CENTRO UNIVERSITÁRIO SENAI CIMATEC

PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL E TECNOLOGIA INDUSTRIAL

Mestrado em Modelagem Computacional e Tecnologia Industrial

Dissertação de Mestrado

Modelo Computacional Baseado em Interface Cérebro-Computador Para Extração de Características e Classificação de Sinais EEG

Apresentada por: Osmar Ferreira Gomes
Orientador: Prof. Dr. Alex Álisson Bandeira Santos
Co-orientador: Prof. Dr. Oberdan Rocha Pinheiro

Agosto de 2020

Osmar Ferreira Gomes

Modelo Computacional Baseado em Interface Cérebro-Computador Para Extração de Características e Classificação de Sinais EEG

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial, Curso de Mestrado em Modelagem Computacional e Tecnologia Industrial do CENTRO UNIVERSITÁRIO SENAI CIMATEC, como requisito parcial para a obtenção do título de **Mestre em Modelagem Computacional e Tecnologia Industrial**.

Área de conhecimento: Sistemas complexos

Orientador: Prof. Dr. Alex Álisson Bandeira Santos
CENTRO UNIVERSITÁRIO SENAI CIMATEC

Co-orientador: Prof. Dr. Oberdan Rocha Pinheiro
CENTRO UNIVERSITÁRIO SENAI CIMATEC

Salvador, BA
CENTRO UNIVERSITÁRIO SENAI CIMATEC
2020

Ficha catalográfica elaborada pela Biblioteca do Centro Universitário SENAI CIMATEC

G633m Gomes, Osmar Ferreira

Modelo computacional baseado em interface cérebro-computador para extração de características e classificação de sinais EEG / Osmar Ferreira Gomes – Salvador, 2020.

125 f. : il. color.

Orientador: Prof. Dr. Alex Álisson Bandeira Santos.
Coorientador: Prof. Dr. Oberdan Rocha Pinheiro.

Dissertação (Mestrado em Modelagem Computacional e Tecnologia Industrial) – Programa de Pós-Graduação, Centro Universitário SENAI CIMATEC, Salvador, 2020.
Inclui referências.

1. Análise de componentes independentes - ICA. 2. Eletroencefalograma. 3. Interface cérebro-computador. I. Centro Universitário SENAI CIMATEC. II. Santos, Alex Álisson Bandeira. III. Pinheiro, Oberdan Rocha. IV. Título.

CDD: 620.00113

NDI - 03

Nota sobre o estilo do PPGMCTI

Esta dissertação de mestrado foi elaborada considerando as normas de estilo (i.e. estéticas e estruturais) propostas aprovadas pelo colegiado do Programa de Pós-graduação em Modelagem Computacional e Tecnologia Industrial e estão disponíveis em formato eletrônico (*download* na Página Web http://ead.fieb.org.br/portal_faculdades/dissertacoes-e-teses-mcti.html ou solicitação via e-mail à secretaria do programa) e em formato impresso somente para consulta.

Ressalta-se que o formato proposto considera diversos itens das normas da Associação Brasileira de Normas Técnicas (ABNT), entretanto opta-se, em alguns aspectos, seguir um estilo próprio elaborado e amadurecido pelos professores do programa de pós-graduação supracitado.

Sistema FIEB | SENAI CIMATEC
PELO FUTURO DA INOVAÇÃO

CENTRO UNIVERSITÁRIO SENAI CIMATEC

Mestrado Acadêmico em Modelagem Computacional e Tecnologia Industrial

A Banca Examinadora, constituída pelos professores abaixo listados, aprova a Defesa de Mestrado, intitulada **"**Modelo Computacional Baseado em Interface Cérebro-Computador para Extração de Características e Classificação de Sinais EEG" apresentada no dia 18 de agosto de 2020, como parte dos requisitos necessários para a obtenção do Título de Mestre em Modelagem Computacional e Tecnologia Industrial.

Assinado digitalmente por: Alex Alisson Bandeira Santos
O tempo: 20-08-2020 10:09:14

Orientador: **Prof. Dr. Alex Álisson Bandeira Santos**
SENAI CIMATEC

Assinado digitalmente por: Oberdan Rocha Pinheiro
O tempo: 18-08-2020 15:53:57

Coorientador: **Prof. Dr. Oberdan Rocha Pinheiro**
SENAI CIMATEC

ROBERTO LUIZ SOUZA MONTEIRO:45488193553
Assinado de forma digital por ROBERTO LUIZ SOUZA MONTEIRO:45488193553
Dados: 2020.08.19 14:49:05 -03'00'

Membro Interno: **Prof. Dr. Roberto Luiz Souza Monteiro**
SENAI CIMATEC

JOSEMAR RODRIGUES DE SOUZA:10504060520
Assinado de forma digital por JOSEMAR RODRIGUES DE SOUZA:10504060520
Dados: 2020.08.18 16:16:02 -03'00'

Membro Externo: **Prof. Dr. Josemar Rodrigues de Souza**
UNEB

Agradecimentos

Eis que a dissertação terminou, todo trabalho desse tipo é uma obra de fôlego e para chegar até aqui muitas pessoas colaboraram. Nesse sentido quero começar agradecendo aos meus pais, (in memoriam), que souberam incentivar em mim e em todos os meus irmãos o valor dos estudos e da boa educação. Quero agradecer também a um grande professor e amigo que tive no Colégio Paulo Afonso (COLEPA), Dr. Orlando Passos de Azevedo, este professor soube mais do que ninguém colocar em seus alunos o gosto e o prazer de fazer ciência. Agradeço aos professores do mestrado PPGMCTI do CIMATEC, agradeço a todos os professores da minha banca de qualificação pela paciência de ler e comentar todas as páginas que poderiam ser melhoradas. Eles, com suas experiências e seus conselhos abalizados, melhoraram ainda mais este trabalho no qual devotamos tanto tempo e tanto interesse. Ao meu amigo Professor Doutor Dionísio Carmo Neto pela ajuda na tradução para o inglês. Ao meu amigo e irmão Professor Doutor Valter da Conceição Rosa com quem, desde 1973 quando nos conhecemos na UFBA, jogando uma partida de xadrez, nunca fiquei uma semana sequer, sem falar sobre ciência, desenvolvimento científico, novas tecnologias e possibilidades no campo da informática. As conversas que mantivemos neste período em que me dediquei a escrever esta dissertação enriqueceram bastante tudo o que ora estamos apresentando.

Ao meu Professor, Doutor Alex Álisson Bandeira Santos, meu orientador, um agradecimento especial por ter confiado em mim e acreditado no meu projeto desde o primeiro dia em que estive em sua sala e falei da minha disposição para retornar ao mestrado. Ao meu Co-Orientador, Professor Doutor Oberdan Rocha Pinheiro, pela abnegação e por ter me aceitado em seu grupo de pesquisa de tecnologias e sinais biológicos. Eu e o professor Oberdan nos conhecemos há bastante tempo, para muito além da amizade não poderia ter encontrado um mentor melhor para juntos encetarmos essa caminhada científica.

Ao meu filho Osmar Ferreira Gomes Filho, aluno de engenharia do Instituto Militar de Engenharia (IME) por ser um aluno abnegado, por sua capacidade de enfrentar e resolver os desafios da lida científica e por estar esperando a conclusão desta dissertação. Espero que o esforço empreendido neste trabalho o estimule a avançar sempre pelos caminhos da ciência. À minha esposa Maria Eunice Silva Gomes e à minha filha Isabela Silva Gomes, pela compreensão e por terem feito todo o esforço para facilitar esse período de muito trabalho e muitas ausências. Sem este apoio ficaria muito difícil chegar até aqui.

Finalmente, quero agradecer a todos que direta ou indiretamente contribuíram para que este trabalho de investigação científica fosse realizado da forma que pretendíamos.

Salvador, Brasil — Osmar Ferreira Gomes

28 de Junho de 2020

Resumo

Qualquer pessoa que seja tolhida de seus movimentos físicos, por um impacto ou um trauma, fica prejudicada de forma dramática. Quanto mais tempo isso permanecer ou maior for o acidente, pior. Os movimentos físicos representam a liberdade, a capacidade de expressar-se, de ser, de movimentar-se e de revelar-se como pessoa. Sucede que, a Incapacidade física, parcial ou total produz um elevado custo social e, uma elevada carga econômica para os familiares. Tais acometimentos levam os pacientes a perderem suas qualidades de vida de modo inevitável e incontestável. Epidemiologicamente, observa-se que o número de pessoas acometidas com doenças neurológicas têm crescido devido a diversos fatores oriundos da sociedade moderna. A ciência contemporânea — incluindo: medicina, neurológica, biomecânica, robótica, informática, física e engenharias — vem progredindo expressivamente no desenvolvimento de conhecimento multidisciplinar. O principal foco tem sido trazer esses pacientes acometidos dessas enfermidades a um patamar superior de qualidade de vida. Isso significa maior cognição, mais liberdade e mais movimentos controlados e controláveis. Epidemiologicamente, observa-se que o número de pessoas acometidas com doenças neurológicas têm crescido devido a diversos fatores oriundos da sociedade moderna. A ciência contemporânea — incluindo: medicina, neurológica, biomecânica, robótica, informática, física e engenharias — vem progredindo expressivamente no desenvolvimento de conhecimento multidisciplinar. O principal foco tem sido trazer esses pacientes acometidos dessas enfermidades a um patamar superior de qualidade de vida. Isso significa maior cognição, mais liberdade e mais movimentos controlados e controláveis. Esta dissertação desenvolve um modelo matemático baseado em interface cérebro-computador que permite o desenvolvimento de novas tecnologias tomando como base sinais de eletroencefalogramas. Tal esforço teórico reflete a necessidade de se produzir cadeiras de rodas mais inteligentes, interfaces flexíveis [para se adaptar a distintos indivíduos]. Num extremo, tentando-se eliminar o ruído de sinais indesejados (os artefatos), se terá a essência, o precioso teor dos sinais pensados que conseguem expressar a origem das variações cerebrais que podem movimentar uma cadeira, por exemplo. Ademais, a cadeira é incômoda, requer ajuda de outros e, precisa de alterações nas estruturas arquitetônicas, para se dizer o mínimo. O método utilizado para essa modelagem baseou-se em usar os sinais de eletroencefalogramas do banco de dados *eegmmidb – EEG Motor Movement-Imagery Dataset,* que está disponível. Tal amostra contém sinais de eletroencefalogramas de 105 indivíduos. Esses sinais foram utilizados para validação do modelo computacional proposto. Os resultados obtidos da simulação, mostraram que o modelo consegue explicar algo como 75% dos movimentos, quando os outros semelhantes alcançavam, no máximo 50%. Assim, este modelo-interface cérebro-computador ofereceu maior robustez revelando que os esforços feitos aqui foram proveitosos. Mais que isso, que outras tentativas podem ser feitas a partir das suposições inerentes ou das associações feitas na construção desse modelo.

Palavras chave: Análise de Componentes Independentes, Eletroencefalograma, Interface cérebro-computador

Abstract

Anyone who is hampered by their physical movements, due to an impact or trauma, they are dramatically impaired. The longer this remains or the greater the accident, the worse. Physical movements represent freedom, the ability to express oneself, to be, to move and to reveal oneself as a person. It turns out that physical disability, partial or total, produces a high social cost and a high economic burden for family members. Such afflictions lead patients to lose their quality of life in an inevitable and indisputable way. Epidemiologically, it is observed that the number of people affected with neurological diseases have grown due to several factors originating in modern society. Contemporary science - including: medicine, neurology, biomechanics, robotics, computers, physics and engineering - has progressed significantly in the development of multidisciplinary knowledge. The main focus has been to bring these patients affected by these diseases to a higher level of quality of life. This means greater cognition, more freedom and more controlled and controllable movements. This essay develops a mathematical model based on a brain-computer interface that allows the development of new technologies based on electroencephalogram signals. Such theoretical effort reflects the need to produce more intelligent wheelchairs, flexible interfaces [to adapt to different individuals]. In an extreme, trying to eliminate the noise of unwanted signals (the artifacts), we will have the essence, the precious content of the thought signals that can express the origin of the brain variations that can move a chair, for example. Furthermore, the chair is uncomfortable, requires help from others, and needs changes in architectural structures, to say the least. The method used for this modeling was based on using the electroencephalogram signals from the eegmmidb database - EEG Motor Movement-Imagery Dataset, which is available. Such sample contains signs of electroencephalograms from 105 individuals. These signals were used to validate the computational model of a proposed model. The results obtained from the simulation showed that the model can explain something like 75% of the movements, when similar ones reached, at most, 50%. Thus, this brain-computer interface model offered greater robustness, revealing that the efforts made here were fruitful. More than that, that other attempts can be made based on the inherent assumptions or associations made in the construction of this model.

Keywords: Independent Component Analysis, Electroencephalography, Brain-Computer Interface

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Siglas

ABNT	*Associação Brasileira de Normas Técnicas*
ACO	*Auto-Condicionamento do Operador*
API	*Application Programming Interface*
AVC	*Acidente Vascular Cerebral*
BCI	*Brain Computer Interface o mesmo que: ICC - Interface Cérebro Computador*
BD	*Banco de Dados*
BSS	*Blind Source Separation (Separação Cega de Fontes)*
CDF	*Função de Distribuição Cumulativa*
CT	*Tumografia Computadorizada*
DCCA	*Detrended Cross-Correlation Analysis*
DFA	*Detrended Fluctuation Analysis*
EDF	*European Data Format*
EEG	*Eletroencefalograma*
EMG	*Eletromiografia - Sinais Elétricos dos Músculos*
EOG	*Eletro-oculografia - Sinais Elétricos do Movimento dos Olhos*
EOS	*Estatística de Ordem Superior, o mesmo que HOS*
EVD	*Eigenvalue Decomposition*
FastICA	*yvärinen's Fixed Point Algorithm*
FFT	*Fast Fourier Transform*
fMRI	*Imagem por Ressonância Magnética Funcional*
GUI	*Graphical User Interface*
Hz	*Unidade de Frequência - Hertz*
IBGE	*Instituto Brasileiro de Geografia e Estatística*
ICA	*Análise de Componentes Independentes*
ICC	*Interface Cérebro-Computador a mesma coisa BCI*
JADE	*joint Approximation Diagonization of Eigenmatrices*
MLP	*Muilti Layer Perceptron*
OMS	*Organização Mundial da Saúde*
ONU	*Organização das Nações Unidas*
PA	*Potencial de Ação*
PCA	*Análise de Componentes Principais*
PPGMCTI ..	*Programa de Pós-gradação em Modelagem Computacional e Tecnologia Industrial*
RNA	*Rede Neural Artificial*
RMS	*Root Mean Square - Valor Eficaz*
RP	*Reconhecimento de Padrões*
SOBI	*Second-Order Blind Identification*
SVM	*Suport Vector Machine*
UFBA	*Universidade Federal da Bahia*

Capítulo Um

Introdução

Esse trabalho de investigação, embora, possa ser aplicado para outras importantes aplicações, tem como finalidade maior, melhorar as condições sociais de pessoas acometidas de graves enfermidades na sua mobilidade, dificultando desta maneira levar vidas sociais plenas.

O ser humano tem necessidade de interagir com o ambiente ao seu redor. Entretanto, pessoas com mobilidade reduzida acabam modificando sua rotina passando a ocupar-se de atividades pouco ativas, reduzindo seu desempenho físico, suas habilidades motoras e sua capacidade de coordenação. Esses efeitos não favorecem à manutenção de um estilo de vida saudável, levando essas pessoas ao isolamento social e à solidão. O aumento das atividades sociais voltadas para indivíduos com mobilidade reduzida, cada vez mais, revela a insuficiência funcional dos equipamentos atuais direcionados para este grupo de indivíduos. Segundo Pinheiro *et al.* (2018), a imaginação do movimento é uma estratégia utilizada em interfaces cérebro-computador que permite à pessoa com graves comprometimentos motores ter algum controle sobre dispositivos inteligentes.

Segundo a Organização Mundial da Saúde (OMS), existem no mundo um bilhão de pessoas com deficiências, representando cerca de 15% da população do planeta. Deste total cerca de 190 milhões de adultos sofrem dificuldades importantes de mobilidade. Há uma estimativa de que cerca de 93 milhões de crianças menores de 15 anos de idade vivem com alguma deficiência moderada ou grave. Ainda segundo a OMS, o número de pessoas que sofrem de deficiências continuará a aumentar à medida que as populações envelhecem e com o aumento global das doenças crônicas. A Assembleia Geral das Nações Unidas (ONU) salientou que cerca de 80% das pessoas com deficiências vivem em países em desenvolvimento e reiterou a necessidade de assegurar que as pessoas com deficiências estejam incluídas em todos os aspectos do desenvolvimento (WHO, 2015).

No Brasil, segundo o IBGE, a proporção de pessoas de 18 anos ou mais de idade, que usavam algum recurso para auxiliar na locomoção, tais como muletas, bengalas ou cadeira de rodas foi de 2.5%. Por região, este percentual variou de 1.7%, na região norte a 2,8%, na região nordeste. Mesmo com o uso de recursos para auxiliar a locomoção, 2,7% das pessoas não conseguiam ou tinham grande dificuldade para se locomover (IBGE, 2014).

Além disso, existem as vítimas de AVC: 15 milhões de pessoas apresentam AVC por ano; destas, cinco milhões morrem em decorrência do evento e grande parte dos sobreviventes apresenta sequelas físicas e / ou mentais, sendo que muitas delas permanecerão em um

cadeira de rodas até o final de suas vidas (RANGEL *et al.*, 2013). Existem ainda pessoas com paralisia cerebral, esclerose lateral amiotrófica, esclerose múltipla, distrofia muscular e outras doenças que atacam o sistema neuromotor.

Nesse sentido, pesquisas científicas que contribuam para melhorar a integração social de pessoas com essas deficiências é de grande utilidade social pois, as tornam mais independentes e contribuem para melhorar de forma significativa as suas vidas sociais e privadas.

1.1 Definição do Problema

As vias neuromusculares são canais de comunicação entre o cérebro e os músculos. Quando uma pessoa deseja realizar um determinado movimento, é através dessas vias que o cérebro envia sinais aos músculos do corpo para realizar o movimento; contudo, em alguns casos, essa comunicação é interrompida devido a danos provocados por doenças que interrompem as vias através das quais o cérebro controla o corpo. Dentre elas, estão a Esclerose Lateral Amiotrófica (ELA), Acidente Vascular Cerebral (AVC), lesão do cérebro ou da medula espinhal, paralisia cerebral, distrofia muscular e esclerose múltipla (WOLPAW; WOLPAW, 2012).

De acordo com Gomes *et al.* (2019) o desenvolvimento de pesquisas que contribuam para amplificação da mobilidade dessas pessoas torna-se essencial para sua ressocialização. O impacto social do uso de um sistema para auxílio à mobilidade de pessoas com comprometimentos motores e neurológicos severos seria significativo, devido ao alto número de portadores de deficiência motora existente. Esse fato justifica o interesse cada vez maior pelo desenvolvimento de tecnologias para auxílio à mobilidade.

É no contexto de casos de incapacidade motora severa, nos quais as pessoas são incapazes de executar qualquer ação motora, mas preservam suas capacidades cognitivas, que este trabalho de investigação se insere. Nessas situações, os sinais cerebrais, tratados através das interface cérebro-computador, representam uma opção capaz de promover a interação da pessoa com o meio em que ela vive.

1.2 Objetivos

Geral:

O objetivo geral deste trabalho de investigação científica é desenvolver uma Interface cérebro-computador tendo como base um mecanismo para a minimização de artefatos, a

partir do *framework* desenvolvido no trabalho de Pinheiro (2016).

Específicos:

Os objetivos específicos são:

- Implementar o módulo de pré-processamento do modelo, tendo como principal atividade o emprego da técnica de eliminação dos artefatos para que sinais mentais imaginados, não sofram influências de outros sinais indesejados. Nesta investigação serão considerados artefatos: as frequências de 50/60 Hz dos aparelhos elétricos, os sinais elétricos dos movimentos dos olhos (EOG), as batidas do coração, e os sinais elétricos dos movimentos musculares (EMG).
- Avaliar a intensidade do sinal dos 64 eletrodos, para identificar a potência média relativa dos sinais imaginados com base no *dataset: eegmmidb - EEG Motor Movement/Imagery Dataset*, objetivando, analisar os sinais mais relevantes.
- Avaliar experimentalmente o modelo proposto.

1.3 Organização da Dissertação de Mestrado

Esta dissertação está dividida em cinco capítulos da seguinte forma:

- **Capítulo 1** - Introdução: Contextualiza o ambiente no qual essa pesquisa foi desenvolvida. Apresenta portanto, a definição do problema, objetivos e justificativas dessa pesquisa e como esta dissertação de mestrado está estrurada.
- **Capítulo 2** - Interface cérebro-computador: Aqui são apresentados os conceitos fundamentais sobre sinais cerebrais, e serão discutidos os conceitos de artefatos, Separação Cega de Fontes (BSS em inglês), Análise de Componentes Independentes (ICA em inglês) e o conceito de branqueamento. Serão descritos também os principais algoritmos de eliminação de artefatos e justificará porque nessa pesquisa foi adotado um determinado algoritmo.
- **Capítulo 3** - Metodologia e Desenvolvimento do Modelo: Este é o capítulo onde se descreve a metodologia e o desenvolvimento do modelo proposto para chegar ao resultado que confirma a hipótese inicial de otimização dos dados de saída.
- **Capítulo 4** - Avaliação e Resultados da Investigação: Neste capítulo, são apresentados os resultados experimentais dessa pesquisa. Aqui também será descrito o treinamento da Rede Neural Artificial (RNA) e seus parâmetros. Portanto, o capítulo três e o quatro se somam para o entendimento desta dissertação de mestrado,
- **Capítulo 5** - Conclusões e Considerações Finais: Aqui são apresentadas a conclusão da pesquisa, as diversas contribuições e serão feitas sugestões sobre a continuidade futura deste trabalho.

Capítulo Dois

Interface Cérebro-Computador

2.1 Sistema Nervoso Central

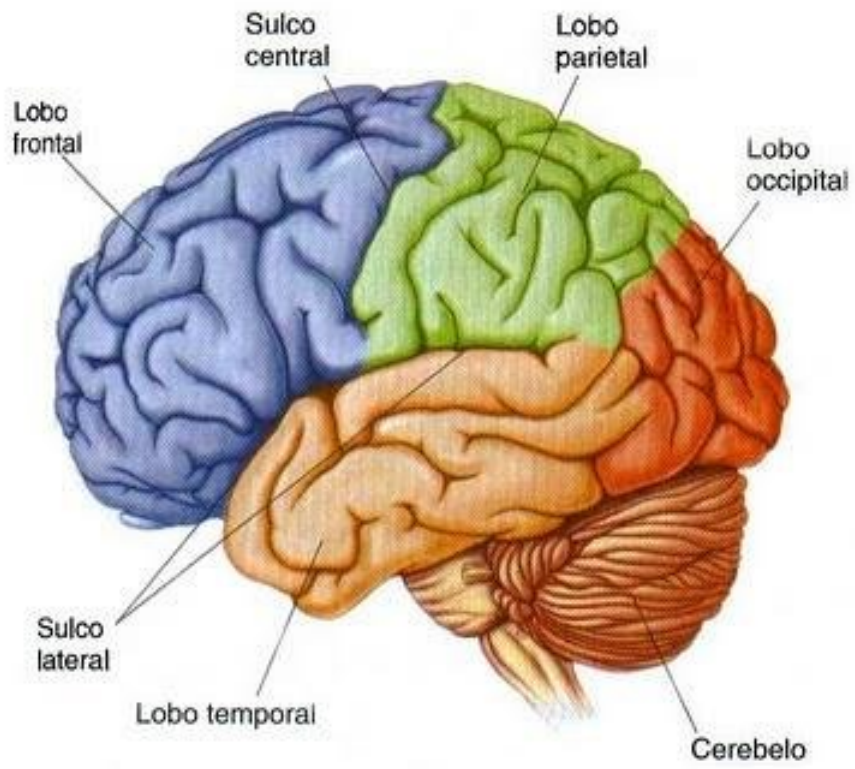
O Sistema Nervoso Central é formado por cérebro, retina e medula espinhal (AMTHOR, 2017), para este estudo vamos nos concentrar no cérebro. Ele está localizado dentro do crânio e pesa em média 1,36 kg. Até bem pouco tempo, a maioria dos livros e artigos estimavam em cem bilhões o número de células neuronais que formavam o cérebro, mais ou menos, a mesma quantidade de estrelas na nossa galáxia, Via Láctea. Entretanto, após a última revisão nesse sentido, uma equipe de pesquisadores da Universidade Federal do Rio de Janeiro (UFRJ), fez um experimento para checar esse valor e chegaram a conclusão que esse número gira em torno de 86 bilhões de neurônios que consomem cerca de 500 Kcal/dia, portanto, um quarto do total de energia consumida pelo corpo, sendo irrigado por um fluxo sanguíneo 750 ml por minuto (HERCULANO-HOUZEL, 2012).

Quando olhamos para um cérebro humano o que vemos na verdade é o neocórtex. Neo significa novo e córtex significa algo como casca ou escudo exterior. Este é dividido em dois hemisférios cerebrais e cada hemisfério para efeito de estudo está dividido em quatro áreas ou lóbulos, também pode ser chamado de lobos. Veja Figura 2.1 frontal, temporal, occipital e parietal (BANSAL; MAHAJAN, 2019).

Descrição, a saber:

- **Lobo Frontal** - É o lobo mais anterior do cérebro, ou seja, o lobo frontal está associado a áreas corticais que lidam com emoções, funções cognitivas como raciocínio, tarefas de memória, planejamento e movimento voluntário dos músculos.
- **Lobo Parietal** - O lobo parietal é anterior ao lobo occipital e responde a estímulo relacionado à sensação de dor, pressão, vibração, temperatura e toque.
- **Lobo Occipital** - O lobo posterior extremo é o lobo occipital, responsável pelos estímulos relacionados à visão e pela detecção e processamento espacial. É composto por área visual primária e secundária. A área primária recebe informações visuais preliminares que são integradas e é tarefa da área visual secundária decodificar o estímulo recebido em informações significativas, relacionando-o com o conhecimento passado.
- **Lobo Temporal** - O lobo temporal combina informação auditiva e visual. O aspecto superior (de cima) e medial (central) do lobo temporal recebe entrada auditiva da

Figura 2.1: Lóbulos Cerebrais



Fonte: (NICIDA, 2015)

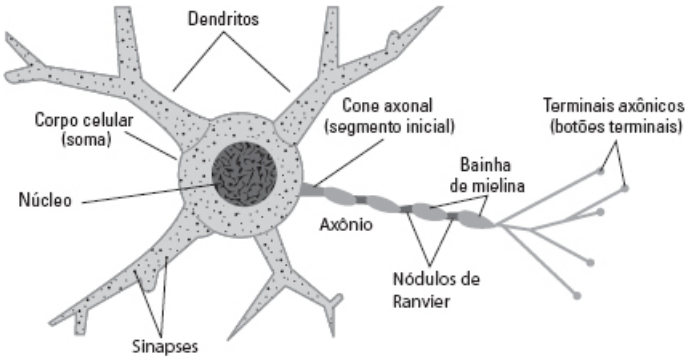
parte do tálamo que transmite informações das orelhas. A parte inferior do lobo temporal faz processamento visual para reconhecimento de objetos e padrões. As partes medial e anterior dele estão envolvidas em reconhecimento visual de ordem alta (sendo capaz de reconhecer faces, por exemplo) bem como do reconhecimento dependente da memória (AMTHOR, 2017).

2.2 Os Neurônios

Os neurônios são as principais células do sistema nervoso central, o outro tipo de célula do sistema nervoso são as células gliais ou glia. Os neurônios são as unidades sinalizadoras do sistema nervoso. Um neurônio típico tem quatro regiões morfologicamente definidas: (1) o corpo celular, (2) os dendritos, (3) o axônio e (4) os terminais pré-sinápticos (Figura 2.2). Cada região tem um papel distinto na geração de sinais e na comunicação com outras células nervosas.

O corpo celular, ou soma, é o centro metabólico da célula. Contém o núcleo, que possui os genes da célula, e o retículo endoplasmático, uma extensão do núcleo onde proteínas

Figura 2.2: Anatomia do Neurônio



Fonte: (AMTHOR, 2016)

celulares são sintetizadas. O corpo celular geralmente origina dois tipos de processos: vários dendritos curtos e um axônio longo e tubular. Os dendritos ramificam-se de forma semelhante a uma árvore e são o principal aparato para recepção de sinais aferentes[1] de outras células nervosas. O axônio tipicamente estende-se até certa distância do corpo celular e carrega sinais a outros neurônios. Um axônio pode transportar sinais elétricos por longas distâncias, de 0,1 mm a 2 m. Esses sinais elétricos, chamados potenciais de ação, são iniciados em uma zona especializada de disparo (zona de gatilho) próxima à origem do axônio, chamada segmento inicial, a partir da qual esses potenciais se propagam através do axônio sem falhas ou distorções, à velocidades de 1 a 100 m/s. A amplitude de um potencial de ação viajando pelo axônio se mantém constante a 100 mV porque o potencial de ação é um impulso tudo-ou-nada que se regenera a intervalos regulares ao longo do axônio (Figura 2.4) (KANDEL *et al.*, 2014).

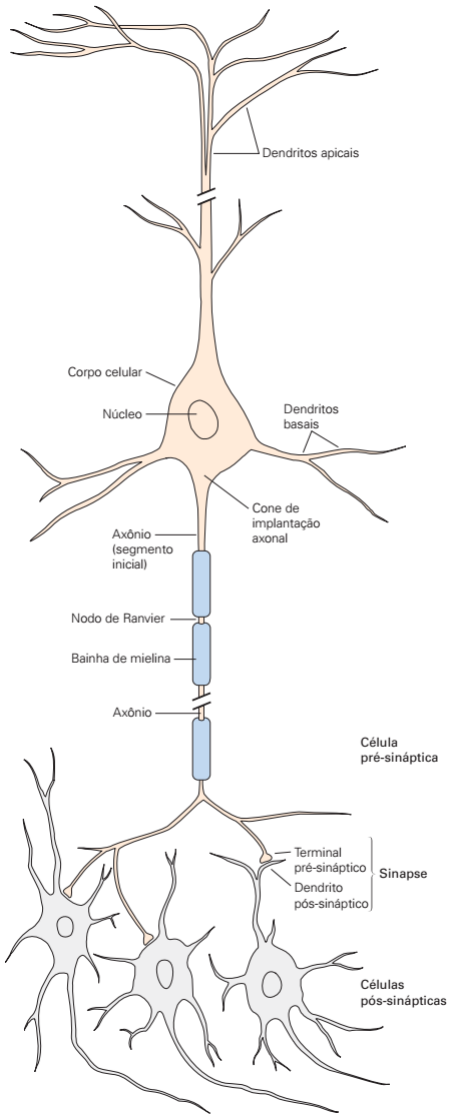
Potenciais de ação são os sinais pelos quais o encéfalo recebe, analisa e transmite a informação. Esses sinais são altamente estereotipados em todo o sistema nervoso, mesmo que iniciados por uma grande variedade de eventos ambientais que nos atingem – da luz ao contato mecânico, de odores a ondas de pressão. Os sinais que transmitem informação sobre visão são idênticos aos que carregam informação sobre odores. Eis um princípio básico da função cerebral: a informação transmitida por um potencial de ação é determinada não pela forma do sinal, mas pela via trafegada pelo sinal no encéfalo. O encéfalo analisa e interpreta os padrões de sinais elétricos aferentes e suas vias, criando nossas sensações de visão, tato, olfato e audição.

Para aumentar a velocidade de condução dos potenciais de ação, grandes axônios são enrolados em uma substância lipídica isolante, a mielina. A bainha de mielina é interrompida a intervalos regulares pelos nodos de Ranvier, pontos do axônio não isolados pela mielina, onde o potencial de ação é regenerado.

[1] Aferente, que conduz um impulso a um centro nervoso (diz-se de fibra nervosa sensível)

Próximo ao seu final, o axônio se divide em finas ramificações que contatam outros neurônios em zonas especializadas de comunicação chamadas sinapses. A célula nervosa que está transmitindo o sinal é chamada célula pré-sináptica; a célula receptora do sinal é a célula pós-sináptica. A célula pré-sináptica transmite sinais por regiões especializadas dilatadas em suas ramificações axonais, chamadas terminais pré-sinápticos ou terminais nervosos. As células pré-sinápticas e pós-sinápticas são separadas por um espaço muito estreito, a fenda sináptica. A maioria dos terminais pré-sinápticos termina nos dendritos dos neurônios pós-sinápticos, mas os terminais podem também fazer contato com o corpo celular ou, menos frequentemente, no início ou extremidade do axônio da célula receptora (Figura 2.3) (KANDEL *et al.*, 2014).

Figura 2.3: Estrutura Detalhada do Neurônio

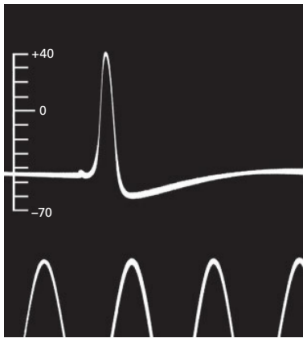


Fonte:(KANDEL *et al.*, 2014)

Este traçado histórico foi o primeiro registro intracelular de um potencial de ação publi-

cado Figura 2.4. Foi registrado em 1939, por Hodgkin e Huxley em um axônio gigante da lula, usando eletrodos capilares de vidro preenchidos com água do mar. Os pulsos temporais são separados por 2 ms. A escala vertical indica o potencial do eletrodo interno em milivolts, sendo a água do mar externa tomada como o potencial zero (KANDEL *et al.*, 2014).

Figura 2.4: Potencial de Ação (PA)



Fonte: (KANDEL *et al.*, 2014)

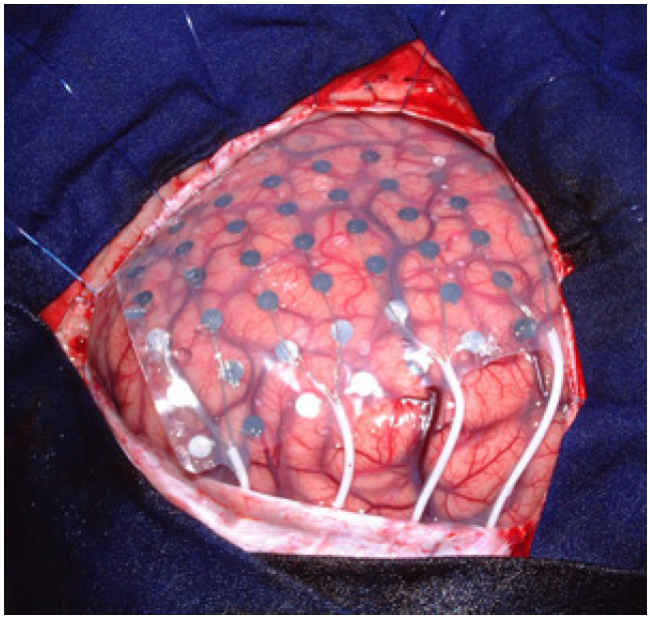
2.3 Eletroencefalografia

A eletroencefalografia é um método para medição de potenciais elétricos no escalpo (couro cabeludo) do indivíduo, resultantes de atividade elétrica neuronal no encéfalo. Os primeiros estudos relacionados aos potenciais elétricos do cérebro foram realizados no ano de 1875 pelo fisiologista inglês Richard Caton (JACKS; MILLER, 2003). Ele fez registros destes potenciais capturado no encéfalo de cães e coelhos utilizando um dispositivo com grande sensibilidade à tensão elétrica. Em 1929, o psiquiatra austríaco Hans Berger realizou os primeiros testes de EEG envolvendo seres humanos e relatou a distinção clara observada entre o EEG do sono e o da vigília (MARK *et al.*, 2008).

Segundo Wolpaw & Wolpaw (2012), Hans Berger foi o primeiro estudioso a usar EEG qualitativamente. Seu uso regular é no diagnóstico de problemas neurológicos, porém uma combinação com métodos de imaginação motora permite investigar e identificar estados mentais.

Segundo Silva & Niedermeyer (1982), a aquisição dos sinais cerebrais utiliza técnicas e métodos para o procedimento que se dividem em duas principais categorias: invasiva (Figura 2.5) e não invasiva (Figura 2.6). A técnica invasiva é utilizada quando o foco está na captação de dados com o mínimo de perda de sinais. Nesse método há uma diminuição de ruídos, Todavia, existe a necessidade de procedimento cirúrgico para que os eletrodos sejam colocados diretamente no cérebro.

Figura 2.5: Eletrodos Intracraniano - Invasivos



Fonte: (WEN, 2006)

Figura 2.6: Eletroencefalograma - não Invasivo

Fonte: (WLADIMIR, 2013)

A vantagem da utilização da técnica não invasiva é que não há necessidade da introdução de eletrodos no interior do crânio ou mesmo de qualquer tipo de procedimento cirúrgico. Esta técnica é menos precisa no que se refere à qualidade do sinal e mesmo quanto ao maior nível de ruído produzido. Por outro lado, ela é mais simples (WOLPAW, 2007). Os sinais são obtidos com uma largura de banda mais limitada, suscetíveis à sobreposição e interferências de sinais diversos. Entretanto, eles possuem aplicabilidade suficiente para ativar comandos computacionais, como por exemplo, mover uma cadeira de rodas inteligente ou mover um braço robótico em uma indústria.

Um gorro ou toca comercial com os eletrodos já instalados e distribuídos pode ser utilizado para facilitar o posicionamento dos mesmos. O sinal de eletroencefalograma é filtrado e amplificado imediatamente após a sua aquisição. A faixa de frequência para análises clínicas é de 0,1 a 50 Hz. Essa baixa amplitude é promovida principalmente, pelas camadas existentes entre o cérebro e o escalpo (WEBSTER, 2009).

Segundo Wolpaw (2007), os estudos com eletroencefalografia invasiva tivereram início na década de 1960 pelo cientista alemão Eberhard Fetz, através da análise dos neurônios localizados no cortex motor primário associados ao processo de movimento corporal. Esses estudos tinham a intenção de levar os movimentos comandados pelo cérebro a dispositivos eletrônicos.

A análise desses sinais permitiu determinar o intervalo entre o início das atividades das células do córtex e dos músculos envolvidos no processo, sendo possível analisar o impacto das regiões cerebrais nas áreas do corpo relativas às posições onde os membros do corpo se encontravam e à sua força motora.

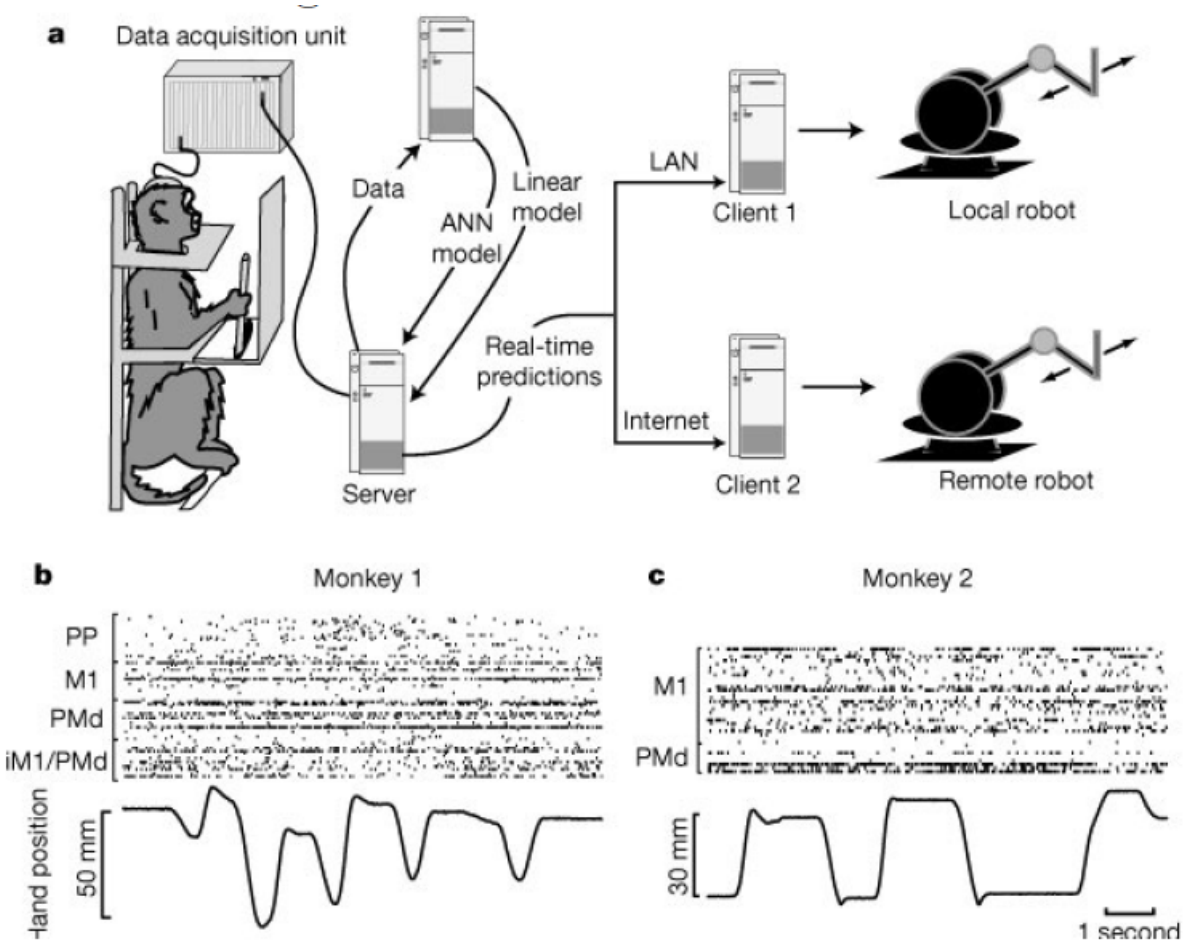
Johan *et al.* (2000) desenvolveram estudos utilizando macacos para controlar um braço robótico, mostrando ser possível também utilizar a mesma técnica em seres humanos. Essa pesquisa abriu caminho para criação de próteses humanas controladas pelo pensamento e para o desenvolvimento de novos tratamentos para pessoas paralisadas.

A Figura 2.7 ilustra o primeiro equipamento robótico usado, ou seja, um braço mecânico que realizava os movimentos pensados por uma macaca. O diagrama esquemático (a) representa o aparelho experimental utilizado para realizar a aquisição dos dados para controlar os movimentos dos dispositivos robotizados locais (LAN) e remotos (internet). (b) e (c) registram simultaneamente a atividade neural.

Schmidt (1980) desenvolveu uma pesquisa utilizando implantes cranianos para permitir o condicionamento de animais para o controle de dispositivos extracorpóreos utilizando padrões neurais. Essas pesquisas demonstraram a importância em tornar a tecnologia de eletroencefalografia aplicável a seres humanos com problemas de paralisia, indicando a possibilidade de desenvolvimento de interfaces funcionais.

Carmena *et al.* (2005) desenvolveram pesquisa utilizando macacos na qual os referidos animais foram condicionados a responder por sinais neuronais à iluminação de um conjunto de oito sinais luminosos, em que cada um estaria associado a um estímulo cerebral específico, com correspondências sucessivas entre os padrões cerebrais e as luzes. A Figura 2.8 ilustra a forma como os ensaios foram desenvolvidos. Cada ensaio iniciou-se com um alvo apresentado em uma posição aleatória na tela. O macaco tinha que usar o polo para colocar o cursor sobre o alvo; se o macaco atravessasse o alvo muito rápido,

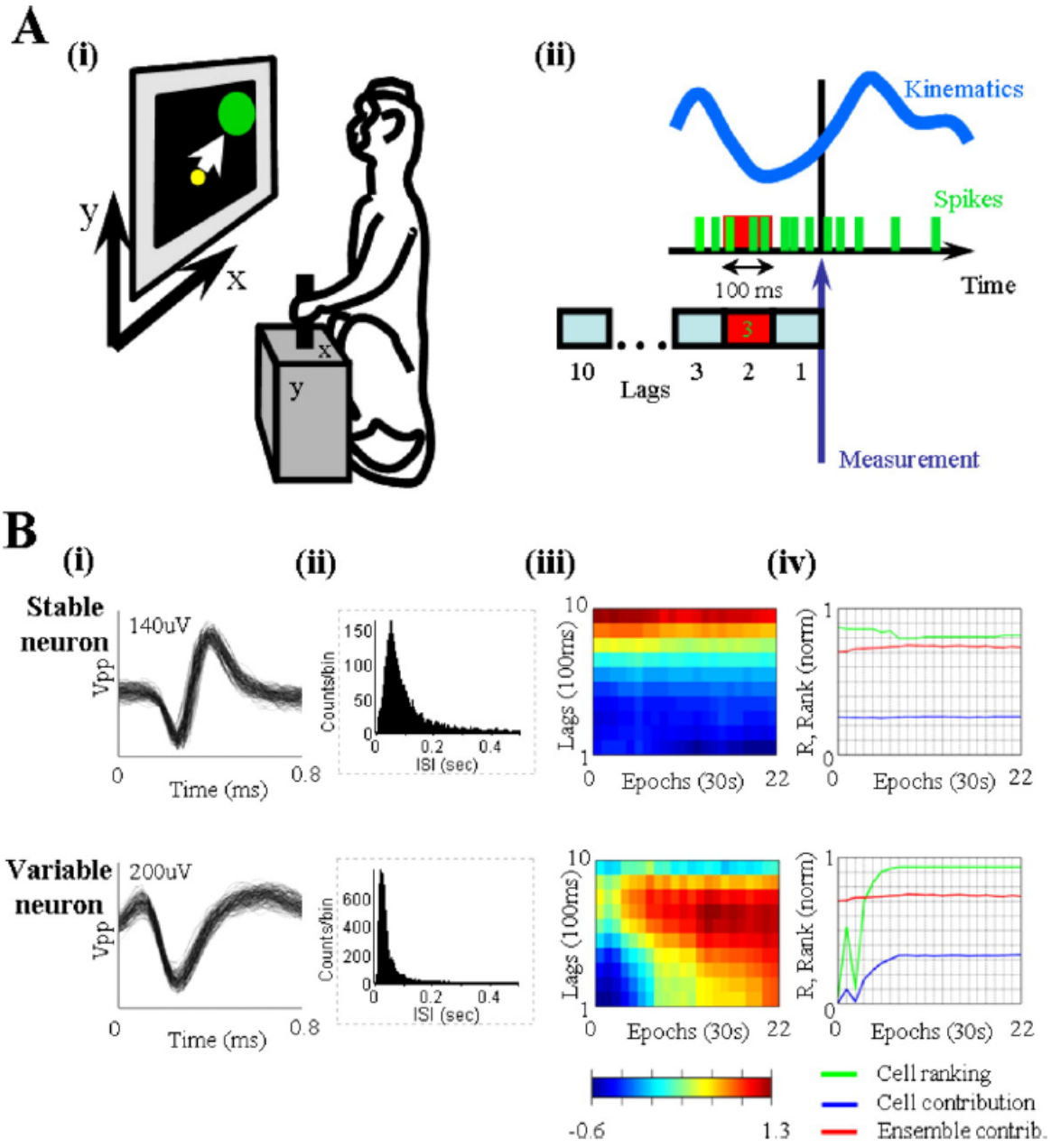
Figura 2.7: Equipamento Robótico



Fonte: (JOHAN *et al.*, 2000)

o alvo desapareceria e o resultado não era registrado. Os macacos tinham cinco segundos para acertar o alvo, com um atraso de 0,5 segundos entre os ensaios. Eles recebiam recompensas para o desempenho correto.

Figura 2.8: Design Experimental



Fonte: (CARMENA *et al.*, 2005)

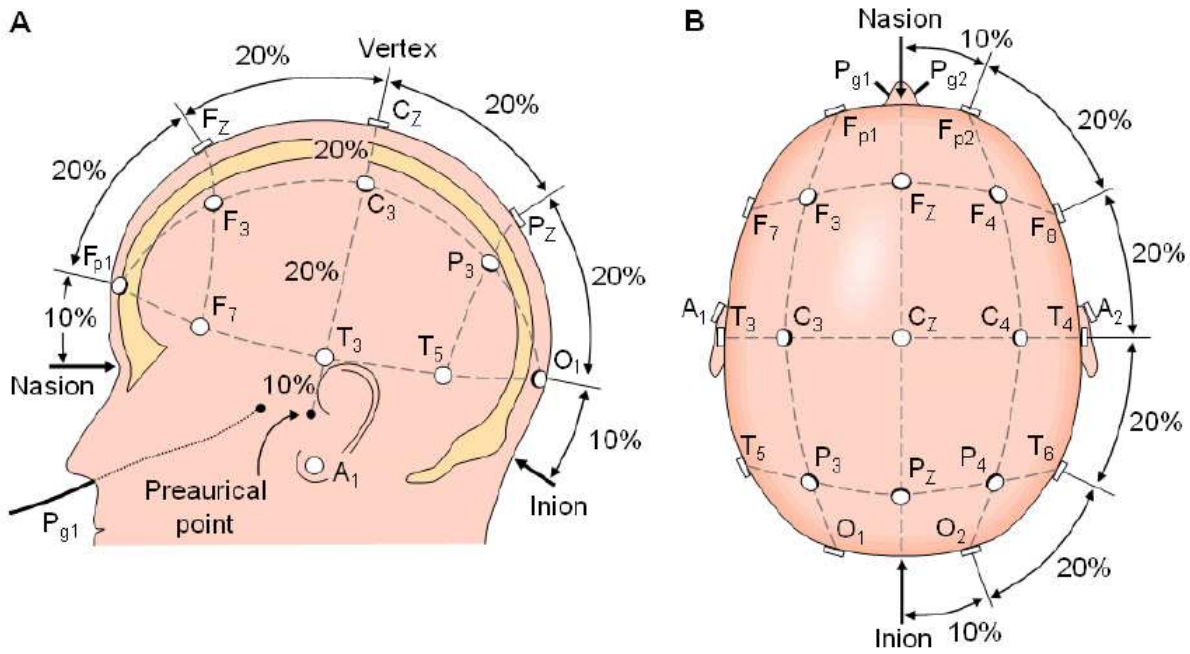
Apesar das leituras de sinais utilizando essa técnica fornecerem informações mais detalhadas sobre o funcionamento dos neurônios, a configuração do córtex cerebral pode variar de pessoa para pessoa (WEN, 2006).

Segundo Webster (2009), o sistema internacional 10-20[2] (Figura 2.9) foi proposto para assegurar a padronização do local de fixação dos eletrodos para captura dos sinais EEG, tais posições são localizadas de acordo as áreas cerebrais. dessa forma, independentemente do

[2]O Sistema internacional 10-20 é utilizado no mapeamento das posições onde serão fixados os eletrodos para registrar os sinais do Eletroencefalograma.

desenvolvimento craniano da pessoa, a posição dos eletrodos mantém-se suficientemente consistente entre os diversos tipos de biotipos pessoais (WEN, 2006).

Figura 2.9: Sistema Internacional 10-20.



Fonte: (WEBSTER, 2009)

Nesse sistema, os números pares indicam eletrodos do lado direito da cabeça e os ímpares do esquerdo. As letras: F, C, P, T e O são utilizadas para identificar as diferentes regiões do córtex, sendo frontal, central, parietal, temporal e occipital, respectivamente (WEBSTER, 2009).

Segundo Wolpaw (2007), os primeiros estudos com eletroencefalografia não invasiva foram realizados durante os anos de 1960 e permitiram que pessoas controlassem os seus níveis de concentração e atenção. As ondas cerebrais são produzidas principalmente quando estamos acordados de olhos fechados e estão associadas à inativação de áreas do córtex que não estão sendo utilizadas (WOLPAW, 2007). Assim, pessoas foram capazes de controlar os padrões das ondas alfa através de técnicas de relaxamento.

As Interfaces cérebro-computador possuem características segundo vários aspectos. No controle temporal síncrono, um estímulo externo, geralmente visual ou auditivo, indica o momento em que a pessoa deve gerar o padrão cerebral que será analisado, causando confinamento deste padrão em uma janela temporal e na maioria das vezes, melhores taxas de acerto (PFURTSCHELLER; NEUPER, 2001). Esse método gera uma dependência do indivíduo em relação ao dispositivo externo. Já no modo de operação assíncrono, a pessoa envia comandos cerebrais para a interface cérebro-computador sem que haja associação temporal a estímulos externos (MILLAN; MOURINO, 2003). Quando o processo

de aquisição dos dados, o pré-processamento do sinal, a extração de características e a classificação são realizados durante o tempo em que o indivíduo está utilizando o sensor, a interface é classificada como online (VIDAURRE *et al.*, 2006). Segundo Vidaurre *et al.* (2006), quando a aquisição dos dados é feita para análise posterior e os sinais capturados não são utilizados para gerar ações imediatas, a interface cérebro-computador é dita offline. Já na aprendizagem homem-máquina, as abordagens utilizadas são as de autocondicionamento do operador (ACO) e reconhecimento de padrões (RP). Na abordagem ACO, a pessoa recebe uma realimentação dos sinais gerados e aprende, ou se condiciona, a controlá-los de forma a produzir um sinal que será mais facilmente reconhecido pela interface cérebro-computador (BIRBAUMER *et al.*, 2000). Já na abordagem por RP, a maior carga de aprendizado, ou treinamento, está na interface cérebro-computador, que deve associar, de maneira correta, os diferentes estados mentais da pessoa nas ações pré-estabelecidas (MILLAN *et al.*, 2002).

2.4 Frequência dos Sinais Cerebrais

As frequências captadas através do eletroencefalograma representam atividades elétricas geradas no córtex cerebral e se correlacionam com estados do comportamento humano, como os níveis de atenção, sono, vigília, concentração ou processos cognitivos (PINTO, 2006).

Os sinais neurais são categorizados por faixas de frequência, sendo que cada uma delas é representada por uma letra do alfabeto grego. As frequências delta, theta, alpha, beta e gamma tem origem cortical e bandas medidas em hertz (Hz) (MARK *et al.*, 2007). A análise dessas frequências é complexa devido a grande quantidade de informações percebidas pelos eletrodos. Da Figura 2.10 até a Figura 2.14 são ilustradas as frequências das ondas: delta theta, alpha, beta e gamma respectivamente.

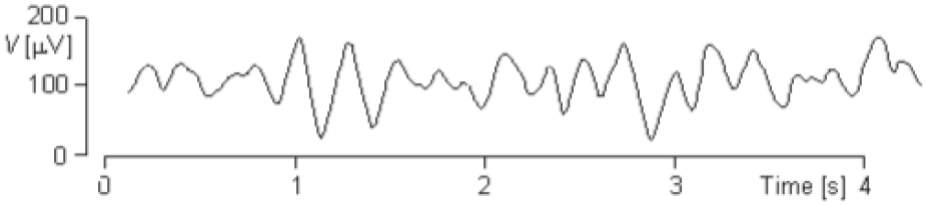
O ritmo das ondas delta variam de 0,1 Hz a 4 Hz, normalmente não ocorrem em adultos durante um estado de vigília. São observadas durante o sono em crianças com menos de um ano de idade (MARK *et al.*, 2007).

Figura 2.10: Frequência da Onda Delta.

Fonte: (MARK *et al.*, 2007)

A onda theta tem frequências entre 4 Hz a 8 Hz, está relacionado à sonolência ou sono leve. É observado assim que a pessoa acorda ou imediatamente antes de dormir. Em geral ocorre nas regiões parietal e temporal em crianças. Em adultos, esse ritmo se manifesta em situações de estresse mental como: desapontamentos ou frustrações.

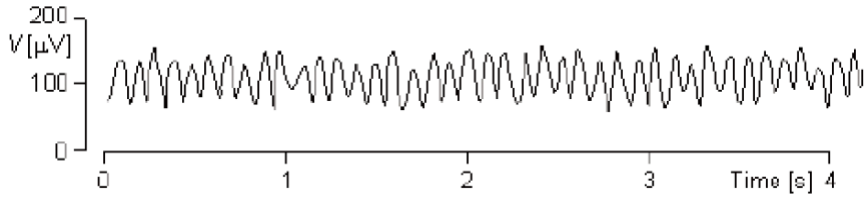
Figura 2.11: Frequência da Onda Theta.



Fonte: (MARK *et al.*, 2007)

Já no ritmo alpha as frequências variam de 8 Hz a 13 Hz e é verificado em uma pessoa normal relaxada, ocorre na região posterior e com maior amplitude sobre a região occipital. O ritma alpha é mais bem visualizado quando o indivíduo está com os olhos fechados, em situação de baixa atividade mental e relaxamento físico (MARK *et al.*, 2007).

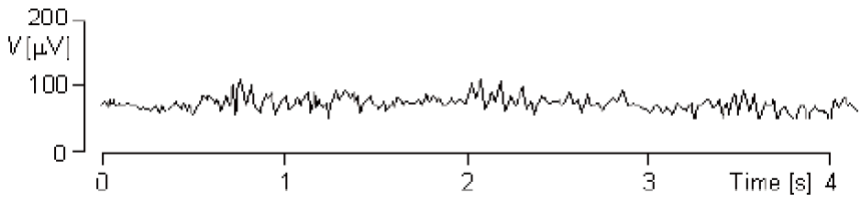
Figura 2.12: Frequência da Onda Alpha.



Fonte: (MARK *et al.*, 2007)

O ritmo beta cujas frequências situam na faixa dos 13 Hz a 30 Hz predominante na região frontal e central, está associado à atividade e concentração.

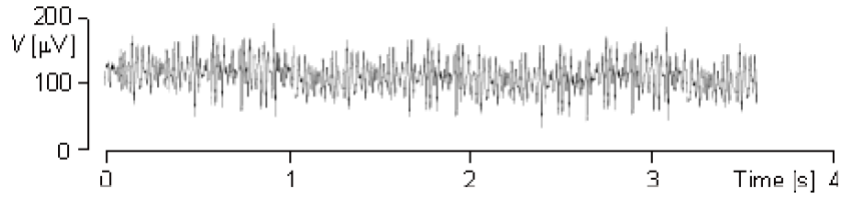
Figura 2.13: Frequência da Onda Beta.



Fonte: (MARK *et al.*, 2007)

O ritmo gama (frequências de 30 Hz a 70 Hz) está relacionado à elevada atividade mental, à resolução de problemas e também ao medo (MARK *et al.*, 2007).

Figura 2.14: Frequência da Onda Gamma.



Fonte: (MARK *et al.*, 2007)

As frequências captadas são amplificadas e filtradas por um sensor de eletroencefalograma para remoção de ruído que possa atrapalhar a leitura dos sinais, que são convertidos de analógicos para digitais utilizando frequência de amostragem. Nesse processamento digital, são aplicados algoritmos matemáticos para transformar os dados obtidos que se apresentam com características diversificadas, transformando-os em informações básicas. Assim, o conjunto de características dos sinais passa por processo de classificação para o reconhecimento de padrões.

2.5 Artefatos

O resultado da atividade elétrica dos neurônios pode ser medida e registrada através do eletroencefalograma. Entretanto, os sinais capitados pelo EEG não são somente os sinais de pensamentos. Junto com eles uma série de outros sinais embaralham a possibilidade de se fazer uma tradução do pensamento imaginado. Ao captar esses sinais cerebrais eles vem juntos com outros sinais.

Dessa forma também vai se refletir nos sinais do EEG, os sinais de 50 Hz e 60 Hz dos equipamentos elétricos, os sinais dos movimentos cardíacos (ECG), os sinais das piscadas dos olhos (EOG), e os sinais musculares (EMG). Todos esses sinais terão suas componentes

nos sinais capitados, dificultando dessa maneira, se chegar aos verdadeiros sinais dos pensamentos imaginados.

Esses outros sinais, que diferem dos sinais que se buscam, é chamado de **artefatos**. Eliminar completamente todos os artefatos não é uma tarefa fácil. Existem algumas estratégias que estão sendo aperfeiçoadas buscando esse objetivo.

Um desses caminhos é o *Blind Source Separation* (BSS) em português: Separação Cega de Fontes. A temática de BSS, atualmente, é uma das que mais atraem a atenção dos pesquisadores da área de processamento de sinais, o que se justifica, em grande medida, por sua significativa aplicabilidade. Essa temática abrange tarefas tão diversas quanto tratamento de sinais de ECG, EEG e fMRI, detecção multiusuário em comunicações sem fio, análise de sinais provenientes de sensores químicos, extração de informação de sinais de astrofísica, modelamento de processos do córtex visual humano, tratamento de sinais de áudio entre outros. Dentro do BSS um dos métodos que mais se destaca para eliminação de artefatos é o Análise de Componentes Independentes (ICA em inglês).

Embora existam outros métodos dentro do BSS, como: a Análise de Componentes Principais (PCA), nesse trabalho vai se dar ênfase aos métodos ICA já que toda essa pesquisa será baseada neste método.

2.6 Separação Cega de Fontes - BSS

A separação cega de fonte (BSS) é um método poderoso de processamento de sinal que foi proposto no final dos anos 1980. Como produto das redes neurais artificiais, processamento estatístico de sinais e teoria da informação. O BSS se tornou um tópico importante em pesquisa e desenvolvimento em muitas áreas, especialmente nas ciências biomédicas, comunicação de sinais de fala, processamento de imagens, ciências da terra, econometria e mineração de dados. Atualmente, o problema do BSS é um ponto de acesso de pesquisa em processamento de sinais, redes neurais artificiais e outras disciplinas, portanto, possui grande valor prático. O BSS, que é um problema tradicional e desafiador no processamento de sinais, envolve extrair e recuperar os sinais de origem subjacentes a partir de dados estatísticos multivariáveis. (YU XIANCHUAN E HU, 2013).

No BSS o método ICA se baseia no problema clássico da festa do coquetel (*cocktail party*), veja a Figura 2.15.

Em uma festa com vários instrumentos tocando ao redor, ouve-se um som resultante da combinação de todos os instrumentos. Se quiser ouvir em separado o que cada instrumento está tocando, o que se faria para separar o som de cada um dos instrumentos presentes? O

problema da festa de coquetel é um problema clássico de Separação Cega de Fontes (BSS do inglês) e uma maneira de resolvê-lo é através da Análise de Componentes Independente (ICA do inglês) (HYVÄRINEN AAPO E OJA, 1999).

Figura 2.15: Problema da Festa de Coquetel

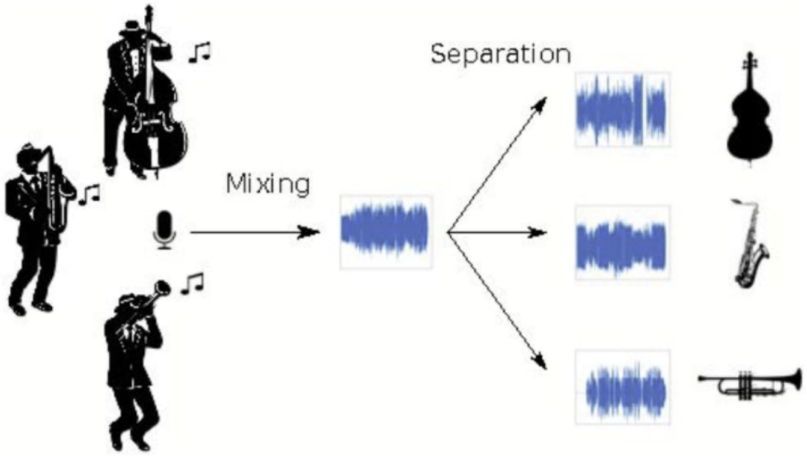


Figura da Internet

A particularidade da separação cega de fontes perante as outras técnicas de filtragens é que, nesse caso, não se precisa conhecer precisamente os sinais das fontes (HYVÄRINEN AAPO E OJA, 1999).

O problema do *cocktail-party* pode ser representado matematicamente assim:

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} e\, s = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}$$

ou seja,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \tag{2.1}$$

Reescrevendo a Equação 2.1 na forma matricial fica:

$$x = As \tag{2.2}$$

Denota-se $\mathbf{x}$ pelo vetor aleatório cujos elementos representam as misturas ou sensores, a matriz $\mathbf{A}$ com os elementos $\mathbf{a_{ij}}$ representam a atenuação ou amplificação sobre o vetor aleatório $\mathbf{s}$ que representam os sinal de fontes $\mathbf{s_1}$, $\mathbf{s_2}$ e $\mathbf{s_3}$.

2.6.1 Análise de Componentes Independentes (ICA)

A análise de componente independentes (ICA) é uma técnica estatística e computacional para revelar fatores ocultos subjacentes a conjuntos de variáveis aleatórias, medições ou sinais.

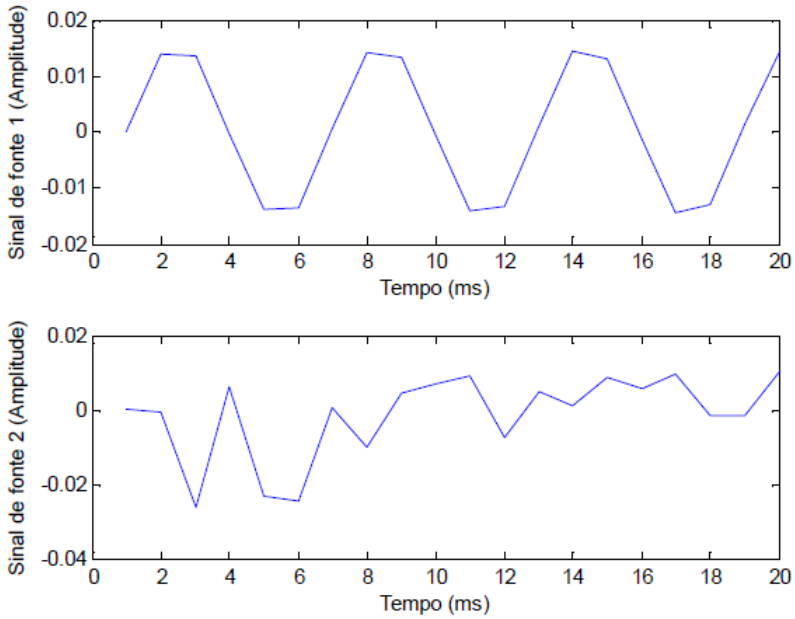
A ICA define um modelo generativo para os dados multivariados observados, que geralmente são dados como um grande banco de dados de amostras. No modelo, as variáveis de dados são assumidas como misturas lineares de algumas variáveis latentes desconhecidas e o sistema de mistura também é desconhecido. As variáveis latentes são consideradas não-gaussianas e independentes entre si, e são chamadas de componentes independentes dos dados observados. Esses componentes independentes, também chamados de fontes ou fatores, podem ser encontrados pela ICA.

A ICA está superficialmente relacionada à análise de componentes principais e análise fatorial (não abordadas nesta dissertação). A ICA é uma técnica muito mais poderosa, e capaz de encontrar os fatores ou fontes subjacentes quando esses métodos clássicos falham completamente (HYVÄRINEN AAPO E KARHUNEN, 2001)

O problema da separação cega de fontes se resume em encontrar uma representação linear na qual os componentes são estatisticamente independentes. Dentre situações práticas, em geral não se pode encontrar uma representação em que os componentes sejam realmente independentes, mas, é possível pelo menos, encontrar componentes tão independentes quanto possível.

Para Ilustrar, será considerado os seguintes sinais representados na Figura 2.16 e Figura 2.17.

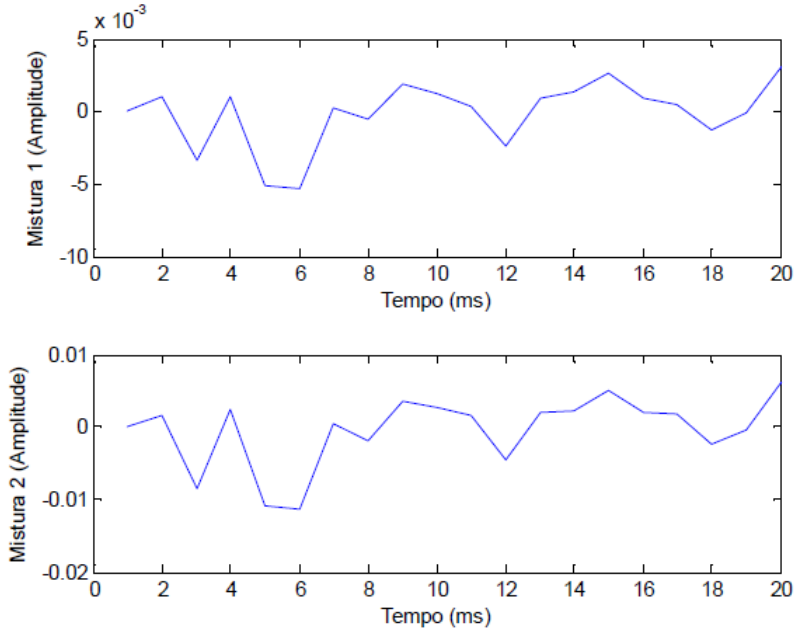
Figura 2.16: Sinais Originais do discurso.



Fonte: (ZHANG KUN E CHAN, 2006)

Os sinais do discurso original são semelhante aos sinais representado na Figura 2.16 e as misturas poderiam se parecer com os sinais na Figura 2.17. Nos gráficos as coordenadas abscissas representam o número de amostra do sinal em cada período de tempo e a ordenada representa a amplitude do sinal. O problema consiste em recuperar os dados na Figura 2.16 utilizando-se apenas os dados da Figura 2.17.

Figura 2.17: Sinais do discurso misturados.



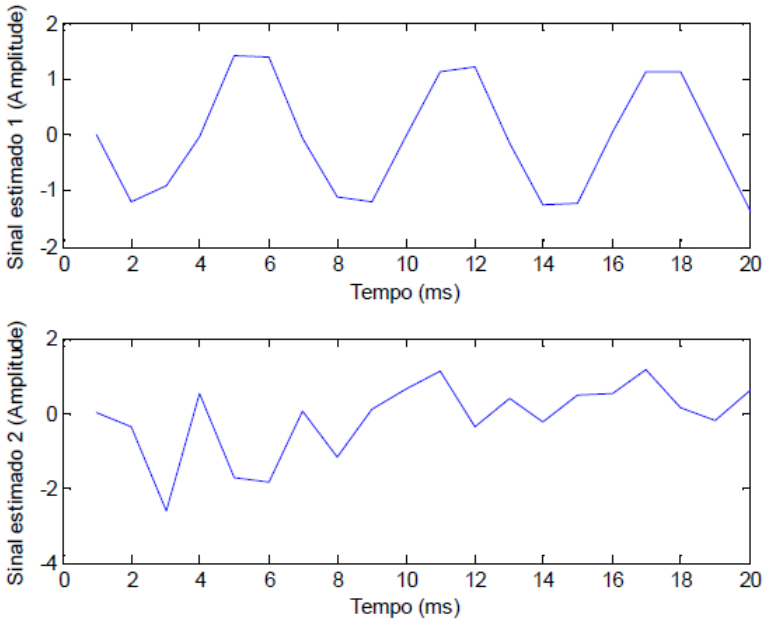
Fonte: (ZHANG KUN E CHAN, 2006)

Sabendo os parâmetros a_{ij} poderia resolver o sistema na Equação 2.1 pelos métodos clássicos. O ponto, porém, é que não se sabe o valor de a_{ij}, de forma que o problema se torna consideravelmente difícil.

Uma abordagem para resolver este problema seria usar alguma informação estatística sobre as propriedades dos sinais $s_i(n)$ para estimar a todos. Na verdade e talvez surpreendentemente, verifica-se que ela seja suficiente para presumir que $s_1(n)$ e $s_2(n)$ a cada instante de tempo n são estatisticamente independentes. A técnica desenvolvida conhecida como ICA pode ser usada para estimar os valores de a_{ij} baseado nas informações de sua independência, o que permite recuperar ou estimar o sinal original $s_1(n)$ e $s_2(n)$ a partir de suas misturas $x_1(n)$ e $x_2(n)$.

A Figura 2.17 representa os sinais de fontes estimados por ICA usando abordagem PCA (ZHANG KUN E CHAN, 2006).

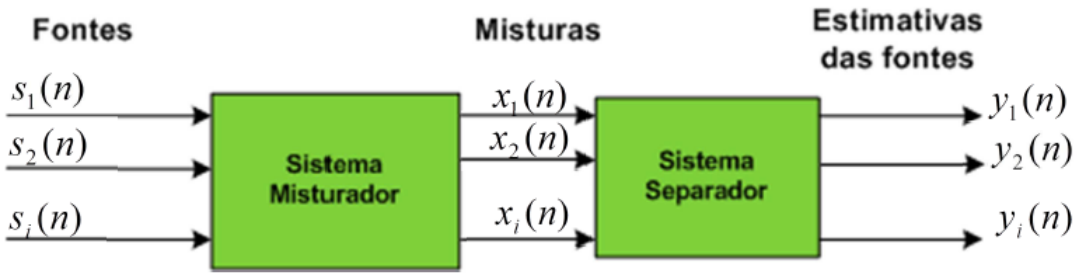
Figura 2.18: Sinais da Fonte Estimados a Partir das Misturas.



Fonte: (ZHANG KUN E CHAN, 2006)

A Figura 2.19 ilustra um diagrama esquemático do problema, onde um conjunto de sinais de fontes são submetidos à ação de um sistema misturador, cujas saídas correspondem a misturas de tais fontes. Deve-se de alguma forma, projetar um sistema separador que seja capaz de estimar as fontes, ou seja, inverter a ação do sistema misturador. Este problema é dito cego em razão da falta de informação que temos sobre as misturas e as fontes.

Figura 2.19: Diagrama esquemático do problema de separação cega linear



Fonte: (ZHANG KUN E CHAN, 2006)

Pode-se também representar de forma simples o processo de misturas das fontes pela expressão da Equação 2.3:

$$x(n) = F(s(n), s(N-1), \ldots, s(n-L), r(n)) \tag{2.3}$$

Onde $F(\cdot)$ corresponde a ação do sistema misturador, L é associado às memórias (amostras atrasadas) no sistema e o vetor r representa o ruído presente nas fontes. Um sistema misturador é dito linear se o mapeamento $F(\cdot)$ atende o princípio da superposição, caso contrário é dito não linear.

Nas situações em que o sistema misturador depende das amostras passadas ($L > 0$) é dito que o sistema misturador é convolutivo (com memória). Entretanto, há situações em que ($L = 0$) o sistema é chamado de instantâneo (COMON, 1994).

Em outras situações onde o número de sensores podem ser maiores que o número de sinais de fontes, tem se o caso sobre-determinado quando o número de sensores é menor que os sinais de fontes, temos o caso subdeterminado.

Apesar de não tratarmos do PCA nesse trabalho, a principal diferença entre o ICA e o PCA é que o PCA usa, unicamente, a estatística de 2ª ordem (média e variância), enquanto o ICA utiliza a estatística de ordens superiores (kurtosis). Por isso, o PCA é usado para variáveis Gaussianas que são de estatística de 2ª ordem. Mas, como a maioria dos sinais são não gaussianos e com ordens estatísticas elevadas, o ICA é uma melhor opção.

ICA consiste em recuperar os sinais originais a partir de uma mistura. Um princípio bastante utilizado para determinar ou inferir nas misturas tem sido a independência estatística, ou seja, o valor de qualquer um dos componentes não fornecer nenhuma informação sobre os valores dos outros componentes.

Normalmente, uma distribuição de probabilidade é caracterizada em termos de sua função de densidade ao invés de CDF (função de distribuição cumulativa ou do inglês: *cumulative distribution function*). Formalmente, a função de densidade de probabilidade é obtida derivando CDF. ICA está intimamente ligado à independência estatística. Matematicamente, a independência estatística é definida em termos de densidade de probabilidade (PAPOULIS, 1991). As variáveis aleatórias x e y (Equação 2.4) são ditas independentes, se e somente se:

$$p_{x,y}(x,y) = p_x(x)p_y(y) \tag{2.4}$$

Dizendo de outra maneira, a densidade conjunta $p_{x,y}(x,y)$ de x e y devem fatorar no produto das suas densidades marginais $p_x(x)$ e $p_y(y)$. Equivalente à independência, esta pode ser definida pela substituição das funções de densidades de probabilidade na Equação 2.4 pelas respectivas funções de distribuição cumulativa, que também deve ser fatoráveis.

Duas variáveis, x e y, são não correlacionadas se a sua covariância for zero Equação 2.5):

$$cov(x,y) = Exy - ExEy = 0 \tag{2.5}$$

Assume-se que a média é zero para todas as variáveis aleatórias. Logo a covariância é igual à correlação (Equação 2.6):

$$cov(x,y) = corr(x,y) = Exy = 0 \tag{2.6}$$

Se as variáveis, x e y, são independentes, então são não-correlacionadas, ou seja,

$$Exy = ExEy \tag{2.7}$$

x e y são independentes. Substitui-se a Equação 2.7 na Equação 2.5 obtém-se a Equação 2.8.

$$cov(x,y) = ExEy - ExEy = 0 \tag{2.8}$$

Porém, se duas variáveis aleatórias forem não-correlacionadas, não implica que sejam

independentes. Por isso, a independência é mais forte que a não correlação. Daí que os sinais a separar de uma mistura tenham que ser mutuamente independentes.

O modelo generativo descreve como os sinais misturados são produzidos e trata-se da base do ICA. Este modelo afirma que os sinais misturados são o produto da combinação linear dos sinais originais (componentes independentes). Para a simplificação do método, não se considera a presença de ruído, diferente de situações reais ou práticas.

2.6.2 Restrições

Para que o modelo ICA possa ser estimado, é preciso fazer algumas restrições:

1. É preciso assumir que as componentes independentes sejam estatisticamente independentes. Duas ou mais variáveis aleatórias são ditas estatisticamente independentes se a informação contida nos valores de qualquer uma delas não fornece informação alguma acerca dos valores de qualquer uma das outras. A independência estatística pode ser definida formalmente através das funções densidade de probabilidade das variáveis aleatórias. Sejam $p(y_1, y_2)$ a função densidade de probabilidade conjunta (fdp) das variáveis aleatórias y_1 e y_2 e $p_i(y_i)$ a função densidade de probabilidade marginal de y_i, ou seja: a função densidade de probabilidade de y_i quando somente esta é considerada. É dito que y_1 e y_2 são estatisticamente independentes se e somente se a função densidade de probabilidade conjunta for favorável da seguinte maneira Equação 2.9:

$$p(y_1, y_2) = p_1(y_1)p_2(y_2) \tag{2.9}$$

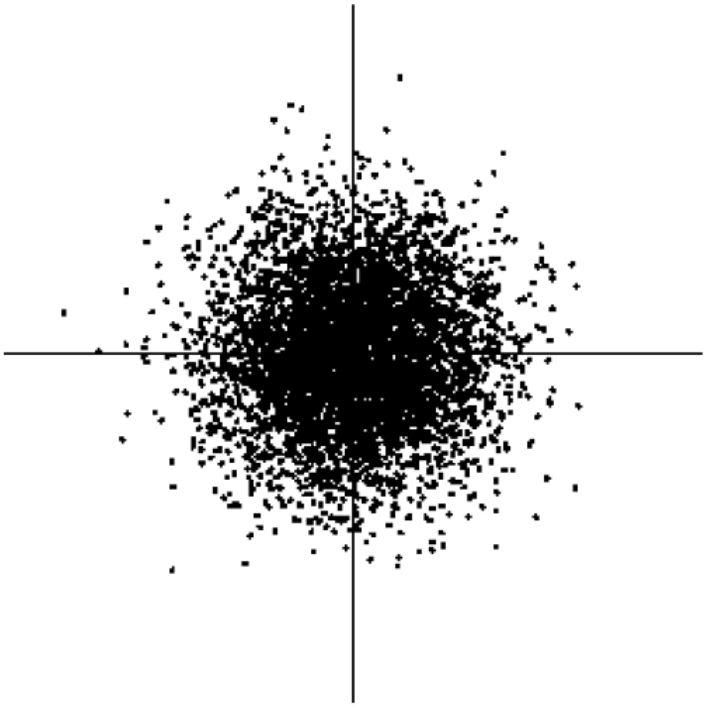
   Uma propriedade importante da independência de variáveis aleatórias é que, dado duas funções h_1 e h_2, dessa forma sempre teremos Equação 2.10:

$$E[h_1(y_1)h_2(y_2)] = E[h_1(y_1)]E[h_2(y_2)] \tag{2.10}$$

2. As componentes independentes precisam ter distribuições de probabilidade não gaussianas. No modelo ICA, não é assumido que as distribuições de probabilidade das componentes independentes são conhecidas, porém, é preciso assumir que elas sejam não gaussianas. As distribuições gaussianas são simétricas (Figura 2.20). Não há uma direção de maior concentração de valores que possa ser privilegiada na estimativa do modelo ICA. Isso significa que as distribuições gaussianas possuem cumulantes de alta ordem iguais a zero; mas essas informações de alta ordem são importantes estimativas do modelo ICA. Embora, seja assumido que as distribuições das componentes independentes sejam não gaussianas, certamente as distribuições

das misturas observadas serão. Para fins de simplicidade, assumi-se que o número de componentes independentes equivale ao número de misturas observadas, ou seja: a matriz de mistura é quadrada e portanto, pode possuir inversa. Se esse não for o caso, haverão misturas redundantes que poderão ser omitidas no modelo.

Figura 2.20: Distribuição de probabilidade gaussiana



Fonte: (HYVARINEN AAPO E KARHUNEN, 2001)

2.6.3 Ambiguidade de ICA

Ambiguidades referentes ao modelo ICA básico:

1. Não é possível determinar as variâncias das componentes independentes. Isto porque tanto A, quanto s são desconhecidos, qualquer escalar α multiplicado em alguma das componentes independentes poderia ser cancelado dividindo a coluna correspondente a_i de A pelo mesmo escalar. Uma solução possível é fixar as energias das variáveis aleatórias; um método simples de se fazer isso é considerar que suas variâncias sejam unitárias, isto é: $E\{s_i^2\} = 1$. Assumir que as componentes independentes possuem variâncias unitárias, é uma das consequências de um dos pré-processamento realizados antes de estimar o modelo (subseção 2.6.4). É bom notar que mesmo assumindo variância unitária para as fontes, ainda permanece a ambiguidade do sinal que significa multiplicar uma componente por -1 porém, esta ambiguidade é considerada insignificante na maioria dos casos.

2. Não se pode determinar a ordem dos componentes independentes. Isso novamente se dá por A e s serem desconhecidos. Uma solução para essa ambiguidade é adicionar uma matriz de permutação P e sua inversa no modelo, tal que (Equação 2.11):

$$x = AP^{-1}Ps \tag{2.11}$$

De forma que as componentes independentes Ps serão as componentes originais s_i. em outra ordem. A matriz AP^{-1} será a nova matriz de mistura a ser estimada.

2.6.4 Pré-Processamento

Antes de aplicar o modelo ICA para a estimativa das componentes independentes, é necessário realizar algumas etapas de pré-processamento no conjunto de misturas observadas. Aqui, iremos discutir algumas técnicas de pré-processamento que tornam as misturas observadas melhor condicionadas e a estimativa do modelo mais simples. Por outro lado, veremos que caso a dimensão dos dados seja muito grande, pode ser útil calcular o PCA a fim de diminuir essa dimensão para um intervalo previamente estabelecido.

2.6.5 Centralização

Esse é o pré-processamento mais básico, embora tenha uma boa contribuição em simplificar a teoria envolvida no modelo ICA e os algoritmos para estimar o modelo.

Sem perda de generalidade, nós podemos assumir que as componentes independentes e as misturas observadas possuem média zero. Para que isso seja verdade, as misturas observadas passam pela fase de centralização, que significa subtrair das misturas a sua média. Denotando as misturas observadas originais por x, as misturas centralizadas x_c são tais que (Equação 2.12):

$$x_c = x - E[x] \tag{2.12}$$

Dessa forma, as componentes independentes também terão média zero já que (Equação 2.13):

$$E[s] = A^{-1}E[x_c] \tag{2.13}$$

O modelo continua sendo estimado da mesma forma, pois não há alteração alguma na matriz de mistura, além disso, após a matriz de mistura ser estimada - considerando que as misturas observadas passaram pela fase de centralização - a média subtraída pode ser reconstruída adicionando a $W^*E[x]$, lembrando que $W = A^{-1}$ e W^* é uma aproximação ótima de W, as componentes independentes de média zero, ou seja, as componentes independentes que foram estimadas no modelo.

2.6.6 Branqueamento (Whitening)

A fase de branqueamento, apesar de ser um pouco mais difícil de ser calculada do que a centralização, ainda é um procedimento simples de ser implementado além de ajudar a diminuir significativamente a complexidade do problema. Sendo assim, o branqueamento ajuda a resolver o problema para o qual ICA é proposto. Esse pré-processamento é aplicado nas misturas centralizadas.

O branqueamento é uma propriedade um pouco mais poderosa em relação à descorrelação. Duas variáveis aleatórias y_1 e y_2 são descorrelacionadas se sua covariância for igual a zero. Isto é (Equação 2.14):

$$cov(y_1, y_2) = E[y_1 Y_2] - E[y_1]E[y_2] = 0 \tag{2.14}$$

Se duas variáveis aleatórias forem independentes, necessariamente elas serão descorrelacionadas. Essa afirmação pode ser verificada se nós tomarmos $h_1 = y_1$ e $h2 = y_2$ na Equação 2.15. Assim teremos:

$$E[y_1 y_2] = E[y_1]E[y_2] \tag{2.15}$$

Que implica em descorrelação das variáveis aleatórias.

Por outro lado, descorrelação não implica independência. Se nós tomarmos duas variáveis aleatórias discretas com uma distribuição tal que o par possui probabilidade 1/4 para os seguintes valores: $(0, 1), (0, -1), (1, 0)$ e $(-1, 0)$. Nós temos que as duas variáveis são descorrelacionadas, porém, a condição da Equação 2.10 é violada como se pode ver na Equação 2.16, portanto, y_1 e y_2 não são independentes.

$$E[y_1^2 y_2^2] = 0 \neq \frac{1}{4} = E[y_1^2]E[y_2^2] \tag{2.16}$$

A diferença entre o branqueamento e a descorrelação é que variáveis brancas, como são chamadas as variáveis aleatórias que passaram pelo processo de branqueamento. Além de serem descorrelacionadas, possuem variância unitária, ou seja, possuem matriz de covariância igual a matriz identidade (Equação 2.17):

$$C_y = E[yy^t] = I \tag{2.17}$$

O processo de branqueamento consiste em aplicar uma determinada transformação linear em uma variável aleatória x, obtendo assim uma nova variável aleatória z, a qual é branca (Equação 2.18):

$$z = Vx \tag{2.18}$$

Um método popular para se realizar o branqueamento é a decomposição por auto-valor (EVD *Eigenvalue Decomposition*) da matriz de covariância (Equação 2.19):

$$E[xx^t] = EDE^t \tag{2.19}$$

Tal que E é a matriz ortogonal dos auto-vetores associados aos auto-valores da matriz de covariância. A matriz de transformação utilizada no branqueamento, geralmente chamada por matriz de branqueamento, é definida por (Equação 2.20):

$$V = ED^{-1/2}E^t \tag{2.20}$$

Pela definição do modelo ICA básico (Equação 2.21 e Equação 2.22), nós temos que:

$$z = VAs \tag{2.21}$$

$$z = A's \tag{2.22}$$

A utilidade do branqueamento reside no fato de que a noma matriz de mistura A' é ortogonal. ou seja, como $A'^{-1} = A'^t$ (no caso de matrizes ortogonais),a estimativa da

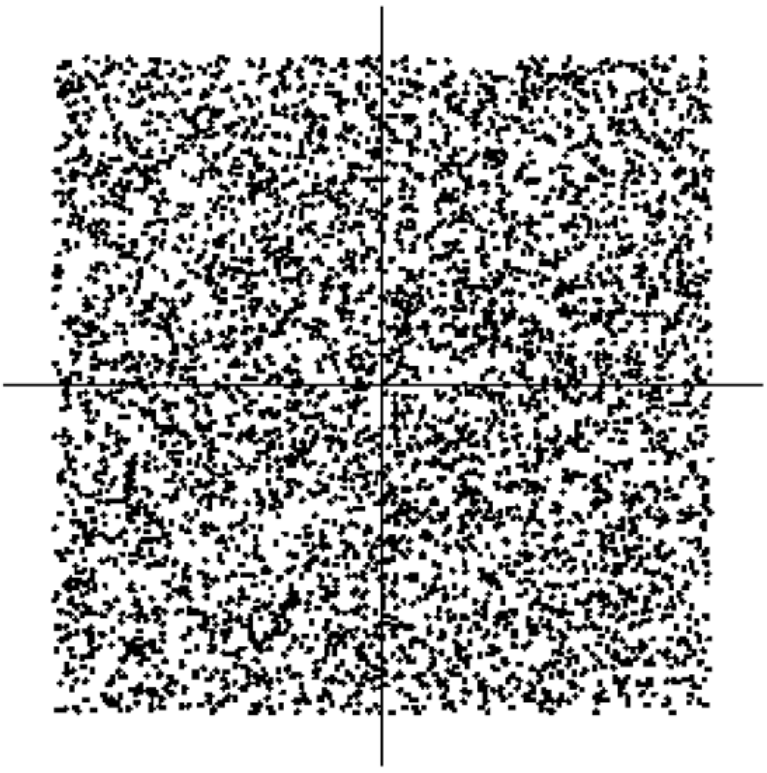
matriz de mistura se restringe ao espaço de matrizes ortogonais. Ao invés de estimar os n^2 parâmetros da matriz de mistura original, somente é preciso estimar os parâmetros da matriz ortogonal A'^{-1}, o que significa $n(n-1)/2$ parâmetros.

Com o objetivo de mostrar essa propriedade será reproduzido a seguir experimento de Hyvarinen Aapo e Karhunen (2001). Consideremos duas variáveis aleatórias x_1 e x_2 com distribuições uniformes $p(s_i)$ (Equação 2.23), tal que:

$$p(s_i) = \begin{cases} \dfrac{1}{2\sqrt{3}} \text{ se } s_i \leq \sqrt{3} \\ 0\,, \text{ caso contrário} \end{cases} \tag{2.23}$$

Os valores da distribuição uniforme foram escolhidos de tal forma que a função densidade de probabilidade conjunta das duas variáveis aleatórias seja propositalmente quadrada Figura 2.21

Figura 2.21: Distribuição de probabilidade Conjunta de x_1 e x_2

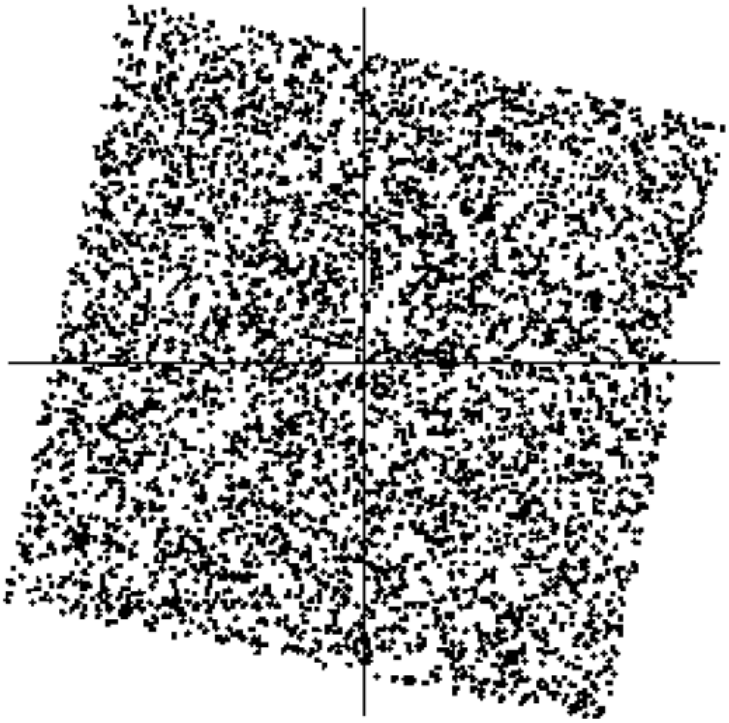


Fonte: (HYVARINEN AAPO E KARHUNEN, 2001)

Após o branqueamento, a função densidade de probabilidade conjunta das variáveis branqueadas é uma versão totalmente rotacionada da função das variáveis originais, como visto na Figura 2.22. Isso se dá porque em um espaço bidimencional, uma transformação

ortogonal ($z = A's$) é determinada por um único parâmetro, que é o ângulo da rotação, ou seja, ao invés de estimar todos os quatro parâmetros de uma matriz 2×2, estima-se somente um parâmetro de rotação, caso as variáveis tenham passado pelo processo de branqueamento.

Figura 2.22: Função densidade de probabilidade conjunta das variáveis branqueadas



Fonte: (HYVARINEN AAPO E KARHUNEN, 2001)

Em algumas aplicações onde a massa de dados é muito grande, pode ser útil extrair as componentes principais dos dados antes de realizar o branqueamento por meio da técnica PCA. A solução para o problema PCA de uma massa de dados x é dada em termos dos autovetores: $e_1, e_2, e_3, \ldots, e_{n-1}, e_n$ da matriz de covariância C_x, onde: $C_x = E[xx^t]$ (OJA, 1983).

Os autovetores são então ordenados de acordo os respectivos autovalores, tal que: $d_1 \geqslant d_2 \geqslant d_3 \ldots, \geqslant d_n$. Os menores autovalores são então descartados de acordo com um valor limite de tolerância definido previamente. Dessa forma, a massa de dados é reduzida de tal forma que os dados remanescentes são as componentes principais dentro dos limites estabelecidos pela tolerância.

2.6.7 Não Gaussianidade e Independência

Nesta seção, será discutida a relação entre as restrições de não gaussianidade e independência estatística das componentes independentes, assim como a maneira como ICA se utiliza dessas restrições para estimar o modelo.

É possível mostrar que para componentes independentes com distribuições de probabilidade gaussianas Equação 2.24, a matriz de mistura não surte efeito algum nas distribuições das misturas Equação 2.25, ou seja, as distribuições são idênticas, o que significa que não há meios de se estimar a matriz de mistura tendo somente as informações fornecidas pelas misturas observadas. Na verdade, é até mais fácil entender porque não se pode estimar a matriz de mistura no caso de componentes independentes gaussianas se as componentes forem brancas.

$$p(s_1, s_2) = \frac{1}{2\pi} exp\left(-\frac{s_1^2 + s_2^2}{2}\right) \tag{2.24}$$

$$p(x_1, x_2) = \frac{1}{2\pi} exp\left(-\frac{x_1^2 + x_2^2}{2}\right) \tag{2.25}$$

Por outro lado, se for considerado que as componentes independentes são estatisticamente independentes e não gaussianas, a técnica ICA é capaz de estimar a matriz de mistura e consequentemente recuperar as componentes baseando-se somente nas informações fornecidas pelas misturas observadas. O Teorema do Limite Central dá uma ideia de como isso é possível.

O Teorema do Limite Central é um resultado clássico da teoria das probabilidades. O teorema afirma que si considerar uma sequência de variáveis aleatórias estatisticamente independentes e igualmente distribuídas $x_1, x_2, x_3, \ldots, x_k$, a soma $y_k = x_1 + x_2 + x_3 + \ldots + x_k$ converge para uma distribuição gaussiana se $k \to \infty$. Na prática, não se tem $k \to \infty$, mas baseados no teorema do limite central, pode-se então dizer que a soma de duas variáveis aleatórias estatisticamente independentes e igualmente distribuídas. Geralmente tem uma distribuição que é mais próxima de uma gaussiana que qualquer uma das distribuições das variáveis originais. Assim, o Teorema do Limite Central pode ser utilizado para estimar as componentes independentes da seguinte forma.

Pelo modelo ICA básico, fica (Equação 2.26):

$$s = A^{-1}x \tag{2.26}$$

Portanto, para se estimar uma das componentes independentes, pode-se considerar uma combinação linear das misturas observadas y, tal que $y = bx$, onde b é um vetor linha a ser determinado. Pelo modelo, tem-se também que $y = bAs$, melhor dizendo, y é uma combinação linear das componentes independentes com o vetor bA. Se denotar esse vetor q, fica (Equação 2.27):

$$y = bx = qs = \sum q_i s_1 \tag{2.27}$$

Se observar a Equação 2.26 e considerar que b corresponde a uma das linhas de A^{-1}. então a combinação linear bx teria como resultado uma das componentes independentes. Consequentemente, se considerar $q = bA$ e b como uma linha da inversa de A, q será necessariamente um vetor com somente um elemento igual a 1 e todos os outros iguais a zero. Se considerar que as componentes independentes são estatisticamente independentes, pelo teorema de limite central, pode se afirmar que qualquer combinação linear das componentes terá distribuição mais próxima da gaussiana do que as distribuições de qualquer uma das componentes. Portanto, $y = qs$, será sempre mais gaussiana do que qualquer s_i, a menos que qs resulte em uma das componentes independentes.

Se os valores de q e s fossem conhecidos, podería variar o valor de q até que y se igualasse a uma das componentes independentes, mas, na prática somente os valores das misturas observadas x são conhecidas. Por outro lado, sabe-se também que $qs = bx$, então se pode calcular um vetor b que maximize a não gaussianidade de bx, esse vetor seria necessariamente igual ao vetor bA, ou seja, $y = bx = qs$, será equivalente a uma das componentes independentes.

Embora a relação entre independência e não gaussianidade. tenha sido esclarecida com auxílio do teorema do limite central, outra questão precisa ser discutida. A combinação linear $y = bx$ seria equivalente a uma das componentes independentes se pudesse calcular um vetor b que maximize a não gaussianidade de bx. Na subseção 2.6.8 pode ser visto como a não gaussianidade pode ser medida.

2.6.8 Medida de Não Gaussianidade

Para utilizar a não gaussianidade na estimativa do modelo ICA, é necessária ter uma medida quantitativa da não gaussianidade de uma variável aleatória y. Apesar de existirem outros métodos relacionados com não gaussianidade de variáveis aleatórias que podem ser utilizados para estimar o modelo ICA, como: máxima verossimilhança e informação mútua, serão apresentadas duas importantes medidas de não gaussianidade: a Kurtosis

(Curtose) e a Negentropia.

2.6.8.1 Kurtosis

Uma medida clássica de não gaussianidade é a *kurtosis*, também chamada de Cumulante de Quarta-ordem.

A função *Kurtosis* de uma variável aleatória y é dada pela seguinte fórmula (Equação 2.28):

$$kurt(y) = E[y^4] - 3(E[y^2])^2 \tag{2.28}$$

Se assumir que y possui variância unitária (se a variável for branca, por exemplo), então a função é simplificada da seguinte maneira (Equação 2.29):

$$kurt(y) = E[y^4] - 3 \tag{2.29}$$

A Equação 2.29 mostra que a função *kurt* é uma versão normalizada do momento de quarta ordem $E[y^4]$.

Usando o valor absoluto da *kurtosis*, chega-se na medida de não gaussianidade. O quadrado do valor absoluto da *kurtosis* também pode ser utilizado. Para o caso de uma variável gaussiiana, o resultado da função *kurtosis* é, na maioria das vezes, igual a zero. Para variáveis não gaussianas, o valor absoluto, ou seu quadrado, da *kurtosis* é diferente de zero.

A razão pela qual a *kurtosis* é amplamente utilizada é pela sua simplicidade. Na prática, a *kurtosis* pode ser calculada utilizando somente o momento de quarta ordem e além disso; para duas variáveis aleatórias independentes de s_1 e s_2, tem-se as seguintes propriedades de linearidade (Equação 2.30 e Equação 2.31):

$$kurt(s_1 + s_2) = kurt(s_1) + kurt(s_2) \tag{2.30}$$

$$kurt(\alpha s_1) = \alpha^4 kurt(s_1) \tag{2.31}$$

Apesar de sua simplicidade, na prática, a *kurtosis* possui algumas desvantagens. A questão é que somente algumas amostras de uma variável aleatória podem ter uma maior influência no valor absoluto da *kurtosis* em relação a todos as outras amostras. Se considerarmos uma variável aleatória com 1000 amostras (com média zero e variância unitária) com valores que variam entre 0 e 1 e algum agente externo contribuísse para que uma amostra possuísse valor igual a 10, o valor da *kurtosis* seria igual a pelo menos $10^4/1000 - 3 = 7$, ou seja: a *kurtosis* é uma medida de não gaussianidade simples mas, não robusta.

2.6.8.2 Negentropia

Nessa seção, será apresentada uma segunda medida de não gaussianidade, chamada *Negentropia*. A negentropia se baseia na quantidade de informação teórica de uma variável dada pela entropia diferencial, tratada simplesmente por entropia.

A entropia de uma variável pode ser interpretada pelo grau de informação que se pode ter ao se observar as variáveis, isto é, quanto mais aleatória for uma variável, maior sua entropia. A entropia H de uma variável aleatória y com função densidade de probabilidade $p_y(\eta)$ é definida pela Equação 2.32:

$$H(y) = -\int p_y(\eta) log(p_y) d\eta \tag{2.32}$$

Um resultado fundamental da teoria da informação é que uma variável gaussiana possui a maior entropia dentre todas as variáveis aleatórias de igual variância. Isto significa que distribuições gaussianas são as mais aleatórias e desestruturadas dentre todas as distribuições, ou seja, pode-se utilizar a entropia como medida de não gaussianidade.

A negentropia é uma versão normalizada da entropia, de tal forma que a negentropia é sempre não negativa e zero para uma variável gaussiana. A negentropia J de uma variável aleatória y é definida da seguinte maneira (Equação 2.33):

$$J(y) = H(y_{gauss}) - H(y) \tag{2.33}$$

onde y_{gauss} é uma variável aleatória com distribuição gaussiana e mesma matriz de correlação (e portanto, covariância) que y.

A negentropia é uma medida de não gaussianidade bem justificada pela teoria estatística e é por vezes considerada um estimador ótimo de não gaussianidade. O problema é

que, como se pode constatar pela própria definição, a negentropia é uma medida de difícil implementação computacional, além de ser necessário ter conhecimento (ou ao menos uma estimativa) da função densidade de probabilidade. A seguir algumas boas aproximações da negentropia que tornam a medida mais praticável.

O primeiro método de aproximação é utilizando cumulantes de alta-ordem. Dessa forma, tem-se a seguinte aproximação (Equação 2.34):

$$J(y) \approx \frac{1}{12} E[y^2]^2 + \frac{1}{48} kurt(y)^2 \tag{2.34}$$

Como se pode constatar na Equação 2.34, a aproximação usando cumulantes de alta-ordem levam ao uso da *kurtosis* apresentada na subseção 2.6.8.1. Como consequência disso, essa aproximação da negentropia não é tão robusta, assim como a *kurtosis*.

Um método mais sofisticado é utilizar esperanças de funções não quadráticas. pode-se utilizar quaisquer duas funções não quadráticas G^1 e G^2 tal que G^1 é impar e G^2 é par. Isso leva à seguinte Equação 2.35:

$$J(y) \approx k_1(E[G^1(y)])^2 + k_2(E[G^2(y)] - E[G^2(v)])^2 \tag{2.35}$$

Onde k_1 e k_2 são constantes positivas e v é uma variável aleatória gaussiana com média zero e variância unitária.

Se usar somente uma função não quadrática G, a aproximação se torna a seguinte Equação 2.36:

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2 \tag{2.36}$$

A questão agora é a escolha de uma função não quadrática G. A escolha de uma G que não cresça tão depressa resulta em estimadores mais robustos. As seguintes funções (Equação 2.37 e (Equação 2.38) tem provado serem boas escolhas:

$$G_1(y) = \frac{1}{a_1} log(cosh(a_1 y)) \tag{2.37}$$

$$G_2(y) = -exp\left(-\frac{y^2}{2}\right) \tag{2.38}$$

onde a_1 é uma constante, tal que: $1 \leq a_1 \leq 2$.

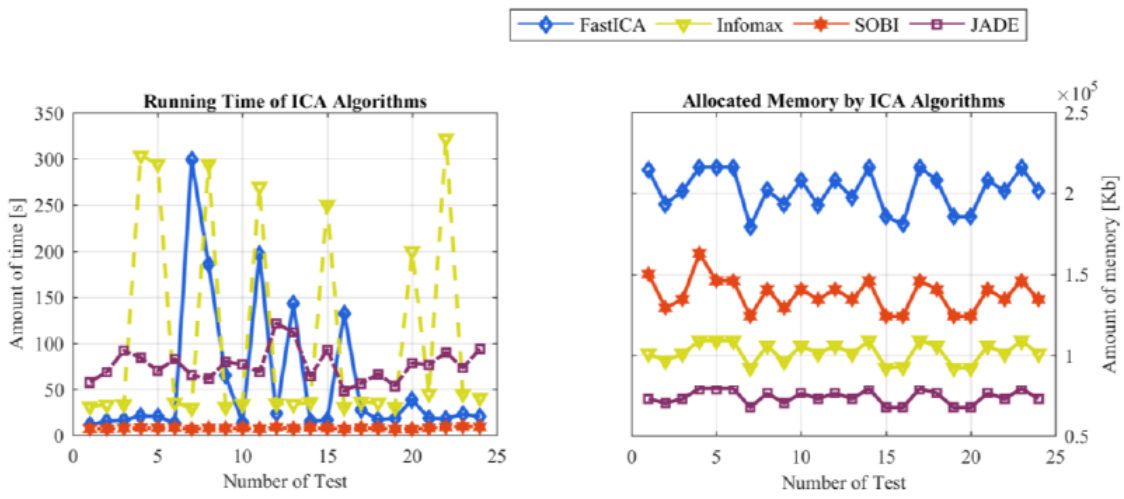
2.6.9 *Algoritmos de Separação de Fontes ICA*

Vários algoritmos foram desenvolvidos para usar o método ICA para separação de fontes, entre eles se destacam: o Algoritmo InfomaX, o Algoritmo FastICA, o Algoritmo SOBI e o Algoritmo JADE.

Nesse trabalho de investigação científica foi usado o algoritmo JADE, não só pela facilidade de implementação no código escrito em MATLAB © como também porque a eficiência desses algoritmos na prática, estão muito próximos, segundo trabalho de Sahonero-Alvarez & Calderon (2017).

No artigo intitulado: *A Comparison of SOBI, FastICA, JADE and Infomax Algorithms* de Sahonero-Alvarez & Calderon (2017) mostra na sua conclusão os seguintes gráficos do estudo comparativo desses algoritmos, Figura 2.23, Figura 2.24, e Figura 2.25.

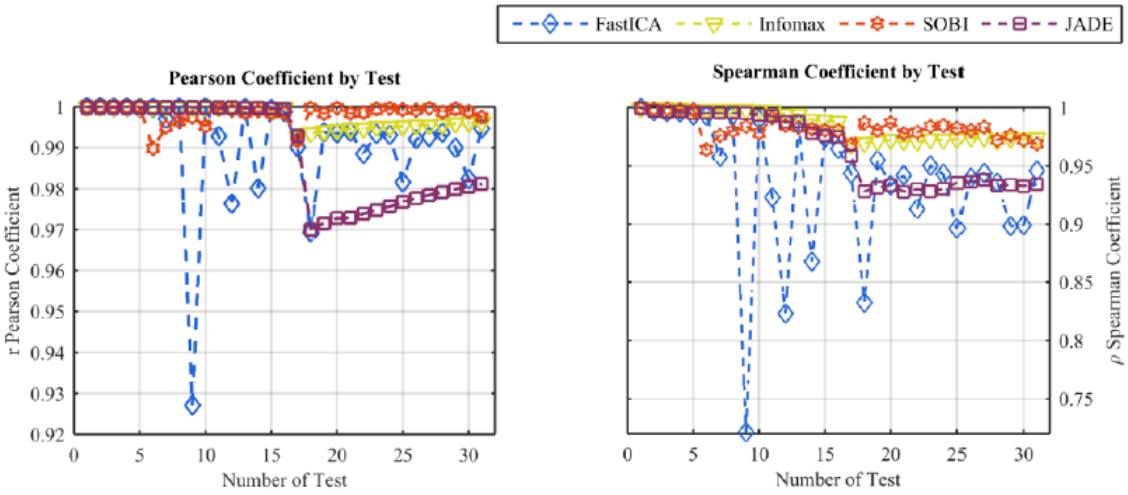
Figura 2.23: Tempo de execução e resultados da memória alocada do voluntário 1



Running Time and Allocated Memory Results of Subject 1

Fonte: (SAHONERO-ALVAREZ; CALDERON, 2017)

Figura 2.24: Coeficientes de Spearman e Pearson de cada algoritmo por teste.



Fonte: (SAHONERO-ALVAREZ; CALDERON, 2017)

Figura 2.25: Tempo de Execução dos Algoritmos

Running Time of Algorithms

Algorithm	Average Time	Worst Time	Best Time	%Dif.
FastICA	51.91 [s]	298.68[s]	11.82 [s]	503.87
Infomax	73.73 [s]	321.89 [s]	25.79 [s]	757.70
SOBI	8.60 [s]	12.28 [s]	6.86 [s]	0
JADE	78.20 [s]	162.86[s]	48.26 [s]	809.65

Fonte: (SAHONERO-ALVAREZ; CALDERON, 2017)

Também o manual do EEGLAB, denominado: EEGLAB Wikitorial, escrito por: Arnaud Delorme e Scott Makeig reporta que testes efetuados pelos algoritmos: InfoMAX, FastICA e JADE em uma base de testes dimensionais simulados e relativamente baixos, para os quais todas as premissas da ICA foram atendidas, todos os três algoritmos retornam componentes quase equivalentes (DELORME; MAKEIG, 2009).

2.6.9.1 InfoMAX

O algoritmo Infomax, que teve relevância no trabalho de Bell & Sejnowski (1995) é um dos mais usados para separação de fontes, sobretudo, porque o mesmo faz parte da EEGLAB, uma importante biblioteca ICA que roda no Matlab, que pode ser usada de forma interativa dentro deste ambiente e para desenvolvedores mais avançados, dispõe de mais de 300 funções que podem ser usadas em desenvolvimento com a linguagem do Matlab.

Para facilitar a pesquisa e desenvolvimento na área de EEG, os desenvolvedores da EEGLAB permitem que essa importante biblioteca seja baixada gratuitamente no endereço Web que segue: https://sccn.ucsd.edu/eeglab/download.php.

Os autores dessa biblioteca desenvolveram um completo manual intitulado: EEGLAB Wikitorial que facilita seu uso interativo bem como uso avançado em desenvolvimento. Este manual pode ser baixado no link: Tutorial do Egglab, e foi escrito por: Delorme & Makeig (2009). Pelo fato do Algoritmo Infomax fazer parte da biblioteca EEGLAB que tem uma interface gráfica interativa oritentada a menus, é o algoritmo ICA mais utilizado atualmente.

2.6.9.2 FastICA - Hyvärinen's Fixed Point Algorithm

O algoritmo FastICA é um método computacional altamente eficiente para realizar a estimativa da ICA. Outra vantagem do algoritmo FastICA é que ele também pode ser usado para executar a busca de projeção, fornecendo um método de análise de dados de uso geral que pode ser usado de forma exploratória e para estimativa de componentes (ou fontes) independentes.

A publicação principal sobre este algorítimo surgiu no trabalho publicado por Aapo Hyvarinen em 1999 intitulado *Fast and Robust Fixed-Point Algorithms for Independent Component Analysis* (HYVARINEN, 1999).

O algoritmo FastICA pode ser derivado tanto para o caso de maximizar a não gaussianidade utilizando o método *kurtosis* quanto o método negentropia. A difereça básica entre os algoritmos será a iteração que calculará o novo w. Segue o algoritmo utilizando o método negentropia:

Figura 2.26: Algoritmo FastICA

1. Escolher um vetor de pesos w inicial (por exemplo, aleatoriamente).
2. $w^* \leftarrow E\{zg(w^t z)\} - E\{g'(w^t z)\}w$.
3. $w \leftarrow \frac{w^*}{\| w^* \|}$.
4. Se não convergiu, voltar ao passo 2.

As Equações: (Equação 2.39 e Equação 2.40), podem ser utilizadas pois, resultam em boas aproximações da negentropia. Além dessas funções, pode-se utilizar também a derivada do momento de quarta ordem, que resultará no método *kurtosis* (Equação 2.41).

$$g1(y) = tanh(a_1 y) \tag{2.39}$$

$$g2(y) = yexp\left(-\frac{y^2}{2}\right) \tag{2.40}$$

$$g3(y) = y^3 \tag{2.41}$$

As derivadas g' são dadas por:

$$g_1'(y) = a_1(1 - tanh^2(a_1 y) \tag{2.42}$$

$$g_2'(y) = (1 - y^2)exp\left(-\frac{y^2}{2}\right) \tag{2.43}$$

$$g_3'(y) = 3y^2 \tag{2.44}$$

A iteração utilizando o kurtosis, é dada da seguinte forma:

$$w^* \leftarrow E\{z(w^t z)^3\} - 3w \tag{2.45}$$

O critério de convergência é o novo e antigo w apontarem para a mesma direção (considerando que w e $-w$ são iguais em virtude das ambiguidades do modelo ICA).

Segue abaixo algumas propriedades do algoritmo FastICA:

1. A convergência é cúbica (ou pelo menos quadrática), sob a suposição do modelo de dados ICA. Isto contrasta com outros algoritmos ICA baseados em métodos de gradiente descendente, onde a convergência é somente linear. Isto significa uma rápida convergência para o algoritmo FastICA. Diversos experimentos em dados em tempo real comprovam esta propriedade.
2. Contrário a algoritmos baseados em gradiente, não há nenhum parâmetro de taxa de aprendizagem para escolher, o que torna o FastICA mais simples.

3. O algoritmos encontra diretamente as componentes independentes de praticamente qualquer distribuição não gaussiana usando qualquer medida de não linearidade g, ao contrário de muitos algoritmos onde a medida de não linearidade precisa ser escolhida especificamente.
4. O desempenho do algoritmo pode ser melhorado com a escolha adequada de uma medida de não linearidade.
5. As componentes independentes podem ser estimadas uma a uma, o que diminui o custo computacional em casos onde somente algumas das componentes independentes precisam ser estimadas.
6. O algoritmo FastICA possui outras vantagens como: Paralelismo, é distribuído, computacionalmente simples e requer pouco espaço de memória.

2.6.9.3 SOBI - Second Order Blind Identification

Um outro algoritmo ICA muito usado com BSS é o SOBI este algoritmo tem origem no trabalho apresentado por Adel Belouchrani. Essa seção é sobretudo baseada em seu artigo sobre este algoritmo (BELOUCHRANI *et al.*, 1997)

O algoritmo SOBI baseia-se na diagonização conjunta aproximada de múltiplas matrizes de correlação com atrasos.

Dado um vetor branqueado $z(k)$, toma-se um conjunto de matrizes de correlação com atraso de $z(k)$, $R_z(\tau)$, $i = 1 \dots k$. O objetivo do algoritmo SOBI é encontrar uma transformação unitária V. tal que (Equação 2.46):

$$V^T R_z(\tau_1) V = D_i \tag{2.46}$$

Para $i = 1 \dots k$. onde D_i é um conjunto de matrizes diagonais.

Como o SOBI trabalha com várias matrizes de correlação, ele reduz a probabilidade de que uma escolha incorreta do atraso τ impeça a implementação da separação cega.

De forma simplificada o algoritmo SOBI pode ser implementado seguindo os seguintes passos:

1. Estimar a matriz de covariância amostral $R_x(0)$ a partir de n amostras;

2. Decompor $R_x(0)$ em $\lambda_1, \ldots, \lambda_n$, os n maiores autovalores e $h_1, \ldots, h_n$, autovetores;
3. Caso seja considerado ruído branco adicionado ao processo, uma estimativa da variância do ruído σ_v^2 pode ser feita pela média dos $m - n$ autovalores de $R_x(0)$;
4. Calcular a matriz de branqueamento como, por exemplo (Equação 2.47):

$$Q = [(\lambda_1 - \sigma_v^2)^{-1/2} h_1, \ldots, (\lambda_n - \sigma_v^2)^{-1/2} h_n]^T \tag{2.47}$$

5. Branquear os dados, determinando o vetor $z(k)$;
6. Formatar a matriz de correlação com atraso $R_z(\tau)$ através de amostras das matrizes de covariância de $z(k)$ para um conjunto fixo de atrasos τ;
7. Obter a matriz unitária U, que é diagonalizadora conjunta de $\{R_z(\tau_j) \mid j = 1, \cdots, F\}$
8. Estimar os sinais de saída conforme Eq:y(k):

$$y(k) = U^T Q x(k) \tag{2.48}$$

e/ou a matriz de mistura:

$$A = Q^+ U \tag{2.49}$$

2.6.9.4 JADE - Joint Approximation Diagonalization of Eigenmatrices.

O JADE é outro algoritmo muito usado em separação cega de fontes (BSS), sobretudo quando se utiliza o modelo ICA. O algoritmo JADE foi proposto por J.F.Cardoso e Souloumiac, A. em 1993.

O objetivo da ICA é recuperar os sinais de fonte pura misturados nos sinais observados. Vários algoritmos diferentes foram propostos para calcular componentes independentes (CIs). Entre os quais se pode citar o FastICA (HYVÄRINEN; OJA, 1997), infoMAX (BELL; SEJNOWSKI, 1995), e a Diagonização Conjunta Aproximada de Autovalores (JADE) (CARDOSO; SOULOUMIAC, 1993). A principal vantagem do JADE sobre outros algoritmos é que é baseado em computação matricial, envolvendo diagonalização de matriz. O JADE é baseado na decomposição em autovalores das matrizes cumulantes dos sinais observados.

O JADE é um bom algoritmo BSS baseado em estatísticas de ordem superior. A principal restrição utilizada pelo algoritmo JADE para solucionar o problema da separação cega é que as fontes a serem separadas apresentam-se estatisticamente independentes. O

algoritmo explora esta independência estatística através de cumulantes de quarta ordem, ou melhor, através da diagonalização conjunta das matrizes de cumulantes de quarta ordem dos dados observados. Assim, o JADE reduz o conjunto de dados observados a um pequeno conjunto de estatísticas. Para realizar a diagonalização conjunta, o algoritmo emprega uma técnica semelhante ao método jacobiano de diagonalização e por este motivo ele é conhecido como algoritmo jacobiano.

Uma das principais vantagens apresentadas pelo JADE é que ele pode se mover em passos macroscópicos através do espaço de parâmetros, solucionando o problema causado pela escolha incorreta do passo de adaptação, comum em algoritmo que utilizam técnicas como o do gradiente natural (CICHOCKI; AMARI, 2002). Por outro lado, a implementação computacional do JADE apresenta grande complexidade, principalmente no que diz respeito aos cálculos das estatísticas de ordem superior.[3].

O Algoritmo JADE proposto inicialmente por Cardoso e Souloumic (CARDOSO; SOULOUMIAC, 1993), é baseado puramente em estatísticas. Este algoritmo pode ser implementado seguindo os seguintes passos.

1. Inicialização;
   (a) Calcular a matriz de covariância dos dados observador;
   (b) Calcular a matriz de branqueamento;
   (c) Branquear os dados observados;
2. Formação Estatística;
   (a) Calcular o conjunto Q_z de matrizes de cumulantes de quarta ordem dos dados branqueados;
   (b) Calcular os pares n mais significantes;
3. Otimização da função custo ortogonal;
   Encontrar a matriz de rotação U, tal que as matrizes de cumulantes sejam tão diagonais quanto possível, ou seja, resolver (Equação 2.50):

$$U = arg\,min \sum off(U^T Q_z U) \tag{2.50}$$

   Este procedimento é descrito como diagonalização conjunta;

[3] Estatística de Ordem Superior (SOS), forma de tratar sinais que não são lineares e que apontam para a não gausseanidade (2.6.7)

4. Separação dos componentes independentes;
   Estimar a matriz de mistura, tal que:

$$A = UQ^{-1} \tag{2.51}$$

   e/ou estimar os componentes independentes, tais que:

$$y(k) = Wx(k) = V^T z(k) \tag{2.52}$$

   Onde:

$$W = U^T Q \tag{2.53}$$

O passo 1, está relacionado a estatística de segunda ordem, sendo implementado através da decomposição em autovalores da matriz de covariância R_x. Devido a consideração de ruído branco uma estimativa da variância do ruído é a média dos $m - n$ menores autovalores de R_x.

No passo 2, o cálculo das automatrizes está relacionado a diagonalização de uma matriz n^2 x n^2 dos elementos de Q_z.

O passo 3 é implementado estendendo-se a técnica jacobiana para várias matrizes. É importante observar ainda quando $n = 2$, a técnica jacobiana não é iterativa, ou seja, uma única rotação de Givens[4] é suficiente para a diagonalização conjunta. Mais uma vez fica claro que o algoritmo JADE é um algoritmo jacobiano, visto que o passo 3 é implementado através de uma técnica jacobiana. Além disso, as rotações não são aplicadas aos dados branqueados, mas as suas matrizes de cumulantes. Assim, o algoritmo atualiza não os dado, mas suas matrizes estatísticas.

O ponto chave do algoritmo JADE está na seleção das matrizes de cumulantes envolvidas na estimativa das fontes originais. As matrizes de cumulantes podem ser exata e conjuntamente diagonalizadas quando o modelo permite, o que não ocorre quando são processados dados do mundo real. Como apenas amostras das estatísticas estão disponíveis - segundo o modelo $x(k) = As(k)$ - não é possível, em geral, assegurar que o algoritmo encontre uma solução adequada (CARDOSO, 1999). Esta é, sem dúvida, mais uma razão que comprova a importância da seleção correta das matrizes de cumulantes, pois no caso de dados do mundo real, a impossibilidade de realizar a diagonalização conjunta exata, corresponde a impossibilidade de encontrar os componentes independentes $y(k)$. Assim, fazer um con-

[4]Em álgebra linear numérica, uma rotação de Givens é uma rotação no plano gerado por dois eixos de coordenadas. As rotações de Givens foram nomeadas em homenagem à Wallace Givens, que apresentou a técnica aos analistas numéricos na década de 1950, enquanto trabalhava no *Argonne National Laboratory*.

junto máximo de matrizes de cumulantes tão diagonais quanto possível, coincide com fazer os componentes de $y(k)$ tão independentes quanto possível (CARDOSO, 1999).

Existem várias formas de estimar o conjunto máximo de matrizes de cumulantes. Por exemplo, para bases canônicas, os elementos das matrizes de cumulantes são iguais aos cumulantes de $z(k)$. Uma outra opção é utilizar uma base simétrica; neste caso, é suficiente estimar e diagonalizar $n + n(n+1)/2$ matrizes de cumulantes simétricas. Existe ainda a opção de reduzir o tamanho das estatísticas necessárias para representar a informação de quarta ordem, o que somente pode ser aplicado se o modelo de separação permitir. Neste caso, as matrizes de cumulantes devem ter a estrutura mostrada na Equação 2.54.

$$Q_z(M) = UA_mU^T \tag{2.54}$$

Desta forma, o mapeamento $M \rightarrow Q_z(M)$ tem rank igual a n ou mais precisamente igual ao número de componentes com *kurtosis* diferente de zero. Isto ocorre porque existem n graus de liberdade para matrizes na forma UAU^T, nomeadamente, os n elementos da diagonal de A (CARDOSO, 1999). A partir desta afirmação e da propriedade de simetria dos cumulantes, pode-se dizer que existem n automatrizes $E_1, E_2, E_3, ..., En$, que são ortonormais e que satisfazem a relação:

$$Q_z(E_i) = \mu_i E_i \tag{2.55}$$

Onde o escalar μ_i é o autovetor correspondente.

Com emprego das automatrizes $E_1, E_2, E_3, ..., En$, verifica-se que toda informação contida em $Q_z(.)$ pode ser representada por n automatrizes. Assim, inserindo $M = u_i u_i^T$ na Equação 2.54 e considerando que U é unitária, verifica-se que o conjunto de auto matrizes é dado por $E_i = u_i u_i^T$ (CARDOSO, 1999)

O algoritmo JADE foi proposto inicialmente para realizar a ICA através da diagonalização conjunta aproximada de auto matrizes, onde a diagonalização conjunta foi somente empregada às n mais significantes auto matrizes de $Q_z(M)$ como uma forma de reduzir a carga computacional. Desta forma, o número de estatísticas é reduzido de n^4 cumulantes ou $n + n(n+1)/2$ matrizes de cumulantes simétricas, a um conjunto de n auto matrizes. Esta redução é obtida sem perdas somente quando o modelo permite, ou seja, não é recomendada a redução a auto matrizes quando são processados conjunto de dados para os quais não está claro se o modelo $x(k) = As(k)$ assegura boa precisão.

O JADE nesse trabalho de investigação está sendo usado para separar fontes de sinais cerebrais mas, o mesmo tem sido usado com sucesso em diversos outros trabalhos investigativos como por exemplo: no pre-processamento de sinais em sistema de sonar passivo (AMORIM *et al.*, 2014). em quimiometria[5] (RUTLEDGE DOUGLAS N E BOUVERESSE, 2013), processamento de sinais em telecomunicações (AMARI *et al.*, 1996) e outros.

Os fontes do algoritmo JADE para Matlab para baixar e executar no ambiente Matlab encontram-se no *GitHub* onde também podemos fazer um download do seu Leia-me.

Mais informações a respeito do algoritmo JADE podem ser encontradas nos artigos: Cardoso & Souloumiac (1993) e Cardoso (1999).

[5] Quimiometria é a aplicação de métodos estatísticos ou matemáticos em dados de origem química.

Capítulo Três

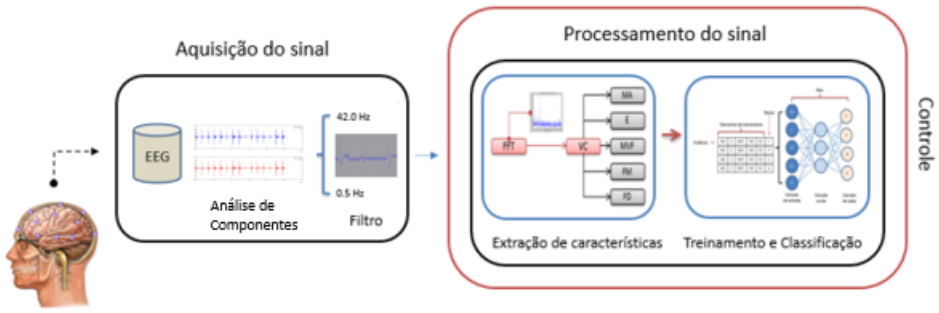
Metodologia e Desenvolvimento do Modelo

Sabe-se que, pessoas amputadas produzem sinais cerebrais semelhantes aos de pessoas não amputadas, ao pensarem nos movimentos dos membros superiores ou inferiores, mesmo que esses movimentos não sejam efetivamente realizados, só pensados (CHINI; BOEMER, 2007). Neste caso, é possível acionar máquinas a partir de pensamentos sobre os movimentos desses membros.

O trabalho de investigação propõe o desenvolvimento de uma interface cérebro-computador não invasiva baseada na imaginação de movimentos do punho esquerdo, punho direito, ambos os punhos e ambos os pés. Nesse projeto, o processo de aquisição dos sinais EEG, o pré-processamento, a extração de características e a classificação dos sinais serão realizados de maneira *offline*, o método utilizado para extração de características será a separação de artefatos através da Análise de Componentes Independente. Os sinais EEG serão filtrados em uma faixa de 0,5 a 42 Hz. Por estarem relacionados à imaginação do movimento, será adotado o modelo assíncrono para o controle temporal.

O modelo proposto mensura a atividade cerebral. Esta atividade é interpretada de modo a detectar diferentes padrões. Esta interação é concebida através da utilização de componentes funcionais intermediários, ilustrados na Figura 3.1.

Figura 3.1: Modelo computacional do projeto

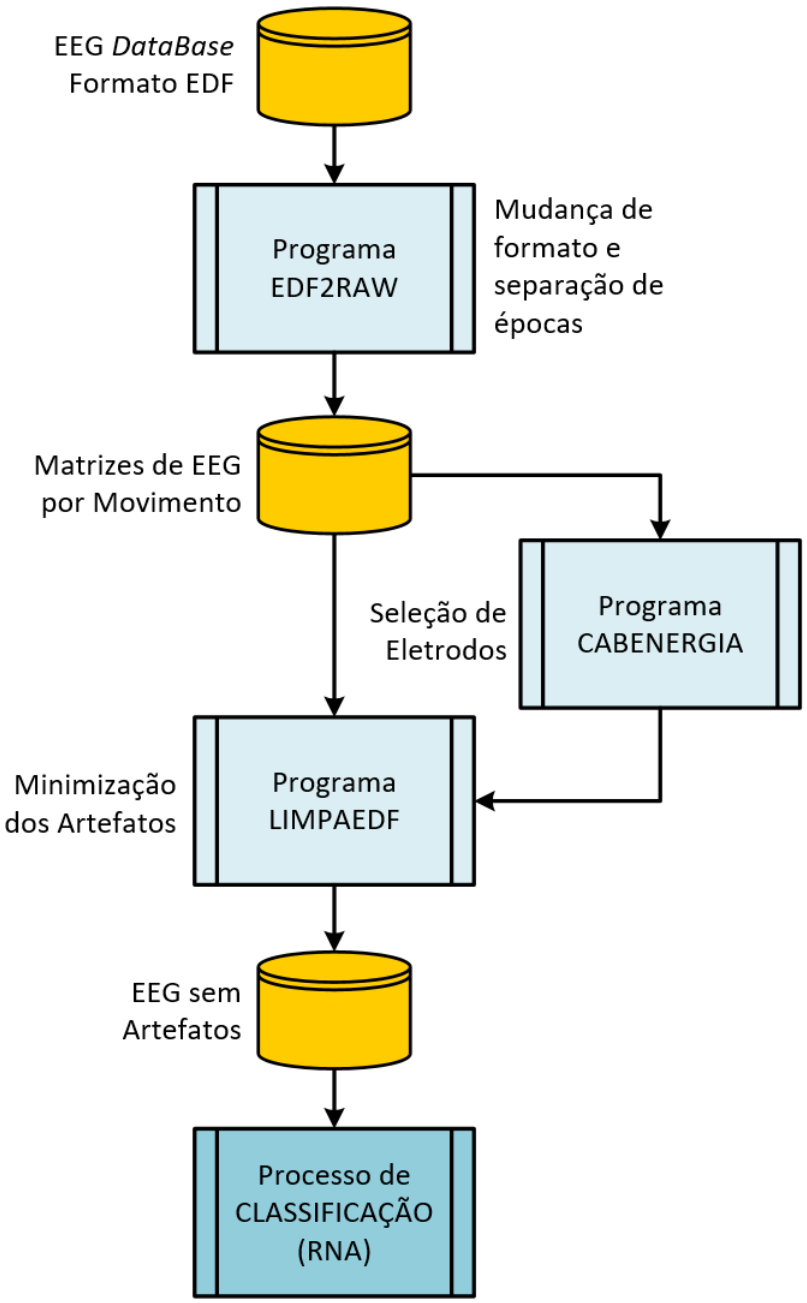


Fonte: Adaptado de (PINHEIRO, 2016)

3.1 Aquisição do sinal

Foi desenvolvida uma estrutura de software para preparar os dados de EEG para serem classificados como pode ser observado na Figura 3.2, nela destaca-se o encadeamento dos módulos de software desenvolvidos, que serão detalhados mais adiante.

Figura 3.2: Arquitetura de software para preprocessamento dos sinais de EEG



Fonte: autor

A atividade cerebral do usuário é medida pelos eletrodos. Os sinais EEG brutos são lidos; em seguida, ocorre a separação dos dados EEG para cada tipo de movimento e tratamento dos artefatos; os sinais EEG serão filtrados por um filtro digital passa-faixa, separando frequências na faixa de 0,5 a 42 Hz.

3.1.1 Base dos dados de origem

Os sinais dos registros EEG utilizados para validar a interface cérebro-computador foram adquiridos do banco de dados *eegmmidb*, capturado utilizando o sistema BCI2000 (SCHALK GERWIN E MCFARLAND, 2004), disponível através do PhysioBank (GOLDBERGER ARY L E AMARAL, 2000). Este banco de dados é composto por mais de 1500 registros de sinais EEG, obtidos a partir de 109 voluntários.

Cada voluntário realizou 14 sessões experimentais: dois ensaios de um minuto, sendo, um minuto com os olhos abertos e um minuto com os olhos fechados (sessões *baseline*), e três ensaios de dois minutos de cada uma das seguintes tarefas:

1. Um alvo aparece no lado esquerdo ou direito da tela. O voluntário abre e fecha o punho correspondente até que o alvo desapareça. Em seguida, o voluntário relaxa.
2. Um alvo aparece no lado esquerdo ou direito da tela. O sujeito imagina abrindo e fechando o punho correspondente até que o alvo desapareça. Em seguida, o voluntário relaxa.
3. Um alvo aparece na parte superior ou inferior da tela. O voluntário abre e fecha ambos os punhos (alvo no topo) ou ambos os pés (alvo na parte inferior) até que o alvo desapareça. Em seguida, o voluntário relaxa.
4. Um alvo aparece na parte superior ou inferior da tela. O voluntário imagina abrindo e fechando ambos os punhos (alvo no topo) ou ambos os pés (alvo na parte inferior) até que o alvo desapareça. Em seguida, o voluntário relaxa.

Em resumo, as 14 sessões experimentais foram as seguintes:

1. Baseline, olhos abertos.
2. Baseline, olhos fechados.
3. Tarefa 1 (abre e fecha o punho esquerdo ou direito).
4. Tarefa 2 (imagina abrindo e fechando o punho esquerdo ou direito).
5. Tarefa 3 (abre e fecha ambos os punhos ou ambos os pés).
6. Tarefa 4 (imagina abrindo e fechando ambos os punhos ou ambos os pés).
7. Tarefa 1 (abre e fecha o punho esquerdo ou direito).
8. Tarefa 2 (imagina abrindo e fechando o punho esquerdo ou direito).

9. Tarefa 3 (abre e fecha ambos os punhos ou ambos os pés).
10. Tarefa 4 (imagina abrindo e fechando ambos os punhos ou ambos os pés).
11. Tarefa 1 (abre e fecha o punho esquerdo ou direito).
12. Tarefa 2 (imagina abrindo e fechando o punho esquerdo ou direito).
13. Tarefa 3 (abre e fecha ambos os punhos ou ambos os pés).
14. Tarefa 4 (imagina abrindo e fechando ambos os punhos ou ambos os pés).

Os registros são disponibilizados no padrão *European Data Format* (EDF) (KEMP; OLIVAN, 2003), contendo registros de 64 eletrodos, cada um amostrado em uma taxa de 160 Hz e um canal de anotação. Cada anotação contém um dos códigos:

T0 corresponde ao período de descanso, com duração de 4,2 segundos.

T1 ocorre quando o alvo aparece do lado esquerdo ou superior da tela e corresponde ao início do movimento real ou imaginário do punho esquerdo ou ambos os punhos. Tem duração de 4,1 segundos. Os movimentos do punho esquerdo ocorrem nas sessões 3, 4, 7, 8, 11, e 12. Os movimentos de ambos os punhos ocorrem nas sessões 5, 6, 9, 10, 13 e 14.

T2 ocorre quando o alvo aparece do lado direito ou inferior da tela e corresponde ao início do movimento real ou imaginário do punho direito ou ambos os pés. Tem duração de 4,1 segundos. Os movimentos do punho direito ocorrem nas sessões 3, 4, 7, 8, 11, e 12. Os movimentos de ambos os pés ocorrem nas sessões 5, 6, 9, 10, 13 e 14

Essas anotações (T0, T1 e T2) são utilizadas para separar, dentro de cada sinal EEG de 2 minutos, o início e o fim de cada tipo de movimento, (executado ou imaginado) para punhos e pés, bem como separar do EEG, os momentos de pausa do voluntário.

Os registros relacionados à imaginação de movimentos (punho esquerdo, punho direito, ambos os pés e ambos os punhos) serão utilizados para validação do modelo e serão extraídos seguindo a orientação disposta na Tabela 3.1.

Os registros de 105 voluntários foram utilizados nos testes. Dados de 4 voluntários (88, 92, 100 e 104) se mostraram corrompidos na etapa de aquisição; por esse motivo, tais registros não foram considerados. Neste trabalho dos 64 eletrodos disponíveis na base, foram utilizados os 17 eletrodos posicionados na região do córtex frontal, numerados de 22 a 38 conforme ilustrado na Figura 3.3.

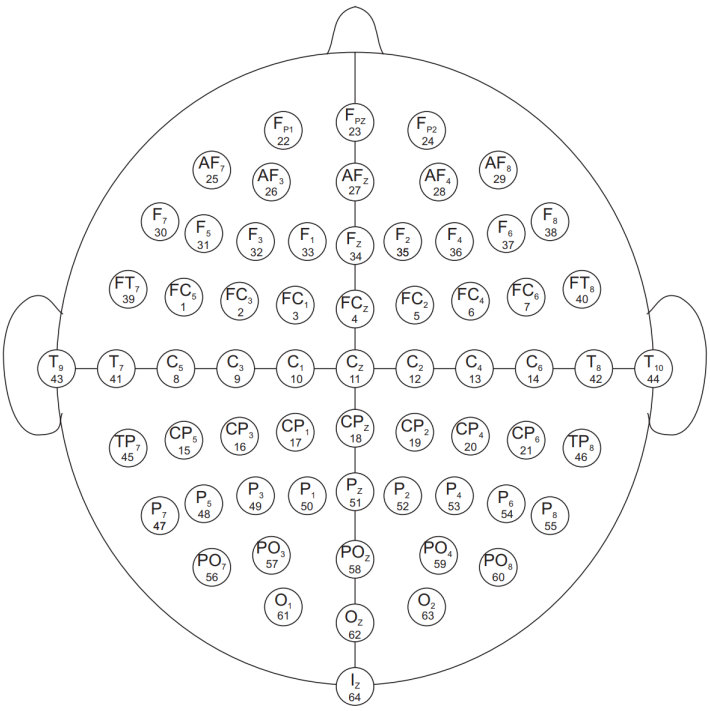
Tabela 3.1: Resumo da Preparação dos Dados

Movimentos Imaginados	**Sessões**	**Tarefas**	**Anotações**
Punho Esquerdo	04; 08; 12	2	T1
Punho Direito	04; 08; 12	2	T2
Ambos os Punhos	06: 10; 14	4	T1
Ambos os Pés	06; 10; 14	4	T2

Fonte: Autor

Foram utilizados os eletrodos do córtex frontal por se mostrarem mais sensíveis a esse tipo de experimento, como mostrado na subseção 3.1.3.

Figura 3.3: Distribuição dos Eletrodos pelo Escalpo segundo o Sistema 10/20



Fonte: (GOLDBERGER ARY L E AMARAL, 2000)

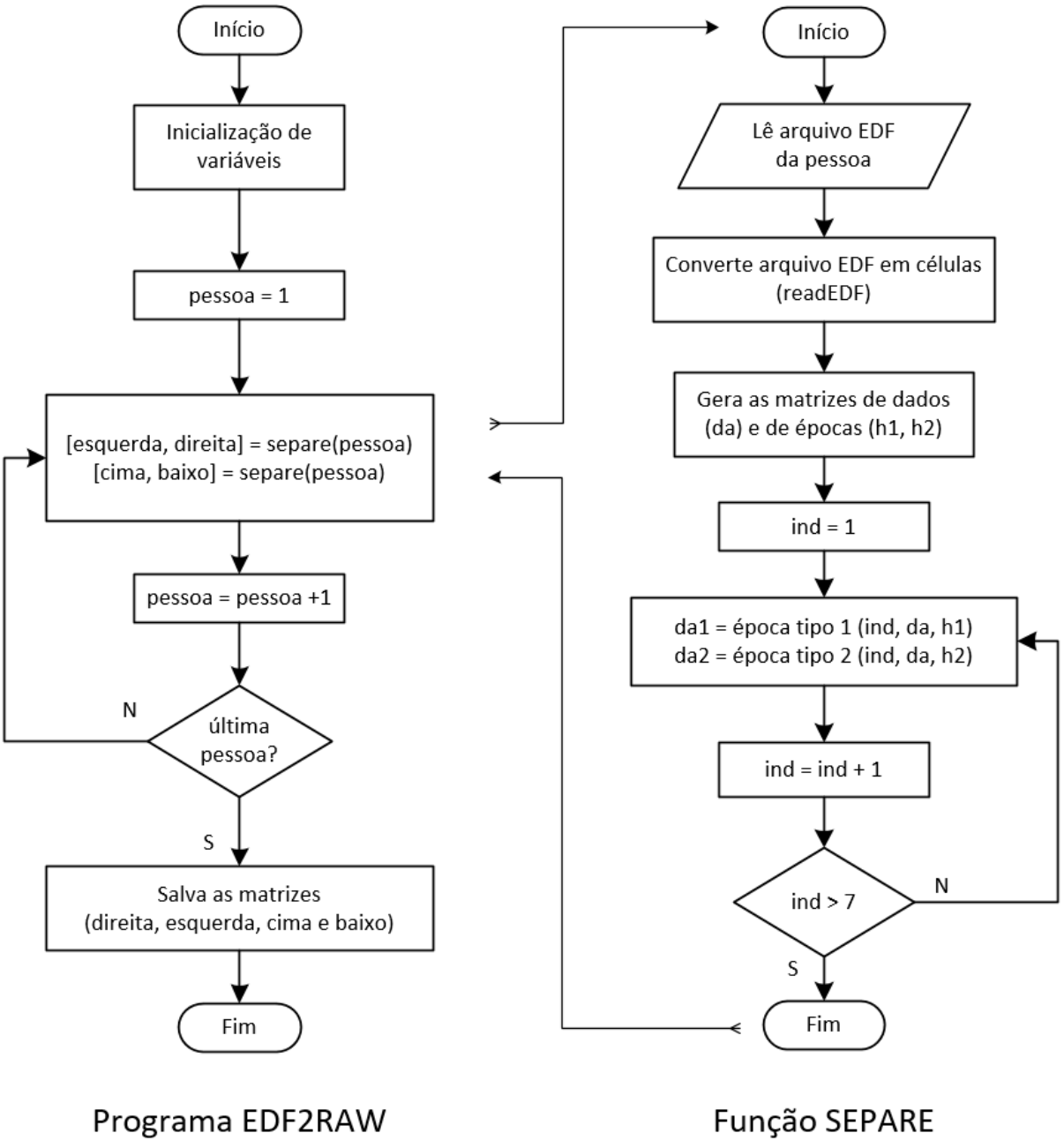
3.1.2 Separação de Épocas dos Sinais EEG

Antes de analisar mais detidamente os programas desenvolvidos, é desejável falar da linguagem usada para modelar esse experimento. Várias linguagens foram pensadas até ficar decidido pela linguagem MatLab, alguns fatores pesaram nessa escolha, entre eles a capacidade de desenvolver programas complexos utilizando poucas linhas de programação, a capacidade de trabalhar com blocos de matrizes enormes de forma confiável, capacidade de implementar de forma fácil toda a álgebra linear e vetorial e ter bibliotecas confiáveis e já consagradas para manipulação de sinais biológicos. Outro fator que pesou, a maioria dos trabalhos científicos que modelam esse tema, tem na plataforma MatLab sua ferra-

menta principal de implementação. Isso sem falar da grande bibliografia a respeito dessa linguagem nos principais idiomas do planeta.

Uma vez decidida a linguagem, seguem descrição dos programas, salientando que os fontes dos mesmos, na linguagem MatLab, se encontram no apêndice A. Os dados capturados dos eletrodos, que estão disponíveis no formato EDF, foram selecionados e convertidos para o formato de dados do Matlab através das rotinas: **EDF2RAW.m** e **SEPARE1.m**, detalhadas nos apêndices: A.1 e A.2, cujo fluxograma pode ser observado na Figura: 3.4

Figura 3.4: Programa edf2raw



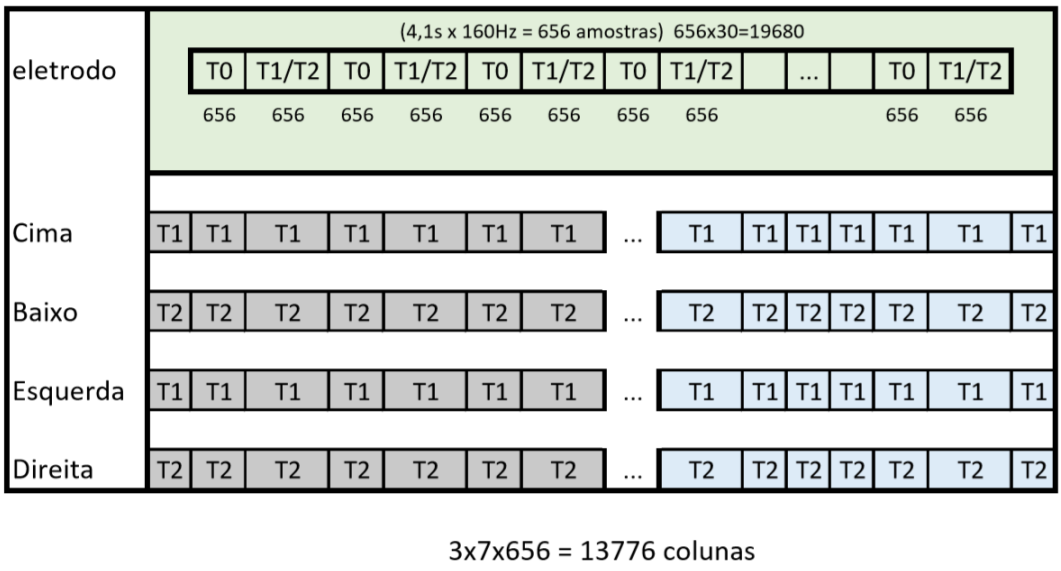
Fonte: autor

A rotina **EDF2RAW** lê os dados no formato EDF, seleciona somente os EEG de movimentos imaginados, e através da rotina **SEPARE1** faz a separação dos movimentos dos punhos e pés segundo o procedimento:

- `edf2raw inicio`
    1. `inicia pessoa`
    2. `próxima pessoa`
    3. `lê sessões 04, 08 e 12 da pessoa`
    4. `separa movimentos de esquerda e direita`
    5. `lê sessões 06, 10 e 14 da pessoa`
    6. `separa movimentos dos pulsos e dos pés`
    7. `armazena resultados`
    8. `loop até a ultima pessoa`
- `edf2raw fim`

Como resultado, gera quatro matrizes, (CIMA, BAIXO, ESQUERDA e DIREITA), cada uma com os sinais dos 64 eletrodos de todos os voluntários, para cada tipo de movimento imaginado; para cima (dois punhos), para baixo (dois pés), para a esquerda (punho esquerdo) e para a direita (punho direito), excluindo os sinais de pausa. Cuja estrutura pode ser observada na Figura 3.5 e Figura 3.6.

Figura 3.5: Separação dos dos sinais EEG



Fonte: autor

Figura 3.6: Detalhe da matriz de movimentos brutos

Esquerda				
	Eletrodo1	...	Eletrodo64	
Época1	656	...	656	Tarefa1
Época2	656	...	656	
Época3	656	...	656	
Época4	656	...	656	
Época5	656	...	656	
Época6	656	...	656	
Época7	656	...	656	
Época1	656	...	656	Tarefa2
Época2	656	...	656	
Época3	656	...	656	
Época4	656	...	656	
Época5	656	...	656	
Época6	656	...	656	
Época7	656	...	656	
Época1	656	...	656	Tarefa3
Época2	656	...	656	
Época3	656	...	656	
Época4	656	...	656	
Época5	656	...	656	
Época6	656	...	656	
Época7	656	...	656	

Fonte: autor

Observa-se que na estrutura original dos dados de EEG, cada eletrodo é observado por 123 segundos divididos em 30 épocas[1] de 4,1 segundos cada, a uma taxa de amostragem de 160 Hz, resultando num total de 19680 amostras.

Cada época é formada por 656 amostras pois cada observação tem duração de 4,1 segundos e são classificadas em época tipo T0, T1 e T2, as épocas tipo T0 foram eliminadas pois, representam períodos de pausa ou descanso.

Foram utilizadas sete épocas tipo T1 e sete épocas do tipo T2.

Nesta etapa de tratamento dos dados brutos, além da eliminação dos tempos de pausa, foram juntados os sinais do mesmo tipo de todas as três tarefas para cada um dos eletrodos,

[1] Época, são trechos especificados de um EEG. Ex: época de três em três segundos

resultando em quatro matrizes, uma para cada tipo de movimento imaginado.

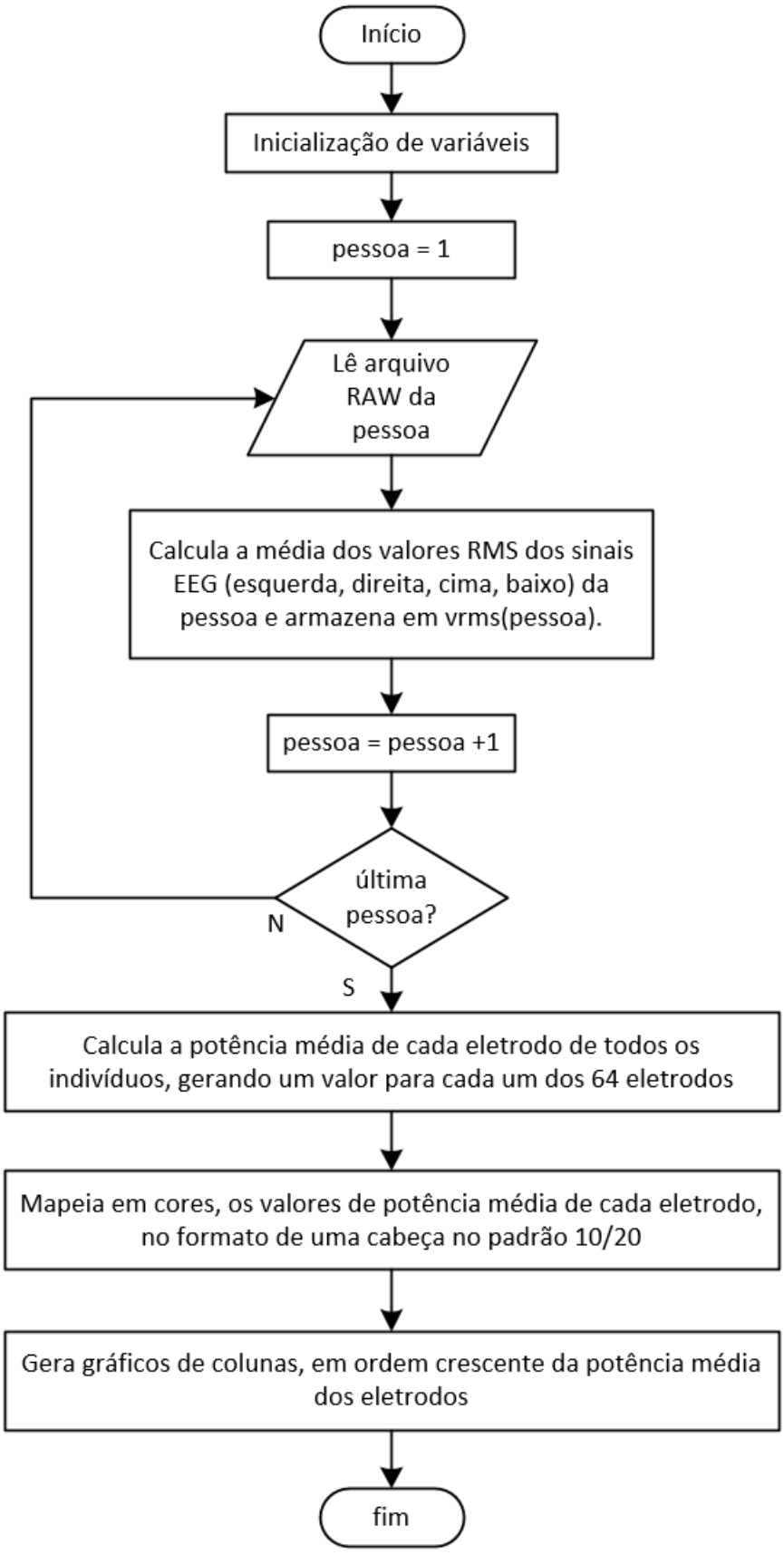
Cada uma dessas matrizes tem 64 linhas, uma para cada eletrodo, e 13776 colunas, correspondentes a três tarefas de sete movimentos de 4,1 segundos cada, como pode ser observado na Figura 3.6.

3.1.3 Seleção dos eletrodos

Inicialmente houve a necessidade de se classificar os 64 eletrodos em relação a intensidade do sinal que recebiam, para verificar sua importância na detecção dos movimentos imaginados, pois, a redução do número de eletrodos reduz o custo computacional.

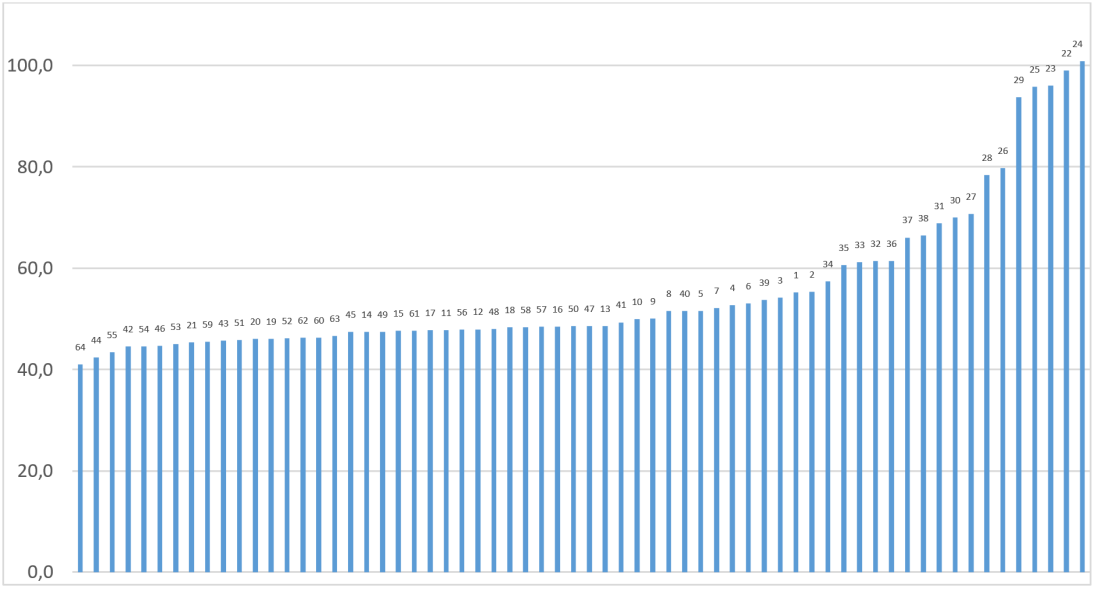
Para isso, foi desenvolvida a rotina **CABENERGIA**, detalhada no apêndice: A.3, cujo fluxograma pode ser observado na Figura: 3.7, que calcula o valor eficaz (RMS) dos sinais de cada eletrodo, considerando todos os 105 voluntários e todos os pensamentos efetuados, num total de 8820 movimentos imaginados, sendo 21 para cada voluntário em 3 sessões de 7 movimentos, para cada um dos 4 sentidos (cima, baixo, esquerda, direita).

Figura 3.7: Programa Cabenergia



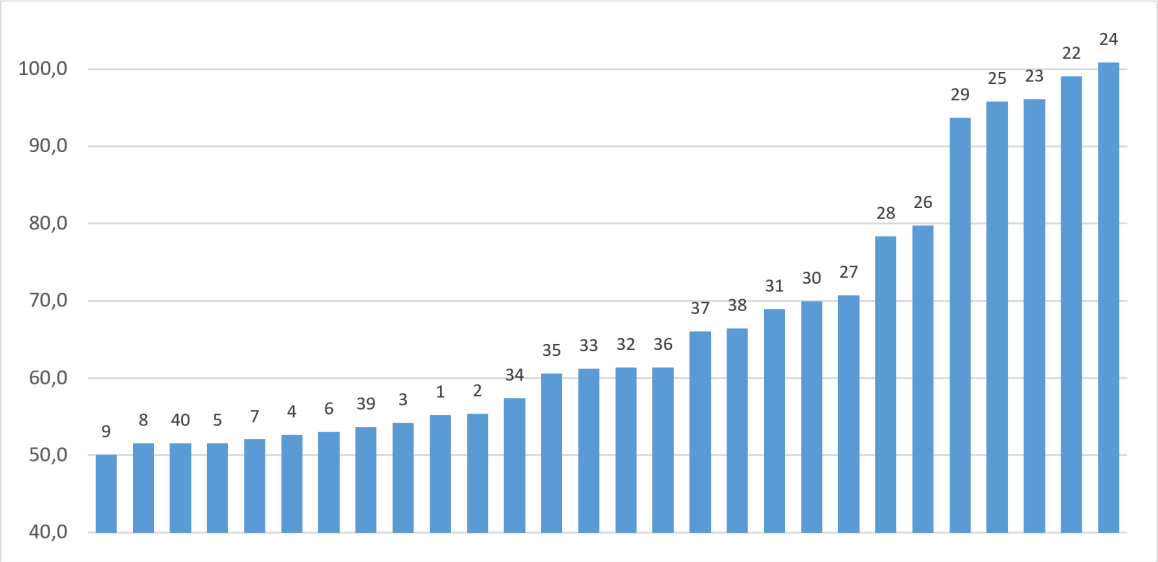
Fonte: autor

Figura 3.8: Tensão média eficaz no 64 eletrodos em ordem crescente



Fonte: autor

Figura 3.9: Tensão média eficaz nos eletrodos mais potentes em ordem crescente

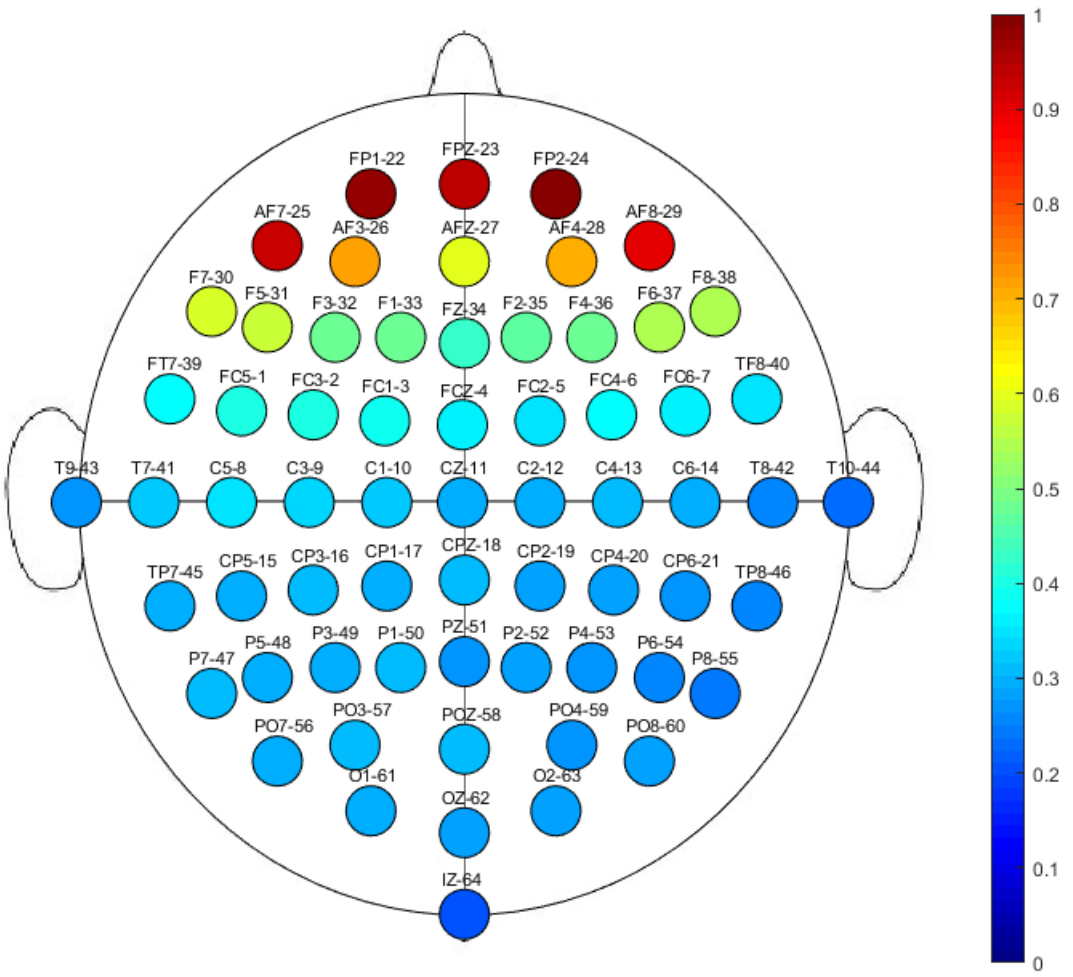


Fonte: autor

Observa-se na Figura 3.8 os valores eficazes dos 64 eletrodos em ordem crescente, e na Figura 3.9 os mais potentes.

Verificando a posição dos sensores na cabeça, Figura 3.10, observa-se que os eletrodos frontais são os mais sensíveis ao experimento, sendo assim foram selecionados para este estudo os eletrodos de 22 ao 38, os outros eletrodos não foram considerados neste experimento.

Figura 3.10: Potência média relativa dos sinais imaginados nos voluntários do Banco de Dados



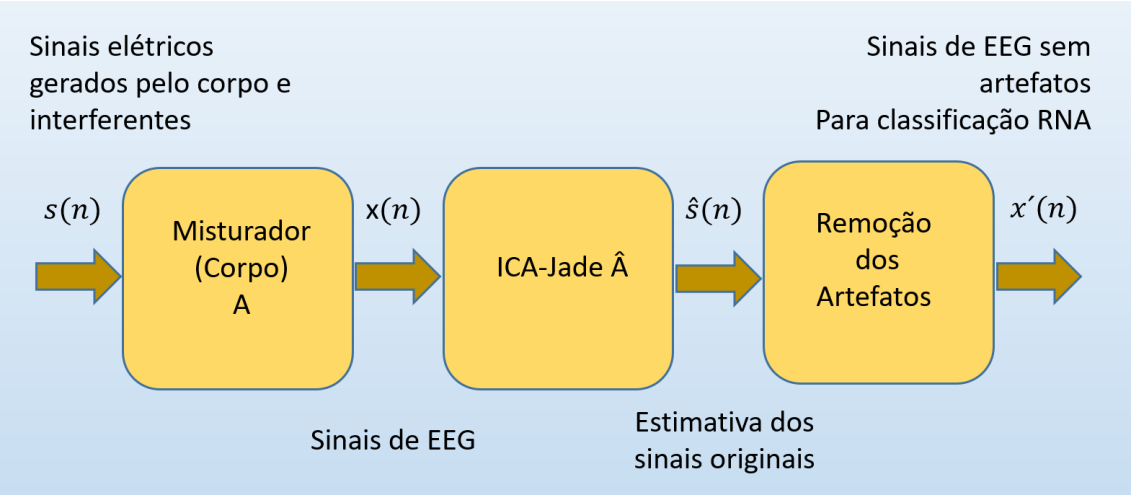
Fonte: adaptação do autor

Observa-se que as funções que elaboraram a Figura 3.10, é uma maneira conveniente de mostrar graficamente os eletrodos que tem os sinais mais potentes dos dados de um determinado experimento de EEG, seja um único EEG, ou um conjunto de EEG como foi o caso desse estudo investigativo. Nesse caso cada eletrodo na cabeça, Figura 3.10, representa o valor RMS dos sinais. Este desenvolvimento, ajudou a selecionar os melhores eletrodos à investigação.

3.1.4 Tratamento dos artefatos

A técnica de separação de componentes independentes pode ser utilizada para remoção de artefatos dos sinais de EEG como pode ser observado na Figura 3.11.

Figura 3.11: Separação de componentes independentes na remoção de artefatos em EEG



Fonte: autor

Os dados dos sinais de EEG não estão puros como desejável, pois os mesmos estão naturalmente contaminados com sinais elétricos dos movimentos cardíacos, musculares e oculares, dentre outros (HYVÄRINEN AAPO E OJA, 1999), assim, é necessário que se minimize as interferências destes sinais/artefatos sobre os sinais de imaginação dos movimentos dos membros, que é o foco desse trabalho.

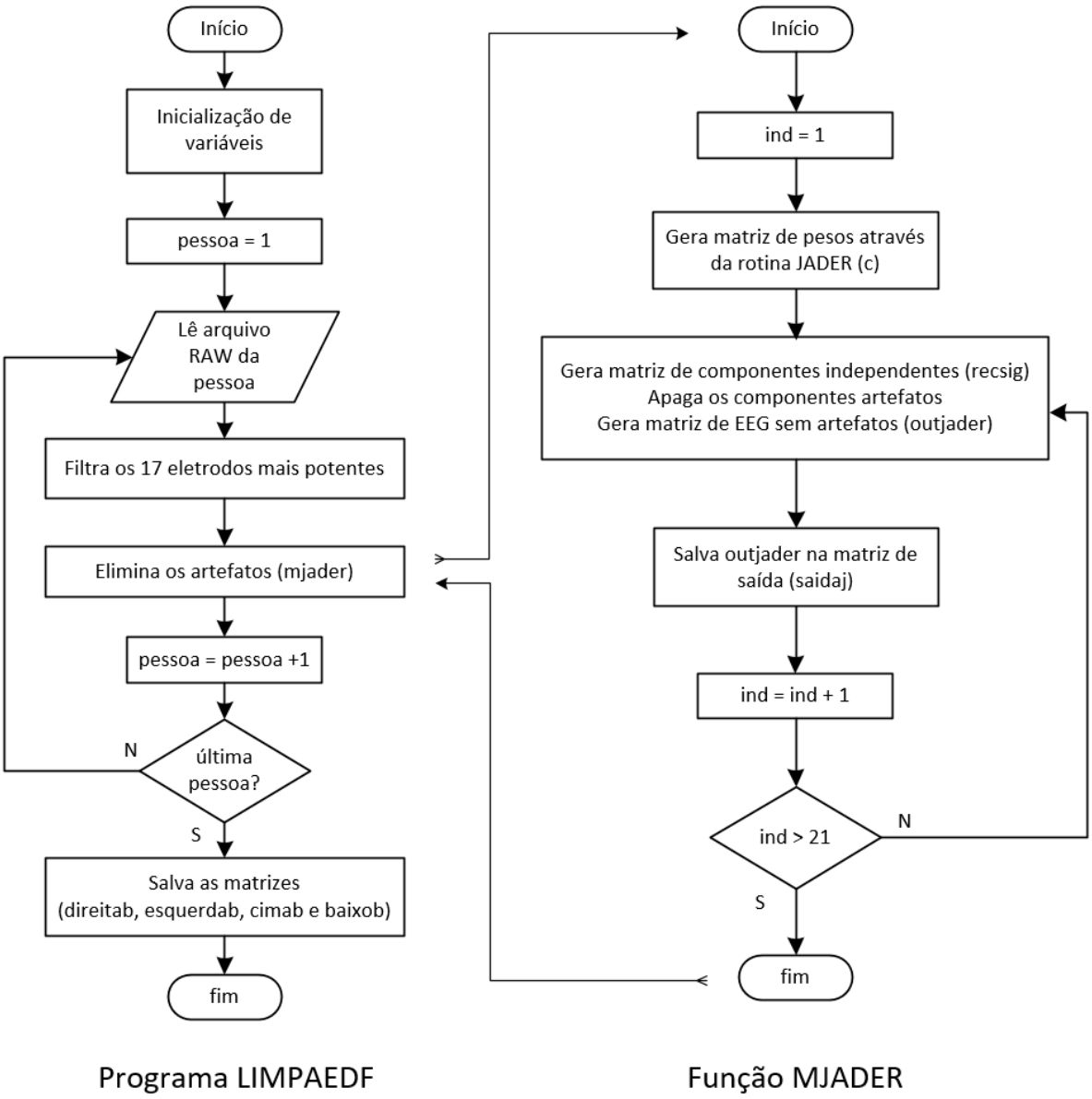
* Separação dos componentes - foram considerados apenas os sinais dos eletrodos frontais que são mais afetados por essa atividade cerebral, conforme demonstramos na Figura 3.10. Para a minimização dos efeitos dos artefatos, foi usada a técnica de separação de componentes independentes (ICA), implementada através do algoritmo JADE, detalhado na subseção 2.6.9.4. O algoritmo JADE (CARDOSO, 1999) de separação de fontes independente (ICA) que faz parte da família de algoritmos BSS, vide a seção 2.6.

* Minimização dos artefatos - para a miminização dos artefatos foram desenvolvidas as rotinas **LIMPAEDF** e **MJADER**, detalhadas nos apêndices: A.4 e A.5, cujo fluxograma pode ser observado na Figura 3.12. A rotina **LIMPAEDF** lê os dados brutos de cada voluntário, apenas os 17 eletrodos do córtex frontal e passa para a rotina **MJADER** que:

  1. Separa os 17 componentes independentes através do algorítimo JADE.
  2. Classifica os componentes independentes por ordem de potência do sinal.
  3. Minimiza os artefatos.
  4. Recompõe os sinais de EEG já sem os artefatos, e devolve para a rotina **LIMPAEDF**, que armazena os resultados para classificação por uma rede neural.

- `limpaedf inicio`
    1. `inicia pessoa`
    2. `próxima pessoa`
    3. `minimiza os artefatos movimentos de esquerda`
    4. `minimiza os artefatos movimentos de direita`
    5. `minimiza os artefatos movimentos de cima`
    6. `minimiza os artefatos movimentos de baixo`
    7. `armazena resultados`
    8. `loop até a ultima pessoa`
- `limpaedf fim`

Figura 3.12: Programa limpaedf



Fonte: autor

Após a separação dos componentes independentes pelo algorítimo JADE, os artefatos foram minimizados através da eliminação dos componentes de maior potência, de acordo com Cardoso (1999). Os demais componentes foram recombinados para gerar os sinais originais de EEG dos eletrodos, considerando que os mesmos agora têm uma menor influência dos artefatos.

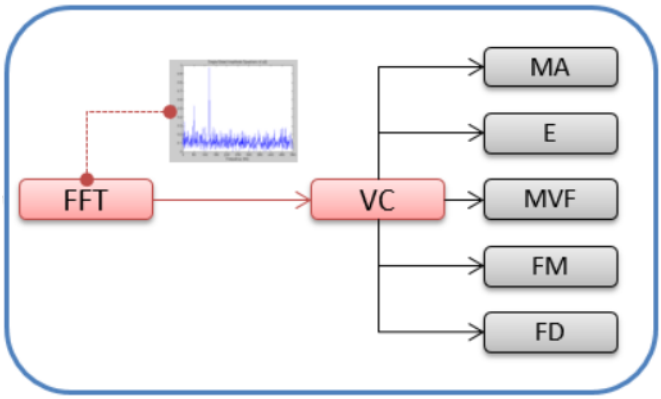
3.2 Processamento do sinal

O módulo de processamento do sinal é responsável pela recepção dos dados filtrados. Essa etapa consiste no processo de extração de características, treinamento e criação dos classificadores especialistas do modelo.

3.2.1 Extração de características

Após a recepção desses dados, calcula-se a Fast Fourier Transform (FFT) a partir dos dados de entrada. O sistema de controle, baseado em informações de amplitude e frequência dos sinais EEG, calcula os atributos que contém os padrões de entrada para o vetor de características (VC), esquema ilustrado na Figura 3.13.

Figura 3.13: Extração de características.



Fonte: (PINHEIRO, 2016)

O vetor de características é composto por alguns atributos que contém o padrão a ser analisado pelo processo de classificação. Esses atributos são:

- **Média aritmética (MA)**: representa o valor médio do sinal. O atributo é calculado

pela Equação 3.1.

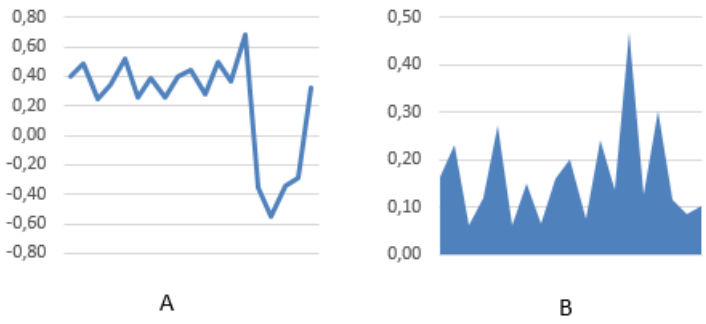
$$m(s) = \frac{1}{N} \sum_{N}^{n=1} S[n] \tag{3.1}$$

na qual s é o valor do sinal da FFT e N o tamanho do vetor de características (S).

- **Energia (E)**: representa a área sob a curva descrita pela função do sinal elevada ao quadrado, calculado pela Equação 3.2. A Figura 3.14 ilustra o sinal e a área sob a curva descrita pelo quadrado da função deste sinal.

$$E_2(s) = \frac{1}{N} \sum_{N}^{n=1} |s[n]|^2 \tag{3.2}$$

Figura 3.14: Sinal original (A) e o quadrado de seu módulo (B).



Fonte: (PINHEIRO, 2016)

- **Máximo valor da FFT (MVF)**: representa o ponto onde ocorre o máximo valor da curva da FFT, calculado pela Equação 3.3.

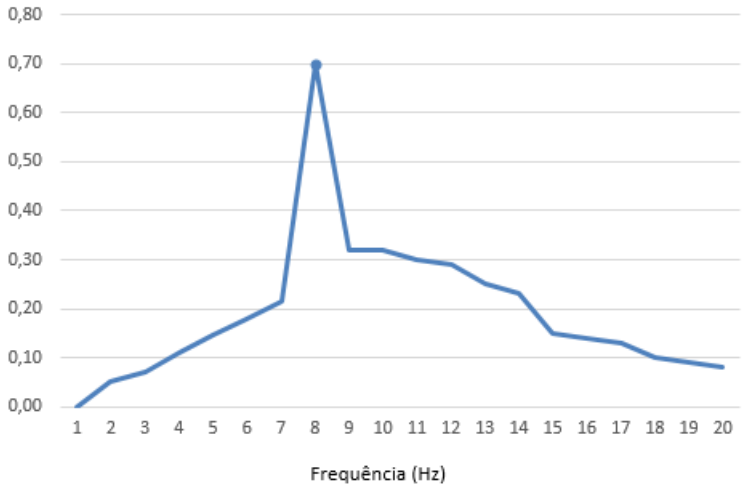
$$MVF(s) = max(S[n]) \tag{3.3}$$

- **Mínimo valor da FFT (FM)**: representa o ponto onde ocorre o valor mínimo da FFT, calculado pela Equação 3.4.

$$FM(s) = min(S[n]) \tag{3.4}$$

- **Frequência dominante (FD)**: representa a frequência que possui maior amplitude, é a coordenada X do ponto onde ocorre o máximo valor da curva da FFT. A Figura 3.15 ilustra a frequência principal: 8 Hz.

Figura 3.15: Frequência dominante.



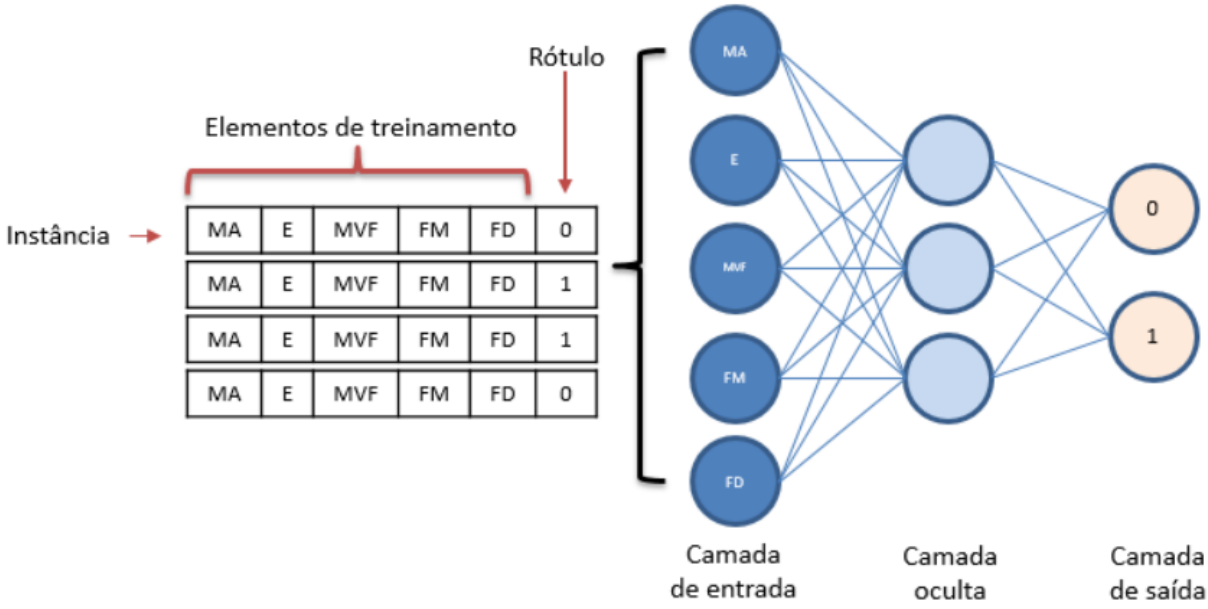
Fonte: (PINHEIRO, 2016)

O resultado deste módulo, é a composição de cinco atributos que formam o vetor de características a partir do sinal de entrada, o qual servirá de entrada para os classificadores (RNA's).

3.2.2 Classificação

O processo de classificação é realizado em duas etapas: treinamento (etapa de aprendizado) e classificação dos dados. Na etapa de treinamento, o modelo do classificador é construído descrevendo um conjunto pré-determinado de classes. A Figura 3.16 apresenta o esquema de comportamento do processo de classificação.

Figura 3.16: Classificação.



Fonte: (PINHEIRO, 2016)

O sistema de controle extrai os atributos do vetor de características (MA, E, MVF, FM e FD) para realizar o treinamento dos classificadores (RNA's), ou seja, construir o modelo de cada classificador, através do conjunto de dados de treinamento. Aos elementos de treinamento, são associados rótulos de classes às quais cada um pertence, sendo: 0 (corresponde a um exemplo falso) e 1 (corresponde a um exemplo verdadeiro). Esta etapa é conhecida como aprendizagem supervisionada, pois o rótulo da classe de cada elemento de treinamento é fornecido ao classificador.

No modelo, teremos quatro classificadores especializados, ou seja, uma RNA será treinada, através de exemplos, para classificar uma determinada imaginação de movimento (Tabela 3.2).

Tabela 3.2: Classificadores especializados.

RNA	Imaginação movimento
Esquerdo	Punho esquerdo
Direito	Punho direito
Punho	Ambos os punhos
Pé	Ambos os pés

Fonte: Autor.

Na etapa de classificação dos dados, o modelo é utilizado para a tarefa de classificação, ou seja, a acurácia preditiva do classificador será estimada. Segundo Han & Kamber (2006), se o conjunto de treinamento for usado para medir a acurácia do classificador, o resultado

será otimista, pois o classificador tende a excessivamente se ajustar aos dados. Nesse caso, será utilizado um conjunto de dados formado com elementos de testes e seus rótulos de classe associados (Figura 3.16). Esses elementos serão selecionados aleatoriamente do conjunto total de dados e são independentes dos elementos de treinamento, ou seja, não devem ser usados para construir o classificador.

A acurácia de um classificador de um dado conjunto de testes é a porcentagem de elementos do conjunto de testes que são corretamente classificados por ele. Para cada elemento de teste, o atributo classe é confrontado com a predição da classe realizada pelo classificador em análise. Se a acurácia do classificador é considerada aceitável, o classificador pode ser usado para classificar elementos futuros dos quais não se conhece a classe (HAN; KAMBER, 2006).

Neste trabalho de investigação, será utilizada, como medida de desempenho dos classificadores (Tabela 3.2) a taxa de acerto. Nesse caso, o sistema de controle determina a classe de cada instância (Figura 3.16): se a classificação está correta, conta-se como um acerto; caso contrário, conta-se como um erro. A taxa de acerto é a proporção de classificações corretas sobre o conjunto inteiro de instâncias. Dessa forma, será mensurado o desempenho global do classificador. Pretende-se conhecer o desempenho do classificador a partir de dados não utilizados na fase de treinamento, pois os dados utilizados na fase de treinamento, já são conhecidos pelo classificador. A taxa de acerto adquirida sobre o conjunto de treinamento é um bom indicador de desempenho na tarefa de classificar dados desconhecidos (não apresentados anteriormente ao classificador). Este conjunto de dados é chamado de conjunto de testes. É importante que os dados de testes não tenham participado da criação do modelo do classificador. Segundo Hall *et al.* (2009), em termos práticos é comum reservar 30% do conjunto de dados para testes e usar o restante (70%) para o treinamento do classificador.

Considerando que temos quatro classificadores e que a saída (resultado) desses classificadores será duas classes (0 - não ou 1 - sim), dois conceitos válidos são:

- **Instâncias positivas**: instâncias da classe principal de interesse, por exemplo, imaginação de movimento do punho esquerdo = 1;
- **Instâncias negativas**: por exemplo, imaginação de movimento do punho esquerdo = 0.

Os verdadeiros positivos referem-se às instâncias positivas que foram classificadas corretamente; por exemplo, instâncias da classe imaginação de movimento do punho esquerdo = 1, que foram corretamente classificadas como tais, enquanto verdadeiros negativos são instâncias negativas que foram classificadas corretamente; por exemplo, instâncias da classe

imaginação de movimento do punho esquerdo = 0, que foram corretamente classificadas. Já os falsos positivos são instâncias negativas que foram classificadas incorretamente; por exemplo, instâncias da classe imaginação de movimento do punho direito = 0, mas que o classificador as rotulou como 1. Os falsos negativos são instâncias positivas que foram classificadas incorretamente; por exemplo, instâncias da classe imaginação de movimento do punho direito = 1, que o classificador as rotulou como 0. Assim, torna-se interessante saber o quão bem o classificador pode reconhecer instâncias da classe, por exemplo, imaginação de movimento do punho esquerdo = 1 (instâncias positivas) e o quão bem ele pode reconhecer instâncias da classe imaginação de movimento do punho esquerdo = 0 (instâncias negativas).

Para esse propósito, a sensibilidade (Equação 3.5) e a especificidade (Equação 3.6) podem ser usadas:

- **Sensibilidade**: taxa de reconhecimento de verdadeiros positivos, ou seja, é a proporção de instâncias positivas que são corretamente identificadas.
- **Especificidade**: taxa de reconhecimento de falsos negativos, ou seja, é a proporção de instâncias negativas que são corretamente identificadas.

Segundo Han & Kamber (2006), essas medidas são definidas como:

$$SSB = \frac{vp}{p} \tag{3.5}$$

$$ESP = \frac{vn}{n} \tag{3.6}$$

Onde **vp** é o número de verdadeiros positivos; **p** é o número de instâncias positivas; **vn** é o número de verdadeiros negativos; **n** é o número de instâncias negativas. Além disso, é possível usar a precisão (Equação 3.7) para obter o percentual de instâncias rotuladas, por exemplo, como: imaginação de movimento do punho esquerdo, que de fato, são instâncias da classe imaginação de movimento do punho esquerdo = 1.

$$PCS = \frac{vp}{vp + fp} \tag{3.7}$$

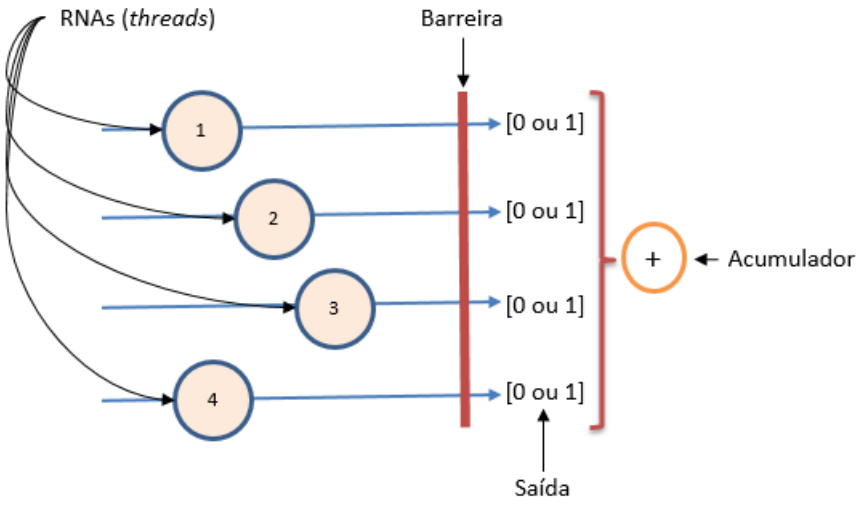
Onde **fp** é o número de falsos positivos. O cálculo da acurácia é definido em função da sensibilidade e da especificidade, calculado pela Equação 3.8.

$$ACR = SSB\frac{p}{p+n} + ESP\frac{n}{p+n} \tag{3.8}$$

O modelo da rede neural artificial utilizado neste trabalho de investigação será a *Muilti Layer Perceptron* (MLP) com o algoritmo de retropropagação do erro (*backpropagation*). Segundo Han & Kamber (2006), a principal característica que torna a MLP interessante para solução de problemas é a capacidade de aprender através de exemplos e de generalizar este aprendizado de forma que seja capaz de reproduzir um resultado a uma entrada diferente da apresentada. Hall *et al.* (2009), afirma que a rede MLP permite a solução de problemas não lineares, como é o caso dos sinas EEG.

O critério de decisão do sistema de controle é baseado na saída (respostas) das quatro RNA's: Esquerdo (*thread* 1), Direito (*thread* 2), Frente (*thread* 3), Parar (*thread* 4). A Figura 3.17 ilustra o esquema de critério de decisão do sistema.

Figura 3.17: Critério de decisão do sistema de controle.



Fonte: Autor.

A instância de dados formada pelo vetor de características será submetida às quatro RNA's. O sistema de controle irá criar 4 *threads*, representando cada uma das RNA's. Essa tarefa implica em um ponto de sincronização entre os processos (Barreira). Isto significa que todos os processos devem chegar a um determinado ponto, informar o resultado da classificação, antes de poderem começar a executar tudo novamente. A tarefa de decisão só acorrerá após todas as RNA's tiverem informados o resultado da classificação. Assim, o sistema de controle acumula (Equação 3.9) o resultado de cada uma das RNA's. Caso o acumulador seja igual a 1 (ACM = 1), o sistema informa a classe da imaginação do movimento (A, B, C, D); caso contrário, o sistema descartará o resultado e iniciará

novamente todo o processo.

$$ACM = R_1 + R_2 + R_3 + R_4 \tag{3.9}$$

Onde R_1 é a saída (resultado) da RNA esquerdo, R_2 é a saída RNA direito, R_3 é a saída da RNA frente e R_4 é a saída RNA parar.

3.3 Considerações finais

O projeto da interface cérebro-computador, desenvolvido neste trabalho de investigação, utilizou o sinal EEG para realizar a medição das atividades cerebrais. A aquisição dos sinais EEG foi realizada através do banco de dados *eegmmidb - EEG Motor Movement/Imagery Dataset*. O passo seguinte foi a separação dos dados de EEG para cada tipo de movimento e tratamento dos artefatos.

O conjunto de dados adquiridos por meio dos sinais EEG alimenta o sistema com uma grande quantidade de informação. O mecanismo de extração de características implementado nesse projeto, utiliza a FFT no sinal EEG. Em seguida, associa um conjunto de informação que caracteriza os fenômenos observados a uma estrutura denominada vetor de características. As etapas de identificação e extração de características dos sinais são de extrema importância para o bom desempenho dos classificadores.

A etapa de classificação dos sinais EEG, refere-se à tradução das características extraídas em comandos. O mecanismo de classificação implementado no projeto utiliza como solução 4 RNA's especialistas, *perceptrons* multicamadas. O objetivo da classificação é associar automaticamente uma instância do vetor de características a uma determinada classe. Esta classe identifica a modalidade da tarefa cognitiva imaginada pelo usuário do sistema.

Capítulo Quatro

Classificação, Avaliação e Resultados

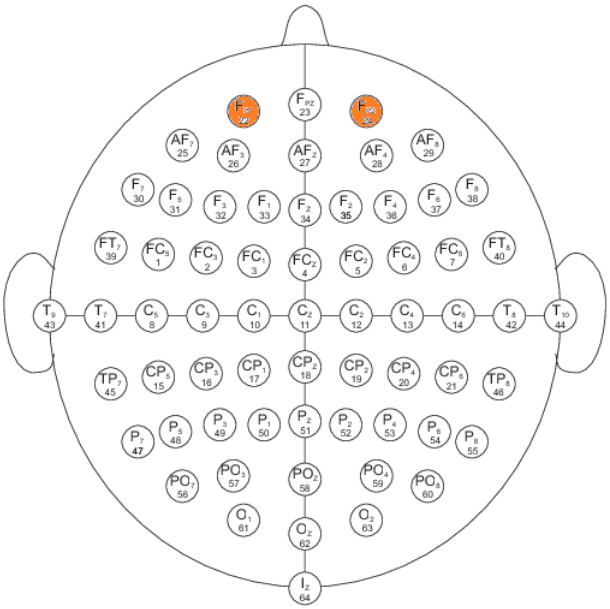
Para a avaliação da interface cérebro-computador, foi utilizada a técnica *holdout*, por se tratar de um método eficiente para a validação de sistemas baseados em aprendizado de máquinas, como é o caso desse projeto investigativo. Segundo Witten *et al.* (2011), essa abordagem visa reservar uma certa quantidade de dados para a fase de teste e usa o restante para a fase de treinamento do classificador. A técnica *holdout* verifica se cada classe da base de dados está proporcionalmente representada no conjunto de treinamento e teste. É importante ressaltar que se todas as instâncias de uma determinada classe ficarem de fora do conjunto de treinamento, dificilmente o classificador apresentará um bom desempenho para instâncias dessa classe na fase de teste. A utilização desse procedimento garante que cada classe seja corretamente representada no conjunto de dados de treinamento e teste por amostras aleatoriamente formadas. O método *holdout* fornece uma proteção contra representações desiguais em ambos os conjuntos (treinamento e teste). O *holdout* repete o processo (treinamento e teste), várias vezes com amostras aleatórias diferentes. Em cada iteração, 70% do conjunto de dados é selecionada aleatoriamente para o treinamento e o restante, ou seja, 30% é usado para teste.

Segundo Witten *et al.* (2011), executar o *holdout* uma única vez pode não ser suficiente para obter-se uma estimativa confiável do erro. Mas executar o *holdout* algumas vezes com o mesmo esquema de aprendizagem e com a mesma base de dados produz, frequentemente, resultados diferentes por conta do efeito da variação aleatória na escolha dos elementos que compõem a base de dados. Nesse trabalho, o processo de validação foi repetido cinco vezes, isto é, cinco vezes para cada um dos quatro classificadores, onde o melhor e o pior valor foram retirados. A taxa de acerto global é a média das taxas de acertos nas diferentes iterações.

Os sinais dos registros EEG utilizados para validar a interface cérebro-computador foram adquiridos através do banco de dados *eegmmidb - EEG Motor Movement/Imagery Dataset*, (https://archive.physionet.org/physiobank/database/eegmmidb/) capturado utilizando o sistema BCI2000 (SCHALK *et al.*, 2004), disponível através do PhysioBank (GOLDBERGER ARY L E AMARAL, 2000). Este banco de dados é composto por mais de 1500 registros de sinais EEG, obtidos a partir de 109 voluntários. Os registros de 105 voluntários foram utilizados nos testes. Dados de 4 voluntários (88, 92, 100 e 104) apresentaram problemas na etapa de aquisição; por esse motivo, tais registros não foram considerados. Neste trabalho de investigação, foram utilizados apenas os eletrodos posicionados na região do córtex pré-frontal (Fp1 e Fp2), ilustrados na Figura 4.1. A coleta dos registros foi realizada com a utilização de um filtro digital passa-faixa, cujas frequências

variam em uma faixa de 0,5 a 42 Hz e frequência de amostragem de 160 Hz.

Figura 4.1: Localização dos eletrodos (Fp1 e Fp2) utilizados na pesquisa.



Modificado de: Goldberger Ary L e Amaral (2000)

As quatro redes neurais artificiais são do tipo *Muilti Layer Perceptron* (MLP), com o algoritmo de retropropagação do erro (*backpropagation*), desenvolvidas utilizando o algoritmo *MultilayerPerceptron* proposto por Hall *et al.* (2009). JavaDoc: Anexo B. Cada rede é composta por 3 camadas (entrada, oculta e saída). O número de neurônios da camada de entrada da RNA é determinado pela dimensionalidade do vetor de características (seção 3.2), sendo 5 neurônios. A quantidade de neurônios na camada de saída da rede foi definida como 2 neurônios, visto que o objetivo da rede é discriminar entre duas respostas (0 ou 1), ou seja, se os dados que representa a imaginação do movimento está presente no sinal (1) ou não (0). A Tabela 4.1 apresenta as metodologias utilizadas para definição do número de neurônios da camada oculta da rede, com a equação, quantidade de neurônios e identificação dos métodos.

Tabela 4.1: Métodos utilizados para definição do número de neurônios da camada oculta.

Identificação	Metodologia	Equação	Número de neurônios
1	Hall (HALL *et al.*, 2009)	4.1	3
2	Han (HAN; KAMBER, 2006)	4.2	11
3	Fletcher (FLETCHER; GOSS, 1993)	4.3	6

Fonte: Autor.

O número de neurônios da camada oculta da rede foi analisado para selecionar o melhor método a ser utilizado pelo sistema. Equações: 4.1, 4.2 e 4.3.

$$N_{hidden} = \sqrt{N_{in} \cdot N_{out}} \tag{4.1}$$

$$N_{hidden} = 2N_{in} + 1 \tag{4.2}$$

$$N_{hidden} = 2\sqrt{N_{in}} + N_{out} \tag{4.3}$$

Onde N_{in} é o número de neurônios da camada de entrada, N_{out} é o número de neurônios da camada de saída da rede e N_{hidden} representa o número de neurônios da camada oculta.

O computador utilizado foi um DELL® modelo Latitude E6430 com processador Intel R CORE TM i7-3540M - 3.00GHz 2 núcleos e 4 *threads*, placa de vídeo NVidia R NVS TM 5200M - 1GB, 64bit - memória RAM com capacidade para 8GB e sistema operacional Linux Ubuntu 14.04.2 LTS.

4.1 Desempenho dos classificadores

Uma vez construída as redes neurais artificias, sua acurácia e capacidade de generalização foram testadas utilizando os dados processados de EEG que compõe o conjunto de amostras para testes. Dessa forma, foi possível analisar a viabilidade do uso das RNA's propostas. Seguindo a recomendação de configuração padrão definida por Hall *et al.* (2009), nesse experimento, a taxa de aprendizado[1] foi configurada em 0.3, o termo de *momentum*[2] foi configurado em 0.2 e o número de épocas[3] foi configurado em 1000. Para a validação dos classificadores, foram selecionadas 9538 instâncias, ou seja, 30% do total de dados originados dos eletrodos Fp1 e Fp2.

[1] A taxa de aprendizado é uma constante de proporcionalidade no intervalo 0 - 1. Uma taxa de aprendizado muito baixa torna o aprendizado da rede muito lento, ao passo que uma taxa de aprendizado muito alta provoca oscilações no treinamento e impede a convergência do processo de aprendizado.

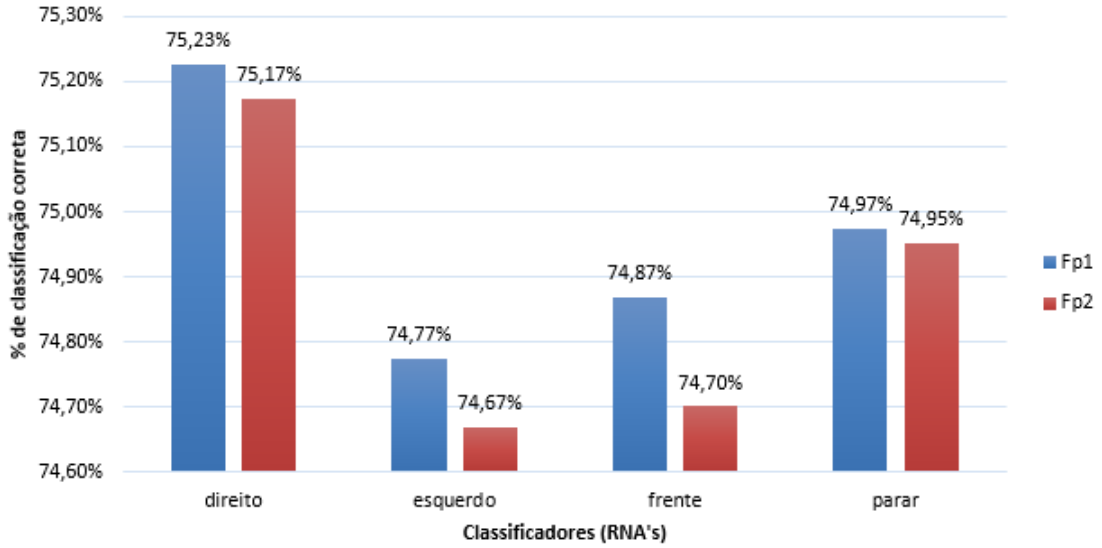
[2] O termo de *momentum* tem por objetivo aumentar a velocidade de treinamento da rede neural e seu valor varia de 0 - 1. É uma constante que determina o efeito das mudanças passadas dos pesos na direção atual do movimento no espaço de pesos.

[3] Define o número de ciclos de treinamento, ou seja, o número de vezes em que o conjunto de treinamento é apresentado à rede. Um número excessivo de ciclos pode levar a rede à perda da capacidade de generalização.

4.1.1 Três neurônios na camada oculta

A Figura 4.2 apresenta o percentual de acerto dos classificadores, utilizando três neurônios na camada oculta da rede, conforme metodologia 1 (Tabela 4.1).

Figura 4.2: Percentual de acerto RNA's com 3 neurônios na camada oculta.



Fonte: Autor.

Na Figura 4.2, observa-se que o classificador "direito" (Fp1) obteve o melhor percentual de acerto: 75,23%. Já o classificador "esquerdo" (Fp2) obteve o menor percentual de acerto: 74,67%. A Tabela 4.2 apresenta os resultados obtidos para os classificadores relacionados ao eletrodo Fp1. Em média, os classificadores relacionados aos dados originados do eletrodo "Fp1" obtiveram melhor desempenho, totalizando: 74,96% de acerto, contra 74,87% do eletrodo "Fp2".

Tabela 4.2: Resultado RNA's com 3 neurônios: Eletrodo Fp1.

Fp1	direito	esquerdo	punho	pe
Corretamente classificadas	7175	7132	7141	7151
Incorretamente classificadas	2363	2406	2397	2387

Fonte: Autor.

O classificador "direito" apresentou maior número de instâncias classificadas corretamente. Esta RNA é relacionada à imaginação de movimento do punho direito. Assim, das 9538 amostras utilizadas nos testes, 7175 foram classificadas corretamente, totalizando 75,23% de acerto. Já o classificador "esquerdo" apresentou maior número de instâncias classificadas incorretamente: 2406, ou seja, o classificador responsável pelo processamento dos registros relacionados à imaginação de movimento do punho esquerdo apresentou pior

desempenho, totalizando: 25,23% de erro.

A Tabela 4.3 apresenta os resultados obtidos para os classificadores relacionados ao eletrodo Fp2. O classificador "direito" apresentou maior número de instâncias classificadas corretamente: 7170, totalizando 75,17% de acerto. O classificador "esquerdo" apresentou maior número de instâncias classificadas incorretamente: 2416, totalizando 25,33% de erro.

Tabela 4.3: Resultado RNA's com 3 neurônios: Eletrodo Fp2.

Fp2	direito	esquerdo	punho	pe
Corretamente classificadas	7170	7122	7125	7149
Incorretamente classificadas	2368	2416	2413	2389

Fonte: Autor.

Observa-se, a partir dos dados apresentados na Figura 4.2, corroborados nas Tabelas 4.2 e 4.3, que a diferença média entre os resultados obtidos pelos classificadores relacionados aos eletrodos Fp1 e Fp2 foi de 0,09%, ou seja, os classificadores configurados com três neurônios na camada oculta, para as RNA's relacionadas ao eletrodo Fp1 apresentaram uma média de 74,96% de acerto. Já para as RNA's relacionadas ao eletrodo Fp2 esse percentual foi de 74,87%. Pode-se afirmar que as RNA's (direito, esquerdo, punho e pe) responsáveis pela classificação dos dados originados do eletrodo Fp1, apresentaram melhores resultados.

4.1.2 Onze neurônios na camada oculta

A Figura 4.3 apresenta o percentual de acerto dos classificadores, utilizando onze neurônios na camada oculta da rede, conforme metodologia 2 (Tabela 4.1). Observa-se que a média dos classificadores relacionados ao eletrodo "Fp2" obtiveram pior desempenho, totalizando: 74,81% de acerto, contra 74,91% para os classificadores relacionados ao eletrodo "Fp1". A diferença média entre os resultados obtidos pelos classificadores relacionados aos eletrodos Fp1 e Fp2 foi de 0,10%. O classificador "punho" (Fp2), obteve o pior percentual de acerto: 74,55%, dentre todos. Já o classificador "direito" (Fp1), obteve o melhor percentual de acerto: 75,20%. A Tabela 4.4 apresenta os resultados obtidos pelos classificadores relacionados ao eletrodo Fp1.

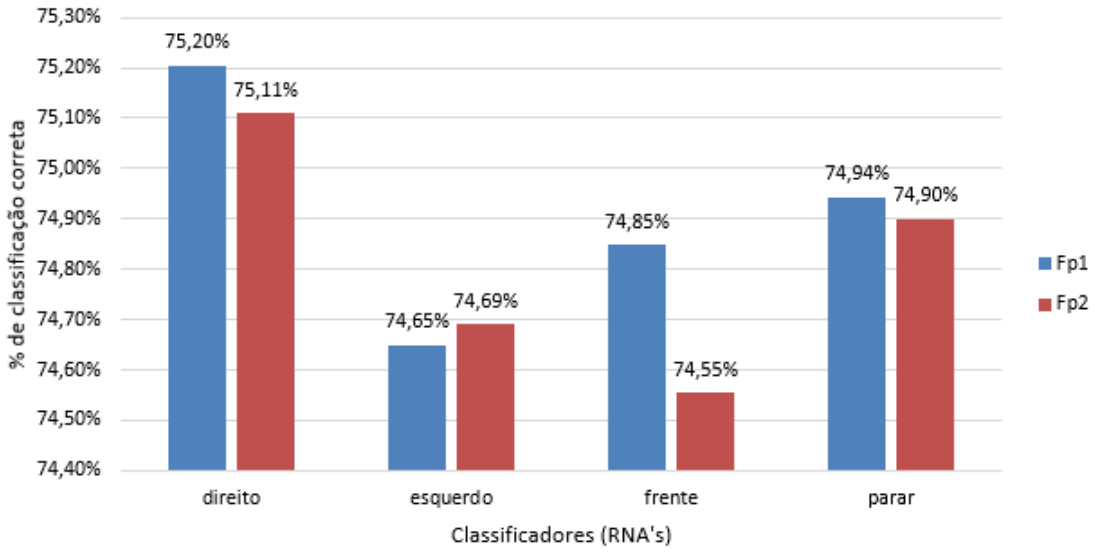
Tabela 4.4: Resultado RNA's com 11 neurônios: Eletrodo Fp1.

Fp1	direito	esquerdo	punho	pe
Corretamente classificadas	7173	7120	7139	7148
Incorretamente classificadas	2365	2418	2399	2390

Fonte: Autor.

Observa-se, a partir dos dados apresentados na Tabela 4.4, que o classificador "direito" apresentou maior número de instâncias classificadas corretamente, ou seja, das 9538 amostras utilizadas nos testes, 7173 foram classificadas corretamente, totalizando 75,20% de acerto. Já o classificador "esquerdo" apresentou maior número de instânncias classificadas incorretamente: 2418, totalizando 25,35% de erro.

Figura 4.3: Percentual de acerto RNA's com 11 neurônios na camada oculta.



Fonte: Autor.

A Tabela 4.5 apresenta os resultados obtidos pelos classificadores relacionados ao eletrodo Fp2. O classificador "direito" apresentou maior número de instâncias classificadas corretamente: 7164, totalizando 75,11% de acerto. O classificador "punho" apresentou maior número de instâncias classificadas incorretamente: 2427, totalizando 25,45% de erro.

Tabela 4.5: Resultado RNA's com 11 neurônios: Eletrodo Fp2.

Fp2	direito	esquerdo	punho	pe
Corretamente classificadas	7164	7124	7111	7144
Incorretamente classificadas	2374	2414	2427	2394

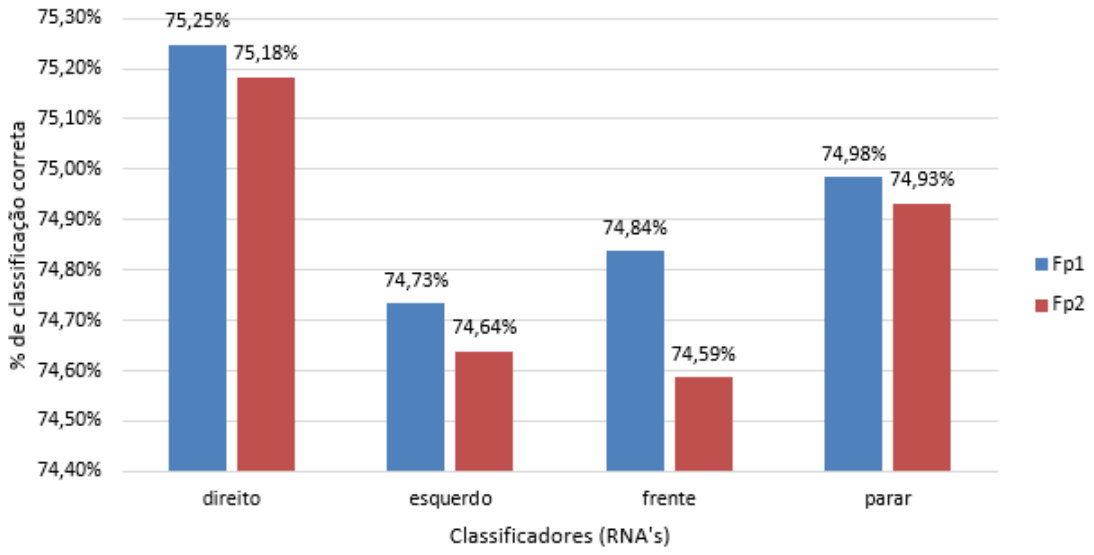
Fonte: Autor.

Os classificadores configurados com onze neurônios na camada oculta, para as RNA's relacionadas ao eletrodo Fp1, apresentaram uma média de 74,91% de acerto. Já para as RNA's relacionadas ao eletrodo Fp2, esse percentual foi de 74,81%. Pode-se afirmar que as RNA's (direito, esquerdo, punho e pe) responsáveis pela classificação dos dados originados do eletrodo Fp1 apresentaram melhores resultados.

4.1.3 *Seis neurônios na camada oculta*

A Figura 4.4 apresenta o percentual de acerto dos classificadores utilizando seis neurônios na camada oculta da rede, conforme metodologia 3 (Tabela 4.1).

Figura 4.4: Percentual de acerto RNA's com 6 neurônios na camada oculta.



Fonte: Autor.

Na Figura 4.4, observa-se que o classificador "direito" (Fp1) obteve o melhor percentual de acerto: 75,25%. Já o classificador "punho" (Fp2) obteve o menor percentual de acerto: 74,59%. A Tabela 4.6 apresenta os resultados obtidos para os classificadores relacionados ao eletrodo Fp1. Em média, os classificadores relacionados aos dados originados do eletrodo "Fp1" obtiveram melhor desempenho, totalizando: 74,95% de acerto, contra 74,83% do eletrodo "Fp2".

Tabela 4.6: Resultado RNA's com 6 neurônios: Eletrodo Fp1.

Fp1	direito	esquerdo	punho	pe
Corretamente classificadas	7177	7128	7138	7152
Incorretamente classificadas	2361	2410	2400	2386

Fonte: Autor.

O classificador "direito" apresentou maior número de instâncias classificadas corretamente: das 9538 amostras utilizadas nos testes, 7177 foram classificadas corretamente, totalizando 75,25% de acerto. Já o classificador "esquerdo" apresentou maior número de instâncias classificadas incorretamente: 2410, totalizando: 25,27% de erro.

A Tabela 4.7 apresenta os resultados obtidos para os classificadores relacionados ao ele-

trodo Fp2. O classificador "direito" apresentou maior número de instâncias classificadas corretamente: 7171, totalizando 75,18% de acerto. O classificador "punho" apresentou maior número de instâncias classificadas incorretamente: 2424, totalizando 25,36% de erro.

Tabela 4.7: Resultado RNA's com 6 neurônios: Eletrodo Fp2.

Fp2	direito	esquerdo	punho	pe
Corretamente classificadas	7171	7119	7114	7147
Incorretamente classificadas	2367	2419	2424	2391

Fonte: Autor.

A diferença média entre os resultados obtidos pelos classificadores relacionados aos eletrodos Fp1 e Fp2 foi de 0,12%, ou seja, os classificadores configurados com seis neurônios na camada oculta, para as RNA's relacionadas ao eletrodo Fp1, apresentaram uma média de 74,95% de acerto. Já para as RNA's relacionadas ao eletrodo Fp2, esse percentual foi de 74,83%. Pode-se afirmar que as RNA's (direito, esquerdo, punho e pe) responsáveis pela classificação dos dados originados do eletrodo Fp1 apresentaram melhores resultados.

4.1.4 Influência do número de neurônios na camada oculta da rede

Comparando os resultados a partir das subseções: 4.1.1, 4.1.2 e 4.1.3, observa-se pela média que os classificadores relacionados ao eletrodo "Fp1" apresentaram 74,94% de instâncias corretamente classificadas. Já os classificadores relacionados ao eletrodo "Fp2" apresentaram 74,84% de instâncias corretamente classificadas. A diferença foi de 0,10%. A Tabela 4.8 apresenta a média percentual de acertos dos classificadores por eletrodos (Fp1 e Fp2) para as três configurações (3, 11 e 6) do número de neurônios definidos na camada oculta da rede.

Tabela 4.8: Média percentual de acertos dos classificadores por eletrodos (Fp1 e Fp2).

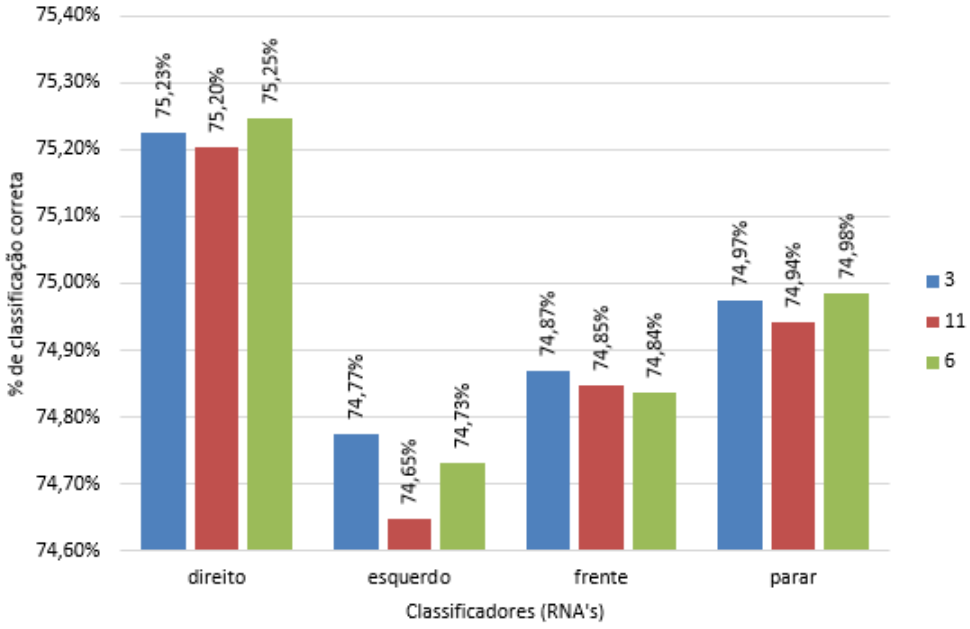
Número de neurônios	**Fp1**	**Fp2**
3	74,96%	74,87%
11	74,91%	74,81%
6	74,95%	74,84%
Média	74,94%	74,84%

Fonte: Autor.

De acordo com os resultados das avaliações (Tabela 4.8), os classificadores relacionados ao eletrodo "Fp1" apresentaram melhores resultados no processo de reconhecimento de padrões a partir do processamento de sinais EEG imaginados do punho direito, esquerdo,

ambos os punhos e ambos os pés, respectivamente relacionados aos classificadores direito, esquerdo, punho e pe. Por esse motivo, o eletrodo recomendado por este trabalho de investigação para tarefa de construção do modelo de classificação foi o "Fp1". A Figura 4.5, apresenta o desempenho das RNA's por número de neurônios na camada oculta para o eletrodo "Fp1".

Figura 4.5: Desempenho das RNA's por número de neurônios na camada oculta.



Fonte: Autor.

Observa-se, a partir da Figura 4.5, que os classificadores configurados com 3 neurônios na camada oculta da rede apresentaram melhores resultados para as RNA's esquerdo (74,77%) e punho (74,87%). Os classificadores configurados com 6 neurônios na camada oculta da rede apresentaram melhores desempenho para as RNA's direito (75,25%) e pe (74,98%). Já os classificadores configurados com 11 neurônios na camada oculta da rede apresentaram piores resultados para todas RNA's: direito (75,20%), esquerdo (74,65%), punho (74,85%) e pe (74,94%), em relação as RNA's configuradas com 3 e 6 na camada oculta da rede. De acordo com essas observações, a recomendação para configuração das RNA's é apresentada na Tabela 4.9.

Tabela 4.9: Configuração das RNA's.

Classificador	**Neurônios camada oculta**
direito	6
esquerdo	3
punho	3
pé	6

Fonte: Autor.

Observa-se que tivemos duas RNA's (direito e pe) configuradas com 6 neurônios na camada oculta e duas RNA's (esquerdo e punho) configuradas com 3 neurônios na camada oculta. Dessa forma, fizemos uma análise das médias percentuais de instâncias corretamente classificadas pelo número de neurônios na camada oculta (Tabela 4.10).

Tabela 4.10: Média percentual de instâncias corretamente classificadas.

Neurônios camada oculta	**Média**
3	74,96%
11	74,91%
6	74,95%

Fonte: Autor.

A partir dos dados apresentados na Tabela 4.10, o melhor resultado foi obtido com a configuração das RNA's com o 3 neurônios na camada oculta da rede. O percentual de instâncias corretamente classificadas foi de 74,96%, porém, o percentual médio apresentado pelas RNA's configuradas com 6 neurônios foi de 74,95%, ou seja, a diferença foi de 0,01%.

Após a conclusão de todos os testes realizados na avaliação e análise dos resultados obtidos, a configuração da topologia das RNA's proposta neste trabalho de investigação, encontra-se na Tabela 4.11.

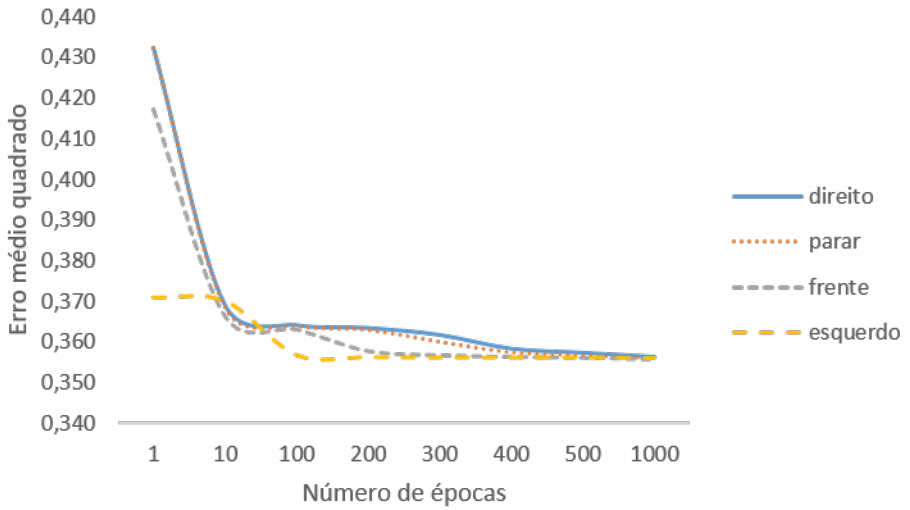
Tabela 4.11: Topologia proposta para as RNA's.

Número de neurônios na camada		
Entrada	Oculta	Saída
5	3	2

Fonte: Autor.

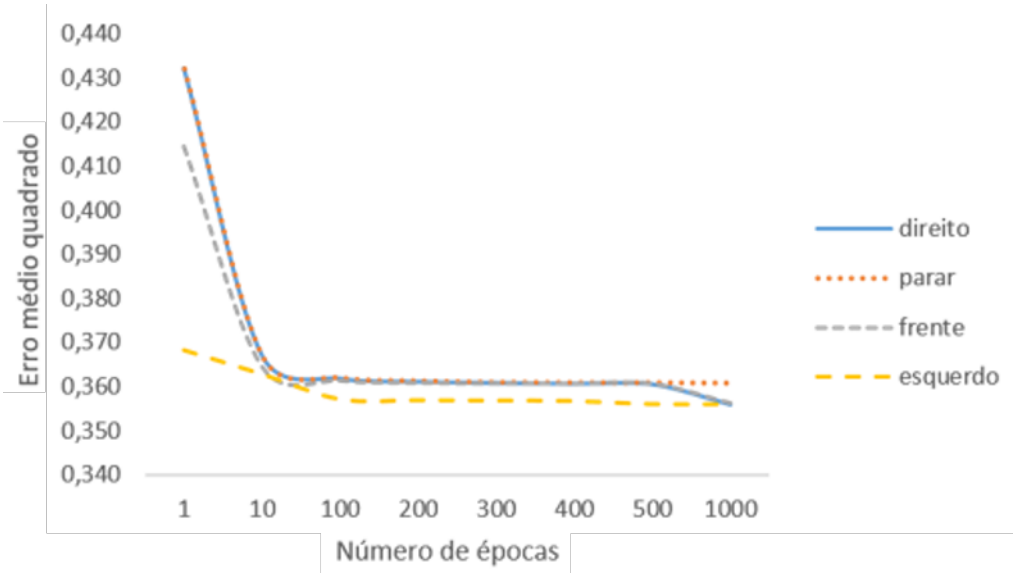
As curvas de aprendizagem médias referentes aos dados originados dos eletrodos Fp1 e Fp2, levando em consideração a topologia das RNA's apresentadas na Tabela 4.11, taxa de aprendizado configurada em 0.3 e o termo de *momentum* em 0.2, são ilustradas nas Figuras 4.6 e 4.7. Observa-se, que o classificador esquerdo (Fp1 e Fp2) resulta em uma convergência mais lenta, isto devido a oscilações no erro médio quadrado durante o processo de aprendizagem.

Figura 4.6: Curvas de aprendizagem dados do eletrodo Fp1.



Fonte: Autor.

Figura 4.7: Curvas de aprendizagem dados do eletrodo Fp2.



Fonte: Autor.

O modelo baseado na topologia proposta neste trabalho de investigação, demonstrou boa capacidade de classificação em comparação as outras configurações estudadas, com média de acertos de 74,96%, sendo que sua menor pontuação foi de 74,77% (classificador esquerdo) e sua maior de 75,23% (classificador direito). Esse valores demonstram que, embora com a utilização de apenas dois eletrodos, foi possível classificar as imaginações de movimentos (punho direito, punho esquerdo, ambos os punhos e ambos os pés) com mais de 70% de acertos.

Observando as avaliações de todos os testes realizados, algumas conclusões podem ser construídas. As RNA's treinadas classificaram corretamente os sinais EEG de indivíduos nos eletrodos: Fp1 e Fp2 na região do córtex pre-frontal, porém, o eletrodo Fp1 apresentou maior percentual de instâncias classificadas corretamente, com um índice de 74,94%.

Os resultados apresentados pelas diferentes configurações em relação ao número de neurônios da camada oculta obtiveram índices satisfatórios, sendo que o menor percentual de instâncias classificadas corretamente foi de 74,91%, na RNA configurada com 11 neurônios na camada oculta. A RNA configurada com 3 neurônios na camada oculta apresentou maior percentual de instâncias classificadas corretamente: 74,96%. Dessa forma, fazendo uma análise mais generalizada, o percentual de instâncias classificadas corretamente variaram pouco nos experimentos realizados. Assim, observa-se que a definição do número de neurônios presentes na camada oculta, pouco influenciaram nos resultados das classificações, porém, o tempo para treinamento da RNA (construção do modelo) aumenta à medida que o número de neurônios da camada oculta também aumenta.

A porcentagem de instâncias classificadas corretamente apresentada pelas RNA's especialistas, configuradas com 3 neurônios na camada oculta, foi: 75,23% para imaginação de movimento do punho direito (classificador Direito), 74,77% para imaginação de movimento do punho esquerdo (classificador Esquerdo), 74,87% para imaginação de movimento de ambos os punhos (classificador Punho) e 74,97% para imaginação de movimento de ambos os pés (classificador Pe). Para esses classificadores, a precisão média foi de 74,96% de acerto. Resultados estes, bem próximos dos obtidos pelos trabalhos apresentados na Tabela 4.12, por exemplo, na pesquisa desenvolvida por Pinheiro *et al.* (2018), onde foram utilizados 2 eletrodos localizados na região do córtex pré-frontal para aquisição de sinais EEG, baseada na imaginação de movimentos para controlar uma cadeira de rodas, a precisão média do modelo foi de 50,40 % . Já no trabalho de Faria *et al.* (2012), os resultados demonstraram uma precisão de 57% de acerto na classificação de expressões faciais. O modelo classificou cinco intenções de movimentos, 14 eletrodos foram utilizados para aquisição de sinais EEG, localizados na região do córtex frontal e motor.

Tabela 4.12: Trabalhos selecionados. Fonte Autor.

Publicação/Ano	Método	Precisão média (%)
Pinheiro *et al.* (2018)	RNA	50%
Bhattacharyya *et al.* (2016)	SVM	75%
Eid *et al.* (2016)	Rastreamento dos olhos	70%
Edelman *et al.* (2016)	Bayes	82%
José & Lopes (2015)	Movimento do lábio	62%
Lay & Pizarro (2015)	BCI2000	81%
El-Madani *et al.* (2015)	Bayes	80%
Kaufmann *et al.* (2014)	Árvore de decisão	87%
Witkowski *et al.* (2014)	BCI2000	63%
Wang *et al.* (2014)	SVM	93%
Carlson & Millan (2013)	Gaussiano	94%
Faria *et al.* (2012)	RNA; SVM; Bayes	57%

Nos trabalhos de Bhattacharyya *et al.* (2016), El-Madani *et al.* (2015), Witkowski *et al.* (2014), Wang *et al.* (2014) e Carlson & Millan (2013), a interface de controle é baseada na imaginação de movimento, realizando a classificação de dados que incluem respostas a duas tarefas mentais distintas, limitando o grau de liberdade das pessoas em duas direções. A interface, desenvolvida por Eid *et al.* (2016), é baseada no rastreamento dos olhos, os pesquisadores preocuparam-se em auxiliar pessoas com graves deficiências motoras.

José & Lopes (2015), desenvolveram uma interface especialmente para pessoas com tetraplegia, o dispositivo de entrada de dados é baseado em um *joystick* controlado pelo lábio inferior. Nesse caso, é necessário considerar que uma interface controlada pelo lábio inferior deve evitar o contato com áreas úmidas da boca por questões de higiene. Este é um fator muito importante, principalmente no caso das pessoas com tetraplegia que precisam da ajuda de outras pessoas para a limpeza da interface. No trabalho de Kaufmann *et al.* (2014), a interface cérebro-computador é baseada em potenciais relacionados a eventos evocados-tátil, dentro dessa área de investigação existe alguns pontos a serem tratados, como: a otimização do tempo de treinamento do usuário para operar o sistema.

As interfaces cérebro-computador desenvolvidas nos trabalhos: Edelman *et al.* (2016), José & Lopes (2015) e Faria *et al.* (2012), utilizam pelo menos 16 eletrodos para aquisição dos sinais EEG. Quanto maior o número de eletrodos para aquisição e envio dos dados, maior será o consumo de energia, consequentemente isso reduz o tempo de utilização do dispositivo em atividades prolongadas.

Observa-se que os projetos, de modo geral, possuem características semelhantes, como a adaptação de diferentes tipos de interfaces de controle para facilitar a interação de pessoas

com diferentes tipos de deficiência, porém, são necessários mais estudos para tratar de casos extremos, como por exemplo: pessoas com quadriplegia, onde existe paralisia completa dos músculos voluntários em todas as partes do corpo, exceto aqueles que controlam o movimento dos olhos, contribuindo negativamente à qualidade de vida dessas pessoas.

Outro ponto importante, ainda em relação a diminuição do número de eletrodos envolvidos no processo de construção das interfaces cérebro-computador, são os aspectos relacionados a representação social e pessoal do indivíduo com deficiência[4]. A necessidade dos eletrodos para aquisição dos sinais EEG incrementa a percepção simbólico da condição de deficiência que o indivíduo enfrenta.

Neste trabalho, a interface cérebro-computador será baseada na imaginação de movimentos do punho esquerdo, punho direito, ambos os punhos e ambos os pés, permitindo a classificação de dados que incluem respostas a quatro tarefas mentais distintas, ampliando o grau de liberdade do indivíduo em quatro direções. Assim, quatro RNA's especialistas (uma para cada imaginação de movimento) foram implementadas, o objetivo da especialização é minimizar a presença de falsos positivos. Com a especialização das redes neurais, a generalização das mesmas será beneficiada, uma vez que é mais fácil discernir entre duas classes de padrões a quatro classes. A justificativa para o uso das redes neurais artificiais como método de reconhecimento de padrões, nesse trabalho de investigação, baseou-se em duas observações, a saber:

- De acordo com Wolpaw & Wolpaw (2012), o uso das redes neurais artificias no reconhecimento de padrões, sugere a possibilidade de identificar nos sinais EEG, características que expressem o comando de movimentos ou a intenção de movimento;
- Dos trabalhos apresentados (Tabela 4.12), o único a realizar testes em pessoas com graves deficiências motoras foi Faria *et al.* (2012), nesse trabalho foi feita a comparação entre diferentes métodos de classificação (RNA; SVM; Bayes). Os resultados obtidos corroboraram que o classificador baseado em RNA foi mais eficiente em relação aos outros métodos: RNA: 57%; SVM: 43,55%; Bayes: 54,64%.

Dessa forma, traremos outra contribuição relacionada ao problema de controle baseado em interfaces cérebro-computador, que é a obtenção de uma alta precisão de classificação, por exemplo: no controle de uma Cadeira de Rodas Inteligente, um erro de classificação (um comando errado) pode causar situações perigosas, por esse motivo é importante garantir uma taxa de erro mínima para manter o indivíduo seguro.

[4] A representação social e pessoal da deficiência é uma espécie de lente pela qual o indivíduo é visto e enxerga seu mundo. Essas percepções podem ser repletas de preconceitos, de ambas as partes, pois é um ser social, participando da construção e manutenção das concepções, status e valores da sociedade (FECHIO *et al.*, 2009).

Capítulo Cinco

Conclusões e Considerações Finais

Este trabalho de investigação apresenta o desenvolvimento de uma interface cérebro-computador não invasiva, com o objetivo de minimizar a dependência social que as pessoas com graves comprometimentos motores têm na sociedade, e na ampliação da mobilidade dessas pessoas. A interface é baseada na imaginação de movimentos que é considerada como um estado cognitivo que corresponde à simulação mental de uma ação motora sem que haja qualquer manifestação motora real. A metodologia proposta é composta de uma etapa de pré-processamento, extração de características e classificação. Na etapa de pré-processamento, os sinais EEG foram filtrados por um mecanismo de minimização de artefatos. O processo de classificação é baseado em aprendizado de máquina, onde o sistema de reconhecimento de padrões (RNA) foi treinado de forma supervisionada para realizar a correta identificação das intenções de movimentos a partir das tarefas mentais.

Os resultados deste trabalho mostram que há uma possibilidade a ser aprofundada, que é o controle de máquinas ou dispositivos através do pensamento, isso certamente afetará muitos que tenham restrições de movimento, dentre outros.

5.1 Conclusões

A interface cérebro-computador proposta nesta pesquisa, através dos experimentos realizados, como detalhado no Capítulo 4, mostrou ser possível identificar nos sinais EEG, a partir dos eletrodos localizados na região do córtex pré-frontal, características que expressem a intenção de quatro movimentos que poderão ser utilizados para acionar um dispositivo robótico.

A capacidade do sistema proposto em discernir entre quatro intenções de movimentos pensados, foi avaliada nos experimentos. As melhores taxas de instâncias classificadas corretamente, foram alcançadas através dos dados do eletrodo Fp1 e pela configuração da RNA com: cinco neurônios na camada de entrada, três neurônios na camada oculta e dois neurônios na camada de saída. Variação entre 74,77% e 75,23%.

Dessa forma, a interface cérebro-computador baseada nos resultados, mostrou-se útil para reconhecer padrões a partir de sinais pensados de EEG. As técnicas desenvolvidas nesta pesquisa são promissoras, o sistema destina-se a contribuir como complemento na ampliação da mobilidade de pessoas com graves comprometimentos motores. Uma fase ex-

perimental mais longa, com pessoas deficientes faz-se necessária, com o objetivo de uma avaliação mais completa.

5.2 Contribuições

Nesse trabalho destacam-se as seguintes contribuições:

- O desenvolvimento de uma interface cérebro-computador, que contribui para ampliação da mobilidade de pessoas com comprometimentos motores e neurológicos severos. Nesse sentido, torna-se possível a ressocialização destas pessoas.
- Elaboração de funções para leitura dos sinais EEG que estão no formato EDF no banco de dados *eegmmidb - EEG Motor Movement/Imaginery Dataset* e conversão para o formato Matlab fazendo uso do processamento paralelo quando se usa processador com mais de um núcleo.
- Desenvolvimento do módulo para tratamento de limpeza de artefatos, que contribuiu para melhoria do índice de acertos, na classificação das intenções de movimentos.
- Desenvolvimento de função que mede a potência dos sinais nos 64 eletrodos cerebrais no padrão 10/20 de forma interativa possibilitando descobrir os eletrodos de maior relevância para o estudo.

5.3 Atividades Futuras de Pesquisa

Nesse trabalho, foi utilizado o algoritmo JADE para limpeza de artefatos, e uma das atividades futuras será testar outros algoritmos ICA como o InfoMax, o FastICA e o SOBI, para melhorar a classificação dos sinais de EEG.

Outra possibilidade a ser testada é fazer um comparativo de precisão entre a RNA atual e uma Máquina de Vetores de Suporte (SVM).

Outra proposta de atividade futura, seria o emprego de outras técnicas para decomposição do sinal EEG no domínio do tempo, por exemplo, a transformada *wavelet* e comparar com a técnica já utilizada.

Referências Bibliográficas

AMARI, Shun-ichi; CICHOCKI, Andrzej; YANG, Howard Hua. A new learning algorithm for blind signal separation. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1996. p. 757–763. 2.6.9.4

AMORIM, Raphael M de *et al.* Teste de robustez da ica no pré-processamento de sinais em sistemas de sonar passivo. Anais do XX Congresso Brasileiro de Automática, 2014. 2.6.9.4

AMTHOR, Frank. *Neuroscience for dummies*. [S.l.]: John Wiley & Sons, 2016. 2.2

_____. *Neurociência para leigos*. [S.l.]: Alta Books Editora, 2017. 2.1, 2.1

BANSAL, Dipali; MAHAJAN, Rashima. *EEG-Based Brain-Computer Interfaces: Cognitive Analysis and Control Applications*. [S.l.]: Academic Press, 2019. 2.1

BELL, Anthony J; SEJNOWSKI, Terrence J. An information-maximization approach to blind separation and blind deconvolution. *computação neural*, MIT Press, v. 7, n. 6, p. 1129–1159, 1995. 2.6.9.1, 2.6.9.4

BELOUCHRANI, Adel *et al.* A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, IEEE, v. 45, n. 2, p. 434–444, 1997. 2.6.9.3

BHATTACHARYYA, S.; SHIMODA, S.; HAYASHIBE, M. A synergetic brain-machine interfacing paradigm for multi-dof robot control. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 46, n. 7, p. 957–968, July 2016. ISSN 2168-2216. 4.12, 4.1.4

BIRBAUMER, N. *et al.* The thought translation device (ttd) for completely paralyzed patients. *IEEE Transactions on Rehabilitation Engineering*, v. 8, n. 2, p. 190–193, Jun 2000. ISSN 1063-6528. 2.3

CARDOSO, Jean-François; SOULOUMIAC, Antoine. Blind beamforming for non-gaussian signals. In: IET. *IEE proceedings F (radar and signal processing)*. [S.l.], 1993. v. 140, n. 6, p. 362–370. 2.6.9.4, 2.6.9.4

CARDOSO, Jean-Fran c c ois. High-order contrasts for independent component analysis. *computação neural*, v. 11, p. 157–192, 1999. 2.6.9.4, 2.6.9.4, 2.6.9.4, 3.1.4, 3.1.4

CARLSON, T.; MILLAN, J. del R. Brain-controlled wheelchairs: A robotic architecture. *IEEE Robotics Automation Magazine*, v. 20, n. 1, p. 65–73, March 2013. ISSN 1070-9932. 4.12, 4.1.4

CARMENA, Jose M. *et al.* Stable ensemble performance with single-neuron variability during reaching movements in primates. *The Journal of Neuroscience*, v. 25, n. 46, p. 10712–10716, 2005. Disponível em: <http://www.jneurosci.org/content/25/46/10712.abstract>. 2.3, 2.8

CHINI, Gislaine Cristina de Oliveira; BOEMER, Magali Roseira. A amputação na percepção de quem a vivencia: um estudo sob a ótica fenomenológica. *Revista Latino-Americana de Enfermagem*, SciELO Brasil, v. 15, n. 2, p. 330–336, 2007. 3

CICHOCKI, Andrzej; AMARI, Shun-ichi. *Adaptive blind signal and image processing: learning algorithms and applications*. [S.l.]: John Wiley & Sons, 2002. 2.6.9.4

COMON, Pierre. Independent component analysis, a new concept? Elsevier, v. 36, n. 3, p. 287–314, 1994. 2.6.1

DELORME, Arnaud; MAKEIG, Scott. Eeglab wikitorial. *Retrieved from10*, v. 1016, 2009. 2.6.9, 2.6.9.1

EDELMAN, B. J.; BAXTER, B.; HE, B. Eeg source imaging enhances the decoding of complex right-hand motor imagery tasks. *IEEE Transactions on Biomedical Engineering*, v. 63, n. 1, p. 4–14, Jan 2016. ISSN 0018-9294. 4.12, 4.1.4

EID, M. A.; GIAKOUMIDIS, N.; SADDIK, A. El. A novel eye-gaze-controlled wheelchair system for navigating unknown environments: Case study with a person with als. *IEEE Access*, v. 4, p. 558–573, 2016. ISSN 2169-3536. 4.12, 4.1.4

EL-MADANI, Ahmad *et al.* Real-time brain computer interface using imaginary movements. *EPJ Nonlinear Biomedical Physics*, v. 9, n. 3, 2015. 4.12, 4.1.4

FARIA, B. M.; REIS, L. P.; LAU, N. Cerebral palsy eeg signals classification: Facial expressions and thoughts for driving an intelligent wheelchair. In: *2012 IEEE 12th International Conference on Data Mining Workshops*. [S.l.: s.n.], 2012. p. 33–40. ISSN 2375-9232. 4.1.4, 4.12, 4.1.4

FECHIO, MB *et al.* A repercussão da lesão medular na identidade do sujeito. *Acta Fisiátrica*, v. 16, n. 1, p. 38–42, 2009. 4

FLETCHER, D.; GOSS, E. Forecasting with neural networks: An application using bankruptcy data. *Information & Management*, v. 24, n. 3, p. 159–167, Mar 1993. 4.1

GOLDBERGER ARY L E AMARAL, Luis AN e Glass Leon e Hausdorff Jeffrey M e Ivanov Plamen Ch e Mark Roger G e Mietus Joseph E e Moody George B e Peng Chung-Kang e Stanley H Eugene. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. Am Heart Assoc, v. 101, p. e215 – e220, 2000. 3.1.1, 3.3, 4, 4.1

GOMES, Osmar; PINHEIRO, Oberdan; BANDEIRA, Alex. Modelo computacional baseado em interface cérebro-computador para classificação de sinais eeg. *E-TECH:Tecnologias para Competitividade Industrial*, Senai - SC, v. 12, p. 23–41, 2019. ISSN 1983–1838. 1.1

HALL, Mark *et al.* The weka data mining software: An update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. ISSN 1931-0145. Disponível em: <http://doi.acm.org/10.1145/1656274.1656278>. 3.2.2, 3.2.2, 4, 4.1, 4.1

HAN, J.; KAMBER, M. *Data Mining: Concepts and Techniques*. USA: Morgan Kaufmann, 2006. 3.2.2, 3.2.2, 4.1

HERCULANO-HOUZEL, Suzana. The remarkable, yet not extraordinary, human brain as a scaled-up primate brain and its associated cost. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 109, n. Supplement 1, p. 10661–10668, 2012. 2.1

HYVARINEN, Aapo. Fast and robust fixed-point algorithms for independent component analysis. *transações IEEE em redes neurais*, v. 10, p. 626–634, 1999. 2.6.9.2

HYVÄRINEN, Aapo; OJA, Erkki. A fast fixed-point algorithm for independent component analysis. *Neural computation*, MIT Press, v. 9, n. 7, p. 1483–1492, 1997. 2.6.9.4

HYVARINEN AAPO E KARHUNEN, J e Oja E. *Independent component analysis and blind source separation*. 2001. 2.20, 2.6.6, 2.21, 2.22

HYVÄRINEN AAPO E KARHUNEN, Juha e Oja Erkki. Independent component analisys,. John Wiley Sons, Inc, v. 1, 2001. 2.6.1

HYVÄRINEN AAPO E OJA, Erkki. Análise de componentes independentes: um tutorial. *Citeseer*, 1999. 2.6, 2.6, 3.1.4

IBGE. *Pesquisa nacional de saúde: 2013: percepção do estado de saúde, estilos de vida e doenças crônicas : Brasil, grandes regiões e unidades da federação*. Rio de Janeiro: [s.n.], 2014. ISBN 9788524043345. Disponível em: <URL:http://loja.ibge.gov.br/pesquisa-nacional-de-saude-2013-percepc-o-do-estado-de-saude-estilos-de-vida-e-doencas-cronicas.html>. 1

JACKS, A. S.; MILLER, N. R. Spontaneous retinal venous pulsation: aetiology and significance. *Journal of Neurology, Neurosurgery & Psychiatry*, v. 74, n. 1, p. 9, 2003. Disponível em: <http://jnnp.bmj.com/content/74/1/9.short>. 2.3

JOHAN, Wessberg *et al.* Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, v. 408, p. 361–365, November 2000. 2.3, 2.7

JOSé, M. A.; LOPES, R. de Deus. Human - computer interface controlled by the lip. *IEEE Journal of Biomedical and Health Informatics*, v. 19, n. 1, p. 302–308, Jan 2015. ISSN 2168-2194. 4.12, 4.1.4

KANDEL, Eric *et al. Princípios de neurociências-5*. [S.l.]: AMGH Editora, 2014. 2.2, 2.3, 2.2, 2.4

KAUFMANN, Tobias; HERWEG, Andreas; KÜBLER, Andrea. Toward brain-computer interface based wheelchair control utilizing tactually-evoked event-related potentials. *Journal of NeuroEngineering and Rehabilitation*, v. 11, n. 1, p. 1–17, 2014. ISSN 1743-0003. Disponível em: <http://dx.doi.org/10.1186/1743-0003-11-7>. 4.12, 4.1.4

KEMP, Bob; OLIVAN, Jesus. European data format 'plus' (edf+), an edf alike standard format for the exchange of physiological data. *Clinical Neurophysiology*, v. 114, n. 9, p. 1755 – 1761, 2003. ISSN 1388-2457. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1388245703001238>. 3.1.1

LAY, Sebastián Monge; PIZARRO, Diego Aracena. Robotic motion control with cognitive and facial detection via emotiv eeg. *Ingeniare. Revista chilena de ingeniería*, scielo, v. 23, p. 496 – 504, 10 2015. ISSN 0718-3305. 4.12

MARK, F. Bear; BARRY, W. Connors; MICHAEL, A. Paradiso. *Neuroscience: Exploring the Brain*. 3rd. ed. [S.l.]: Lippincott Williams & Wilkins, 2007. ISBN 978-0781760034. 2.4, 2.10, 2.11, 2.4, 2.12, 2.13, 2.4, 2.14

_____. *Neurociências: Desvendando o Sistema Nervoso*. Porto Alegre: Artmed, 2008. ISBN 9788536313337. 2.3

MILLAN, J. del R. *et al.* A local neural classifier for the recognition of eeg patterns associated to mental tasks. *IEEE Transactions on Neural Networks*, v. 13, n. 3, p. 678–686, May 2002. ISSN 1045-9227. 2.3

MILLAN, J. R.; MOURINO, J. Asynchronous bci and local neural classifiers: an overview of the adaptive brain interface project. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 11, n. 2, p. 159–161, June 2003. ISSN 1534-4320. 2.3

NICIDA, Denise Pirillo. *Neurociência: compreendendo o funcionamento do sistema nervoso*. 2015. Neurociências da Aprendizagem. Disponível em: <http://www.neuroeducacao.com.br/neurociencias.asp>. Acesso em: 17 JUN 2017. 2.1

OJA, Erkki. *Subspace methods of pattern recognition*. [S.l.]: Research Studies Press, 1983. v. 6. 2.6.6

PAPOULIS, Athanasios. Probability, random variables, and stochastic processes. McGraw-Hill, v. 19842, p. 345–348, 1991. 2.6.1

PFURTSCHELLER, G.; NEUPER, C. Motor imagery and direct brain-computer communication. *Proceedings of the IEEE*, v. 89, n. 7, p. 1123–1134, Jul 2001. ISSN 0018-9219. 2.3

PINHEIRO, Oberdan Rocha. *SmartChair: Cadeira de Rodas Inteligente com Interface Flexivel*. Tese (Doutorado) — Centro Universitário Senai - CIMATEC, Av. Orlando Gomes, 1845 Piatã, Salvador - BA 41650-010, 2016. Disponível em: <http://www.senaicimatec.com.br/en/teses_pos/pinheiro-oberdan-rocha/>. Acesso em: 3 jul. 2019. 1.2, 3.1, 3.13, 3.14, 3.15, 3.16

Pinheiro, O. R.; Alves, L. R. G.; Souza, J. R. D. Eeg signals classification: Motor imagery for driving an intelligent wheelchair. *IEEE Latin America Transactions*, v. 16, n. 1, p. 254–259, 2018. 1, 4.1.4, 4.12

PINTO, Luiz Carlos. *Neurofisiologia Clínica - Princípios Básicos e Aplicações*. [S.l.]: Atheneu, 2006. ISBN 9788538801436. 2.4

RANGEL, Edja Solange Souza; BELASCO, Angêlica Gonçalves Silva; DICCINI, Solange. Qualidade de vida de pacientes com acidente vascular cerebral em reabilitação. *Acta Paulista de Enfermagem*, scielo, v. 26, p. 205 – 212, 2013. ISSN 0103-2100. 1

RUTLEDGE DOUGLAS N E BOUVERESSE, D Jouan-Rimbaud. Independent components analysis with the jade algorithm. *Trends in Analytical chemistry*, Elsevier, v. 50, p. 22–32, 2013. 2.6.9.4

SAHONERO-ALVAREZ, Guillermo; CALDERON, Humberto. Uma comparação dos algoritmos sobi, fastica, jade e infomax. In: *Anais da 8ª Conferência Internacional de Complexidade, Informática e Cyberneti, Orlando, FL, EUA*. [S.l.: s.n.], 2017. p. 21–24. 2.6.9, 2.23, 2.24, 2.25

SCHALK, G. *et al.* Bci2000: a general-purpose brain-computer interface (bci) system. *IEEE Transactions on Biomedical Engineering*, v. 51, n. 6, p. 1034–1043, June 2004. ISSN 0018-9294. 4

SCHALK GERWIN E MCFARLAND, Dennis J e Hinterberger Thilo e Birbaumer Niels e Wolpaw Jonathan R. Bci2000: um sistema de interface cérebro-computador de uso geral (bci). *IEEE Transactions on engenharia biomédica*, v. 51, p. 1034–1043, 2004. 3.1.1

SCHMIDT, Edward M. Single neuron recording from motor cortex as a possible source of signals for control of external devices. *Biomedical Engineering*, v. 8, n. 4, p. 339–349, July 1980. 2.3

SILVA, Fernando Lopes; NIEDERMEYER, Ernst. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. New Jersey: [s.n.], 1982. 2.3

VIDAURRE, C. *et al.* A fully on-line adaptive bci. *IEEE Transactions on Biomedical Engineering*, v. 53, n. 6, p. 1214–1219, June 2006. ISSN 0018-9294. 2.3

WANG, H. *et al.* An asynchronous wheelchair control by hybrid eeg–eog brain–computer interface. *Cognitive Neurodynamics*, v. 8, n. 5, p. 399–409, 2014. Disponível em: <http://doi.org/10.1007/s11571-014-9296-y>. 4.12, 4.1.4

WEBSTER, John G. *Medical Instrumentation Application and Design*. 4th edition. ed. Nova York: John Wiley & Sons, 2009. ISBN 978-0471676003. 2.3, 2.3, 2.9, 2.3

WEN, Dr. *Monitorização contínua do paciente por VÍDEO-EEG*. 2006. Eplepsia. Disponível em: <http://www.epilepsia-cirurgia.com.br/etapas_investigacao.htm>. Acesso em: 12 novembro 2015. 2.5, 2.3

WHO. *WHO global disability action plan 2014-2021. Better health for all people with disability*. Switzerland: [s.n.], 2015. ISBN 978 92 4 150961 9. Disponível em: <URL:http://apps.who.int/iris/bitstream/10665/199544/1/9789241509619_eng.pdf?ua=1>. 1

WITKOWSKI, Matthias *et al.* Enhancing brain-machine interface (bmi) control of a hand exoskeleton using electrooculography (eog). *Journal of NeuroEngineering and Rehabilitation*, v. 11, n. 1, p. 1–6, 2014. ISSN 1743-0003. Disponível em: <http://dx.doi.org/10.1186/1743-0003-11-165>. 4.12, 4.1.4

WITTEN, Ian H.; FRANK, Eibe; HALL, Mark A. *Data mining: practical machine learning tools and techniques*. USA: Morgan Kaufmann, 2011. 4

WLADIMIR. *Cientistas controlam o corpo de uma pessoa com cérebro de outra*. 2013. Nerdices. Disponível em: <http://nerdices.com.br/42/2013/08/28/cientistas-controlam-corpo-cerebro/>. Acesso em: 10 novembro 2019. 2.6

WOLPAW, Jonathan; WOLPAW, Elizabeth Winter. *Brain-Computer Interfaces: Principles and Practice*. New York: Oxford University Press, 2012. 1.1, 2.3, 4.1.4

WOLPAW, Jonathan R. Brain–computer interfaces as new brain output pathways. *Physiology*, v. 579, n. 3, p. 613–619, Jan 2007. 2.3, 2.3

YU XIANCHUAN E HU, Dan e Xu Jindong. *Blind Source Separation: Theory and Applications*. [S.l.]: John Wiley Sons, 2013. 2.6

ZHANG KUN E CHAN, Lai-Wan. Ica by pca approach: Relating higher-order statistics to second-order moments. In: *International Conference on Independent Component Analysis and Signal Separation*. [S.l.]: Springer, 2006. p. 311–318. 2.16, 2.17, 2.6.1, 2.18, 2.19

Apêndice A - Fontes dos Programas em Matlab

A.1 Programa EDF2RAW.m

Transformação do formato EDF para formato de Matrizes

```
% Escola: Centro Universitário Senai-Cimatec
% Curso:  Mestrado em Modelagem Computacional - Sistemas Complexos
% Aluno:  Osmar Ferreira Gomes
%
% Rotina para conversão dos dados de EEG da base Physionet
% https://physionet.org/physiobank/database/eegmmidb/
%
% Conversão da estrututa EDF para estrutura de Matrizes do Matlab
% Variaveis de entrada: Numero da pessoa observada
% Variáveis de saída:
% Matrizes com dados separados por tipo de movimento pensado
%
% Direções dos movimentos: Esquerda, Direita, Cima e Baixo
% Renomear S106raw.mat para S088raw.mat
% 107 -> 092, 108 -> 100, 109 -> 104

close; clear;

perro=[88 92 100 104];  % pessoas com erro
maxp=109;               % número de pessoas

for ipessoa=setdiff(1:maxp,perro)

  pessoa=strcat('S',num2str(ipessoa,'%03u'));

  fprintf('Separando épocas da pessoa %s\n', pessoa);

  arq=strcat('arqEDF\', pessoa ,'\', pessoa,'R04.edf'); % esquerdo/direito
  [T104, T204]=separe1(arq);    % T104 movimentos tipo 1 arq04
                                % T204 movimentos tipo 2 arq04
  arq=strcat('arqEDF\', pessoa ,'\', pessoa,'R08.edf'); % esquerdo/direito
  [T108, T208]=separe1(arq);    % T108 movimentos tipo 1 arq08
                                % T208 movimentos tipo 2 arq08
  arq=strcat('arqEDF\', pessoa ,'\', pessoa,'R12.edf'); % esquerdo/direito
  [T112, T212]=separe1(arq);    % T112 movimentos tipo 1 arq12
                                % T212 movimentos tipo 2 arq12
  esquerda=[T104; T108; T112]'; % saida somente movimento para esquerda
  direita= [T204; T208; T212]'; % saida somente movimento para direita

  arq=strcat('arqEDF\', pessoa ,'\', pessoa,'R06.edf'); % cima/baixo
  [T106, T206]=separe1(arq);

  arq=strcat('arqEDF\', pessoa ,'\', pessoa,'R10.edf'); % cima/baixo
  [T110, T210]=separe1(arq);

  arq=strcat('arqEDF\', pessoa ,'\', pessoa,'R14.edf'); % cima/baixo
  [T114, T214]=separe1(arq);

  cima= [T106; T110; T114]';
  baixo=[T206; T210; T214]';

  save(['arqRAW\' pessoa 'raw.mat'],'cima','baixo','esquerda','direita')
end
```

A.2 Função SEPARE1.m

Separação das ÉPOCAS

```
function [da1,da2]=separe1(arqedf)

  [d, h]=readEDF(arqedf);           % converte o arquivo edf em celula
  da=cell2mat(d);                   % matriz de dados brutos 19680x64
  he=cell2mat(h.annotation.event); % matriz de epocas 1x30
  freq=160;                         % frequencia de amostragem (Hz)
  iepoc=h.annotation.starttime;     % inicio do evento (s)
  amep=freq*iepoc+1;                % posicao inicial das epocas

  h1=amep(strfind(he,'1')/2); % posições iniciais das epocas tipo T1 em da
  h2=amep(strfind(he,'2')/2); % posições iniciais das epocas tipo T2 em da

  % separa as epocas em duas matrizes por tipo
  for ind=1:7
    epi=(ind-1)*656+1;
    da1(epi:656*ind,:)=da(h1(ind):h1(ind)+655,:); % 7 epocas tipo T1
    da2(epi:656*ind,:)=da(h2(ind):h2(ind)+655,:); % 7 epocas tipo T2
  end
  % save vsepare % provisório, salva as variaveis para debug
end
```

A.3 Programa CABENERGIA.m

Potência média relativa dos sinais pensados nos voluntários do Banco de Dados

```
%------- classificação dos eletrodos por potência média do sinal EEG
% ------ Nome: CebEnergia.m
close all;
%----- cálculo da potencia media dos eletrodos ---------%
maxp=105;
vrms=zeros(maxp,64); % matriz 105x64
for ipessoa=1:maxp
  pessoa=strcat('S',num2str(ipessoa,'%03u')); % individuo, ex 'S036'
  load(['arqRAW\' pessoa 'raw.mat'])
  % dados brutos: cima, baixo, esquerda, direita 64x13776=676x21
  mrms=rms(esquerda'); % 1x64
  mrms=mrms+rms(direita');
  mrms=mrms+rms(cima');
  mrms=mrms+rms(baixo');
  mrms=mrms/4; % media dos valores rms por individuo, todos os movimentos
  vrms(ipessoa,:)=mrms; % 105x64
end

% potencia média de cada eletrodo, todos os individuos
media=ceil(mean(vrms));

%------------------- mostra a cabeça ------------------%
hs=zeros(64,1); vet=zeros(64,1); cla;
load posicao.mat
apos=ceil(cell2mat(pos(:,1:4))/2);
imshow('Head5.jpg', 'Border','tight'); % carrega orelha e nariz
rectangle('Position',[60 40 387 413],'Curvature',[1,1]); % cabeca
for ind=1:64
  hs(ind)=rectangle('Position',apos(ind,1:4),'Curvature',[1,1]);
  text(apos(ind,1)+1, apos(ind,2)-5, strcat(pos{ind,5},'-',...
  num2str(ind)), 'FontSize', 8)
end % nome dos eletrodos

%------- mostra as cores na escala de potência --------%
media=media-min(media)+16; % ajuste de escala
maxcolor=max(media); % topo da escala de cores
colorbar; % mostra regua de cores
cmap=colormap(jet(maxcolor)); % gera mapa de cores
for ind=1:64 % colore cada eletrodo com a cor da media de potencia
  set(hs(ind),'FaceColor',cmap(media(ind),:))
end
%-------------------------------------------------------%
```

A.4 Programa LIMPAEDF.m

Minimização dos artefatos

```
%-----------------------------------------------------------------
% tratamento de artefatos pelo algoritimo jader
%-----------------------------------------------------------------

close; clear; tic

maxp=105;               % número de pessoas
eletrodos=[22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38];
nartef=4;               % numero de componentes artefatos

for ipessoa=1:maxp

  pessoa=strcat('S',num2str(ipessoa,'%03u'));
  load(['arqRAW\' pessoa 'raw.mat'])

  fprintf('Tratando artefatos da pessoa %s\n', pessoa);

  fprintf('Punho esquerdo\n');
  esquerdaf=esquerda(eletrodos,:);
  esquerdab=mjader(esquerdaf,nartef);

  fprintf('Punho direito\n');
  direitaf=direita(eletrodos,:);
  direitab=mjader(direitaf,nartef);

  fprintf('Ambos os punhos\n');
  cimaf=cima(eletrodos,:);
  cimab=mjader(cimaf,nartef);

  fprintf('Ambos os pés\n');
  baixof=baixo(eletrodos,:);
  baixob=mjader(baixof,nartef);

  fprintf('Salvando arquivos\n\n');

  save(['arqMAT\' pessoa 'direc.mat'],...
  'cimab','baixob','esquerdab','direitab','eletrodos','nartef')

  %dlmwrite(['arqCSV\' pessoa '\' pessoa 'esquerdab.csv'], esquerdab, ';')
  %dlmwrite(['arqCSV\' pessoa '\' pessoa 'direitab.csv'], direitab, ';')
  %dlmwrite(['arqCSV\' pessoa '\' pessoa 'cimab.csv'], cimab, ';')
  %dlmwrite(['arqCSV\' pessoa '\' pessoa 'baixob.csv'], baixob, ';')

  toc

end
```

A.5 Função Mjader.m

Separação de componentes

```
function saidaj = mjader(entradaj,nartef)
    for ind=1:21                % numero de epocas no arquivo
    inpjader=entradaj(:,(ind-1)*656+1:ind*656);  %isolando cada epoca
    c = jader(inpjader);        % gerando matriz de pesos
    recsig=c*inpjader;          % gerando os componentes
    vrms=rms(recsig,2);
    for imax=1:nartef
      [~,I]=max(vrms);
      vrms(I)=0; recsig(I,:)=0; % deletando componentes
    end
    outjader=c\recsig;          % sinais de eletrodos sem artefatos
    saidaj(:,:,ind)=outjader;   % armazenando
  end
end
```

Anexo B

Class MultilayerPerceptron

public class **MultilayerPerceptron**[1]

extends Classifier

implements OptionHandler, WeightedInstancesHandler, Randomizable

A Classifier that uses backpropagation to classify instances.

This network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are all sigmoid (except for when the class is numeric in which case the the output nodes become unthresholded linear units).

Valid options are:

-L (learning rate) Learning Rate for the backpropagation algorithm. (Value should be between 0 - 1, Default = 0.3).

-M (momentum) Momentum Rate for the backpropagation algorithm. (Value should be between 0 - 1, Default = 0.2).

-N (number of epochs) Number of epochs to train through. (Default = 500).

-V (percentage size of validation set) Percentage size of validation set to use to terminate training (if this is non zero it can pre-empt num of epochs. (Value should be between 0 - 100, Default = 0).

-S (seed) The value used to seed the random number generator (Value should be >= 0 and and a long, Default = 0).

-E (threshold for number of consequetive errors) The consequetive number of errors allowed for validation testing before the netwrok terminates. (Value should be >0, Default = 20).

[1]Disponível na URL: http://weka.sourceforge.net/doc.stable/weka/classifiers/functions/MultilayerPerceptron.html. Acessado em 02/02/2016.

-G GUI will be opened. (Use this to bring up a GUI).

-A Autocreation of the network connections will NOT be done. (This will be ignored if -G is NOT set)

-B A NominalToBinary filter will NOT automatically be used. (Set this to not use a NominalToBinary filter).

-H (comma seperated numbers for nodes on each layer) The hidden layers to be created for the network. (Value should be a list of comma separated Natural numbers or the letters 'a' = (attribs + classes) / 2, 'i' = attribs, 'o' = classes, 't' = attribs .+ classes) for wildcard values, Default = a).

-C Normalizing a numeric class will NOT be done. (Set this to not normalize the class if it's numeric).

-I Normalizing the attributes will NOT be done. (Set this to not normalize the attributes).

-R Reseting the network will NOT be allowed. (Set this to not allow the network to reset).

-D Learning rate decay will occur. (Set this to cause the learning rate to decay).

Author: Malcolm Ware (mfw4@cs.waikato.ac.nz)

B.1 Constructor Summary

MultilayerPerceptron() The constructor.

B.2 Method Summary

B.2.1 main

public static void main(java.lang.String[] argv)

Main method for testing this class.

Parameters: argv - should contain command line options (see setOptions)

B.2.2 setDecay

public void setDecay(boolean d)

Parameters: d - True if the learning rate should decay.

B.2.3 getDecay

public boolean getDecay()

Returns: the flag for having the learning rate decay.

B.2.4 setReset

public void setReset(boolean r)

This sets the network up to be able to reset itself with the current settings and the learning rate at half of what it is currently. This will only happen if the network creates NaN or infinite errors. Also this will continue to happen until the network is trained properly. The learning rate will also get set back to it's original value at the end of this. This can only be set to true if the GUI is not brought up.

Parameters: r - True if the network should restart with it's current options and set the learning rate to half what it currently is.

B.2.5 getReset

public boolean getReset()

Returns: The flag for reseting the network.

B.2.6 setNormalizeNumericClass

public void setNormalizeNumericClass(boolean c)

Parameters: c - True if the class should be normalized (the class will only ever be normalized if it is numeric). (Normalization puts the range between -1 - 1).

B.2.7 getNormalizeNumericClass

public boolean getNormalizeNumericClass()

Returns: The flag for normalizing a numeric class.

B.2.8 setNormalizeAttributes

public void setNormalizeAttributes(boolean a)

Parameters: a - True if the attributes should be normalized (even nominal attributes will get normalized here) (range goes between -1 - 1).

B.2.9 getNormalizeAttributes

public boolean getNormalizeAttributes()

Returns: The flag for normalizing attributes.

B.2.10 setNominalToBinaryFilter

public void setNominalToBinaryFilter(boolean f)

Parameters: f - True if a nominalToBinary filter should be used on the data.

B.2.11 getNominalToBinaryFilter

public boolean getNominalToBinaryFilter()

Returns: The flag for nominal to binary filter use.

B.2.12 setSeed

public void setSeed(int l)

This seeds the random number generator, that is used when a random number is needed for the network.

Specified by: setSeed in interface Randomizable

Parameters: l - The seed.

B.2.13 getSeed

public int getSeed()

Description copied from interface: Randomizable Gets the seed for the random number generations

Specified by: getSeed in interface Randomizable

Returns: The seed for the random number generator.

B.2.14 setValidationThreshold

public void setValidationThreshold(int t)

This sets the threshold to use for when validation testing is being done. It works by ending testing once the error on the validation set has consecutively increased a certain number of times.

Parameters: t - The threshold to use for this.

B.2.15 getValidationThreshold

public int getValidationThreshold()

Returns: The threshold used for validation testing.

B.2.16 setLearningRate

public void setLearningRate(double l)

The learning rate can be set using this command. NOTE That this is a static variable so it affect all networks that are running. Must be greater than 0 and no more than 1.

Parameters: l - The New learning rate.

B.2.17 getLearningRate

public double getLearningRate()

Returns: The learning rate for the nodes.

B.2.18 setMomentum

public void setMomentum(double m)

The momentum can be set using this command. THE same conditions apply to this as to the learning rate.

Parameters: m - The new Momentum.

B.2.19 getMomentum

public double getMomentum()

Returns: The momentum for the nodes.

B.2.20 setAutoBuild

public void setAutoBuild(boolean a)

This will set whether the network is automatically built or if it is left up to the user. (there is nothing to stop a user from altering an autobuilt network however).

Parameters: a - True if the network should be auto built.

B.2.21 getAutoBuild

public boolean getAutoBuild()

Returns: The auto build state.

B.2.22 setHiddenLayers

public void setHiddenLayers(java.lang.String h)

This will set what the hidden layers are made up of when auto build is enabled. Note to have no hidden units, just put a single 0, Any more 0's will indicate that the string is badly formed and make it unaccepted. Negative numbers, and floats will do the same. There are also some wildcards. These are 'a' = (number of attributes + number of classes) / 2, 'i' = number of attributes, 'o' = number of classes, and 't' = number of attributes + number of classes.

Parameters: h - A string with a comma seperated list of numbers. Each number is the number of nodes to be on a hidden layer.

B.2.23 getHiddenLayers

public java.lang.String getHiddenLayers()

Returns: A string representing the hidden layers, each number is the number of nodes on a hidden layer.

B.2.24 setGUI

public void setGUI(boolean a)

This will set whether A GUI is brought up to allow interaction by the user with the neural network during training.

Parameters: a - True if gui should be created.

B.2.25 getGUI

public boolean getGUI()

Returns: The true if should show gui.

B.2.26 setValidationSetSize

public void setValidationSetSize(int a)

This will set the size of the validation set.

Parameters: a - The size of the validation set, as a percentage of the whole.

B.2.27 getValidationSetSize

public int getValidationSetSize()

Returns: The percentage size of the validation set.

B.2.28 setTrainingTime

public void setTrainingTime(int n)

Set the number of training epochs to perform. Must be greater than 0.

Parameters: n - The number of epochs to train through.

B.2.29 getTrainingTime

public int getTrainingTime()

Returns: The number of epochs to train through.

B.2.30 blocker

public void blocker(boolean tf)

A function used to stop the code that called buildclassifier from continuing on before the user has finished the decision tree.

Parameters: tf - True to stop the thread, False to release the thread that is waiting there (if one).

B.2.31 getCapabilities

public Capabilities getCapabilities()

Returns default capabilities of the classifier.

Specified by: getCapabilities in interface CapabilitiesHandler

Overrides: getCapabilities in class Classifier

Returns: the capabilities of this classifier

See Also: Capabilities

B.2.32 buildClassifier

public void buildClassifier(Instances i) throws java.lang.Exception

Call this function to build and train a neural network for the training data provided.

Specified by: buildClassifier in class Classifier

Parameters: i - The training data.

Throws: java.lang.Exception - if can't build classification properly.

B.2.33 distributionForInstance

public double[] distributionForInstance(Instance i) throws java.lang.Exception

Call this function to predict the class of an instance once a classification model has been built with the buildClassifier call.

Overrides: distributionForInstance in class Classifier

Parameters: i - The instance to classify.

Returns: A double array filled with the probabilities of each class type.

Throws: java.lang.Exception - if can't classify instance.

B.2.34 listOptions

public java.util.Enumeration listOptions()

Returns an enumeration describing the available options.

Specified by: listOptions in interface OptionHandler

Overrides: listOptions in class Classifier

Returns: an enumeration of all the available options.

B.2.35 getOptions

public java.lang.String[] getOptions()

Gets the current settings of NeuralNet.

Specified by: getOptions in interface OptionHandler

Overrides: getOptions in class Classifier

Returns: an array of strings suitable for passing to setOptions()

B.2.36 toString

public java.lang.String toString()

Overrides: toString in class java.lang.Object

Returns: string describing the model.

B.2.37 globalInfo

public java.lang.String globalInfo()

This will return a string describing the classifier.

Returns: The string.

B.2.38 learningRateTipText

public java.lang.String learningRateTipText()

Returns: a string to describe the learning rate option.

B.2.39 momentumTipText

public java.lang.String momentumTipText()

Returns: a string to describe the momentum option.

B.2.40 autoBuildTipText

public java.lang.String autoBuildTipText()

Returns: a string to describe the AutoBuild option.

B.2.41 seedTipText

public java.lang.String seedTipText()

Returns: a string to describe the random seed option.

B.2.42 validationThresholdTipText

public java.lang.String validationThresholdTipText()

Returns: a string to describe the validation threshold option.

B.2.43 trainingTimeTipText

public java.lang.String trainingTimeTipText()

Returns: a string to describe the learning rate option.

B.2.44 nominalToBinaryFilterTipText

public java.lang.String nominalToBinaryFilterTipText()

Returns: a string to describe the nominal to binary option.

B.2.45 hiddenLayersTipText

public java.lang.String hiddenLayersTipText()

Returns: a string to describe the hidden layers in the network.

Modelo Computacional Baseado em Interface Cérebro-Computador Para Extração de Características e Classificação de Sinais EEG

Osmar Ferreira Gomes

Salvador, Agosto de 2020.