas atividades motoras. O indivíduo deveria efetuar duas tarefas mentais distintas, imaginar o movimento das mãos direita e esquerda uma de cada vez. Para isso, eram emitidos sinais sonoros e visuais em uma tela posicionada em frente ao usuário, alertando-o para executar a tarefa, e se a tarefa estava sendo executada corretamente, esse método é denominado realimentação. Assim, estes indivíduos percebiam quando erravam ou acertavam a tarefa e podiam aprender através de seus erros. Esta retroalimentação do desempenho da tarefa permite que o indivíduo modele sua atividade cerebral de maneira que este EEG se torna algo que não é simplesmente a imaginação de movimento de partes do corpo e estes sinais de EEG são mais fáceis de se classificar comparados a um EEG obtido sem realimentação do usuário. A Figura 16 representa o experimento.

O experimento possui 9 segundos de duração e foi repetido 140 vezes. O tempo começa a ser contabilizado no instante de 1 segundo e acaba em 10 segundos, resultando nos 9 segundos citados anteriormente. No instante de 2 segundos um sinal sonoro é emitido para o usuário se preparar para executar a atividade. No instante 3 uma seta aparece na tela para direita ou para esquerda para definir qual a tarefa mental o usuário deverá executar, essa tarefa se encerra no instante de 10 segundos. O valor 1 foi atribuído para a imaginação do movimento da mão esquerda e o valor 2 atribuído para a imaginação da mão direita, assim, formando o vetor  $y_{train}$ .

O EEG é amostrado a uma frequência de 128 Hz, como a duração do experimento possui 9 segundos, então, multiplicando a frequência de amostragem pela duração do experimento, obtemos as 1152 linhas que compõem os arranjos desse banco de dados. Os outros componentes das matrizes são as 3 colunas que representam o número de eletrodos

Figura 16 – Representação do experimento realizado para a obtenção da base de dados BCI *competition* II (BLANKERTZ et al., 2004).



e as 140 camadas que representam as repetições do experimento.

### 3.1.3 Banco de dados BCI *competition* III

O terceiro banco de dados utilizado no trabalho também foi disponibilizado pelo professor da Universidade de Graz (Áustria) José Millán (MILLAN, 2004) para o BCI *competition* III. Este banco de dados é constituído de dados de três indivíduos. Diferente do banco de dados anterior, este não possui a informação de que os usuários estão imaginando as tarefas de forma correta. O EEG foi adquirido através de 32 eletrodos posicionados no topo da cabeça dos usuários. Os dados foram obtidos com uma taxa de amostragem de 512 Hz. As tarefas mentais foram divididas em:

- i. Imaginação contínua e repetitiva do movimento de sua mão esquerda durante 15 s. Rótulo: 2;
- ii. Imaginaçãoo contínua e repetitiva do movimento de sua mão direita durante 15 s. Rótulo: 3; e
- iii. Geração de palavras começando com uma mesma letra qualquer durante 15 s. Rótulo: 7.

As tarefas possuem duração de aproximadamente 15 s e foram realizadas aleatoriamente pelos usuários até totalizarem uma seção de quatro minutos. Os usuários executaram um total de 4 seções, com intervalos entre 5 e 10 minutos. Entre as seções 3 delas possuem rótulos e a quarta seção não possui, pois, a mesma foi destinada a competição.

Os dados obtidos foram salvos em quatro arquivos para cada indivíduo. Esses arquivos são compostos por duas matrizes, a matriz  $X$  é composta por 122.368 linhas, que correspondem as amostras do EEG e por 32 colunas que representam os eletrodos. A outra matriz é a  $Y$  que corresponde aos rótulos do EEG com 122.368 linhas e uma coluna. Essa matriz define as amostras pertencentes a suas respectivas atividades executadas pelo usuário.

### **3.2 Pré-processamento dos dados de EEG**

Antes de classificar os sinais EEG é necessário que se faça o pré-processamento dos mesmos. Os dados obtidos são amostrados no domínio do tempo e, como visto anteriormente, as informações que devem ser classificadas estão contidas em faixas de frequência do sinal, assim, o EEG deve ser representado no domínio da frequência. Além disso o sinal pode conter artefatos que devem ser removidos no pré-processamento.

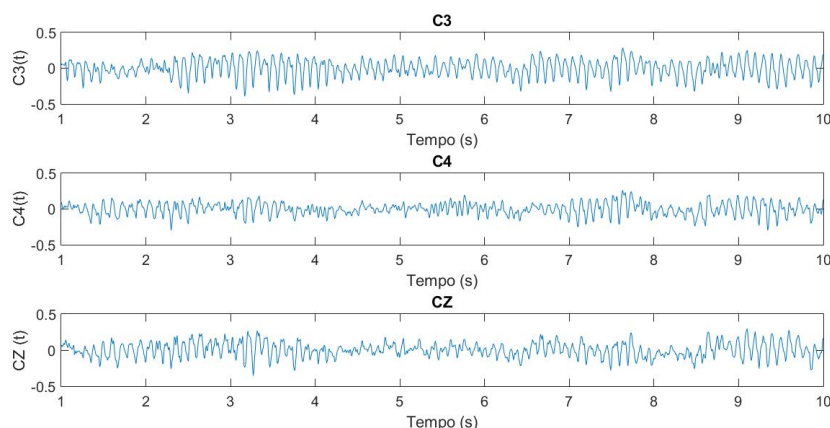
#### **3.2.1 Pré-processamento do Banco de dados BCI *competition II***

Como citado anteriormente o banco de dados BCI *competition II* possui sinais considerados bons, pois os dados eram obtidos com realimentação, além disso, o sinal foi previamente filtrado nas frequências entre 0,5 e 30 Hz. A filtragem foi feita para eliminar a componente CC e os artefatos de origem técnica com a frequência da rede elétrica de 50 Hz. Um exemplo de sinal EEG no domínio do tempo está descrito na Figura 17. A plotagem desse sinal foi feita através do *software* Matlab e o código utilizado para esse fim está descrito no Anexo A.

Este sinal foi obtido através de três eletrodos posicionados sobre a cabeça do usuário. Sendo C3 localizado no lado esquerdo, CZ no centro e C4 no lado direito da cabeça. Os sinais que foram utilizados neste trabalho são referentes aos eletrodos C3 e C4, que por estarem posicionados sobre os hemisférios esquerdo e direito respectivamente possuem um sinal melhor em relação ao CZ que está no centro da cabeça. Os dados são

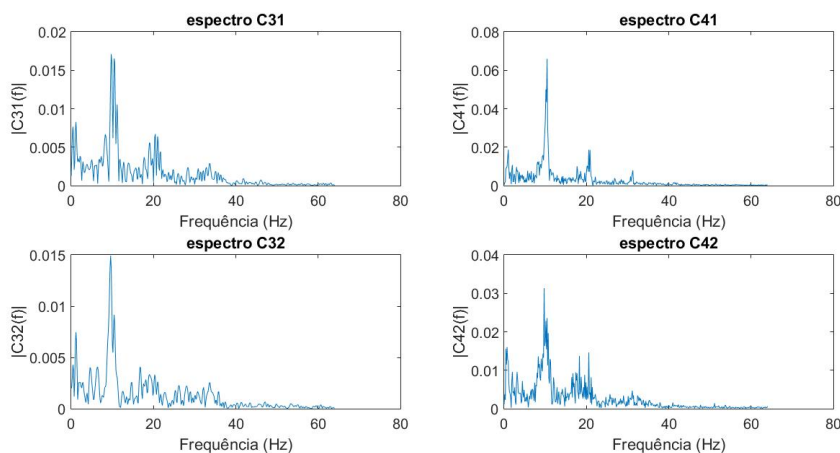


Figura 17 – Representação dos sinais no domínio do tempo (Elaborado pelo autor).



divididos em duas partes, antes da execução da atividade motora e durante a atividade motora, gerando quatro sinais C31 e C41 representam o instante antes da execução da atividade e C32 e C42 durante a atividade. Assim, através do código presente no Anexo B executado no *software* Matlab, foi realizada a transformada discreta de Fourier, com a função transformada rápida de Fourier (FFT, do inglês *fast Fourier transform*), desses sinais. A representação dos sinais no domínio da frequência está descrita na Figura 18.

Figura 18 – Representação dos sinais no domínio da frequência (Elaborado pelo autor).



O código do Anexo B foi feito com o intuito de demonstrar a informação presente na transformada de Fourier. Podemos observar que a amplitude ou energia do sinal é maior na faixa de frequência de 8 a 12 Hz. Isso ocorre, pois, a atividade motora está ligada ao ritmo cerebral Mu.

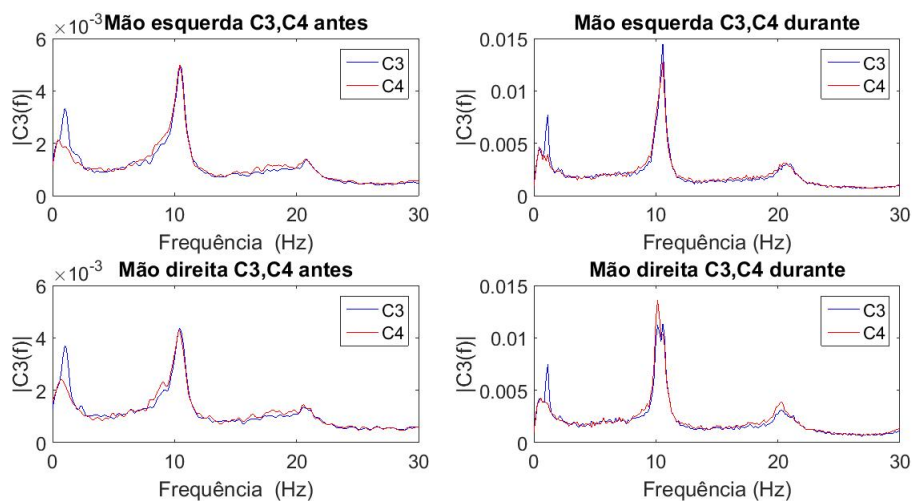
Antes de enviar os dados para classificação foi utilizada uma técnica de filtragem com o método CAR. Após a filtragem são efetuados os mesmos passos citados anteriormente até se obter a transformada de Fourier de C31, C32, C41 e C42. Esses dados

foram salvos em C3M1, C3M2, C4M1 e C4M2, para serem utilizados na classificação. Os passos para obter esses dados para a classificação estão presentes no código do Anexo C.

Como citado anteriormente a informação a ser classificada está contida na faixa de frequência entre 8 e 12 Hz. A energia do sinal nessa faixa de frequência sofre uma queda no momento da imaginação da atividade. Quando se imagina mover a mão direita a energia do eletrodo C3 diminui, isso ocorre, pois, o lado esquerdo do cérebro controla o lado direito do corpo. O mesmo ocorrerá no momento em que a atividade imaginada for a de mover a mão esquerda, assim, a energia do eletrodo C4 posicionado na direita vai diminuir.

Através do código do Anexo D, é demonstrado o que ocorre com os sinais antes e durante a imaginação. Para isso separamos os sinais salvos C3M1, C3M2, C4M1 e C4M2 em duas partes, a primeira parte é referente a imaginação da mão esquerda e a segunda referente a imaginação da mão esquerda. Aplicando a média a cada um dos sinais e a transformada de Fourier dessa média, foram obtidos os espectros representados pela Figura 19.

Figura 19 – Representação dos sinais no domínio da frequência antes e durante a imaginação das atividades motoras (Elaborado pelo autor).



Com base na Figura 19, podemos observar que a energia dos eletrodos C3 e C4 antes da imaginação é aproximadamente a mesma. A partir do momento em que se imagina o movimento, podemos observar que para a imaginação de movimento da mão esquerda, tem-se uma redução da energia em C4 em comparação ao C3. Isso também ocorre para o caso de imaginação da mão direita, a energia de C3 diminui em comparação a de C4.

Esta análise confirma a existência do padrão a ser classificado no banco de dados.

Porém, este padrão, que é a variação na energia da banda Mu, foi facilmente observável apenas no espectro médio calculado com as 140 repetições do experimento de imaginação motora. Na maior parte das vezes este padrão é imperceptível no espectro obtido em uma única realização do experimento, o que resulta no desafio da classificação destes sinais.

### 3.2.2 Pré-processamento do Banco de dados BCI *competition III*

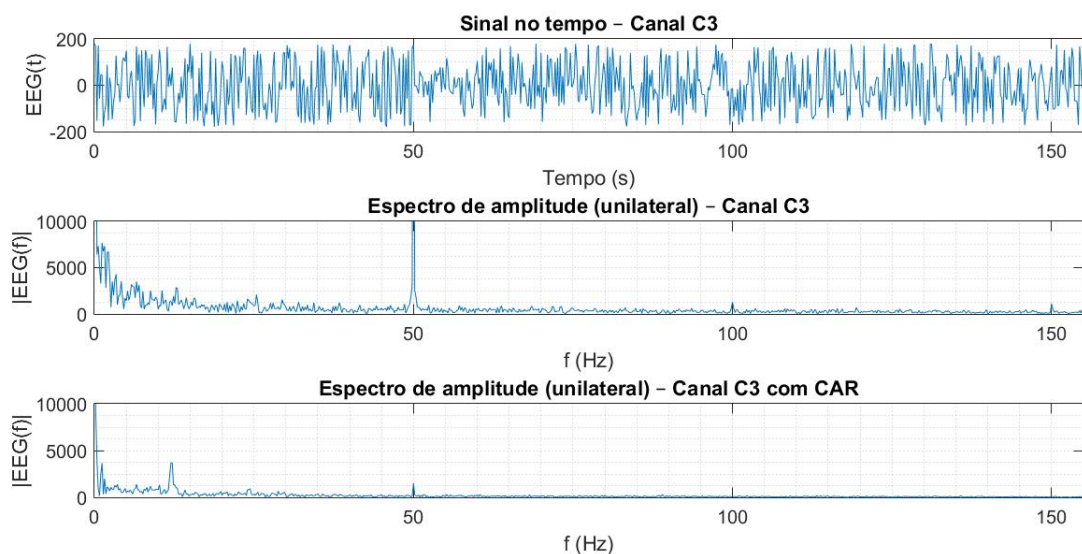
As atividades deste banco de dados possuem duração de aproximadamente 15 s, porém foram executadas aleatoriamente e gravadas continuamente resultando em um arquivo contendo aproximadamente 4 minutos para cada indivíduo. Assim, foi necessário organizar os dados separando as atividades. Este processo já havia sido realizado anteriormente pelo orientador que os organizou utilizando o formato de estruturas. A primeira estrutura é a *subject* representando os três usuários que executaram o experimento. As estruturas que representam as atividades são *tasc2*, *tasc3* e *tasc7*, essa estrutura contém aproximadamente 12 gravações de EEG. Por fim a última estrutura é a estrutura denominada EEG que possui 10240 amostras e 32 eletrodos. Assim, a estrutura fica *subject(j).tasc2(i).EEG(10240,32)*, onde *j* varia de 1 a 3 e *i* varia de 1 a 12, para este caso foi utilizada a *tasc2* como exemplo, para obter as outras atividades, deve se apenas substituir o *tasc2* por *tasc3* ou *tasc7*.

A partir da estrutura, foram escolhidos os dados do usuário 3, pois o mesmo possui exatamente 12 gravações de EEG para cada atividade. Ainda, foram utilizadas apenas as amostras entre 5 e 15 segundos, descartando a parte inicial e final da gravação. Isso é feito para evitar informações indesejadas do sinal. Com isso foram geradas 3 arranjos tridimensionais, uma para cada tarefa mental com 2561 linhas, 32 colunas e 12 camadas.

Diferente do anterior, este banco de dados não foi filtrado previamente. Assim, o mesmo é um sinal ruidoso, então antes de enviar os dados para classificação foi efetuado a filtragem através do CAR. Após efetuar a filtragem, foi realizada a transformada discreta de Fourier dos sinais e os mesmos foram salvos. Foram salvos os dados dos sinais dos eletrodos C3, C4 e F7, pois C3 e C4 estão sobre o córtex motor responsável pelo processamento das atividades motoras e F7 está sobre área de Broca, responsável pelo processamento da fala. Com os sinais desses eletrodos foi gerando C3ME, C4ME e F7ME representando a imaginação do movimento da mão esquerda, C3MD, C4MD e F7MD representando a imaginação do movimento da mão direita e por fim C3P, C4P e FP representando a imaginação das palavras. O código do Anexo E representa essa tarefa.

Com o código do Anexo F, podemos gerar os gráficos presentes na Figura 20 que demonstra o sinal obtido através do eletrodo C3, onde o terceiro usuário está efetuando a atividade de imaginar a mão esquerda. Os gráficos representam o sinal de C3 no domínio do tempo e no domínio da frequência antes de ser filtrado e depois de ser filtrado. Pode-se observar que após o uso do CAR a frequência de 50 Hz da rede foi reduzida e suas harmônicas praticamente eliminadas, ainda se tem uma diminuição nos demais ruídos, fazendo com que a energia das frequências em torno de 8 a 12 Hz fique mais evidente.

Figura 20 – Representação do sinal do eletrodo C3 (Elaborado pelo autor).



### 3.3 Análise dos classificadores

Nessa etapa foi realizadas a análise dos classificadores através do banco de dados de Iris de Fisher. Junto a análise é feita a configuração das funções do *software* Matlab que representam os classificadores.

Para todos os classificadores foi utilizado o método de validação cruzada do tipo k-fold para validar a capacidade de generalização dos classificadores. Ainda, foi feita a normalização dos dados. O objetivo da normalização é alterar os valores das colunas numéricas no conjunto de dados para usar uma escala comum, sem distorcer as diferenças nos intervalos de valores ou perder informações (MELO, 2018). A normalização foi executada fazendo a média de cada uma das entradas e subtraindo as entradas de suas respectivas médias. Essa subtração resultou em valores positivos e negativos, com isso,

foram encontrados os valor máximo e mínimo entre a subtração das entradas com suas respectiva médias. Assim, dividiu-se todos os valores maiores que zero pelo valor máximo da subtração da média com a entrada, e os valores menores que zero pelo valor mínimo da subtração da média com a entrada.

### 3.3.1 Análise do k-vizinhos mais próximos

O classificador k-nn é implementado pela função do *software* Matlab `knnclassify`. Essa função classifica as linhas de um arranjo de dados com base nas linhas do arranjo de treinamento. O arranjo de treinamento e o arranjo de dados a ser classificado devem possuir o mesmo número de colunas.

Para testar essa função, o primeiro passo foi criar um vetor que gera números aleatórios para embaralhar as linhas do banco de dados Iris de Fisher incluindo os rótulos. Assim para toda a simulação o classificador recebera os dados em uma sequência diferente. Após embaralhar os dados os mesmos foram normalizados e divididos em 5 partes sendo 1/5 para classificação e 4/5 para treinamento. Esses dados são enviados para o classificador que está configurado para variar o número de vizinhos de 1 a 50 e com o método de distância euclidiana para calcular a distância dos vizinhos mais próximos. Escolheu-se variar o número de vizinhos para observar qual valor teria a melhor média de acertos.

A classificação dos dados é repetida por 5 vezes, assim, realizando a validação cruzada. Com os 5 resultados obtidos na classificação, foi feita a média e o desvio padrão dos mesmos, estes resultados são demonstrados em uma tabela e em um gráfico de barras. O código que efetua essa atividade está presente no Anexo G.

### 3.3.2 Análise da rede neural artificial

A função utilizada para implementar uma rede neural foi a `newff`. Essa função reproduz uma rede neural *feedforward* multicamadas. A mesma permite que sejam configurados o número de neurônios das camadas escondidas e da camada de saída, ainda podemos definir o número de camadas escondidas. Nesta função é necessário que se defina a função de ativação para cada uma das camadas, e também o algoritmo de treinamento. Neste trabalho foram utilizados dois algoritmos de treinamento *backpropagation* que são

o Levenberg-Marquardt (LM), que é o padrão da função e a regularização bayesiana (RB).

As funções de ativação utilizadas na RNA foram a tangente hiperbólica, para a camada oculta e a função rampa simétrica para a camada de saída. As duas funções são limitadas entre -1 e 1. A função tangente hiperbólica foi escolhida para a camada oculta, pois, a mesma é uma função não-linear. A presença da não-linearidade é importante porque, do contrário, a relação de entrada-saída da rede poderia ser reduzida àquela de um perceptron de camada única (HAYKIN, 2001). Para valores muito altos na entrada do neurônio a função tangente hiperbólica tende a saturar o gradiente, isso causa uma demora no processo de classificação. Para evitar a saturação do gradiente foi utilizada a função rampa simétrica na camada de saída da rede, assim evitando a lentidão no processo.

Com as funções de ativação definidas, foram definidos o número de camadas escondidas e o número de neurônios em cada camada. A rede foi definida com uma única camada escondida, com o número de neurônios variando de 2 a 50. Isso foi feito para testar o desempenho da RNA conforme seus neurônios. Para os bancos de dados que possuem três classes (Flor de Iris e BCI *Competition* III) a camada de saída foi definida com dois neurônios. Assim, as saídas foram definidas por  $[-1, -1]$ ,  $[1, -1]$  e  $[1, 1]$ . Já para o banco de dados BCI *Competition* II a camada de saída foi definida com apenas um neurônio, em que os valores 1 e -1 representam as duas classes.

O critério de parada da RNA que foi utilizado para todos os bancos de dados e para os dois algoritmos de treinamento foi o número máximo de épocas e o erro de resposta da rede. O número máximos de épocas foi definido com o valor de 1000 épocas e o erro de resposta da rede foi definido com o valor de  $10^{-7}$ .

Com a RNA configurada, foi feita a preparação dos dados para serem classificados. Os dados foram misturados por um vetor de números aleatórios e esses dados foram normalizados. Após a normalização é aplicada a validação cruzada, da mesma forma que foi feita para o k-nn, e a classificação dos dados. O resultado desse processo é a média de acerto para cada um dos neurônios e seu desvio padrão, esses dados foram demonstrados em tabelas e gráficos de barras.

A representação da RNA para os dois algoritmos de treinamento está descrita na Figura 21. Onde as entradas C1, C2, C3 e C4 são as colunas do banco de dados de Fisher que possuem as dimensões das pétalas e sépalas das três espécies de flores de Iris. As saídas Y1 e Y2 estão representadas na Tabela 1.

Os métodos de configuração e preparação dos dados da RNA são os mesmos para os algoritmos de treinamento LM e RB. O Anexo H apresenta o código que efetuou essa

Figura 21 – Representação da RNA para o banco de dados de Fisher (Elaborado pelo autor).

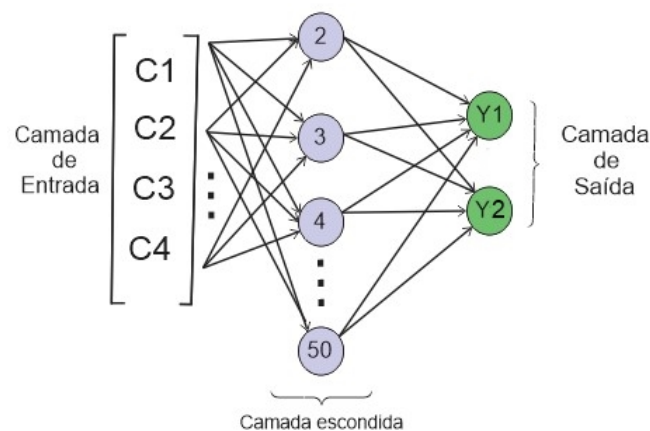


Tabela 1 – Representação das saídas da RNA para o banco de dados de Fisher (Elaborado pelo autor).

Y1	Y2	Espécie
-1	-1	Setosa
1	1	Versicolor
1	-1	Virginica

etapa para a RNA com algoritmo de treinamento LM e o Anexo I apresenta o código que efetuou essa etapa para a RNA com algoritmo de treinamento RB.

### 3.4 Classificação dos dados BCI *competition II*

#### 3.4.1 Classificação dos dados BCI *competition II* com k-vizinhos mais próximos

A configuração da função que representa o k-nn para a classificação do banco de dados BCI *competition II* foi a mesma utilizada para a análise dos classificadores com o banco de dados de Fisher. A diferença neste caso é o tratamento dos dados antes da classificação. Estes dados foram filtrados pelo CAR e separados em duas partes, antes da imaginação da imaginação da atividade motora e durante a imaginação da atividade motora. Utilizando os eletrodos C3 e C4 durante a atividade cerebral que são C3M2 e C4M2. As dimensões de C3M2 e C4M2 são 449 linhas por 140 colunas, as linhas representam o EEG e as 140 colunas as repetições do experimento. A informação desses sinais está contida na frequência de 8 a 12 Hz representando o ritmo  $\mu$  e também na frequência de 14

a 30 Hz representando o ritmo  $\beta$ . Estudos recentes demonstraram a dissociação entre os ritmos  $\mu$  e  $\beta$  do eletroencefalograma (EEG) durante tarefas de imagens motoras (KIM et al., 2016). Assim, gerando mais dados para serem aplicados a entrada da RNA. Podemos observar na Figura 19 que a imaginação de movimento da mão esquerda não alterou a banda  $\beta$ , mas a imaginação de movimento da mão direita alterou e isso pode ser usado como entrada para a RNA. Assim, foram selecionados os pontos do EEG que representam essas frequências. Utilizando um vetor que gera números aleatórios, foi feita a mistura das colunas de C3M2 e C4M2. Em seguida foram feitas as médias de C3M2 e C4M2, isso foi feito pois os sinais de EEG são diferentes em cada uma das 140 repetições. Com isso foi criado um vetor  $r_q$  contendo as médias de C3M2 e C4M2, com dimensões de 4 linhas e 140 colunas. Esses dados foram normalizados e divididos em 4 partes para realizar a validação cruzada. Utilizando 1/4 dos dados para classificar e 3/4 para treinar. Os dados divididos são referentes às colunas do vetor  $r_q$ . Com isso os dados são enviados para a classificação. A classificação se repete por 4 vezes completando a validação cruzada. Com os 4 resultados obtidos na classificação, foi feita a média e o desvio padrão dos mesmos, estes resultados foram demonstrados em uma tabela e em um gráfico de barras. O código que efetua essa atividade está presente no Anexo J.

### 3.4.2 Classificação do BCI *competition II* com redes neurais artificiais

Os passos utilizados no k-nn para se obter o vetor  $r_q$ , são repetidos para a RNA. Antes de enviar  $r_q$  para a classificação, foram ajustados os valores dos rótulos, pois o vetor  $y_{train}$  possui valores entre 1 e 2, e as funções de ativação da RNA possuem intervalos entre -1 e 1, assim subtraímos 1,5 de  $y_{train}$ , e utilizando a função *round* do *software* Matlab para arredondar o valor para o primeiro número inteiro. Como o resultado dessa subtração foi -0,5 e 0,5, os valores foram arredondados para -1 e 1. Com os dados de  $r_q$  e os rótulos ajustados, foi feita a normalização dos valores de  $r_q$ . Em seguida foi efetuada a validação cruzada da mesma forma que foi feita para o k-nn.

A RNA foi definida com uma única camada escondida, com o número de neurônios variando de 2 a 50 para os dois algoritmos de treinamento. Isso foi feito para testar o desempenho da rede conforme seus neurônios. A camada de saída foi definida com 1 neurônio, pois, para esse caso existem duas atividades, assim, a saída vai ter valores entre -1 e 1. As entradas da RNA são as quatro linhas do vetor  $r_q$ . Com isso foi efetuada a classificação dos dados. O resultados desse processo é a média de acertos para cada um



dos neurônios e seu desvio padrão, esses dados foram demonstrados em tabelas e gráficos de barras.

A representação da RNA para os dois algoritmos de treinamento está descrita na Figura 22. Onde as entradas  $\mu_{C3}$  e  $\mu_{C4}$  são os valores médios das amplitudes no intervalo de frequências entre 8 e 12 Hz dos canais C3 e C4 do EEG presente no banco de dados BCI *competition II*. E  $\beta_{C3}$  e  $\beta_{C4}$  são os valores médios das amplitudes no intervalo de frequências entre 14 e 30 Hz dos canais C3 e C4 do EEG presente no banco de dados BCI *competition II*. A saída Y1 é representada na Tabela 2. Os espectros de frequências foram calculados para cada canal através da FFT utilizando 896 amostras. Então, obteve-se uma resolução de 7 amostras para cada um hertz (448/64), já que o espectro unilateral compreendeu 448 amostras e a taxa de amostragem dos sinais foi de 128 Hz, sendo a frequência máxima medida limitada a 64 Hz. Então, a amplitude da banda  $\mu$  de um determinado canal foi calculada como o valor médio das posições entre 57 e 85 no vetor obtido da FFT já que  $57/7 = 8,14$  Hz e  $85/7 = 12,14$  Hz. A amplitude da banda  $\beta$  foi calculada como o valor médio das posições entre 113 e 211 no vetor obtido da FFT já que  $113/7 = 16,14$  Hz e  $211/7 = 30,14$  Hz.

Figura 22 – Representação da RNA para o banco de dados BCI *competition II* (Elaborado pelo autor).

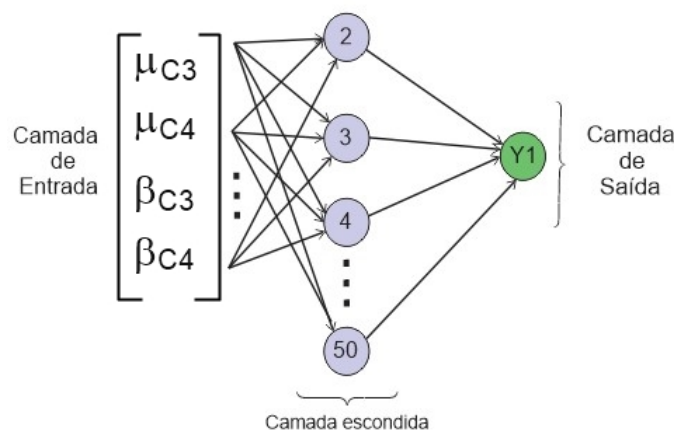


Tabela 2 – Representação das saídas da RNA para o banco de dados BCI *competition II* (Elaborado pelo autor).

Y1	Tarefas
1	Mão direita
-1	Mão esquerda

Os métodos de configuração e preparação dos dados da RNA são os mesmos para

o algoritmo de treinamento Levenberg-Marquardt e o algoritmo de treinamento regularização bayesiana. O Anexo K apresenta o código que efetuou essa etapa com o algoritmo de treinamento Levenberg-Marquardt e o Anexo L apresenta o código que efetuou essa etapa com o algoritmo de treinamento regularização bayesiana.

### **3.5 Classificação dos dados BCI *competition* III**

#### **3.5.1 Classificação do BCI *competition* III com k-vizinhos mais próximos**

Para esse banco de dados o k-nn foi configurado com o número de vizinhos variando de 1 a 25, devido ao número de amostras de treinamento. Após a configuração do k-nn, fez-se a seleção dos dados para serem enviados aos classificadores. A informação de interesse presente nos dados salvos do banco de dados BCI *competition* III estão situadas no intervalo de frequência de 8 a 12 Hz e 14 a 30 Hz, selecionando apenas os pontos do EEG que representam esses intervalos, foram criados 3 vetores. Esses vetores foram separados pelas atividades mentais, sendo compostos pelas médias dos sinais dos eletrodos C3, C4 e F7, o vetor ME contém as informações da atividade de mover a mão esquerda, o vetor MD contém as informações da atividade de mover a mão direita e o vetor P contém as informações da atividade de imaginar palavras. Com esses vetores foi criado um arranjo denominado EEG com as seguintes dimensões, 6 linhas e 36 colunas, as linhas representam os sinais dos três eletrodos nos intervalos de frequência  $\mu$  e  $\beta$ , e as colunas representam as repetição das atividades a serem classificadas. Com isso foi criado um vetor com os rótulos, atribuindo o valor de 1 para a atividade de imaginação de palavra, -1 para a imaginação do movimento da mão esquerda e 0 para a imaginação da mão direita. Após isso foi utilizado um vetor de números aleatórios para fazer com que os dados pertencentes ao arranjo EEG e seus alvos sejam apresentados ao classificador de forma aleatória, assim foi criado o arranjo rq. O arranjo rq foi normalizado e dividido em 4 partes para realizar a validação cruzada. Utilizando 1/4 dos dados para classificar e 3/4 para treinar. Os dados divididos são referentes às colunas do arranjo rq. Com isso os dados são enviados para a classificação.

Com os dados prontos fez-se a classificação dos dados, os resultados obtidos são a média de acertos e o seu desvio padrão que foram representados em uma tabela e um gráfico de barras. O código que apresenta essa etapa está descrito no Anexo M.

### 3.5.2 Classificação do BCI *competition* III com redes neurais artificiais

Como feito para o k-nn foi criado o arranjo  $r_q$  com dimensões de 6 linhas representando os 3 eletrodos em seus diferentes intervalos de frequência e 36 colunas que representam as repetição das atividades a serem classificadas. Com isso foi criado um vetor com os rótulos, atribuindo o valor de [1 -1] para a atividade de imaginação de palavra, [-1 -1] para a imaginação do movimento da mão esquerda e [1 1] para a imaginação da mão direita. Após isso foi utilizado um vetor de números aleatórios para fazer com que os dados pertencentes ao arranjo  $r_q$  e seus rótulos sejam apresentados ao classificador de forma aleatória. Com isso foi feita a normalização dos dados.

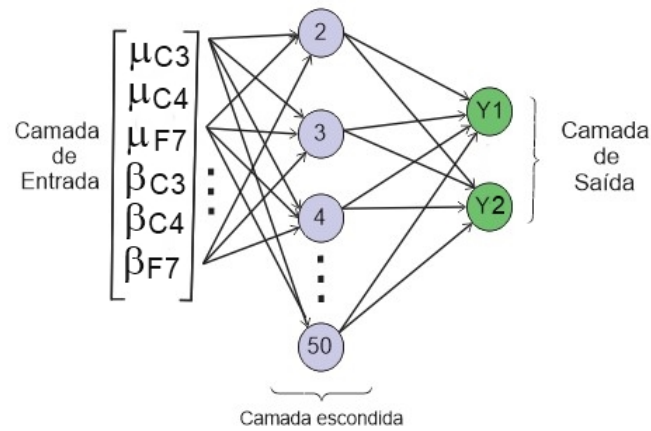
Com os dados prontos fez-se a validação cruzada da mesma forma que foi feito para o classificador k-nn, e assim foi feita a classificação dos dados, os resultados obtidos são a média de acertos e o seu desvio padrão que foram demonstrados em tabelas e em gráficos de barras.

A representação da RNA para os dois algoritmos de treinamento está descrita na Figura 23. Onde as entradas  $\mu C3$ ,  $\mu C4$  e  $\mu F7$  são os valores médios das amplitudes no intervalo de frequências entre 8 e 12 Hz dos canais C3, C4 e F7 do EEG presente no banco de dados BCI *competition* III. E  $\beta C3$ ,  $\beta C4$  e  $\beta F7$  são os valores médios das amplitudes no intervalo de frequências entre 14 e 30 Hz dos canais C3, C4 e F7 do EEG presente no banco de dados BCI *competition* III. As saídas Y1 e Y2 estão representadas na Tabela 3. Neste banco de dados os espectros de frequências foram calculados para cada canal através da FFT utilizando 2560 amostras. Então, obteve-se uma resolução de 5 amostras para cada um hertz (1280/256), já que o espectro unilateral compreendeu 1280 amostras e a taxa de amostragem dos sinais foi de 512 Hz, sendo a frequência máxima medida limitada a 256 Hz. Então, a amplitude da banda  $\mu$  de um determinado canal foi calculada como o valor médio das posições entre 40 e 61 no vetor obtido da FFT já que  $40/5 = 8$  Hz e  $61/5 = 12,2$  Hz. A amplitude da banda  $\beta$  foi calculada como o valor médio das posições entre 70 e 150 no vetor obtido da FFT já que  $70/5 = 14$  Hz e  $150/5 = 30$  Hz.

Tabela 3 – Representação das saídas da RNA para o banco de dados BCI *competition* III (Elaborado pelo autor).

Y1	Y2	Tarefas
1	1	Mão direita
-1	-1	Mão esquerda
1	-1	Palavras

Figura 23 – Representação da RNA para o banco de dados BCI *competition III* (Elaborado pelo autor).



Os métodos de configuração e preparação dos dados da rede são os mesmos para o algoritmo de treinamento Levenberg-Marquardt e o algoritmo de treinamento regularização bayesiana. O Anexo N apresenta o código que efetuou essa etapa para a RNA com o algoritmo de treinamento Levenberg-Marquardt e o Anexo O apresenta o código que efetuou essa etapa para a RNA com o algoritmo de treinamento regularização bayesiana.

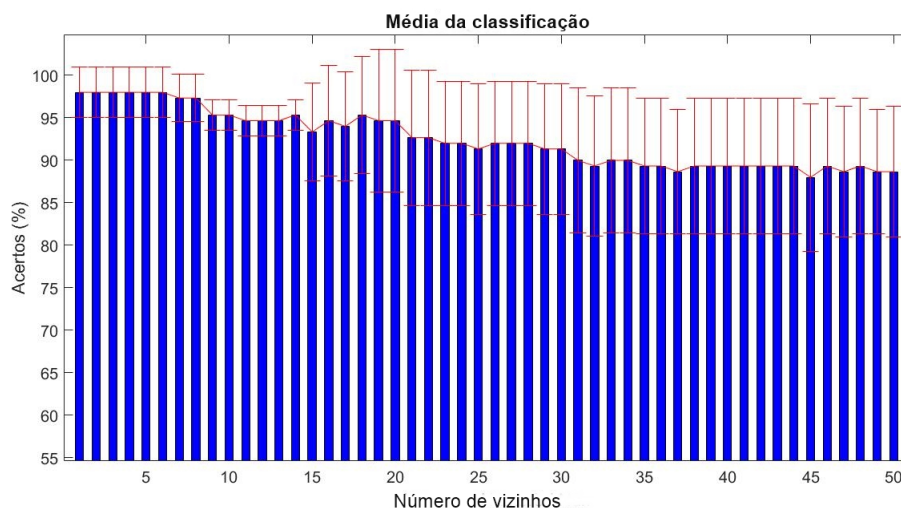
## 4 RESULTADOS

Neste capítulo foram apresentados os resultados obtidos nas simulações realizadas. A primeira seção deste capítulo representa a etapa de análise dos classificadores. Essa análise foi realizada com o banco de dados de Fisher. Na seção 4.2 e 4.3 foram apresentados respectivamente os resultados das classificações dos bancos de dados BCI *competition* II e III. Os resultados foram apresentados em tabelas e gráficos de barras para todos os classificadores.

### 4.1 Resultado da análise dos classificadores com banco de dados Iris de Fisher

A análise dos classificadores foi feita com o banco de dados Iris de Fisher. Com o código do Anexo G executado no *software* Matlab obtemos os resultados de média de acerto e desvio padrão para essa etapa que estão presentes na Tabela 4 e no gráfico da Figura 24 para o classificador k-nn.

Figura 24 – Resultado da análise do classificador k-nn com banco de dados de Fisher (Elaborado pelo autor).



Pode-se observar que o classificador k-nn configurado com número de vizinhos mais próximos no intervalo de 1 a 6 apresenta 98% de média de acerto e 2,9814 de desvio padrão, sendo essa métrica a mais elevada para o classificador k-nn nesta etapa de análise. Ainda, pode-se observar que o aumento do número de vizinhos faz com que a taxa de acerto seja reduzida em comparação com a utilização de um número menor de vizinhos. Esse resultado valida o uso do classificador k-nn, pois, garante que o mesmo não está

Tabela 4 – Resultado da análise do classificador k-nn com banco de dados de Fisher (Elaborado pelo autor).

Número de vizinhos	Média de acertos (%)	Desvio padrão	Número de vizinhos	Média de acertos (%)	Desvio padrão
1	98	2,9814	26	92	7,3030
2	98	2,9814	27	92	7,3030
3	98	2,9814	28	92	7,3030
4	98	2,9814	29	91,3333	7,6739
5	98	2,9814	30	91,3333	7,6739
6	98	2,9814	31	90	8,4984
7	97,3333	2,7889	32	89,3333	8,2999
8	97,3333	2,7889	33	90	8,4984
9	95,3333	1,8257	34	90	8,4984
10	95,3333	1,8257	35	89,3333	7,9582
11	94,6667	1,8257	36	89,3333	7,9582
12	94,6667	1,8257	37	88,6667	7,3030
13	94,6667	1,8257	38	89,3333	7,9582
14	95,3333	1,8257	39	89,3333	7,9582
15	93,3333	5,7735	40	89,3333	7,9582
16	94,6667	6,4979	41	89,3333	7,9582
17	94	6,4118	42	89,3333	7,9582
18	95,3333	6,9121	43	89,3333	7,9582
19	94,6667	8,3666	44	89,3333	7,9582
20	94,6667	8,3666	45	88	8,6923
21	92,6667	7,9582	46	89,3333	7,9582
22	92,6667	7,9582	47	88,6667	7,6739
23	92	7,3030	48	89,3333	7,9582
24	92	7,3030	49	88,6667	7,3030
25	91,3333	7,6739	50	88,6667	7,6739

gerando resultados aleatórios. Um resultado aleatório neste caso que são analisados 3 rótulos diferentes seria uma taxa de acerto de 33,33%.

Com base no Anexo H, foi efetuada a análise da rede neural artificial com algoritmo de treinamento Levenberg-Marquardt. Os resultados para as etapas de treinamento e classificação, estão presentes nas Tabelas 5 e 6. Ainda, para demonstrar esses resultados foi feito um gráfico de barras com as médias de acertos e o seu respectivo desvio padrão, esse gráfico está presente na Figura 25. A análise da rede neural artificial com algoritmo de treinamento de regularização bayesiana foi feita de acordo com o código do Anexo I. Os resultados para as etapas de treinamento e classificação, estão presentes nas Tabelas 7 e 8. Ainda, para demonstrar esses resultados foi feito um gráfico de barras com as médias de acertos e o seu respectivo desvio padrão, esse gráfico está presente na Figura 26.

Da mesma forma que o classificador k-nn a RNA com os algoritmos de treinamento Levenberg-Marquardt (LM) e regularização bayesiana (RB) apresentaram médias de acerto acima de 33,3333%, assim, a RNA não está acertando de forma aleatória, com isso é válido o uso da RNA para classificar os bancos de dados.

Comparando os resultados obtidos, pode-se observar que tanto o classificador k-nn quanto a RNA com os algoritmos de treinamento LM e RB apresentaram médias de acerto superiores do que 33,3333%. Ainda, podemos dizer que na fase de treinamento a RNA com algoritmo de treinamento RB obteve resultados de média de acerto superiores do que a RNA com algoritmo de treinamento de LM. Porém, para a etapa de classificação

Figura 25 – Resultado da análise da rede neural artificial com algoritmo de treinamento Levenberg-Marquardt com banco de dados de Fisher (Elaborado pelo autor).

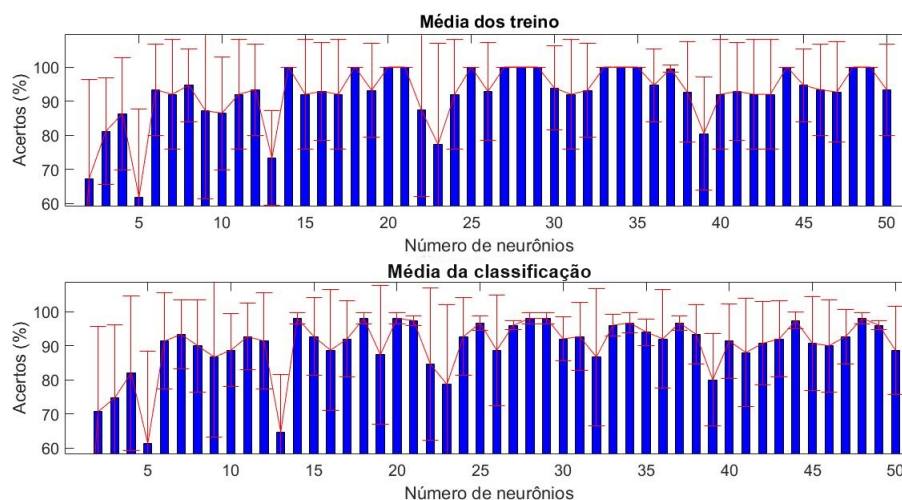


Tabela 5 – Resultado do treino para a análise da rede neural artificial com algoritmo de treinamento Levenberg-Marquardt com banco de dados de Fisher (Elaborado pelo autor).

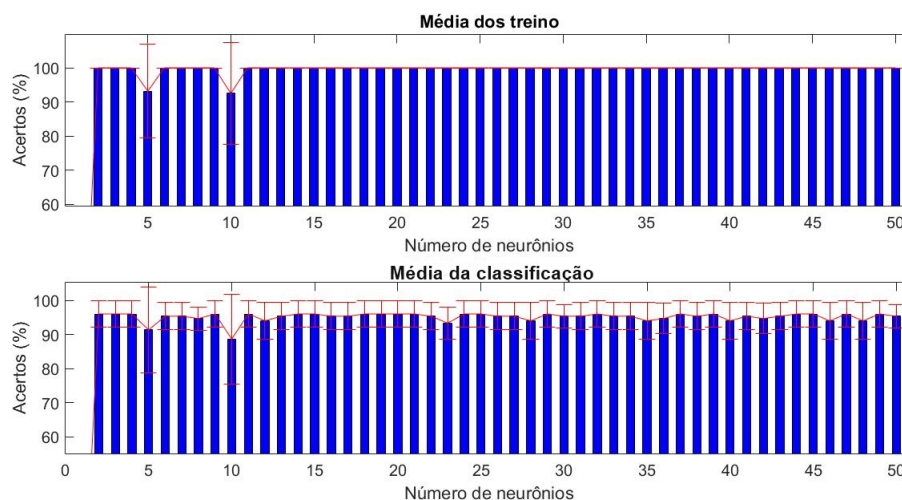
Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	92,8333	14,3333
2	67,3333	29,0756	27	100	0
3	81,1667	15,5956	28	100	0
4	86,3333	16,4029	29	100	0
5	61,8333	25,9080	30	93,8333	12,3333
6	93,3333	13,3333	31	92	16
7	92	16	32	93,1667	13,6667
8	94,6667	10,6667	33	100	0
9	87,1667	25,6667	34	100	0
10	86,5000	16,5529	35	100	0
11	92	16	36	94,6667	10,6667
12	93,3333	13,3333	37	99,5000	1
13	73,3333	13,7941	38	92,6667	14,6667
14	100	0	39	80,5000	16,4958
15	92	16	40	92	16
16	92,8333	14,3333	41	92,8333	14,3333
17	92	16	42	92	16
18	100	0	43	92	16
19	93,1667	13,6667	44	100	0
20	100	0	45	94,6667	10,6667
21	100	0	46	93,3333	13,3333
22	87,3333	25,3333	47	92,6667	14,6667
23	77,3333	29,6947	48	100	0
24	92	16	49	100	0
25	100	0	50	93,3333	13,3333

a RNA com algoritmo de treinamento LM e o k-nn obtiveram médias de acerto superior do que a RNA com algoritmo de treinamento RB. Assim, na etapa de análise a RNA com algoritmo de treinamento LM obteve média de acerto superior aos demais, obtendo uma taxa de acerto de 98% com 14, 18, 20, 28, 29 e 48 neurônios. Além disso, o desvio padrão para os neurônios citados anteriormente é de 1,6330, esse desvio é próximo de zero.

Tabela 6 – Resultado da classificação para a análise da rede neural artificial com algoritmo de treinamento Levenberg-Marquardt com banco de dados de Fisher (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	88,6667	16,1383
2	70,6667	21,5613	27	96	1,3333
3	74,6667	25,0688	28	98	1,6330
4	82	22,6667	29	98	1,6330
5	61,3333	27,0473	30	92	6,5320
6	91,3333	14,0791	31	92,6667	9,9778
7	93,3333	10,1105	32	86,6667	20,1108
8	90,0000	13,4990	33	96	3,2660
9	86,6667	23,3809	34	96,6667	2,9814
10	88,6667	10,6667	35	94	3,8873
11	92,6667	9,7525	36	92	14,3914
12	91,3333	14,0791	37	96,6667	2,1082
13	64,6667	16,9444	38	93,3333	8,6923
14	98	1,6330	39	80	13,4990
15	92,6667	11,4310	40	91,3333	10,8730
16	88,6667	17,7138	41	88	15,8605
17	92	11,0755	42	90,6667	12,1838
18	98	1,6330	43	92	11,0755
19	87,3333	20,3743	44	97,3333	2,4944
20	98	1,6330	45	90,6667	13,7275
21	97,3333	1,3333	46	90	13,4990
22	84,6667	22,3706	47	92,6667	8,0000
23	78,6667	23,2475	48	98	1,6330
24	92,6667	11,4310	49	96	1,3333
25	96,6667	2,1082	50	88,6667	12,9271

Figura 26 – Resultado da análise da rede neural artificial com algoritmo de treinamento regularização bayesiana com banco de dados de Fisher (Elaborado pelo autor).



## 4.2 Resultado da classificação do banco de dados BCI *competition II*

Com os classificadores analisados, passa-se para a fase de classificação. O primeiro banco de dados a ser classificado foi o BCI *competition II*. Nos Anexos J, K e L, estão respectivamente os códigos do k-nn, da RNA com algoritmo de treinamento LM e RB que foram utilizados para essa classificação. A Tabela 9 e a Figura 27 apresentam os dados da classificação com o classificador k-nn, essa tabela é composta por média de



Tabela 7 – Resultado do treino para a análise da rede neural artificial com algoritmo de treinamento regularização bayesiana com banco de dados de Fisher (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	100	0
2	100	0	27	100	0
3	100	0	28	100	0
4	100	0	29	100	0
5	93,1667	13,6667	30	100	0
6	100	0	31	100	0
7	100	0	32	100	0
8	100	0	33	100	0
9	100	0	34	100	0
10	92,5000	15,0000	35	100	0
11	100	0	36	100	0
12	100	0	37	100	0
13	100	0	38	100	0
14	100	0	39	100	0
15	100	0	40	100	0
16	100	0	41	100	0
17	100	0	42	100	0
18	100	0	43	100	0
19	100	0	44	100	0
20	100	0	45	100	0
21	100	0	46	100	0
22	100	0	47	100	0
23	100	0	48	100	0
24	100	0	49	100	0
25	100	0	50	100	0

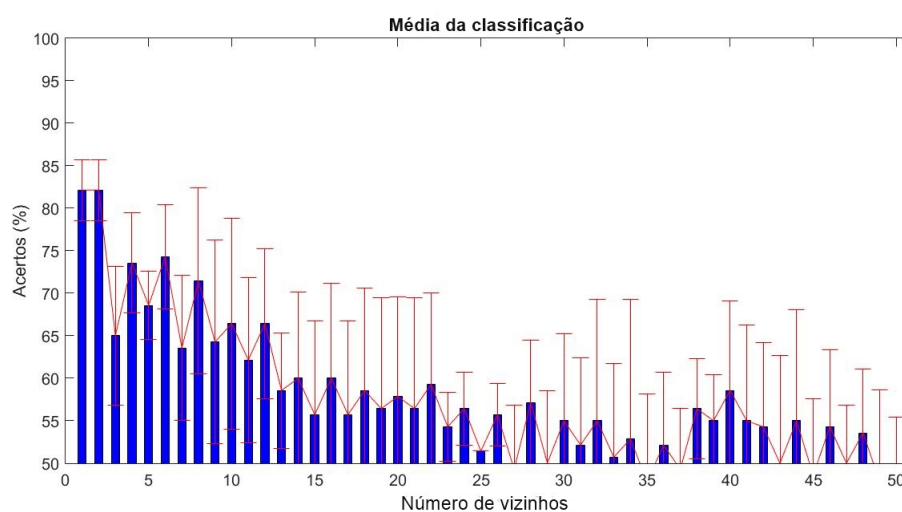
Tabela 8 – Resultado da classificação para a análise da rede neural artificial com algoritmo de treinamento regularização bayesiana com banco de dados de Fisher (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	95,3333	4
2	96	3,8873	27	95,3333	4
3	96	3,8873	28	94	5,3333
4	96	3,8873	29	96	3,8873
5	91,3333	12,5786	30	95	3,3993
6	95,3333	4	31	95	4
7	95,3333	4	32	96	3,8873
8	94,6667	3,3993	33	95,3333	4
9	96	3,8873	34	95,3333	4
10	88,6667	13,0979	35	94	5,3333
11	96	3,8873	36	94,6667	4,5216
12	94	5,3333	37	96,	3,8873
13	95,3333	4	38	95,3333	4
14	96	3,8873	39	96	3,8873
15	96	3,8873	40	94	5,3333
16	95,3333	4	41	95,3333	4
17	95,3333	4	42	94,6667	4,5216
18	96	3,8873	43	95,3333	4
19	96	3,8873	44	96	3,8873
20	96	3,8873	45	96	3,8873
21	96	3,8873	46	94	5,3333
22	95,3333	4	47	96	3,8873
23	93,3333	4,7140	48	94	5,3333
24	96	3,8873	49	96	3,8873
25	96	3,8873	50	95,3333	3,3993

acerto e desvio padrão.

Os resultados para as etapas de treinamento e classificação para a RNA com algoritmo de treinamento LM, estão presentes nas Tabelas 10 e 11. Ainda, para demonstrar esses resultados foi feito um gráfico de barras com as médias de acertos e o seu respec-

Figura 27 – Resultado da classificação com o classificador k-nn para o banco de dados BCI *competition II* (Elaborado pelo autor).



tivo desvio padrão, esse gráfico está presente na Figura 28. Da mesma forma que foi feito anteriormente, os resultados para a RNA com algoritmo de treinamento a RB para as etapas de treinamento e classificação, estão presentes nas Tabelas 12 e 13. Ainda, para demonstrar esses resultados foi feito um gráfico de barras com as médias de acertos e o seu respectivo desvio padrão, esse gráfico está presente na Figura 29.

Diferente do banco de Iris de Fisher, esse banco de dados possui um sinal de EEG, esse tipo de sinal é mais complexo para ser classificado, devido as suas características de baixa amplitude, assim, o mesmo é afetado por ruídos. Nesse banco foram classificadas duas atividades em dois rótulos, assim, considera-se que os resultados com média de acerto menores ou iguais a 50% são considerados aleatórios.

Com base nos resultados obtidos pode-se observar que todos os classificadores obtiveram médias de acertos superiores a 50%, exceto para a RNA com algoritmo de treinamento LM utilizando dois neurônios na camada escondida.

Com os resultados do classificador k-nn, pode-se observar que a média de acerto foi superior as demais para 1 e 2 vizinhos mais próximos obtendo 82,1429% de taxa de acerto com desvio padrão de 3,5952. Ainda, o classificador k-nn apresentou uma queda na taxa de acerto conforme o aumento do número de vizinhos. Em relação a RNA, para o algoritmo de treinamento LM a maior taxa de acerto foi de 79,2857% com desvio padrão de 4,2258, enquanto para o algoritmo de treinamento RB foi obtida a maior taxa de acerto de 72,1429% com desvio padrão de 3,7115, para os neurônios 40, 43, 44, 45, 46, 48 e 50.

Comparando os resultados pode-se observar que a RNA com algoritmo de treinamento LM obteve maior taxa de acerto na fase de treinamento para o número de neurônios

Tabela 9 – Resultado da classificação com o classificador k-nn para o banco de dados BCI *competition* II (Elaborado pelo autor).

Número de vizinhos	Média de acertos (%)	Desvio padrão	Número de vizinhos	Média de acertos (%)	Desvio padrão
1	82,1429	3,5952	26	55,7143	3,6886
2	82,1429	3,5952	27	49,2857	7,5142
3	65	8,2065	28	57,1429	7,3771
4	73,5714	5,8902	29	50	8,5714
5	68,5714	4,0406	30	55	10,2685
6	74,2857	6,1721	31	52,1429	10,2685
7	63,5714	8,5317	32	55	14,2619
8	71,4286	10,9420	33	50,7143	11,0349
9	64,2857	12,0091	34	52,8571	16,4130
10	66,4286	12,4267	35	47,8571	10,2685
11	62,1429	9,7241	36	52,1429	8,5317
12	66,4286	8,8448	37	49,2857	7,1429
13	58,5714	6,8014	38	56,4286	5,8902
14	60	10,1686	39	55	5,4085
15	55,7143	11,0657	40	58,5714	10,5624
16	60	11,1879	41	55	11,2788
17	55,7143	11,0657	42	54,2857	9,8974
18	58,5714	12,0091	43	50	12,6706
19	56,4286	13,0671	44	55	13,0671
20	57,8571	11,7514	45	48,5714	9,0351
21	56,4286	13,0671	46	54,2857	9,0351
22	59,2857	10,7855	47	50	6,8014
23	54,2857	4,0406	48	53,5714	7,5142
24	56,4286	4,2857	49	47,8571	10,7855
25	51,4286	0	50	50	5,4710

na camada escondida maior que 3 em relação ao algoritmo de treinamento RB. A taxa de acerto para a etapa de classificação também foi superior para o algoritmo de treinamento LM com 79,2857% de acerto em relação ao do algoritmo de treinamento RB que obteve 72,1429% como maior taxa de acerto. Porém, o classificador k-nn foi o que obteve maior taxa de acerto em relação aos outros com média de 82,1429%.

Comparando o resultado obtido com o classificador k-nn com o apresentado por Benevides (2007) utilizando RNA para o banco de dados BCI *competition* II foi superior, tendo como média de acerto 93,3333%. Porém, no trabalho comparado foram utilizadas outras técnicas de pré-processamento dos dados.

### 4.3 Resultado da classificação do banco de dados BCI *competition* III

O banco de dados BCI *competition* III, apresenta três tarefas mentais, gerando três rótulos a serem classificados. Assim como o banco de dados de Fisher, as médias de acerto menores ou iguais a 33,3333%, são considerados resultados aleatórios.

Nos Anexos J, K e L, estão respectivamente os códigos do k-nn, RNA com algoritmo de treinamento LM e RNA com algoritmo de treinamento RB que foram utilizados para essa classificação. A Tabela 14 e a Figura 30 apresentam os dados da classificação com o k-nn, essa tabela é composta por média de acertos e desvio padrão. Os resultados

Figura 28 – Resultados da classificação do banco de dados BCI *competition* II com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor).

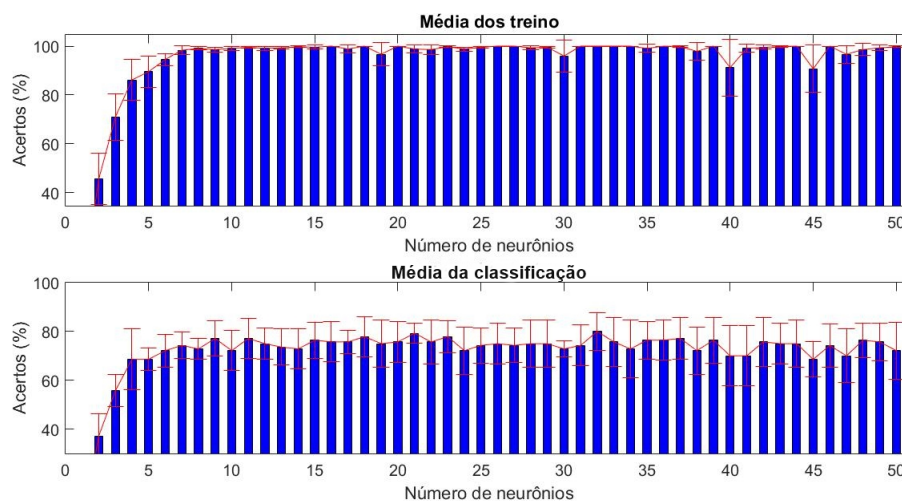


Tabela 10 – Resultados do treino para classificação do banco de dados BCI *competition* II com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	100	0
2	45,4762	10,5059	27	100	0
3	70,9524	9,5595	28	99,2857	0,7897
4	86,1905	8,2341	29	99,5238	0,4762
5	89,5238	6,4594	30	95,9524	6,4725
6	94,5238	2,4629	31	100	0
7	98,3333	1,8288	32	100	0
8	99,0476	0,6734	33	100	0
9	98,5714	1,0648	34	100	0
10	99,0476	0,9524	35	99,0476	1,6496
11	99,5238	0,4762	36	100	0
12	99,0476	0,9524	37	99,7619	0,4124
13	99,2857	0,4124	38	97,8571	3,7115
14	99,7619	0,4124	39	100	0
15	99,5238	0,8248	40	91,1905	11,7297
16	100	0	41	99,0476	1,6496
17	98,8095	1,5613	42	99,5238	0,8248
18	100	0	43	99,7619	0,4124
19	96,6667	4,6899	44	100	0
20	100	0	45	90,7143	9,7677
21	98,8095	1,5613	46	100	0
22	98,5714	1,9634	47	96,6667	3,5952
23	99,7619	0,4124	48	98,5714	2,4744
24	98,8095	0,7897	49	99,2857	1,2372
25	99,5238	0,4762	50	99,7619	0,4124

para as etapas de treinamento e teste para a RNA com algoritmo de treinamento LM, estão presentes nas Tabelas 15 e 16. Ainda, para demonstrar esses resultados foi feito um gráfico de barras com as médias de acertos e o seu respectivo desvio padrão, esse gráfico está presente na Figura 31. Da mesma forma que foi feito anteriormente, os resultados utilizando RNA com algoritmo de treinamento RB para as etapas de treinamento e teste, estão presentes nas Tabelas 17 e 18. Ainda, para demonstrar esses resultados foi feito um

Tabela 11 – Resultados da classificação do do banco de dados BCI *competition II* com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	75	8,4213
2	37,1429	9,2582	27	74,2857	6,9985
3	55,7143	6,5465	28	75	9,5565
4	68,5714	12,4540	29	75	9,7677
5	68,5714	4,5175	30	72,8571	3,1944
6	72,1429	6,8139	31	74,2857	8,3299
7	74,2857	5,3452	32	80	7,8246
8	72,8571	4,2857	33	75,7143	10
9	77,1429	7,2843	34	72,8571	11,8666
10	72,1429	8,1754	35	76,4286	7,6599
11	77,1429	8,3299	36	76,4286	8,1754
12	75	6,5075	37	77,1429	8,5714
13	73,5714	7,3886	38	72,1429	9,7677
14	72,8571	8,2065	39	76,4286	9,3405
15	76,4286	7,3886	40	70	12,3718
16	75,7143	8,2065	41	70	12,3718
17	75,7143	4,7380	42	75,7143	10
18	77,8571	8,1754	43	75	8,4213
19	75	9,7677	44	75	9,5565
20	75,7143	8,4515	45	68,5714	7,2843
21	79,2857	4,2258	46	74,2857	8,8063
22	75,7143	9,1473	47	70	10,9731
23	77,8571	6,5075	48	76,4286	7,1071
24	72,1429	9,7677	49	75,7143	7,6931
25	74,2857	7,2843	50	72,1429	11,6715

Figura 29 – Resultados da classificação do banco de dados BCI *competition II* com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).

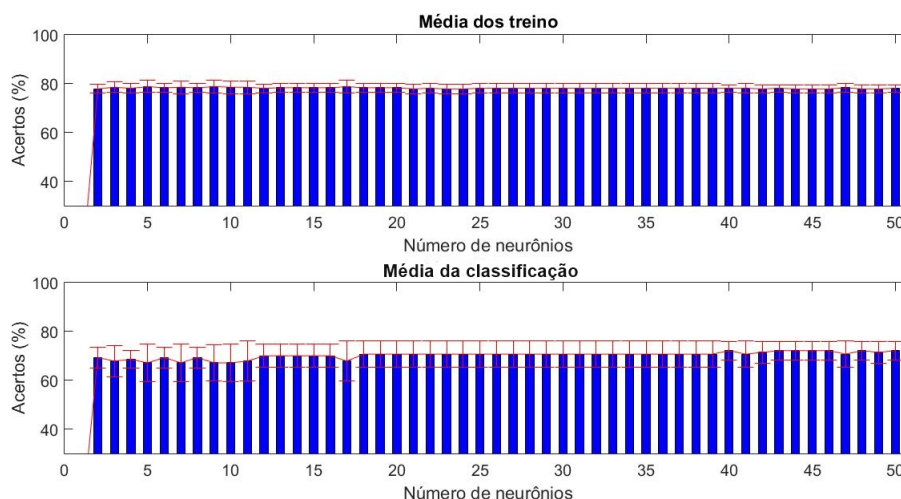


gráfico de barras com as médias de acertos e o seu respectivo desvio padrão, esse gráfico está presente na Figura 32.

Com base nos resultados obtidos pode-se observar que todos os classificadores obtiveram médias de acertos superiores a 33,3333%, exceto para a RNA com algoritmo de treinamento LM utilizando 16 neurônios na camada escondida e para o classificador k-nn com 14, 19 e 24 vizinhos.

Tabela 12 – Resultados do treino para classificação do banco de dados BCI *competition II* com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	78,0952	2,0203
2	77,8571	1,8288	27	78,0952	2,0203
3	78,5714	2,0757	28	78,0952	2,0203
4	78,0952	2,0203	29	78,0952	2,0203
5	78,8095	2,4629	30	78,0952	2,0203
6	78,3333	1,7003	31	78,0952	2,0203
7	78,5714	2,5644	32	78,0952	2,0203
8	78,3333	1,7003	33	78,0952	2,0203
9	78,8095	2,5533	34	78,0952	2,0203
10	78,5714	2,5644	35	78,0952	2,0203
11	78,5714	2,5644	36	78,0952	2,0203
12	78,0952	1,7817	37	78,0952	2,0203
13	78,3333	1,8288	38	78,0952	2,0203
14	78,3333	1,8288	39	78,0952	2,0203
15	78,3333	1,8288	40	78,0952	1,5058
16	78,3333	1,8288	41	78,0952	2,0203
17	78,8095	2,5533	42	77,8571	1,7003
18	78,3333	1,8288	43	78,0952	1,5058
19	78,3333	1,8288	44	77,8571	1,7003
20	78,3333	1,8288	45	77,8571	1,7003
21	77,8571	1,9489	46	77,8571	1,7003
22	78,0952	2,0203	47	78,3333	1,8288
23	77,8571	1,9489	48	77,8571	1,7003
24	77,8571	1,9489	49	77,8571	1,7003
25	78,0952	2,0203	50	78,0952	1,5058

Tabela 13 – Resultados da classificação do do banco de dados BCI *competition II* com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	70,7143	5,4865
2	69,2857	4,2258	27	70,7143	5,4865
3	67,8571	6,5075	28	70,7143	5,4865
4	68,5714	3,4993	29	70,7143	5,4865
5	67,1429	7,6931	30	70,7143	5,4865
6	69,2857	4,2258	31	70,7143	5,4865
7	67,1429	7,6931	32	70,7143	5,4865
8	69,2857	4,2258	33	70,7143	5,4865
9	67,1429	7,4231	34	70,7143	5,4865
10	67,1429	7,6931	35	70,7143	5,4865
11	67,8571	8,1754	36	70,7143	5,4865
12	70	4,7380	37	70,7143	5,4865
13	70	4,7380	38	70,7143	5,4865
14	70	4,7380	39	70,7143	5,4865
15	70	4,7380	40	72,1429	3,7115
16	70	4,7380	41	70,7143	5,4865
17	67,8571	8,1754	42	71,4286	4,5175
18	70,7143	5,4865	43	72,1429	3,7115
19	70,7143	5,4865	44	72,1429	3,7115
20	70,7143	5,4865	45	72,1429	3,7115
21	70,7143	5,4865	46	72,1429	3,7115
22	70,7143	5,4865	47	70,7143	5,4865
23	70,7143	5,4865	48	72,1429	3,7115
24	70,7143	5,4865	49	71,4286	4,5175
25	70,7143	5,4865	50	72,1429	3,7115

Com os resultados do classificador k-nn, pode-se observar que a média de acerto foi superior as demais para 1 e 2 vizinhos mais próximos obtendo 66,6667% de taxa de acerto com desvio padrão de 15,7135. Ainda, o classificador k-nn apresentou uma queda na taxa de acerto conforme o aumento do número de vizinhos. Em relação a RNA, para o

algoritmo de treinamento LM a maior taxa de acerto foi de 66,6667% com desvio padrão de 7,8567, para 45 e 47 neurônios na camada escondida. E para o algoritmo de treinamento RB foi obtida a maior taxa de acerto de 63,8889% com desvio padrão de 4,8113, para 4 neurônios na camada escondida. Comparando os resultados dos classificadores pode-se observar que a maior taxa de acerto foi obtida com o k-nn e com a RNA com algoritmo de treinamento LM, porem, o resultado do desvio padrão para o k-nn é superior ao desvio padrão obtido com a RNA com algoritmo de treinamento LM. Assim, considerando a maior taxa de acerto e o menor desvio padrão pode-se considerar que a RNA com algoritmo de treinamento RB possui um resultado melhor que os demais com taxa de acerto de 63,8889% com desvio padrão de 4,8113.

Comparando o resultado obtido com a RNA com algoritmo de treinamento RB com o apresentado por Benevides et al (2012) e pelos resultados finais da competição BCI III (BLANKERTZ, 2005). Esses resultados foram obtidos classificando os dados de EEG simulados em tempo real. A comparação dos dados esta presente na Tabela 19. Com base na comparação pode-se observar que o resultado obtido nesse trabalho foi inferior apenas ao resultado obtido por Benevides et al (2012).

Figura 30 – Resultado da classificação com o classificador k-nn para o banco de dados BCI *competition* III (Elaborado pelo autor).

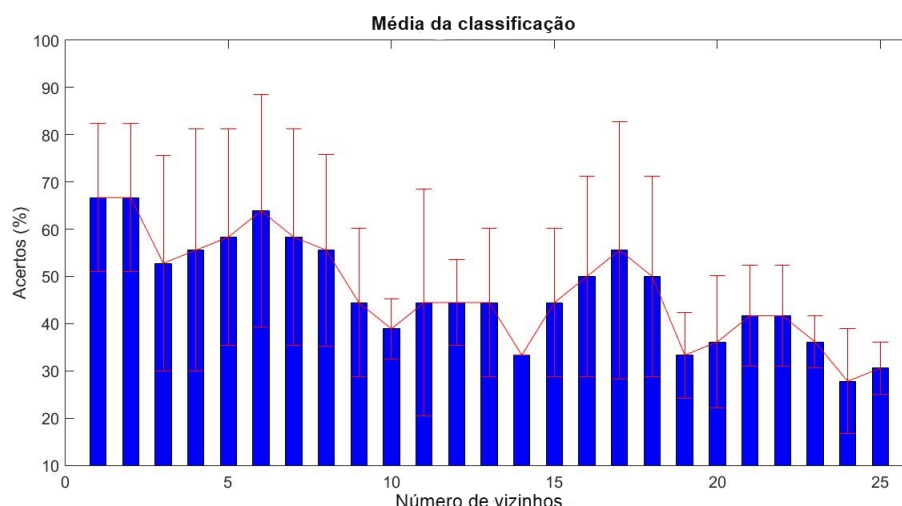


Figura 31 – Resultados da classificação do banco de dados BCI *competition III* com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor).

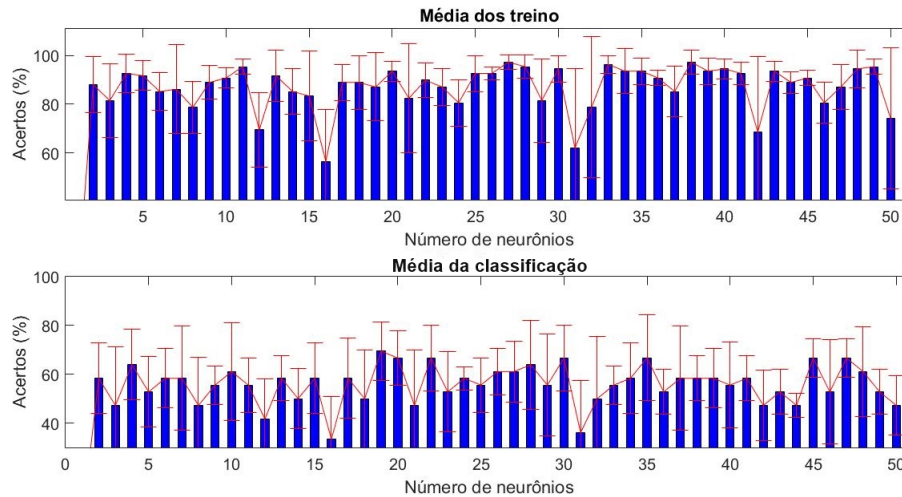


Figura 32 – Resultados da classificação do banco de dados BCI *competition III* com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).

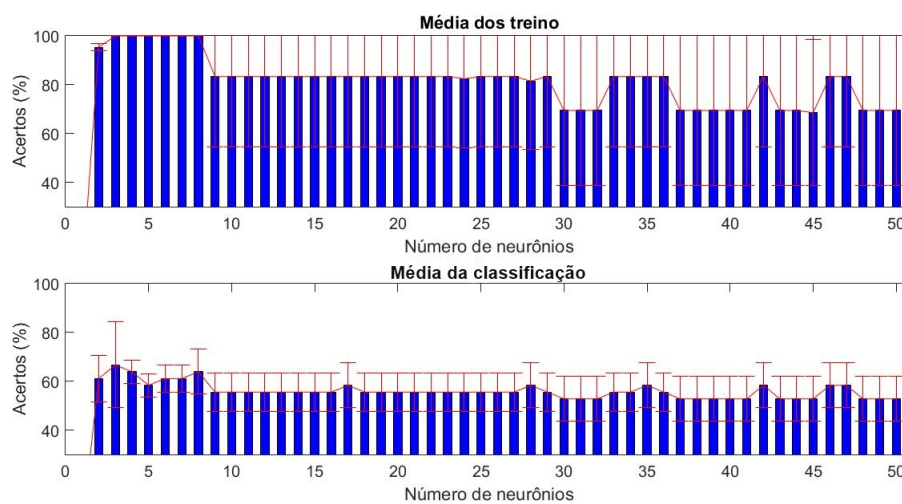




Tabela 14 – Resultado da classificação com o classificador k-nn para o banco de dados BCI *competition* III (Elaborado pelo autor).

Número de vizinhos	Média de acertos (%)	Desvio padrão
1	66,6667	15,7135
2	66,6667	15,7135
3	52,7778	22,9061
4	55,5556	25,6600
5	58,3333	22,9061
6	63,8889	24,6373
7	58,3333	22,9061
8	55,5556	20,2860
9	44,4444	15,7135
10	38,8889	6,4150
11	44,4444	24,0027
12	44,4444	9,0722
13	44,4444	15,7135
14	33,3333	0
15	44,4444	15,7135
16	50	21,2762
17	55,5556	27,2166
18	50	21,2762
19	33,3333	9,0722
20	36,1111	13,9812
21	41,6667	10,6381
22	41,6667	10,6381
23	36,1111	5,5556
24	27,7778	11,1111
25	30,5556	5,5556

Tabela 15 – Resultados do treino para classificação do banco de dados BCI *competition III* com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	92,5926	2,6189
2	87,9630	11,5277	27	97,2222	3,0709
3	81,4815	15,0445	28	95,3704	4,8113
4	92,5926	7,8567	29	81,4815	17,1734
5	91,6667	6,0717	30	94,4444	5,5556
6	85,1852	7,8567	31	62,0370	32,4339
7	86,1111	18,2151	32	78,7037	28,9713
8	78,7037	10,5977	33	96,2963	3,7037
9	88,8889	6,9290	34	93,5185	9,2128
10	90,7407	4,1409	35	93,5185	5,4779
11	95,3704	3,0709	36	90,7407	3,2075
12	69,4444	15,1298	37	85,1852	10,4757
13	91,6667	10,5977	38	97,2222	4,8113
14	85,1852	9,4426	39	93,5185	5,4779
15	83,3333	18,4257	40	94,4444	4,1409
16	56,4815	21,3365	41	92,5926	4,5361
17	88,8889	7,4074	42	68,5185	31,0427
18	88,8889	11,1111	43	93,5185	4,0360
19	87,0370	13,9812	44	88,8889	4,5361
20	93,5185	4,0360	45	90,7407	3,2075
21	82,4074	22,2800	46	80,5556	8,4356
22	89,8148	7,1122	47	87,0370	9,2593
23	87,0370	7,6354	48	94,4444	7,6354
24	80,5556	9,5779	49	95,3704	3,0709
25	92,5926	7,4074	50	74,0741	28,9269

Tabela 16 – Resultados da classificação do do banco de dados BCI *competition III* com rede neural artificial com algoritmo de treinamento Levenberg-Marquardt (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	61,1111	9,6225
2	58,3333	14,4338	27	61,1111	12,4226
3	47,2222	24,0563	28	63,8889	18,2151
4	63,8889	14,4338	29	55,5556	20,7870
5	52,7778	14,4338	30	66,6667	13,6083
6	58,3333	12,1081	31	36,1111	21,3365
7	58,3333	21,3365	32	50	25,4588
8	47,2222	19,8373	33	55,5556	7,8567
9	55,5556	7,8567	34	58,3333	14,4338
10	61,1111	20,0308	35	66,6667	17,5682
11	55,5556	11,1111	36	52,7778	9,2128
12	41,6667	16,4336	37	58,3333	21,3365
13	58,3333	9,2128	38	58,3333	9,2128
14	50	12,4226	39	58,3333	12,1081
15	58,3333	14,4338	40	55,5556	17,5682
16	33,3333	17,5682	41	58,3333	9,2128
17	58,3333	16,4336	42	47,2222	14,4338
18	50,0000	20,0308	43	52,7778	9,2128
19	69,4444	12,1081	44	47,2222	4,8113
20	66,6667	11,1111	45	66,6667	7,8567
21	47,2222	22,7371	46	52,7778	21,3365
22	66,6667	13,6083	47	66,6667	7,8567
23	52,7778	16,4336	48	61,1111	18,4257
24	58,3333	4,8113	49	52,7778	9,2128
25	55,5556	11,1111	50	47,2222	12,1081

Tabela 17 – Resultados do treino para classificação do banco de dados BCI *competition III* com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	83,3333	28,8675
2	95,3704	1,6038	27	83,3333	28,8675
3	100	0	28	81,4815	27,8394
4	100	0	29	83,3333	28,8675
5	100	0	30	69,4444	30,8070
6	100	0	31	69,4444	30,8070
7	100	0	32	69,4444	30,8070
8	100	0	33	83,3333	28,8675
9	83,3333	28,8675	34	83,3333	28,8675
10	83,3333	28,8675	35	83,3333	28,8675
11	83,3333	28,8675	36	83,3333	28,8675
12	83,3333	28,8675	37	69,4444	30,8070
13	83,3333	28,8675	38	69,4444	30,8070
14	83,3333	28,8675	39	69,4444	30,8070
15	83,3333	28,8675	40	69,4444	30,8070
16	83,3333	28,8675	41	69,4444	30,8070
17	83,3333	28,8675	42	83,3333	28,8675
18	83,3333	28,8675	43	69,4444	30,8070
19	83,3333	28,8675	44	69,4444	30,8070
20	83,3333	28,8675	45	68,5185	29,9176
21	83,3333	28,8675	46	83,3333	28,8675
22	83,3333	28,8675	47	83,3333	28,8675
23	83,3333	28,8675	48	69,4444	30,8070
24	82,4074	28,3732	49	69,4444	30,8070
25	83,3333	28,8675	50	69,4444	30,8070

Tabela 18 – Resultados da classificação do do banco de dados BCI *competition III* com rede neural artificial com algoritmo de treinamento regularização bayesiana (Elaborado pelo autor).

Número de neurônios	Média de acertos (%)	Desvio padrão	Número de neurônios	Média de acertos (%)	Desvio padrão
1	-	-	26	55,5556	7,8567
2	61,1111	9,6225	27	55,5556	7,8567
3	66,6667	17,5682	28	58,3333	9,2128
4	63,8889	4,8113	29	55,5556	7,8567
5	58,3333	4,8113	30	52,7778	9,2128
6	61,1111	5,5556	31	52,7778	9,2128
7	61,1111	5,5556	32	52,7778	9,2128
8	63,8889	9,2128	33	55,5556	7,8567
9	55,5556	7,8567	34	55,5556	7,8567
10	55,5556	7,8567	35	58,3333	9,2128
11	55,5556	7,8567	36	55,5556	7,8567
12	55,5556	7,8567	37	52,7778	9,2128
13	55,5556	7,8567	38	52,7778	9,2128
14	55,5556	7,8567	39	52,7778	9,2128
15	55,5556	7,8567	40	52,7778	9,2128
16	55,5556	7,8567	41	52,7778	9,2128
17	58,3333	9,2128	42	58,3333	9,2128
18	55,5556	7,8567	43	52,7778	9,2128
19	55,5556	7,8567	44	52,7778	9,2128
20	55,5556	7,8567	45	52,7778	9,2128
21	55,5556	7,8567	46	58,3333	9,2128
22	55,5556	7,8567	47	58,3333	9,2128
23	55,5556	7,8567	48	52,7778	9,2128
24	55,5556	7,8567	49	52,7778	9,2128
25	55,5556	7,8567	50	52,7778	9,2128

Tabela 19 – Comparação dos resultados da classificação do banco de dados BCI *competition* III (adaptado de BENEVIDES et al., 2012).

Autores	Taxa de acerto para o usuário 3 (%)
Autor	63,8889
Benevides et al	65,66
Ferran Galan	47,5
Xiang Liao	52
Walter	32,3

## 5 CONSIDERAÇÕES FINAIS

O objetivo principal deste trabalho foi a classificação das tarefas mentais presentes no eletroencefalograma. Como visto na metodologia utilizando a função que representa a transformada de tempo discreto de Fourier foi feita a representação do EEG no domínio da frequência. Com isso foi identificadas as informações presentes no EEG a serem classificadas.

Antes de classificar as informações presentes no EEG, foi feita a análise dos classificadores com o banco de dados Iris de Fisher. Nessa etapa todos os classificadores obtiveram taxa de acerto superior a 33,3333%, esse resultado demonstra que os classificadores não estão classificando os dados de forma aleatória e podem ser utilizados para a classificação dos bancos de dados BCI II e BCI III. Nessa fase de análise de classificadores a rede neural artificial com algoritmo de treinamento Levenberg-Marquardt obteve taxa de acerto superior aos demais classificadores, obtendo uma taxa de acerto de 98% com desvio padrão de 1,6330.

O primeiro banco de dados a ser classificado foi o BCI *competition* II. Para essa etapa os classificadores utilizados obtiveram médias de acertos superiores a 50%, exceto para a RNA com algoritmo de treinamento LM utilizando dois neurônios na camada escondida, nesses casos a RNA está classificando de forma aleatória. Comparando os resultados dos classificadores o classificador k-nn foi o que obteve maior taxa de acerto em relação aos outros com média de 82,1429%. Ainda, comparando o resultado obtido com o classificador k-nn para a classificação do banco de dados BCI *competition* II com o resultado apresentado por Benevides (2007) que é de 93,3333% para taxa de acerto. Com isso pode-se observar que o resultado obtido por Benevides (2007) é superior ao obtido neste trabalho.

Por fim, foi feita a classificação do banco de dados BCI *competition* III, observou-se que todos os classificadores obtiveram médias de acertos superiores a 33,3333%, exceto para a RNA com algoritmo de treinamento LM utilizando 16 neurônios na camada escondida e para o classificador k-nn com 14, 19 e 24 vizinhos, nesses casos os classificadores estão classificando de forma aleatória. Comparando a média de acerto e desvio padrão entre os classificadores, a RNA com algoritmo de treinamento RB apresentou uma taxa de acerto de 63,8889% e desvio padrão de 4,8113. Esse resultado foi utilizado para fazer a comparação com os resultados obtidos na competição BCI III e para o resultado apresentado por Benevides et al (2012). Com isso pode-se observar que a taxa de acerto obtida

nesse trabalho foi inferior somente a taxa de acerto obtida por Benevides et al (2012) que obteve taxa de acerto de 65,66%. Os demais resultados dessa comparação estão presentes na Tabela 19.

## REFERÊNCIAS

- ALMEIDA, M. **Filtragem Digital de Sinais Biomédicos. 116 p.** Dissertation (Master) — Dissertação de Mestrado — Universidade Federal de Santa Catarina (UFSC), Florianópolis, 1997.
- ANDRADE, V. S. Reconhecimento de texturas utilizando padrão binário local e classificador knn. **Revista Tecnológica da Fatec Americana**, v. 6, n. 1, p. 95–100, 2018.
- ANGHINAH, R. et al. Artefatos biológicos no eeg quantitativo. **Arquivos de neuro-psiquiatria**, SciELO Brasil, v. 64, n. 2A, p. 264–268, 2006.
- BABLANI, A.; EDLA, D. R.; DODIA, S. Classification of eeg data using k-nearest neighbor approach for concealed information test. **Procedia computer science**, Elsevier, v. 143, p. 242–249, 2018.
- BEAR, M. F.; CONNORS, B. W.; PARADISO, M. A. **Neurociencias: Desvendando o Sistema Nervoso. 3ª ed.** Sao Paulo, Brasil: Artmed R, 2008.
- BENEVIDES, A. Classificação de padrões mentais via redes neurais artificiais e transformada wavelet. **Faculdade de Engenharia Eletrica, CentroTecnológico da Universidade Federal do Espírito Santo, Vitoria**, 2007.
- BENEVIDES, A. **Proposta de um Modelo de Fontes Cerebrais para Classificação de Padrões Mentais. 222 p.** Dissertation (Master) — Dissertação de Mestrado — Universidade Federal do Espírito Santo, 2009.
- BENEVIDES, A. B.; FILHO, T. F. B.; FILHO, M. S. Pseudo-online classification of mental tasks using kullback-leibler symmetric divergence. **Journal of Medical and Biological Engineering**, v. 32, n. 6, p. 411–416, 2012.
- BLANKERTZ, B. **BCI competition III Final results.** 2005. Available from Internet: <<http://www.bbc.de/competition/iii/results/index.html>>.
- BLANKERTZ, B. et al. The bci competition 2003: progress and perspectives in detection and discrimination of eeg single trials. **IEEE transactions on biomedical engineering**, IEEE, v. 51, n. 6, p. 1044–1051, 2004.
- BOLZAN, M. J. A. Análise da transformada em ondeletas aplicada em sinal geofísico. **Revista Brasileira de Ensino de Física**, SciELO Brasil, v. 26, n. 1, p. 37–41, 2004.
- BORN, R. S. **Filtros adaptativos aplicados a sinais biomédicos.** Monografia — Universidade Federal de Pelotas, 2000.
- CARVALHO, R. R. M. et al. Estudo de máquina de aprendizagem extrema aplicada à aproximação de função. In: **apresentado na 74ª Semana Oficial de Engenharia e Agronomia, no Congresso Técnico Científico da Engenharia e da Agronomia, Belém, Brasil.** [S.l.: s.n.], 2017.
- CORREIA, L. R. Utilização de redes neurais na identificação de tendências de desempenho das ofertas públicas iniciais. **Trabalho de conclusão de curso, Universidade Federal do Paraná**, 2015.

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936.

FORESEE, F. D.; HAGAN, M. T. Gauss-newton approximation to bayesian learning. In: IEEE. **Proceedings of International Conference on Neural Networks (ICNN'97)**. [S.l.], 1997. v. 3, p. 1930–1935.

FUKUNAGA, K.; NARENDRA, P. M. A branch and bound algorithm for computing k-nearest neighbors. **IEEE transactions on computers**, IEEE, v. 100, n. 7, p. 750–753, 1975.

GOMES, M. da M. Bases fisiológicas do eletroencefalograma. **Revista Brasileira de Neurologia**, v. 51, n. 1, 2015.

GONÇALVES, R. M. et al. Modelagem preditiva de linha de costa utilizando redes neurais artificiais. **Boletim de Ciências Geodésicas**, SciELO Brasil, v. 16, n. 3, p. 420–444, 2010.

GUYTON, A. C.; HALL, J. E. **Tratado de fisiologia médica**. 11<sup>a</sup>. ed. Rio de Janeiro, RJ: Elsevier Editora, 2006.

HAGAN, M. T.; MENHAJ, M. B. Training feedforward networks with the marquardt algorithm. **IEEE transactions on Neural Networks**, IEEE, v. 5, n. 6, p. 989–993, 1994.

HAYKIN, S. **Redes neurais: princípios e prática**. Segunda edição. Porto Alegre, Rio Grande do Sul, Brasil: Bookman, 2001. 900 p.

HAYKIN, S. S.; VEEN, B. V. **Sinais e sistemas**. Porto Alegre, Rio Grande do Sul, Brasil: Bookman, 2001. 668 p.

JARDIM, D. d. S. **Remoção de artefatos em EEG para Neurofeedback**. Dissertation (Master) — Dissertação de Mestrado – Universidade Federal de Pernambuco, 2017.

JUNIOR, A. N. **Aplicação de Redes Neurais utilizando o software MATLAB**. Monografia – Fundação de Ensino Eurípides Soares da Rocha, Centro Universitário Eurípides de Marília (UNIVEM), Marília, 2005.

KIM, Y. et al. Motor imagery classification using mu and beta rhythms of eeg with strong uncorrelating transform based complex common spatial patterns. **Computational intelligence and neuroscience**, Hindawi, v. 2016, 2016.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. **Quarterly of applied mathematics**, v. 2, n. 2, p. 164–168, 1944.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. **Journal of the society for Industrial and Applied Mathematics**, SIAM, v. 11, n. 2, p. 431–441, 1963.

MELO, G. S. **Aplicação de Aprendizado de Máquina para Previsão de Fluxo de Caixa em ATMs**. Monografia – Universidade Federal de Pernambuco, Centro de Informática, Recife, 2018.



MILLAN, J. R. On the need for on-line learning in brain-computer interfaces. In: IEEE. **2004 IEEE International Joint Conference on Neural Networks**. [S.l.], 2004. v. 4, p. 2877–2882.

PERALES, T. R. **Utilização de Redes Neurais Artificiais no Diagnóstico de Cardiopatias**. 168 p. Dissertation (Master) — Dissertação de Mestrado — Departamento de Engenharia Elétrica, Universidade Estadual de Londrina, 2011.

PFURTSCHELLER, G.; SILVA, F. L. D. Event-related eeg/meg synchronization and desynchronization: basic principles. **Clinical neurophysiology**, Elsevier, v. 110, n. 11, p. 1842–1857, 1999.

RAMOS, C. D. **Caracterização do eletroencefalograma normal em situação de vigília: elaboração da base de dados e análise quantitativa**. Universidade Federal de Uberlândia, 2017.

RUFINO, M. P. **Implementação de uma Rede Neural Artificial para classificação de sinais mioelétricos**. Trabalho de conclusão de curso — Universidade Federal da Paraíba Centro de Energias Alternativas e Renováveis Departamento de Engenharia Elétrica, 2017.

SILVA, C. D. **Processamento de sinais de EEG para classificação de tarefas motoras em sistemas de interface cérebro-máquina**. Tese de Pós-Graduação — Universidade Federal de Santa Catarina UFSC, Departamento de Automaçõ e Sistemas, Florianópolis, 2017.

TAVARES, L. D. **Treinamento Multiobjetivo de Extreme Learning Machines Utilizando Decomposição em Valores Singulares**. Tese de Pós-Graduação — Universidade Federal de Minas Gerais, Escola de Engenharia, Belo Horizonte, 2015.

## ANEXO A — CÓDIGO PARA SE OBTER O SINAL EEG DO BANCO DE DADOS BCI II NO DOMÍNIO DO TEMPO

```
clear all
clc
load Dataset
T=[1:0.00781928:10];
A = x_train(:, :, 140);
C3= A(:, 1);
CZ= A(:, 2);
C4= A(:, 3);

subplot (311)
plot(T,C3)
title('C3')
xlabel('Tempo (s)')
ylabel('C3(t)')
set(gca, 'FontSize', 15)

subplot(312)
plot(T,C4)
title('C4')
xlabel('Tempo (s)')
ylabel('C4(t)')
set(gca, 'FontSize', 15)

subplot (313)
plot(T,CZ)
title('CZ')
xlabel('Tempo (s)')
ylabel('CZ (t)')
set(gca, 'FontSize', 15)
```

## ANEXO B — CÓDIGO PARA SE OBTER O SINAL EEG DO BANCO DE DADOS BCI II NO DOMÍNIO DA FREQUÊNCIA

```

clear all
clc
load Dataset
Fs=128;
T=1/Fs;
L=1152;
A = x_train(:, :, 140);
B = x_test(:, :, :);
Y = y_train(:, :);
C3= A(:, 1);
CZ= A(:, 2);
C4= A(:, 3);
NFFT= 896;
%FFT calnal C3 (coluna 1 da matriz)
C3ft=fft(C3(1:256),NFFT)/L;
f=(Fs/2)* linspace(0,1,NFFT/2+1);
subplot(221)
plot(f,2*abs(C3ft(1:NFFT/2+1)))
title('espectro C31')
xlabel('Frequency (Hz)')
ylabel('IC31(f)l')

set(gca,'FontSize',15)
%FFT C31 (coluna 2 da matriz)
C31ft=fft(C3(257:end),NFFT)/L;
f=(Fs/2)* linspace(0,1,NFFT/2+1);
subplot(222)
plot(f,2*abs(C31ft(1:NFFT/2+1)))
title('espectro C41')
xlabel('Frequency (Hz)')
ylabel('IC41(f)l')

set(gca,'FontSize',15)
%FFT C4 (coluna 3 da matriz)
C4ft=fft(C4(1:256),NFFT)/L;
f=(Fs/2)* linspace(0,1,NFFT/2+1);
subplot(223)
plot(f,2*abs(C4ft(1:NFFT/2+1)))
title('espectro C32')
xlabel('Frequency (Hz)')
ylabel('IC32(f)l')

set(gca,'FontSize',15)
%FFT C41 (coluna 4 da matriz)
C41ft=fft(C4(257:end),NFFT)/L;
f=(Fs/2)* linspace(0,1,NFFT/2+1);
subplot(224)
plot(f,2*abs(C41ft(1:NFFT/2+1)))
title('espectro C42')
xlabel('Frequency (Hz)')
ylabel('IC42(f)l')

set(gca,'FontSize',15)

```

## ANEXO C — CÓDIGO PARA SALVAR OS DADOS DO EEG DO BANCO DE DADOS BCI II

```

clear all
clc
load Dataset
Fs=128; %frequencia de amostragem
T=1/Fs; % Periodo do sinal
L=128*9; % Comprimento do sinal 128Hz*9s
f=(Fs/2)*linspace(0,1,100/2+1);

for i = 1:140
    A = x_train(:, :, i);
    %CAR
    CT= mean(A,2); linha .
    C3= A(:,1)-CT;
    CZ= A(:,2)-CT;
    C4= A(:,3)-CT;

    %FFT calnal C3 (coluna 1 da zmatriz)
    C3ft1 = fft(C3(1:256),NFFT)/L; %C3 menor que NFFT, então C3 será preenchido à direita até o comprimento NFFT.
    C3M1(:, i) = abs(C3ft1(1:NFFT/2+1));

    %FFT C31 (coluna 2 da zmatriz)
    C3ft2=fft(C3(257:end),NFFT)/L;%L divide a amplitude da fft
    C3M2(:, i) = abs(C3ft2(1:NFFT/2+1));

    %FFT C4 (coluna 3 da zmatriz)
    C4ft1=fft(C4(1:256),NFFT)/L;
    C4M1(:, i) = abs(C4ft1(1:NFFT/2+1));

    %FFT C41 (coluna 4 da zmatriz)
    C4ft2=fft(C4(257:end),NFFT)/L;
    C4M2(:, i) = abs(C4ft2(1:NFFT/2+1));
end

save('EEGFFT.mat', 'C3M1', 'C3M2', 'C4M1', 'C4M2');

```

## ANEXO D — CÓDIGO REPRESENTA O GRÁFICO DA IMAGINAÇÃO DE ATIVIDADE CEREBRAIS DO EEG DO BANCO DE DADOS BCI II

```

clc
clear all
load dataset
Fs=128;
T=1/Fs;
L=1152;
NFFT= 896;
f=(Fs/2)* linspace(0,1,NFFT/2+1);
K= find(y_train==1);
Q= find(y_train==2);
load EEGFFT
C3ME1=C3M1(:,K);
C3ME2=C3M2(:,K);
C4ME1=C4M1(:,K);
C4ME2=C4M2(:,K);
C3MD1=C3M1(:,Q);
C3MD2=C3M2(:,Q);
C4MD1=C4M1(:,Q);
C4MD2=C4M2(:,Q);
C31E= mean (C3ME1,2);

C32E= mean (C3ME2,2);

C41E= mean (C4ME1,2);

C42E= mean (C4ME2,2);

subplot(221)
plot(f,2*C31E,'b'); hold on;
plot(f,2*C41E,'r')
title('MÃo esquerda C3,C4 antes')
xlabel('FrequÃªncia (Hz)')
ylabel('IC3(f)l')
legend('C3','C4')
xlim([0 30])
set(gca,'FontSize',18)
subplot(222)
plot(f,2*C32E,'b'); hold on;
plot(f,2*C42E,'r')
title('MÃo esquerda C3,C4 durante')
xlabel('FrequÃªncia (Hz)')
ylabel('IC3(f)l')
legend('C3','C4')
xlim([0 30])
set(gca,'FontSize',18)
%%direita
C31D= mean (C3MD1,2);

C32D= mean (C3MD2,2);

C41D= mean (C4MD1,2);

C42D= mean (C4MD2,2);

subplot(223)
plot(f,2*C31D,'b'); hold on;
plot(f,2*C41D,'r')
title('MÃo direita C3,C4 antes')
xlabel('FrequÃªncia (Hz)')
ylabel('IC3(f)l');
legend('C3','C4')
xlim([0 30])
set(gca,'FontSize',18)
subplot(224)
plot(f,2*C32D,'b'); hold on;

```

```
plot(f,2*C42D,'r')
title('MÃeõ direita C3,C4 durante ')
xlabel('FrequÃncia (Hz)')
ylabel('|C3(f)|')
legend('C3','C4')
xlim([0 30])
set(gca,'FontSize',18)
```

## ANEXO E — CÓDIGO QUE SALVA OS DADOS DO BCI III

```

close all, clear all
load Dados3.mat
for i=1:12
eeg2(:, :, i) = Dados3.subject(3).task2(i).EEG(2560:5120, :);
eeg3(:, :, i) = Dados3.subject(3).task3(i).EEG(2560:5120, :);
eeg7(:, :, i) = Dados3.subject(3).task7(i).EEG(2560:5120, :);
end
eeg2;
eeg3;
eeg7;
for i = 1:12
    A = eeg2(:, :, i);
    CT3= mean(A(:, [2:29 31 32]), 2);
    C33=A(:, 8) - CT3;
    C43=A(:, 23) - CT3;
    F73=A(:, 3) - CT3;
    [M,L] = size(C33);
    fs = 512;
    %C3
    Freqs = fft(C33);
    Espectro = fftshift(Freqs);
    C3ME(:, i) = 2*abs(Espectro(M/2:M, :));
    %C4
    Freqs = fft(C43);
    Espectro = fftshift(Freqs);
    C4ME(:, i) = 2*abs(Espectro(M/2:M, :));
    %F7
    Freqs = fft(F73);
    Espectro = fftshift(Freqs);
    F7ME(:, i) = 2*abs(Espectro(M/2:M, :));
    %%%%%%%%%%%%%%%%%%%%%%%%%%
    Fase_Freqs = rad2deg(angle(Espectro(M/2:M, :)));
    vetor_freq_uni = linspace(0, fs/2, M/2+1);
end

for i=1:12
A = eeg3(:, :, i);
    CT3= mean(A(:, [2:29 31 32]), 2);
    C33=A(:, 8) - CT3;
    C43=A(:, 23) - CT3;
    F73=A(:, 3) - CT3;
[M,L] = size(C33);
    fs = 512;
    %C3
    Freqs = fft(C33);
    Espectro = fftshift(Freqs);
    C3MD(:, i) = 2*abs(Espectro(M/2:M, :));
    %C4
    Freqs = fft(C43);
    Espectro = fftshift(Freqs);
    C4MD(:, i) = 2*abs(Espectro(M/2:M, :));
    %F7
    Freqs = fft(F73);
    Espectro = fftshift(Freqs);
    F7MD(:, i) = 2*abs(Espectro(M/2:M, :));
    %%%%%%%%%%%%%%%%%%%%%%%%%%

    Fase_Freqs = rad2deg(angle(Espectro(M/2:M, :)));
    vetor_freq_uni = linspace(0, fs/2, M/2+1);
end

for i=1:12
A = eeg7(:, :, i);
    CT7= mean(A(:, [2:29 31 32]), 2);
    C37=A(:, 8) - CT7;
    C47=A(:, 23) - CT7;
    F77=A(:, 3) - CT7;
[M,L] = size(C37);

```

```

fs = 512;
%C3
Freqs = fft(C37);
Espectro = fftshift(Freqs);
C3P(:,i)= 2*abs(Espectro(M/2:M,:));
%C4
Freqs = fft(C47);
Espectro = fftshift(Freqs);
C4P(:,i)= 2*abs(Espectro(M/2:M,:));
%F7
Freqs = fft(F77);
Espectro = fftshift(Freqs);
F7P(:,i)= 2*abs(Espectro(M/2:M,:));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Fase_Freqs = rad2deg(angle(Espectro(M/2:M,:)));
vetor_freq_uni = linspace(0,fs/2,M/2+1);
end
save('EEGFFT2.mat','C3ME','C4ME','F7ME','C3MD','C4MD','F7MD','C3P','C4P','F7P');

```



## ANEXO F — CÓDIGO QUE GERA O SINAL DO ELETRODO C3 DO BANCO DE DADOS BCI III

```

close all, clear all
load Dados3.mat

eeg = Dados3.subject(3).task2(1).EEG(2560:5120,:);
[M,L] = size(eeg);
fs = 512;
Freqs = fft(Freqs);
Espectro = fftshift(Freqs);
Amp_Freqs = 2*abs(Espectro(M/2:M,:));
Fase_Freqs = rad2deg(angle(Espectro(M/2:M,:)));
vetor_freq_uni = linspace(0,fs/2,M/2+1);

for i=1:12
eeg1(:, :, i) = Dados3.subject(3).task2(i).EEG(2560:5120,:);
end
eeg1;
for i = 1:12
    A = eeg1(:, :, i);
    CT3= mean(A(:, [2:29 31 32]), 2);
    C3=A(:, 8)-CT3;
    [N,P] = size(C3);
    fs = 512;
    %C3
    C33 = fft(C3);
    EspectroC33 = fftshift(C33);
    vetor_freq = linspace(0,fs/2,N/2+1);
    C3ME= 2*abs(EspectroC33(N/2:N,:));
end

subplot(312)
plot(vetor_freq_uni, Amp_Freqs(:, 8));
title('Espectro de amplitude (unilateral) â Canal C3');
xlabel('f (Hz)'); ylabel('IEEG(f)'); ylim([0 10000]); grid minor;
xlim([0 155])
set(gca, 'FontSize', 15)
subplot(311)
plot(vetor_freq_uni, Fase_Freqs(:, 8));
title('Sinal no tempo â Canal C3');
xlabel('Tempo (s)'); ylabel('IEEG(t)'); grid minor;
xlim([0 155])
set(gca, 'FontSize', 15)
subplot(313)
plot(vetor_freq, C3ME);
title('Espectro de amplitude (unilateral) â Canal C3 com CAR');
xlabel('f (Hz)'); ylabel('IEEG(f)'); ylim([0 10000]); grid minor;
xlim([0 155])
set(gca, 'FontSize', 15)

```

## ANEXO G — CÓDIGO DO K-NN PARA CLASSIFICAÇÃO DO BANCO DE DADOS DE FISHER

```

clc
clear all
load meas
r1 = randi([1 150],1,150);
ry=meas(r1,5);
rw=meas(r1,1:4)';
%% Normaliza
[sm sn]= size(rw);
mrw=mean(rw');
B=zeros(sm,sn);
B(1,:)=rw(1,:)-mrw(1);
B(2,:)=rw(2,:)-mrw(2);
B(3,:)=rw(3,:)-mrw(3);
B(4,:)=rw(4,:)-mrw(4);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end
end
%%xw
x1=B(:,1:30);
x2=B(:,31:60);
x3=B(:,61:90);
x4=B(:,91:120);
x5=B(:,121:150);
x6=x1;
x7=x2;
x8=x3;
x9=x4;

y1=ry(1:30,1)';
y2=ry(31:60,1)';
y3=ry(61:90,1)';
y4=ry(91:120,1)';
y5=ry(121:end,1)';
y6=y1;
y7=y2;
y8=y3;
y9=y4;
entrada={x1 x2 x3 x4 x5 x6 x7 x8 x9};
saida={y1 y2 y3 y4 y5 y6 y7 y8 y9};
%validaÃ§Ão cruzada
for j=1:5
    treino=[entrada{1,j+1} entrada{1,j+2} entrada{1,j+3} entrada{1,j+4}]';
    treinomao=[saida{1,j+1} saida{1,j+2} saida{1,j+3} saida{1,j+4}];
    teste = [entrada{1,j}];
    testemao=[saida{1,j}];
    for i= 1:50
cknnw= knnclassify(teste ,treino ,treinomao ,i);
Xw =abs(round(cknnw') - testemao);
for k=1:30
    if Xw(k)> 1
        Xw(k)=1;
    end
end
end
Y2w(j,i) = 100 - (((sum(Xw))/30)*100);
end

```

```
end
Mediatestew= mean(Y2w)
desviotestew= std(Y2w)
bar(Mediatestew,0.5,'b'); hold on
errorbar([1:50],Mediatestew,desviotestew,'r');hold off
ylim([50 100])
xlim([0 51])
title('MÃ©dia dos testes ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca,'FontSize',15)
```

## ANEXO H — CÓDIGO DO LEVENBERG-MARQUARDT PARA CLASSIFICAÇÃO DO BANCO DE DADOS DE FISHER

```

clc
clear all
load meas
r1 = randi([1 150],1,150);
A=[-ones(1,50); -ones(1,50)];
C=[ones(1,50); ones(1,50)];
D=[ones(1,50); -ones(1,50)];
y=[A C D];
ry = y(:,r1);
rw=meas(r1,1:4)';
%% Normaliza
[sm sn]= size(rw);
mrw=mean(rw');
B=zeros(sm,sn);
B(1,:)=rw(1,:)-mrw(1);
B(2,:)=rw(2,:)-mrw(2);
B(3,:)=rw(3,:)-mrw(3);
B(4,:)=rw(4,:)-mrw(4);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end
end
%% separa os dados para valida
x1=B(:,1:30);
x2=B(:,31:60);
x3=B(:,61:90);
x4=B(:,91:120);
x5=B(:,121:end);
x6=x1;
x7=x2;
x8=x3;
x9=x4;

y1=ry(:,1:30);
y2=ry(:,31:60);
y3=ry(:,61:90);
y4=ry(:,91:120);
y5=ry(:,121:end);
y6=y1;
y7=y2;
y8=y3;
y9=y4;

entrada=[x1 x2 x3 x4 x5 x6 x7 x8 x9];
saida={y1 y2 y3 y4 y5 y6 y7 y8 y9};
for j=1:5
    %valida os dados cruzada
    treino=[entrada(:,j+1) entrada(:,j+2) entrada(:,j+3) entrada(:,j+4)];
    treinomao=[saida{1,j+1} saida{1,j+2} saida{1,j+3} saida{1,j+4}];
    teste= [entrada{1,j}];
    testemao=[saida{1,j}];
    for i=2:50
        rede = newff(minmax(B),[i 2],{'tansig' 'satlins'});
        rede = train(rede, treino, treinomao);
        Ctrain = sim(rede, treino);
        Ctrain= round(Ctrain);
        cont_acerto_setosa = 0;
    end
end

```

```

    cont_acerto_versicolor = 0;
    cont_acerto_virginica = 0;
    for k= 1:120
        if Ctrain(1,k) == -1 & Ctrain (2,k) == -1 & treinomao(1,k) == -1 & treinomao(2,k) == -1
            cont_acerto_setosa = cont_acerto_setosa + 1;
        elseif Ctrain(1,k) == 1 & Ctrain (2,k) == 1 & treinomao(1,k) == 1 & treinomao(2,k) == 1
            cont_acerto_versicolor = cont_acerto_versicolor + 1;
        elseif Ctrain(1,k) == 1 & Ctrain (2,k) == -1 & treinomao(1,k) == 1 & treinomao(2,k) == -1
            cont_acerto_virginica = cont_acerto_virginica + 1;
        end
    end
    total_acertos1 = cont_acerto_setosa + cont_acerto_versicolor + cont_acerto_virginica;
    Y1(j,i) = (total_acertos1/120)*100;
    Y = sim(rede , teste);
    Y=round(Y);
    cont_acerto_teste_setosa = 0;
    cont_acerto_teste_versicolor = 0;
    cont_acerto_teste_virginica = 0;
    for k= 1:30
        if Y(1,k) == -1 & Y(2,k) == -1 & testemao(1,k) == -1 & testemao(2,k) == -1
            cont_acerto_teste_setosa = cont_acerto_teste_setosa + 1;
        elseif Y(1,k) == 1 & Y (2,k) == 1 & testemao(1,k) == 1 & testemao(2,k) == 1
            cont_acerto_teste_versicolor = cont_acerto_teste_versicolor + 1;
        elseif Y(1,k) == 1 & Y (2,k) == -1 & testemao(1,k) == 1 & testemao(2,k) == -1
            cont_acerto_teste_virginica = cont_acerto_teste_virginica + 1;
        end
    end
    total_acertos1_teste = cont_acerto_teste_setosa + cont_acerto_teste_versicolor + cont_acerto_teste_virginica;
    Y2(j,i) = ((total_acertos1_teste)/30)*100;

end
end
Mediatreino= mean(Y2,1)
desviotreino= std(Y2,1)
Mediateste= mean(Y1,1)
desvioste= std(Y1,1)
subplot(211)
bar(Mediatreino ,0.5, 'b'); hold on
errorbar ([1:50],Mediatreino ,desviotreino , 'r');hold off
ylim([50 100])
xlim([0 51])
title('MÃ©dia dos treino ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel(' Acertos (%)');
set(gca, 'FontSize', 15)
subplot(212)
bar(Mediateste ,0.5, 'b'); hold on
errorbar ([1:50],Mediateste ,desvioste , 'r');hold off
ylim([50 100])
xlim([0 51])
title('MÃ©dia dos testes ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel(' Acertos (%)');
set(gca, 'FontSize', 15)

```

## ANEXO I — CÓDIGO DO DA REGULARIZAÇÃO BAYESIANA PARA CLASSIFICAÇÃO DO BANCO DE DADOS DE FISHER

```

clc
clear all
load meas
r1 = randi([1 150],1,150);
A=[-ones(1,50); -ones(1,50)];
C=[ones(1,50); ones(1,50)];
D=[ones(1,50); -ones(1,50)];
y=[A C D];
ry = y(:,r1);
rw=meas(r1,1:4)';
%% Normaliza
[sm sn]= size(rw);
mrw=mean(rw');
B=zeros(sm,sn);
B(1,:)=rw(1,:)-mrw(1);
B(2,:)=rw(2,:)-mrw(2);
B(3,:)=rw(3,:)-mrw(3);
B(4,:)=rw(4,:)-mrw(4);
mx=max(B');
mi=min(B');
for i=1:sm
for j=1:sn
if B(i,j)> 0
B(i,j)=B(i,j)/mx(i);
elseif B(i,j)< 0
B(i,j)=B(i,j)/(-mi(i));
else
B(i,j)=B(i,j);
end
end
end
%% separa os dados para valida
x1=B(:,1:30);
x2=B(:,31:60);
x3=B(:,61:90);
x4=B(:,91:120);
x5=B(:,121:end);
x6=x1;
x7=x2;
x8=x3;
x9=x4;

y1=ry(:,1:30);
y2=ry(:,31:60);
y3=ry(:,61:90);
y4=ry(:,91:120);
y5=ry(:,121:end);
y6=y1;
y7=y2;
y8=y3;
y9=y4;
entrada=[x1 x2 x3 x4 x5 x6 x7 x8 x9];
saida={y1 y2 y3 y4 y5 y6 y7 y8 y9};
for j=1:5
%valida cruzada
treino=[entrada(:,j+1) entrada(:,j+2) entrada(:,j+3) entrada(:,j+4)];
treinomao=[saida{1,j+1} saida{1,j+2} saida{1,j+3} saida{1,j+4}];
teste= [entrada{1,j}];
testemao=[saida{1,j}];
for i=2:50
rede = newff(minmax(B),[i 2],{'tansig' 'satlins'},'trainbr');
rede = train(rede,treino,treinomao);
Ctrain = sim(rede,treino);
Ctrain= round(Ctrain);
cont_acerto_setosa = 0;

```

```

cont_acerto_versicolor = 0;
cont_acerto_virginica = 0;
for k= 1:120
if Ctrain(1,k) == -1 & Ctrain (2,k) == -1 & treinomao(1,k) == -1 & treinomao(2,k) == -1
cont_acerto_setosa = cont_acerto_setosa + 1;
elseif Ctrain(1,k) == 1 & Ctrain (2,k) == 1 & treinomao(1,k) == 1 & treinomao(2,k) == 1
cont_acerto_versicolor = cont_acerto_versicolor + 1;
elseif Ctrain(1,k) == 1 & Ctrain (2,k) == -1 & treinomao(1,k) == 1 & treinomao(2,k) == -1
cont_acerto_virginica = cont_acerto_virginica + 1;
end
end
total_acertos1 = cont_acerto_setosa + cont_acerto_versicolor + cont_acerto_virginica;
Y1(j,i) = (total_acertos1/120)*100;
Y = sim(rede , teste);
Y=round(Y);
cont_acerto_teste_setosa = 0;
cont_acerto_teste_versicolor = 0;
cont_acerto_teste_virginica = 0;
for k= 1:30
if Y(1,k) == -1 & Y(2,k) == -1 & testemao(1,k) == -1 & testemao(2,k) == -1
cont_acerto_teste_setosa = cont_acerto_teste_setosa + 1;
elseif Y(1,k) == 1 & Y (2,k) == 1 & testemao(1,k) == 1 & testemao(2,k) == 1
cont_acerto_teste_versicolor = cont_acerto_teste_versicolor + 1;
elseif Y(1,k) == 1 & Y (2,k) == -1 & testemao(1,k) == 1 & testemao(2,k) == -1
cont_acerto_teste_virginica = cont_acerto_teste_virginica + 1;
end
end
total_acertos1_teste = cont_acerto_teste_setosa + cont_acerto_teste_versicolor + cont_acerto_teste_virginica;
Y2(j,i) = (total_acertos1_teste/30)*100;

end
end
Mediatreino= mean(Y2,1)
desviotreino= std(Y2,1)
Mediatreino= mean(Y1,1)
desviotreino= std(Y1,1)
subplot(211)
bar(Mediatreino ,0.5, 'b'); hold on
errorbar ([1:50],Mediatreino ,desviotreino , 'r');hold off
ylim([50 100])
xlim([0 51])
title('MÃ©dia dos treino ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca, 'FontSize', 15)
subplot(212)
bar(Mediateste ,0.5, 'b'); hold on
errorbar ([1:50],Mediateste ,desvioste , 'r');hold off
ylim([50 100])
xlim([0 51])
title('MÃ©dia dos testes ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca, 'FontSize', 15)

```

## ANEXO J — CÓDIGO DO K-NN PARA CLASSIFICAÇÃO DO BCI *COMPETITION II*

```

clc
clear all
load Dataset
load EEGFFT
r1 = randi([1 140],1,140);
ry = y_train(r1,:)-1;
rw=mean(C3M2(70:77,r1));
rv=mean(C4M2(70:77,r1));
rw1=mean(C3M2(113:211,r1));% beta
rv1=mean(C4M2(113:211,r1));
rq =[rw;rv;rw1;rv1];

%% Normalizar
[sm sn]= size(rq);
mrq=mean(rq');
B=zeros(sm,sn);
B(1,:)=rq(1,:)-mrq(1);
B(2,:)=rq(2,:)-mrq(2);
B(3,:)=rq(3,:)-mrq(3);
B(4,:)=rq(4,:)-mrq(4);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end
end
%% separa os dados para valida os
x1=B(:,1:35);
x2=B(:,36:70);
x3=B(:,71:105);
x4=B(:,106:end);
x5=x1;
x6=x2;
x7=x3;

y1=ry(1:35,1)';
y2=ry(36:70,1)';
y3=ry(71:105,1)';
y4=ry(106:end,1)';
y5=y1;
y6=y2;
y7=y3;

entrada=[x1 x2 x3 x4 x5 x6 x7];
saida={y1 y2 y3 y4 y5 y6 y7};
%valida os dados cruzada
for j=1:4
    treino=[entrada{1,j+1} entrada{1,j+2} entrada{1,j+3}]';
    treinomao=[saida{1,j+1} saida{1,j+2} saida{1,j+3}];
    teste= [entrada{1,j}]';
    testemao=[saida{1,j}];
    for i=1:50
        cknw= knnclassify(teste,treino,treinomao,i);
        Xw = round(cknw') - testemao;
        Y2w(j,i) = 100 - ((sum(abs(Xw))/35)*100);
    end
end
end

```



```
Mediatestew= mean(Y2w)
desviotestew= std(Y2w)
bar(Mediatestew,0.5,'b'); hold on
errorbar([1:50],Mediatestew,desviotestew,'r');hold off
ylim([30 100])
xlim([0 51])
title('Média dos testes ');
xlabel('Número de neurónios ');
ylabel('Acertos (%)');
set(gca,'FontSize',15)
```

## ANEXO K — CÓDIGO DE LEVENBERG-MARQUARDT PARA CLASSIFICAÇÃO DO BCI *COMPETITION II*

```

clc
clear all
load Dataset
load EEGFFT
r1 = randi([1 140],1,140);
ry = round(y_train(r1,:)-1.5);
rw=mean(C3M2(57:85,r1));
rv=mean(C4M2(57:85,r1));
rw1=mean(C3M2(113:211,r1));% beta
rv1=mean(C4M2(113:211,r1));
rq =[rw;rv;rw1;rv1];
%% Normalizar
[sm sn]= size(rq);
mrq=mean(rq');
B=zeros(sm,sn);
B(1,:)=rq(1,:)-mrq(1);
B(2,:)=rq(2,:)-mrq(2);
B(3,:)=rq(3,:)-mrq(3);
B(4,:)=rq(4,:)-mrq(4);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end
end
%% separa os dados para valida
x1=B(:,1:35);
x2=B(:,36:70);
x3=B(:,71:105);
x4=B(:,106:end);
x5=x1;
x6=x2;
x7=x3;
y1=ry(1:35,1)';
y2=ry(36:70,1)';
y3=ry(71:105,1)';
y4=ry(106:end,1)';
y5=y1;
y6=y2;
y7=y3;
entrada=[x1 x2 x3 x4 x5 x6 x7];
saida={y1 y2 y3 y4 y5 y6 y7};
for j=1:4
    %valida cruzada
    treino=[entrada{1,j+1} entrada{1,j+2} entrada{1,j+3}];
    treinomao=[saida{1,j+1} saida{1,j+2} saida{1,j+3}];
    teste= [entrada{1,j}];
    testemao=[saida{1,j}];
    for i=2:50
        rede = newff(minmax(B),[i 1],{'tansig' 'satlins'});
        rede = train(rede, treino, treinomao);
        Ctrain = sim(rede, teste);
        Ctrain=round(Ctrain);
        Xtrain = (Ctrain - treinomao);
        for k=1:105
            if Xtrain(k)>= 1
                Xtrain(k)=1;
            end
        end
    end
end

```

```

elseif Xtrain(k)< 0
    Xtrain(k)=1;
end
end
end
Y1(j,i) = 100 - (sum(Xtrain)/105)*100;
Y = sim(rede, teste);
Y=round(Y);
X = (Y - testemao);
    for k=1:35
if X(k)>= 1
    X(k)=1;

elseif X(k)< 0
    X(k)=1;
end
end
end
Y2(j,i) = 100 - ((sum(X)/35)*100);
end
end
Mediateste= mean(Y2,1)
desvioteste= std(Y2,1)
Mediatreino= mean(Y1,1)
desviotreino= std(Y1,1)
subplot(211)
bar(Mediatreino,0.5,'b'); hold on
errorbar([1:50],Mediatreino,desviotreino,'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos treino ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca,'FontSize',15)
subplot(212)
bar(Mediateste,0.5,'b'); hold on
errorbar([1:50],Mediateste,desvioteste,'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos testes ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca,'FontSize',15)

```

## ANEXO L — CÓDIGO DA REGULARIZAÇÃO BAYESIANA PARA CLASSIFICAÇÃO DO BCI *COMPETITION II*

```

clc
clear all
load Dataset
load EEGFFT
r1 = randi([1 140],1,140);
ry = y_train(r1,:)-1;
rw=mean(C3M2(57:85,r1));
rv=mean(C4M2(57:85,r1));
rw1=mean(C3M2(113:211,r1));% beta
rv1=mean(C4M2(113:211,r1));
rq =[rw;rv;rw1;rv1];
%% Normalizar
[sm sn]= size(rq);
mrq=mean(rq');
B=zeros(sm,sn);
B(1,:)=rq(1,:)-mrq(1);
B(2,:)=rq(2,:)-mrq(2);
B(3,:)=rq(3,:)-mrq(3);
B(4,:)=rq(4,:)-mrq(4);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end
end
%% separa os dados para valida
x1=B(:,1:35);
x2=B(:,36:70);
x3=B(:,71:105);
x4=B(:,106:end);
x5=x1;
x6=x2;
x7=x3;
y1=ry(1:35,1)';
y2=ry(36:70,1)';
y3=ry(71:105,1)';
y4=ry(106:end,1)';
y5=y1;
y6=y2;
y7=y3;
entrada=[x1 x2 x3 x4 x5 x6 x7];
saida={y1 y2 y3 y4 y5 y6 y7};
for j=1:4
    %valida cruzada
    treino=[entrada{1,j+1} entrada{1,j+2} entrada{1,j+3}];
    treinomao=[saida{1,j+1} saida{1,j+2} saida{1,j+3}];
    teste= [entrada{1,j}];
    testemao=[saida{1,j}];
    for i=2:50
        rede = newff(minmax(B),[i 1],{'tansig' 'satlins'},'trainbr');
        rede = train(rede, treino, treinomao);
        Ctrain = sim(rede, teste);
        Ctrain=round(Ctrain);
        Xtrain = (Ctrain - treinomao);
        for k=1:105
            if Xtrain(k)>= 1
                Xtrain(k)=1;
            end
        end
    end
end

```

```

elseif Xtrain(k)< 0
    Xtrain(k)=1;
end
end
end
Y1(j,i) = 100 - (sum(Xtrain)/105)*100;
Y = sim(rede, teste);
Y=round(Y);
X = (Y - testemao);
    for k=1:35
if X(k)>= 1
    X(k)=1;

elseif X(k)< 0
    X(k)=1;
end
end
end
Y2(j,i) = 100 - ((sum(X)/35)*100);
end
end
Mediateste= mean(Y2,1)
desvioteste= std(Y2,1)
Mediatreino= mean(Y1,1)
desviotreino= std(Y1,1)
subplot(211)
bar(Mediatreino,0.5,'b'); hold on
errorbar([1:50],Mediatreino,desviotreino,'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos treino ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca,'FontSize',15)
subplot(212)
bar(Mediateste,0.5,'b'); hold on
errorbar([1:50],Mediateste,desvioteste,'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos testes ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca,'FontSize',15)

```

## ANEXO M — CÓDIGO DO K-NN PARA CLASSIFICAÇÃO DO BCI

### COMPETITION III

```

clc
clear all
load EEGFFT2

ME=[mean(C3ME(40:61,:)); mean(C4ME(40:61,:)); mean(F7ME(40:61,:));mean(C3ME(70:150,:)); mean(C4ME(70:150,:)); mean(F7ME(70:150,:))];
MD=[mean(C3MD(40:61,:)); mean(C4MD(40:61,:)); mean(F7MD(40:61,:));mean(C3MD(70:150,:)); mean(C4MD(70:150,:)); mean(F7MD(70:150,:))];
P=[mean(C3P(40:61,:)); mean(C4P(40:61,:)); mean(F7P(40:61,:));mean(C3P(70:150,:)); mean(C4P(70:150,:)); mean(F7P(70:150,:))];
%atribuindo -1 como alvo de ME, 0 para MD e 1 para P
A= -ones(12,1);
B= zeros(12,1);
C= ones(12,1);
y= [A;B;C];
EEG=[ME MD P];
r1 = randi([1 36],1,36);
ry=y(r1,:);
rq= EEG(:,r1);
%% Normalizar
[sm sn]= size(rq);
mrq=mean(rq');
B=zeros(sm,sn);
B(1,:)=rq(1,:)-mrq(1);
B(2,:)=rq(2,:)-mrq(2);
B(3,:)=rq(3,:)-mrq(3);
B(4,:)=rq(4,:)-mrq(4);
B(5,:)=rq(5,:)-mrq(5);
B(6,:)=rq(6,:)-mrq(6);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end
end

%% separa os dados para valida os
x1=B(:,1:9);
x2=B(:,10:18);
x3=B(:,19:27);
x4=B(:,28:end);
x5=x1;
x6=x2;
x7=x3;

y1=ry(1:9,1)';
y2=ry(10:18,1)';
y3=ry(19:27,1)';
y4=ry(28:end,1)';
y5=y1;
y6=y2;
y7=y3;

entrada=[x1 x2 x3 x4 x5 x6 x7];
saida={y1 y2 y3 y4 y5 y6 y7};
%valida os cruzada
for j=1:4
    treino=[entrada{1,j+1} entrada{1,j+2} entrada{1,j+3}]';
    treinomao=[saida{1,j+1} saida{1,j+2} saida{1,j+3}];
    teste= [entrada{1,j}]';

```

```

    testemao=[saida{1,j}];
    for i= 1:25
cknnw= knnclassify(teste ,treino ,treinomao , i);
Xw = abs(round(cknnw') - testemao);
    for k=1:9
        if Xw(k)> 1
            Xw(k)=1;
        end
    end
    Y2w(j , i) = 100 - ((sum((Xw))/9)*100);
    end
end
Mediatestew= mean(Y2w)
desviotestew= std(Y2w)
bar(Mediatestew,0.5,'b'); hold on
errorbar([1:25],Mediatestew ,desviotestew ,'r');hold off
ylim([10 100])
xlim([0 51])
title('MÃ©dia dos testes ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca,'FontSize',15)

```

## ANEXO N — CÓDIGO DE LEVENBERG-MARQUARDT PARA CLASSIFICAÇÃO DO BCI *COMPETITION III*

```

clc
clear all
load EEGFFT2

ME=[mean(C3ME(40:61,:)); mean(C4ME(40:61,:)); mean(F7ME(40:61,:));mean(C3ME(70:150,:)); mean(C4ME(70:150,:)); mean(F7ME(70:150,:))];
MD=[mean(C3MD(40:61,:)); mean(C4MD(40:61,:)); mean(F7MD(40:61,:));mean(C3MD(70:150,:)); mean(C4MD(70:150,:)); mean(F7MD(70:150,:))];
P=[mean(C3P(40:61,:)); mean(C4P(40:61,:)); mean(F7P(40:61,:));mean(C3P(70:150,:)); mean(C4P(70:150,:)); mean(F7P(70:150,:))];
%atribuindo -1 -1 como alvo de ME, 11 para MD e 1-1 para P
A= [-ones(12,1) -ones(12,1)];
D= [ones(12,1) ones(12,1)];
C= [ones(12,1) -ones(12,1)];
y= [A;D;C];
EEG=[ME MD P];
r1 = randi([1 36],1,36);
ry=y(r1,:);
rq= EEG(:,r1);

%% Normalizar
[sm sn]= size(rq);
mrq=mean(rq');
B=zeros(sm,sn);
B(1,:)=rq(1,:)-mrq(1);
B(2,:)=rq(2,:)-mrq(2);
B(3,:)=rq(3,:)-mrq(3);
B(4,:)=rq(4,:)-mrq(4);
B(5,:)=rq(5,:)-mrq(5);
B(6,:)=rq(6,:)-mrq(6);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end
end
%% separa os dados para validação
x1=B(:,1:9);
x2=B(:,10:18);
x3=B(:,19:27);
x4=B(:,28:end);
x5=x1;
x6=x2;
x7=x3;

y1=ry(1:9,:);
y2=ry(10:18,:);
y3=ry(19:27,:);
y4=ry(28:end,:);
y5=y1;
y6=y2;
y7=y3;

entrada={x1 x2 x3 x4 x5 x6 x7};
saida={y1 y2 y3 y4 y5 y6 y7};

for j=1:4
    %validação cruzada
    treino=[entrada(:,j+1) entrada(:,j+2) entrada(:,j+3)];
    treinomao=[saida(:,j+1) saida(:,j+2) saida(:,j+3)];
    teste= [entrada(:,j)];

```



```

testemao=[saida(:,j)];
for i=2:50
    rede = newff(minmax(B),[i 2],{'tansig' 'satlins'});
    rede = train(rede,treino,treinomao);
    Ctrain = sim(rede,treino);
    Ctrain=round(Ctrain);
    cont_acerto_ME = 0;
    cont_acerto_MD = 0;
    cont_acerto_P = 0;
    for k= 1:27
        if Ctrain(1,k) == -1 & Ctrain (2,k) == -1 & treinomao(1,k) == -1 & treinomao(2,k) == -1
            cont_acerto_ME = cont_acerto_ME + 1;
        elseif Ctrain(1,k) == 1 & Ctrain (2,k) == 1 & treinomao(1,k) == 1 & treinomao(2,k) == 1
            cont_acerto_MD = cont_acerto_MD + 1;
        elseif Ctrain(1,k) == 1 & Ctrain (2,k) == -1 & treinomao(1,k) == 1 & treinomao(2,k) == -1
            cont_acerto_P = cont_acerto_P + 1;
        end
    end
    total_acertos1 = cont_acerto_ME + cont_acerto_MD + cont_acerto_P;
    Y1(j,i) = (total_acertos1/27)*100;
    Y = sim(rede,teste);
    Y=round(Y);
    cont_acerto_ME_teste = 0;
    cont_acerto_MD_teste = 0;
    cont_acerto_P_teste = 0;
    for k= 1:9
        if Y(1,k) == -1 & Y(2,k) == -1 & testemao(1,k) == -1 & testemao(2,k) == -1
            cont_acerto_ME_teste = cont_acerto_ME_teste + 1;
        elseif Y(1,k) == 1 & Y(2,k) == 1 & testemao(1,k) == 1 & testemao(2,k) == 1
            cont_acerto_MD_teste = cont_acerto_MD_teste + 1;
        elseif Y(1,k) == 1 & Y(2,k) == -1 & testemao(1,k) == 1 & testemao(2,k) == -1
            cont_acerto_P_teste = cont_acerto_P_teste + 1;
        end
    end
    total_acertos1_teste = cont_acerto_ME_teste + cont_acerto_MD_teste + cont_acerto_P_teste;
    Y2(j,i) = ((total_acertos1_teste)/9)*100;

end
end
Mediateste= mean(Y2,1)
desvioteste= std(Y2,1)
Mediatreino= mean(Y1,1)
desviotreino= std(Y1,1)
subplot(211)
bar(Mediatreino,0.5,'b'); hold on
errorbar([1:50],Mediatreino,desviotreino,'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos treino');
xlabel('NÃºmero de neurÃ³nios');
ylabel('Acertos (%)');
set(gca,'FontSize',15)
subplot(212)
bar(Mediateste,0.5,'b'); hold on
errorbar([1:50],Mediateste,desvioteste,'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos testes');
xlabel('NÃºmero de neurÃ³nios');
ylabel('Acertos (%)');
set(gca,'FontSize',15)

```

## ANEXO O — CÓDIGO DA REGULARIZAÇÃO BAYESIANA PARA CLASSIFICAÇÃO DO BCI *COMPETITION III*

```

clc
clear all
load EEGFFT2

ME=[mean(C3ME(40:61,:)); mean(C4ME(40:61,:)); mean(F7ME(40:61,:));mean(C3ME(70:150,:)); mean(C4ME(70:150,:)); mean(F7ME(70:150,:))];
MD=[mean(C3MD(40:61,:)); mean(C4MD(40:61,:)); mean(F7MD(40:61,:));mean(C3MD(70:150,:)); mean(C4MD(70:150,:)); mean(F7MD(70:150,:))];
P=[mean(C3P(40:61,:)); mean(C4P(40:61,:)); mean(F7P(40:61,:));mean(C3P(70:150,:)); mean(C4P(70:150,:)); mean(F7P(70:150,:))];
%atribuindo -1 -1 como alvo de ME, 11 para MD e 1-1 para P
A= [-ones(12,1) -ones(12,1)];
D= [ones(12,1) ones(12,1)] ;
C= [ones(12,1) -ones(12,1)];
y= [A;D;C];
EEG=[ME MD P];
r1 = randi([1 36],1,36);
ry=y(r1,:);
rq= EEG(:,r1);

%% Normalizar
[sm sn]= size(rq);
mrq=mean(rq');
B=zeros(sm,sn);
B(1,:)=rq(1,:)-mrq(1);
B(2,:)=rq(2,:)-mrq(2);
B(3,:)=rq(3,:)-mrq(3);
B(4,:)=rq(4,:)-mrq(4);
B(5,:)=rq(5,:)-mrq(5);
B(6,:)=rq(6,:)-mrq(6);
mx=max(B');
mi=min(B');
for i=1:sm
    for j=1:sn
        if B(i,j)> 0
            B(i,j)=B(i,j)/mx(i);
        elseif B(i,j)< 0
            B(i,j)=B(i,j)/(-mi(i));
        else
            B(i,j)=B(i,j);
        end
    end
end

%% separa os dados para valida
x1=B(:,1:9);
x2=B(:,10:18);
x3=B(:,19:27);
x4=B(:,28:end);
x5=x1;
x6=x2;
x7=x3;

y1=ry(1:9,:);
y2=ry(10:18,:);
y3=ry(19:27,:);
y4=ry(28:end,:);
y5=y1;
y6=y2;
y7=y3;

entrada={x1 x2 x3 x4 x5 x6 x7};
saida={y1 y2 y3 y4 y5 y6 y7};

for j=1:4
    %valida cruzada
    treino=[entrada(:,j+1) entrada(:,j+2) entrada(:,j+3)];
    treinomao=[saida(:,j+1) saida(:,j+2) saida(:,j+3)];

```

```

teste= [entrada(:,j)];
testemao=[saida(:,j)];
for i=2:50
    rede = newff(minmax(B),[i 2],[ 'tansig' 'satlins' ],'trainbr ');
    rede = train(rede, treino, treinomao);
    Ctrain = sim(rede, treino);
    Ctrain=round(Ctrain);
    cont_acerto_ME = 0;
    cont_acerto_MD = 0;
    cont_acerto_P = 0;
    for k= 1:27
        if Ctrain(1,k) == -1 & Ctrain (2,k) == -1 & treinomao(1,k) == -1 & treinomao(2,k) == -1
            cont_acerto_ME = cont_acerto_ME + 1;
        elseif Ctrain(1,k) == 1 & Ctrain (2,k) == 1 & treinomao(1,k) == 1 & treinomao(2,k) == 1
            cont_acerto_MD = cont_acerto_MD + 1;
        elseif Ctrain(1,k) == 1 & Ctrain (2,k) == -1 & treinomao(1,k) == 1 & treinomao(2,k) == -1
            cont_acerto_P = cont_acerto_P + 1;
        end
    end
    total_acertos1 = cont_acerto_ME + cont_acerto_MD + cont_acerto_P;
    Y1(j, i) = (total_acertos1/27)*100;
    Y = sim(rede, teste);
    Y=round(Y);
    cont_acerto_ME_teste = 0;
    cont_acerto_MD_teste = 0;
    cont_acerto_P_teste = 0;
    for k= 1:9
        if Y(1,k) == -1 & Y(2,k) == -1 & testemao(1,k) == -1 & testemao(2,k) == -1
            cont_acerto_ME_teste = cont_acerto_ME_teste + 1;
        elseif Y(1,k) == 1 & Y(2,k) == 1 & testemao(1,k) == 1 & testemao(2,k) == 1
            cont_acerto_MD_teste = cont_acerto_MD_teste + 1;
        elseif Y(1,k) == 1 & Y(2,k) == -1 & testemao(1,k) == 1 & testemao(2,k) == -1
            cont_acerto_P_teste = cont_acerto_P_teste + 1;
        end
    end
    total_acertos1_teste = cont_acerto_ME_teste + cont_acerto_MD_teste + cont_acerto_P_teste;
    Y2(j, i) = ((total_acertos1_teste)/9)*100;

end
end
Mediateste= mean(Y2,1)
desvioteste= std(Y2,1)
Mediatreino= mean(Y1,1)
desviotreino= std(Y1,1)
subplot(211)
bar(Mediatreino,0.5, 'b'); hold on
errorbar([1:50], Mediatreino, desviotreino, 'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos treino ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca, 'FontSize', 15)
subplot(212)
bar(Mediateste,0.5, 'b'); hold on
errorbar([1:50], Mediateste, desvioteste, 'r');hold off
ylim([30 100])
xlim([0 51])
title('MÃ©dia dos testes ');
xlabel('NÃºmero de neurÃ³nios ');
ylabel('Acertos (%)');
set(gca, 'FontSize', 15)

```